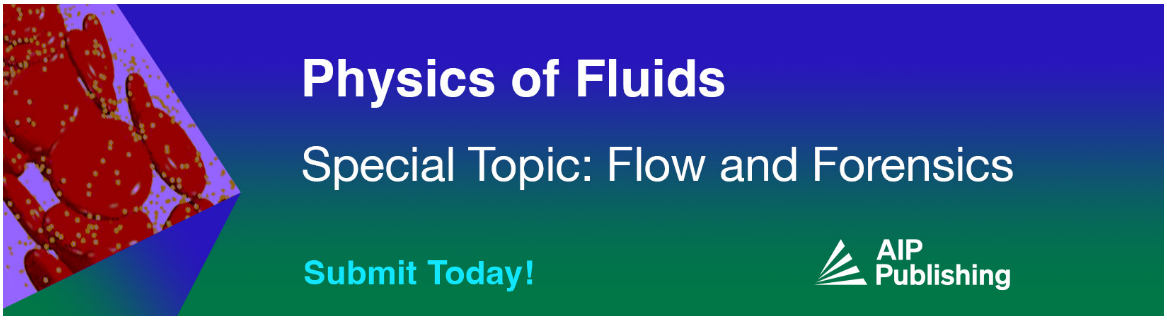
Physics of Fluids

RESEARCH ARTICLE | JANUARY 25 2022

Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network

Hongping Wang (王洪平) ; Yi Liu (刘毅); Shizhao Wang (王士召) ✉

Physics of Fluids 34, 017116 (2022)
https://doi.org/10.1063/5.0078143



Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network

Cite as: Phys. Fluids **34**, 017116 (2022); doi: 10.1063/5.0078143
Submitted: 10 November 2021 · Accepted: 4 January 2022 ·
Published Online: 25 January 2022

View Online Export Citation CrossMark

Hongping Wang (王洪平),[1,2] Yi Liu (刘毅),[1,2] and Shizhao Wang (王士召)[1,2,a)]

AFFILIATIONS

[1]The State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Engineering Science, University of Chinese Academy of Sciences, Beijing 100049, China

Note: This paper is part of the special topic, Artificial Intelligence in Fluid Mechanics.
[a)]**Author to whom correspondence should be addressed:** wangsz@lnm.imech.ac.cn

ABSTRACT

The velocities measured by particle image velocimetry (PIV) and particle tracking velocimetry (PTV) commonly provide sparse information on flow motions. A dense velocity field with high resolution is indispensable for data visualization and analysis. In the present work, a physics-informed neural network (PINN) is proposed to reconstruct the dense velocity field from sparse experimental data. A PINN is a network-based data assimilation method. Within the PINN, both the velocity and pressure are approximated by minimizing a loss function consisting of the residuals of the data and the Navier–Stokes equations. Therefore, the PINN can not only improve the velocity resolution but also predict the pressure field. The performance of the PINN is investigated using two-dimensional (2D) Taylor's decaying vortices and turbulent channel flow with and without measurement noise. For the case of 2D Taylor's decaying vortices, the activation functions, optimization algorithms, and some parameters of the proposed method are assessed. For the case of turbulent channel flow, the ability of the PINN to reconstruct wall-bounded turbulence is explored. Finally, the PINN is applied to reconstruct dense velocity fields from the experimental tomographic PIV (Tomo-PIV) velocity in the three-dimensional wake flow of a hemisphere. The results indicate that the proposed PINN has great potential for extending the capabilities of PIV/PTV.

I. INTRODUCTION

Flow measurement is essential for better understanding complex flow physics, validating numerical computations, and proving analytical and mathematical models or theories (Scharnowski and Kähler, 2020). Both particle image velocimetry (PIV, Adrian, 1984) and particle tracking velocimetry (PTV, Adamczyk and Rimai, 1988) are widely used to quantitatively describe the motion of flows in the field of experimental fluid mechanics. The greatest advantage is that PIV and PTV can noninvasively measure the instantaneous flow field simultaneously at many points (Westerweel *et al.*, 2013). The flow fields obtained through PIV/PTV allow the turbulence properties and coherent structures to be visualized and analyzed even at high Reynolds numbers (Adrian *et al.*, 2000; Adrian, 2005; Hutchins and Marusic, 2007; Humble *et al.*, 2009; Ye *et al.*, 2016; Wang *et al.*, 2019). Considering the complexity and three-dimensionality of flow motions, the development of a high-fidelity three-dimensional (3D) measurement technique to resolve all scales is natural and necessary.

Much effort has been devoted to achieving 3D measurements (Gao *et al.*, 2013; Westerweel *et al.*, 2013; Scarano, 2013; Discetti and Coletti, 2018; Raffel *et al.*, 2018). The defocusing PIV introduced by Willert and Gharib (1992) uses a three-pinhole aperture in conjunction with defocused particle imaging to code the depth of particles. This technique is primarily limited by the low seeding rates and low light intensity due to the small pinholes (Cierpka and Kähler, 2012). Holography can be used to record particle sizes and 3D positions in a technique referred to as holographic PIV (Hinsch, 2002; Shao *et al.*, 2020). Unfortunately, the axial resolution is much coarser than that in the other two directions (Gao *et al.*, 2013). The scanning PIV proposed by Brücker (1995) combines classical PIV with volume scanning using a scanning light sheet. Although this method can be used to obtain three-dimensional three-component (3D3C) instantaneous velocity fields, the technological limitation of the high-speed camera allows the study of only relatively low-speed flows. Ganapathisubramani *et al.* (2005) utilized three cameras to measure the full velocity components in two differentially separated planes. This dual-plane PIV can obtain the entire velocity gradient tensor only on the measured plane. The most important milestone in experimental fluid mechanics is the advent of tomographic PIV (Tomo-PIV, Elsinga *et al.*, 2006). In this approach, tracer particles are illuminated by a thick laser light sheet, and the light scattered by these particles is recorded by multiple cameras from different view angles (four cameras are usually required). The spatial intensity distribution is reconstructed according to the geometrical projection. Then, volumetric cross-correlation analysis or a particle tracking technique is applied to obtain the displacement field (Westerweel *et al.*, 2013). The ghost particles, generated due to the limited number of cameras, bring a negative effect of smoothing and reducing the particle displacement variations (Elsinga *et al.*, 2011). To overcome this weakness, many approaches have been proposed to remove or weaken ghost particles (de Silva *et al.*, 2013; Elsinga and Tokgoz, 2014; Schanz *et al.*, 2016; Wang *et al.*, 2016). 3D PTV is also a multiview technique (Maas *et al.*, 1993; Malik *et al.*, 1993); however, its seeding concentration of 0.005 particles per pixel (ppp) is one order of magnitude lower than that of Tomo-PIV. Recently, Shake-the-Box (Schanz *et al.*, 2016), combining the iteration triangulation method and image matching method (Wieneke, 2013), was specifically designed to track the particles in the temporal domain. The Shake-the-Box method enables temporal–spatial reconstruction with a particle image concentration as high as 0.125 ppp. Light-field PIV relies on light-field photography to capture 3D information of tracer particles (Shi *et al.*, 2017, 2018). The greatest advantage of light-field PIV is that it needs only one camera.

From the above discussion, 3D flow measurement relies heavily on the development of hardware and requires higher costs than planar PIV. Although the Shake-the-Box method has successfully reduced the occurrence of ghost particles and improved the measurement resolution, an emerging problem is how to accurately interpolate the sparse measured data into a structured mesh to observe and analyze the flow motions. With the development of technology and the improvement of computational power, data assimilation (DA) and machine learning (ML) may become another powerful way to improve the spatial resolution of PIV measurements or even infer the 3D flow field from sparse measured data (Chandramouli *et al.*, 2019). First, Navier–Stokes (N–S) equations can be utilized to enhance PIV or PTV data. Sciacchitano *et al.* (2012) solved the unsteady incompressible N–S equations to fill gappy PIV data. The vortex-in-cell method proposed by Schneiders and Scarano (2016) solves the vorticity transport equation by incorporating time-resolved volumetric PTV measurements. This method can be applied to reconstruct the dense velocity field from a sparse PTV field. By imposing physical constraints, FlowFit (Gesemann *et al.*, 2016) can simultaneously recover the velocity, acceleration, and pressure fields from a noisy PTV field using B-splines and penalties. Based on the DA framework, the 2D2C, 2D3C, and 3D3C velocity fields are used as observational data to improve the mean flow reconstruction (Symon *et al.*, 2017), to denoise and defilter PIV data (Gillissen *et al.*, 2019), and to determine the pressure fields (He *et al.*, 2020). Second, deep neural networks (DNNs) have been successfully applied to enhance the resolution of flow fields in studies on fluid mechanics (Fukami *et al.*, 2019; Zhou *et al.*, 2019; Deng *et al.*, 2019; Liu *et al.*, 2020; Kim *et al.*, 2021; Gao *et al.*, 2021; Zhou *et al.*, 2021; Zhang *et al.*, 2021). Once a DNN is trained using flow fields from high-fidelity simulations or experiments, it can be used to accurately predict high-resolution fields from low-resolution fields. Moreover, such a DNN can also be used to predict the aerodynamic performance of tire grooves (Uddin *et al.*, 2019) and the tire pattern noise in the early design stage (Lee *et al.*, 2021). These ML-based approaches provide new avenues for processing PIV velocity fields.

ML is a kind of data-driven method, which means that the result is determined by the quality of the training data instead of the known physical laws. Introducing physical laws into neural networks is necessary and significant. A physics-informed neural network (PINN, Raissi *et al.*, 2019; Raissi *et al.*, 2020) is a novel framework that combines data-driven neural networks and physical laws. The former can be regarded as a universal function approximator (Hornik *et al.*, 1989) represented by a fully connected neural network or a residual neural network, and the latter is achieved by embedding partial differential equations (PDEs) into the loss of the neural network using automatic differentiation. PINNs have been utilized to learn the velocity and pressure fields from flow visualization (Raissi *et al.*, 2020) or solve the incompressible N–S equations based on the velocity-pressure form and velocity-vorticity form, ranging from laminar flows to turbulent channel flows (Rao *et al.*, 2020; Jin *et al.*, 2021). Xu *et al.* (2021) employed a PINN and treated the governing equations as a parameterized constraint to recover the missing information of flow fields. Arzani *et al.* (2021) used a PINN to obtain the near-wall hemodynamics and wall shear stress from sparse velocity measurements. A PINN can even infer the full 3D velocity and pressure fields from snapshots of 3D temperature fields obtained by tomographic background-oriented Schlieren (Cai *et al.*, 2021b); this inverse problem is extremely ill-posed because only temperature fields are given. The Python library DeepXDE was designed by Lu *et al.* (2021) for solving differential equations in the PINN framework. Although a PINN in the current state cannot replace traditional approaches due to the limited accuracy and long training period (Markidis, 2021), it can still be considered a novel DA approach (Duraisamy *et al.*, 2019; Zhang *et al.*, 2020; Karniadakis *et al.*, 2021; Cai *et al.*, 2021a) and has received increasing research attention in the fluid mechanics field.

In the present work, we use a PINN to reconstruct dense velocity fields from sparse PIV/PTV data. There are two most significant differences compared with adjoint-based DA (Yang *et al.*, 2015; Mons *et al.*, 2016; Lemke and Sesterhenn, 2016; Symon *et al.*, 2017; He *et al.*, 2020). First, the numerical simulation solver is approximately replaced by a network, and the accuracy of the solver is embedded in the loss function. Second, the computational domain is the same as the measurement domain rather than the full computational mesh, as in the numerical simulation. Although the 3D velocity and pressure can be inferred from experimental data via a PINN (Jin *et al.*, 2021; Cai *et al.*, 2021a; Karniadakis *et al.*, 2021), whether PINNs are suitable for noisy PIV/PTV data and how the measurement error influences the PINN precision is still not clear. The rest of the paper is organized as follows. In Sec. II, we first introduce the PINN and the influences of activation functions, initial conditions, optimization algorithms, and data parameters on the performance of the proposed method. In Sec. III, we use direct numerical simulation (DNS) data of turbulent channel flow to numerically assess the performance of the PINN. In Sec. IV, the proposed method is applied to practical Tomo-PIV data of the 3D wake flow of a hemisphere. Finally, we offer a discussion and conclusions on the PINN in Sec. V.

II. METHODOLOGY

A. PINN

A PINN can seamlessly integrate known information (boundary conditions, initial conditions, and measured data) and physical laws by embedding PDEs into the loss function (Karniadakis *et al.*, 2021; Jin *et al.*, 2021). In the present work, we will extend the PINN to optimize the field data measured by PIV or PTV.

A PINN is essentially a DNN that can be used to approximate the solution determined by the data and PDEs; its architecture is shown in Fig. 1. A residual neural network (He *et al.*, 2016) is adopted in the present work, for which the relationship between the inputs and outputs can be expressed as

$$(\mathbf{U}, p) = \mathscr{F}_{NN}(\mathbf{X}, t; \mathbf{\Theta}). \tag{1}$$

Here, $\mathscr{F}_{NN}$ represents the neural network, whose inputs are space coordinates $\mathbf{X} = (x, y, z)$ and time t. The parameter $\mathbf{\Theta}$ represents the trainable variables. The outputs of the neural network are velocity vector $\mathbf{U} = (u, v, w)$ and pressure p. Through this network, the relationship between input $(\mathbf{X}, t)$ and output $(\mathbf{U}, p)$ is constructed. We use the parameters N_{layer} and N_{cell} to denote the number of hidden layers and the number of neurons in each hidden layer, respectively. The mathematical expression of the k-th hidden layer for the residual neural network is written as

$$Y^k = \sigma(Y^{k-1}W^{k-1} + b^{k-1}) + Y^{k-1}, \tag{2}$$

where W and b are the weights and biases, respectively, and Y represents the output of each hidden layer. The dimensionality of the output space for each hidden layer is N_{cell}. The choice of the activation function σ will be discussed in detail in Sec. II B.

Once the network is created, the partial derivatives $\partial/\partial\mathbf{X}$ and $\partial/\partial t$ can be computed based on the chain rule, which has been implemented by automatic differentiation in TensorFlow or PyTorch. Taking TensorFlow 2.1 as an example, the first-order derivative u_x is estimated using the function $tf.GradientTape().gradient(u, x)$. Higher-order derivatives can be estimated by calling this function multiple times. Therefore, the residuals of the N–S equations can be expressed by the network as

$$\begin{aligned}
e_1 &= u_t + uu_x + vu_y + wu_z + p_x - \frac{1}{Re}(u_{xx} + u_{yy} + w_{zz}),\\
e_2 &= v_t + uv_x + vv_y + wv_z + p_y - \frac{1}{Re}(v_{xx} + v_{yy} + v_{zz}),\\
e_3 &= w_t + uw_x + vw_y + ww_z + p_z - \frac{1}{Re}(w_{xx} + w_{yy} + w_{zz}),\\
e_4 &= u_x + v_y + w_z.
\end{aligned} \tag{3}$$

Here, Re represents the Reynolds number, and e_1, e_2, e_3, and e_4 are the residuals of the N–S equations. To make the network satisfy the governing equations, the loss function in the PINN is defined as follows:

$$\mathscr{L} = \mathscr{L}_{data} + \alpha\mathscr{L}_{eqns}, \tag{4}$$

where α is a weighting coefficient and $\mathscr{L}_{data}$ and $\mathscr{L}_{eqns}$ are computed as

$$\begin{aligned}
\mathscr{L}_{data} &= \sum_{j=1}^{N_{data}} |\mathbf{U}_{data}(\mathbf{X}^j, t^j) - \mathbf{U}_{pred}(\mathbf{X}^j, t^j)|^2,\\
\mathscr{L}_{eqns} &= \sum_{i=1}^{4}\sum_{j=1}^{N_{eqns}} |e_i(\mathbf{X}^j, t^j)|^2.
\end{aligned} \tag{5}$$

Here, $\mathscr{L}_{data}$ represents the loss between the measured data $\mathbf{U}_{data}$ and the predicted data $\mathbf{U}_{pred}$ and $\mathscr{L}_{eqns}$ denote the total residual of the N–S equations. The parameters N_{data} and N_{eqns} denote the numbers of measured data and equation points. The equation points are randomly selected and uniformly distributed in the computational domain. Note that the residuals of the N–S equations at the measured points are also estimated. Once the loss function is defined, the variables $\mathbf{\Theta}$ are optimized by minimizing the loss function, as shown in Fig. 1. The value of α will be discussed in Sec. II C when considering measurement noise. The PINN program is coded based on the open-source ML platform TensorFlow v2.1.0 within Python and has been shared on https://github.com/hpwang87/PINN-TF2.1. We also referred to other

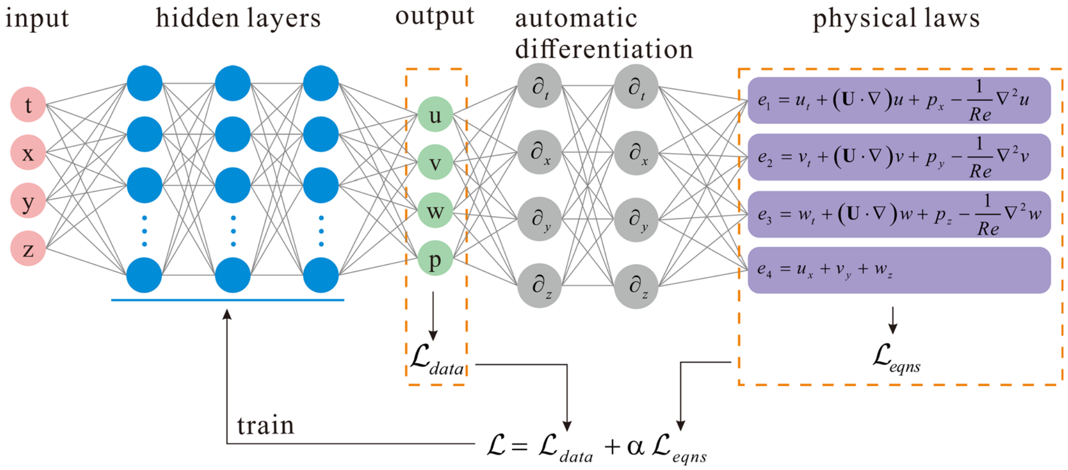

input
hidden layers
output
automatic differentiation
physical laws
t
x
y
z
u
v
w
p
∂_t
∂_x
∂_y
∂_z
$e_1 = u_t + (\mathbf{U}\cdot\nabla)u + p_x - \frac{1}{Re}\nabla^2 u$
$e_2 = v_t + (\mathbf{U}\cdot\nabla)v + p_y - \frac{1}{Re}\nabla^2 v$
$e_3 = w_t + (\mathbf{U}\cdot\nabla)w + p_z - \frac{1}{Re}\nabla^2 w$
$e_4 = u_x + v_y + w_z$
$\mathcal{L}_{data}$
$\mathcal{L}_{eqns}$
train
$\mathcal{L} = \mathcal{L}_{data} + \alpha\,\mathcal{L}_{eqns}$

FIG. 1. Architecture of a PINN. The inputs are space coordinates $\mathbf{X} = (x, y, z)$ and time t, and the outputs are velocity $\mathbf{U} = (u, v, w)$, and pressure p. The physical laws are represented by incompressible N–S equations and expressed using automatic differentiation operators. In the present work, we only considered the data measured by PIV/PTV.

open-source PINN libraries written by Raissi *et al* (2019) and Lu *et al.* (2021). The weights and biases of the network are initialized using the default Glorot uniform initializer and zeros, respectively. Unless otherwise specified, the network is trained and tested on a workstation equipped with an Intel Xeon W2245 3.90 GHz CPU and a single NVIDIA Quadro RTX 5000 GPU with 16 GB. The training variables are updated using the adaptive moment estimation (Adam) optimizer with an initial learning rate of 1×10^{-3}. The learning rate decays exponentially in a staircase manner. After 10 000 epochs with 10 steps per epoch in training, the loss converges to a very low value, and then, the network can be used to predict the velocity and pressure once the input $(\mathbf{X}, t)$ is given. Note that the pressure is solved without any known information. Before using the PINN to tackle experimental data, we will first explore the influences of activation functions, initial conditions, optimization algorithms, and data parameters on the performance of the proposed method.

B. Activation functions, initial conditions, and optimization algorithms

Here, we consider 2D Taylor's decaying vortices to check the influence of activation functions, initial conditions, and optimization algorithms on the PINN. An exact 2D solution to the N–S equations given by Taylor (1923) is as follows (Kim and Moin, 1985; Ethier and Steinman, 1994):

$$\begin{aligned} u(x,y,t) &= -\cos(x)\sin(y)e^{-2t}, \\ v(x,y,t) &= \sin(x)\cos(y)e^{-2t}, \\ p(x,y,t) &= -0.25\big(\cos(2x)+\cos(2y)\big)e^{-4t}. \end{aligned} \tag{6}$$

The computation is carried out in the domain of $0 \le x, y \le 2\pi$, and $0 \le t \le 2$. The Reynolds number *Re* is equal to 1. As shown in Fig. 2, ten thousand equation points ($N_{eqns} = 1 \times 10^4$) are randomly distributed in the space-time domain as red dots. The blue dots represent the data points where the velocity has been given by Eq. (6), these data points are also randomly selected in the space-time domain. The training dataset consists of the equation points and data points. Note that the data loss is estimated at data points, while the residuals of the N–S equations are computed at both the equation points and data points.

The activation function plays a major role in the success of data approximation with neural networks. Without activation functions, the network is only a linear transformation. Currently, the widely used activation functions include the rectified linear unit (ReLU), leaky ReLU, tanh, sigmoid, swish, and adaptive swish functions (Ramachandran *et al.*, 2017; Markidis, 2021). The adaptive swish function is expressed by

$$\sigma(x) = \frac{x}{1+e^{-ax}}, \tag{7}$$

where $a = n\beta$, β is a trainable parameter, and n is a scale factor. According to Jagtap *et al.* (2020), without the scale factor n, the adaptive parameter a will present a slow convergence toward the optimal value due to the small learning rate. A larger $n \ge 1$ can accelerate the convergence procedure to find an optimal a. Therefore, the scale factor n is set to 10, and the initial value of β is set to 0.1 (Jagtap *et al.*, 2020). Figure 3 presents the training loss (a) and the validation loss (b) of different activation functions for the case of $N_{eqns} = 10000$, $N_{data} = 2048$, $N_{cell} = 64$, and $N_{layer} = 7$. Ten percent of the input is reserved to be used as the validation data. For each activation function, the PINN is trained for 10 000 epochs. As seen in this figure, the performance of adaptive swish is the best among all the activation functions. The solutions with ReLU and leaky ReLU do not converge to the exact solutions because of the discontinuity of these functions (Raissi *et al.*, 2020; Markidis, 2021). The sigmoid function also performs poorly because it is more appropriate for probability prediction. We also present the final training loss, final validation loss, and total training time for the different activation functions in Table I. Among all the activation functions, adaptive swish has the smallest difference between the training loss and the validation loss. However, the computation time with adaptive swish is approximately 3–4 times as long as that with ReLU. In the present work, we choose adaptive swish with a scale factor of $n = 10$ as the activation function of the PINN.

To determine the weights and biases of the network, an optimization process is applied to minimize the loss function. The Adam optimizer is a stochastic gradient descent method based on adaptive estimation of the first moment (the mean) and second moment (the uncentered variance) (Kingma and Ba, 2014) and has been widely used in deep learning. However, a large number of iterations are needed to converge to an exact solution. Therefore, a more efficient

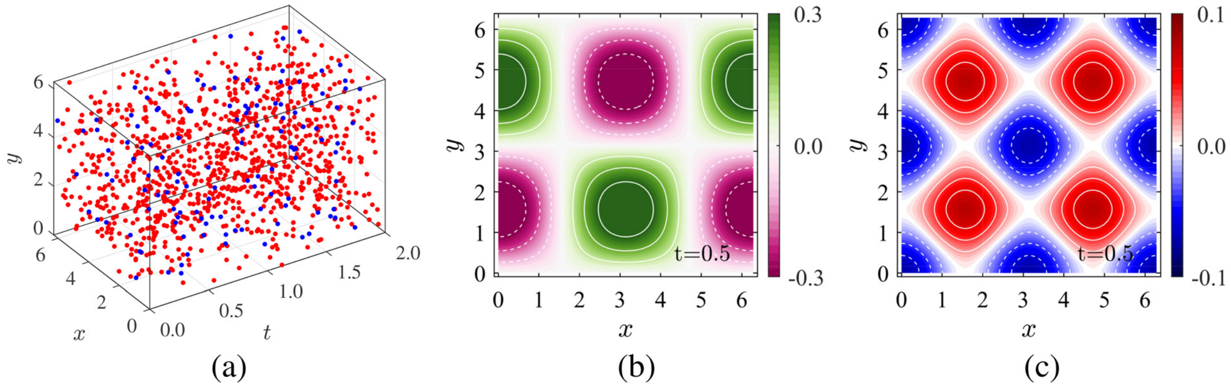


FIG. 2. (a) Distribution of the equation points (red dots) and the data points (blue dots) in the space-time domain for the case of 2D Taylor's decaying vortices. (b) Streamwise velocity of the analytical solution at $t = 0.5$. (c) Pressure of the analytical solution at $t = 0.5$.

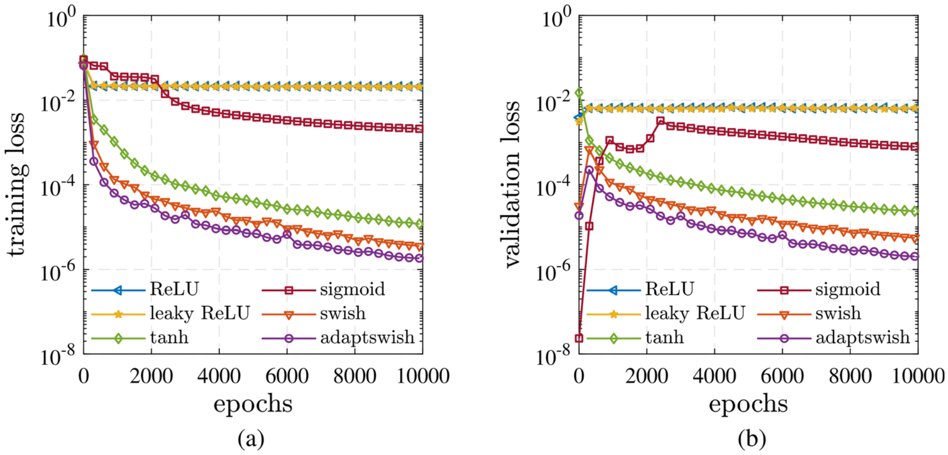


FIG. 3. Training loss (a) and validation loss (b) for different activation functions; only the Adam optimizer with an initial learning rate of 1×10^{-3} is used. 10% of the input is reserved to be used as validation data.

TABLE I. Comparison of training loss, validation loss, and computation time for different activation functions with 10 000 training epochs.

	ReLU	Leaky ReLU	Tanh	Sigmoid	Swish	Adapt swish
Final training loss	2.1×10^{-2}	2.1×10^{-2}	1.2×10^{-5}	2.1×10^{-3}	3.6×10^{-6}	1.8×10^{-6}
Final validation loss	6.5×10^{-3}	6.5×10^{-3}	2.3×10^{-5}	7.9×10^{-4}	5.6×10^{-6}	2.0×10^{-6}
Total training time (s)	3181	3123	4741	4373	10 503	11 851

optimizer, limited memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) (Liu and Nocedal, 1989), which considers the Hessian matrix, is used to optimize the variables after Adam optimization (Raissi *et al.*, 2019; Markidis, 2021; Jin *et al.*, 2021). The Adam optimizer is used first to avoid local minima. Figure 4 shows the training error for the adaptive swish function with the Adam optimizer (before 5000 epochs) followed by the L-BFGS-B optimizer, which is implemented by a wrapper named autograd-minimize (https://pypi.org/project/autograd-minimize/). After the first 5000 epochs, the loss with the Adam optimizer is reduced to 1×10^{-5}, while the L-BFGS-B optimizer can reduce the loss from 1×10^{-5} to 5×10^{-7}. The performance of the L-BFGS-B optimizer is obviously much better than that of the Adam optimizer. However, we note that there are two limitations for the L-BFGS-B optimizer. First, it cannot be used at the beginning of the training because it may converge to a local minimum (Markidis, 2021). Second, it is only practical for small-scale ML applications since the L-BFGS-B optimizer is a full batch approach (Bollapragada *et al.*, 2018). Therefore, most of the cases in the present work still use the Adam optimizer. Additionally, we also investigate the influence of the initial conditions of the network on training. The upper and lower boundaries of the blue shaded region shown in Fig. 4 indicate the maximum and minimum of the loss from ten independent runs with independent initial variables. Different initial values have a negligible effect on the training loss.

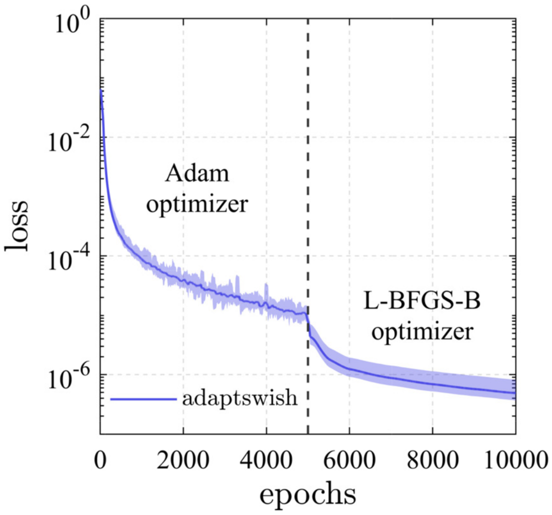


FIG. 4. Training loss for the adaptive swish function. The Adam optimizer is used before 5000 epochs (the dashed black line), followed by the L-BFGS-B optimizer to fine-tune the network. The upper and lower boundaries of the shaded region indicate the maximum and minimum of the loss from ten independent runs with independent initial variables.

C. On the parameters N_{data}, N_{cell}, and α

In this section, we mainly consider the influence of N_{data}, N_{cell}, and α on the prediction accuracy with and without random noise. The streamwise u-component and pressure predicted by the PINN with different N_{data} are presented in Fig. 5, and noise is not taken into account. One advantage of the PINN is that it can output pressure without providing any information. The pressure is optimized by minimizing the residuals of the N–S equations. The number of neurons in each layer and the number of layers are set to $N_{cell} = 64$ and $N_{layer} = 7$ for all cases, and the number of data points varies from 32 to 256 from left to right. Compared with the exact solution presented in Fig. 2, two

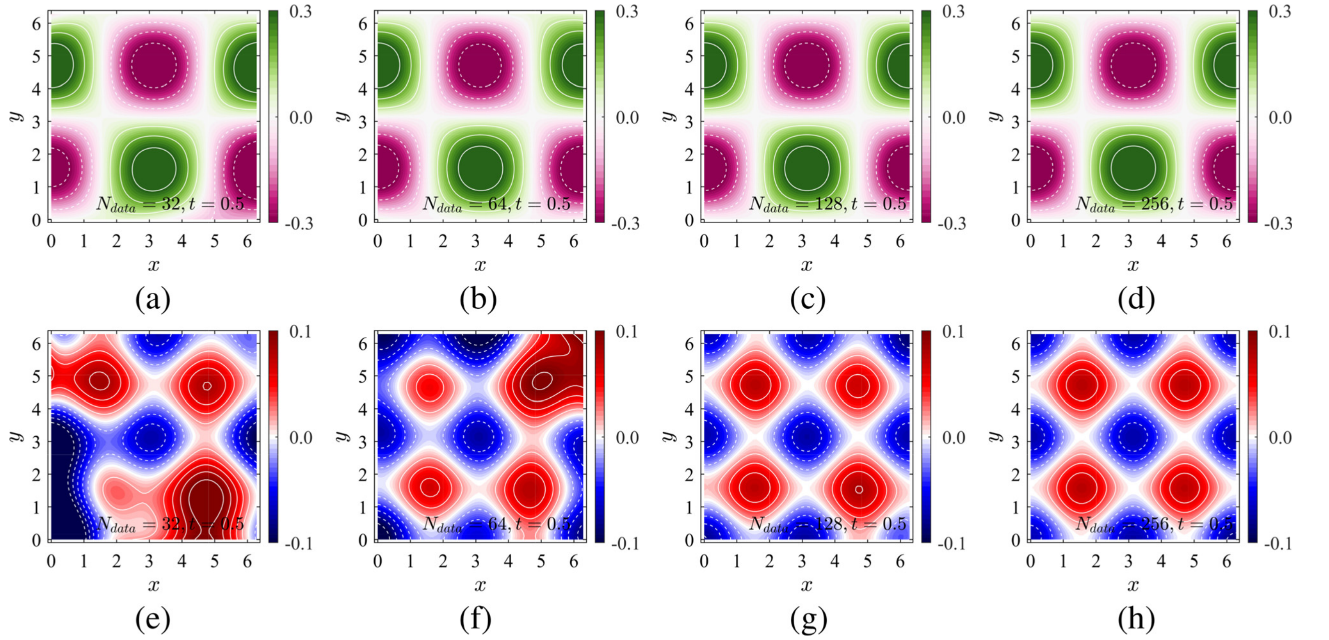


FIG. 5. Streamwise *u*-component (a)–(d) and pressure (e)–(h) predicted by the PINN with $N_{data} = 32$ (a) and (e), 64 (b) and (f), 128 (c) and (g), and 256 (d) and (h). N_{cell} and N_{layer} are equal to 64 and 7 for all cases, and the time instant $t = 0.5$. Noise is not taken into account.

remarks can be offered. First, the predicted velocity fields show better agreement with the exact velocity distribution than the pressure. This implies that the velocity more easily converges than the pressure due to the velocity constraint in the PINN. The error can be propagated and amplified from velocity to pressure (de Kat and van Oudheusden, 2012; Pan *et al.*, 2016). Second, increasing N_{data} can significantly reduce the error in the predicted field. To quantitatively estimate the error caused by the PINN, the relative two-norm (L_2) error of u is defined as

$$\varepsilon_u = \frac{||u_e - u_p||_2}{||u_e||_2}. \tag{8}$$

Here, u_e and u_p are the exact velocity and the predicted velocity, respectively. Figure 6 shows the relative L_2 errors as a function of N_{data} and N_{cell}, and noise is not considered. In general, the error deceases with increasing N_{data} and N_{cell}. For the velocity, the error converges to less than 1% when N_{data} is larger than 1000, while for the pressure, the error converges to approximately 10%. Increasing the scale of the network (represented by N_{cell}) can also reduce the prediction error; however, this works only when the PINN is fed sufficient data.

The influence of measurement noise on the performance of the PINN is also evaluated. Gaussian random noise with a standard deviation proportional to the velocity fluctuation is added to the training data. Figure 7 displays the relative error in the velocity (a) and pressure (b) as a function of noise level and weighting coefficient α. The noise level varies from 0 to 20% with an interval of 2.5%, and α varies from 1 to 80 with an interval of 10. The number of data points and the number of neurons in each layer are $N_{data} = 8192$ and $N_{cell} = 64$ in this test case. The measurement noise indeed deteriorates the performance of the PINN, as both the velocity and pressure relative errors

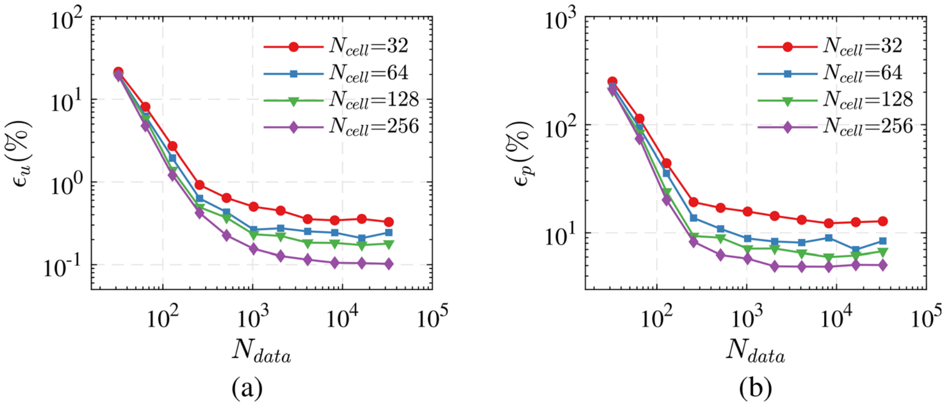


FIG. 6. Charts of the relative L_2 errors of the streamwise u-component (a) and pressure (b) vs N_{data} and N_{cell}. N_{layer} is fixed to 7, and training is conducted for 10 000 epochs in all cases. The error at each point is obtained by averaging three independent runs. No noise is added to the training data.

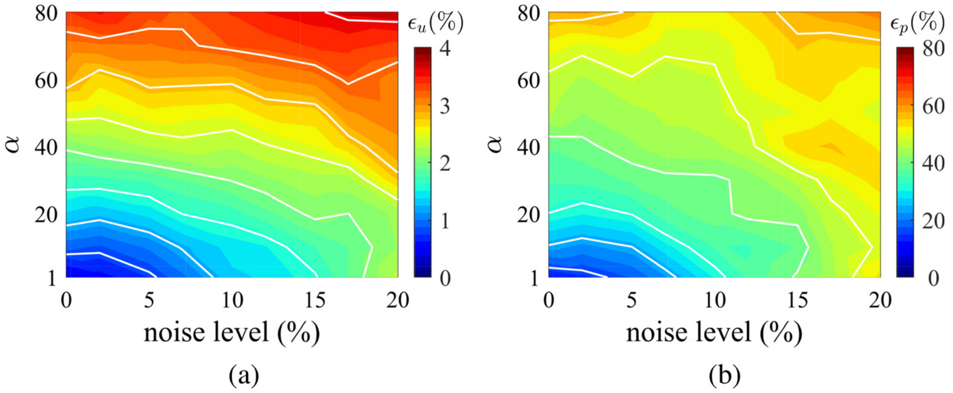


FIG. 7. Contour maps of the relative L_2 error for velocity (a) and pressure (b) as a function of noise level and weight coefficient α. The number of neurons in each layer is 64, and the data number is set to 8192. The intervals of the noise level and weight coefficient are 2.5% and 10, respectively. The error at each point is obtained by averaging three independent runs.

increase with the increasing noise level. At a specific noise level, increasing α, which implies that the constraint imposed by the N–S equations is enhanced, does not improve the accuracy of the PINN. Jin *et al.* (2021) proposed a dynamic weight strategy to estimate α in an N–S flow net, where a PINN is employed as a computational fluid dynamics solver to solve N–S equations. In their work, the dynamic weight strategy performs well since the boundary and initial conditions are accurate without error. However, for velocity reconstruction using experimental data, the optimal α is selected as 1, as shown in Fig. 7.

Figure 8 shows the relative error in the velocity (a) and pressure (b) as a function of N_{data} and N_{cell} for $\alpha = 1$. Noise of 10% of the velocity fluctuation is added to the training data. When considering measurement noise, the characteristics of the PINN, as shown in Fig. 8, are still essentially similar to those in Fig. 6. On the one hand, increasing N_{cell}, the scale of the network, has little influence on the velocity accuracy; however, it can significantly improve the performance of the pressure prediction when N_{data} is larger than 10^4. The velocity error is much lower than the pressure error since the loss of data is imposed on the velocity. It is difficult to further reduce the velocity error by increasing the scale of the network because the loss of data (containing noise) is much larger than the residuals of the equations. On the other hand, increasing the amount of data is still an effective way to suppress the negative effect brought by measurement noise. In addition, the relative error in pressure is almost one order of magnitude higher than that in velocity.

III. NUMERICAL ASSESSMENTS ON WALL-BOUNDED TURBULENCE

A. Data description

A DNS dataset of turbulent channel flow at $Re_\tau = 550$ is used to numerically assess the performance of the PINN for wall-bounded turbulence since it is inhomogeneous and anisotropic. We want to know whether the PINN can accurately reconstruct the turbulence when given measured data. The streamwise, spanwise, and wall-normal lengths of the measurement volume are $L_x = \delta$, $L_z = \delta$, and $L_y = 0.2\delta$, respectively, and the wall-normal region starts from 0. The parameter δ denotes the thickness of the boundary layer. Three hundred instantaneous flow fields with time interval $\Delta t^+ = 0.3$ are used to generate the dataset. Δt^+, which is ten times the simulation time step, is normalized by the skin friction velocity and the inner length scale. Figure 9 displays an instantaneous flow field with complex vortical structures in the cropped volume. These vortices are identified by the Q-criterion, where Q denotes the vortex identification criterion estimated as the second invariant of the velocity gradient tensor. This DNS dataset is used to mimic experimental data.

The data structures for the measured velocities obtained by PIV and PTV are different. On the one hand, the cross-correlation method

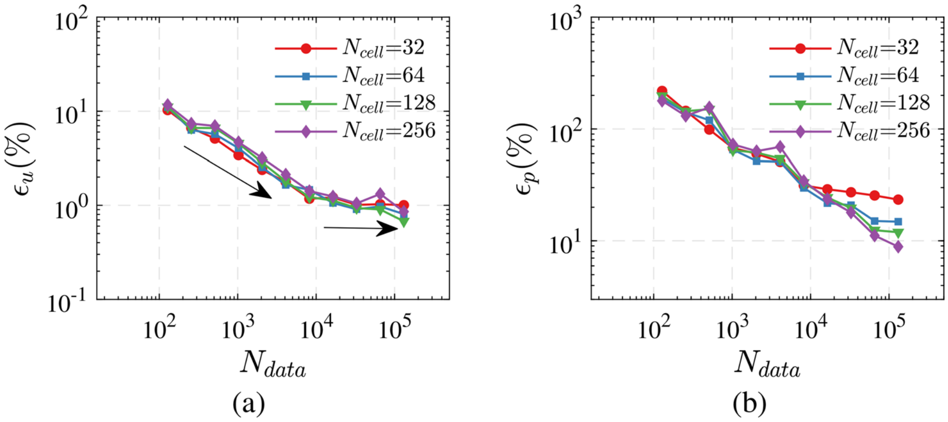


FIG. 8. Charts of the relative L_2 errors of the streamwise u-component (a) and pressure (b) vs N_{data} and N_{cell}. N_{layer} is fixed to 7, and training is conducted for 10 000 epochs in all cases. The error at each point is obtained by averaging three independent runs. Gaussian random noise of 10% of the velocity fluctuation is added to the velocity.

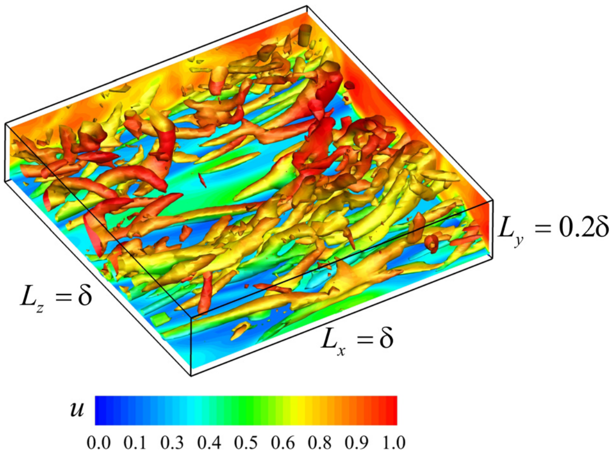


FIG. 9. Instantaneous flow field in the DNS dataset. The isosurfaces colored by the streamwise velocity u represent vortical structures identified by the Q-criterion with $Q = 6$. The contour map of the streamwise velocity is also presented in the $x - z$ plane at $y = 0.01$ ($y^+ = 5.5$).

of PIV results in a regular velocity field with a rectangular or square grid. In particular, PIV is incapable of resolving the near-wall velocity due to the large interrogation window and the high-velocity gradient. On the other hand, the PTV data points are randomly and uniformly distributed in the whole measured volume. Therefore, three kinds of data points, denoted by $S1$, $S2$, and $S3$, are considered as follows.

$S1$ The first kind of data point, which is shown in Fig. 10(a), is generated to simulate the regular PIV velocity field. The spacing step is set to 15 wall units in all three directions, and there are no data in the region of $y^+ < 15$. This configuration is extracted from the real experimental setup by Schröder *et al.* (2011). The size of the data points in one volume is $35 \times 35 {\times} 6$. The velocity at these points is equal to the DNS result without considering measurement noise.

$S2$ The second kind of data point, which is shown in Fig. 10(b), is generated to simulate the random velocity field obtained by PTV. The number of data points is equal to that in PIV. The velocity at these points is equal to the DNS result without considering measurement noise.

$S3$ The third kind of data point, which is shown in Fig. 10(c), is generated in the same way as $S1$ except that the measurement noise is considered. The velocity is contaminated by adding spatially correlated Gaussian random noise to the velocity fluctuation (Azijli and Dwight, 2015; Wang *et al.*, 2018) to simulate the error caused by the cross-correlation in PIV. The standard deviation of the Gaussian noise is set to 5% of the local magnitude of each velocity component, and then, the velocity field is smoothed by adopting Gaussian filtering with a 3×3 kernel.

B. Flow field prediction from sparse 3D3C data

The dense velocity field can be obtained by mathematically interpolating the sparse data. Therefore, the proposed PINN approach is also compared with linear interpolation and natural neighbor interpolation. The latter involves taking a weighted average over natural neighbors of a point (Ledoux and Gold, 2005). This method has advantages over linear interpolation and is provided by the griddata function in SciPy and MATLAB. Figure 10 shows the vortical structures identified using the Q-criterion for different methods at the same time instant as the DNS result given in Fig. 9. The vortical structures are colored based on the streamwise velocity u. From top to bottom, the velocity is reconstructed into a DNS mesh using linear interpolation, natural interpolation, and the proposed PINN approach. From left to right, the figures show the corresponding reconstructions from different data points denoted by $S1$, $S2$, and $S3$. For the PINN method, the parameters are summarized in Table II; this configuration yields a total of 1 266 919 trainable variables. From the visual comparison, the coherent structures for $S1$ and $S2$ are similar to each other due to the same number of data points, and the flow field interpolated by the natural method is much smoother than that from the linear interpolation. For $S3$, the vortical structures are inevitably reduced due to the spatially correlated noise introduced by the PIV algorithm. Therefore, the threshold of the Q-criterion is reduced from 6 (for $S1$ and $S2$) to 3 (for $S3$). The fields reconstructed by the PINN are closer to the DNS result (Fig. 9) and smoother than the linear interpolation and the natural interpolation based on a visual comparison. More vortical structures are resolved by the PINN, although the PINN still cannot fully recover all the vortical structures. At the $x - z$ plane of $y^+ = 5.5$, the PINN fields clearly capture the low- and high-speed streaks in the volume.

We also plot the time series of streamwise velocity u, as shown in Fig. 11. The velocity is extracted at the center of the streamwise–spanwise plane, and the wall-normal position y^+ is 50 and 8 for the top and bottom panels. Note that there are no data in the region of $y^+ \leq 15$ for the simulated PIV data denoted by $S1$ and $S3$. Both the results from $S1$ and $S2$ show good agreement with the reference data series. When considering the measurement noise, the reconstructed velocities present a large deviation from the DNS data, especially at $y^+ = 8$. Another important feature is that small scales are lost by the PINN. This may be caused by two reasons. One is that the scale of the present PINN is not sufficient to resolve the small scales in turbulence. The other is that the loss of the present PINN needs to further converge to a lower level.

In addition to inspecting the results by visual comparison of the vortical structures and time series, the velocity statistics obtained by the PINN with different kinds of data points ($S1$, $S2$, and $S3$) are also compared with the DNS profiles, as shown in Fig. 12. The statistics are obtained using all 300 snapshots and averaged along with the streamwise and spanwise directions. Without considering the noise, the mean streamwise velocity U and the streamwise velocity fluctuation $\langle u' \rangle$ are consistent with the DNS profiles, while the velocity fluctuations $\langle v' \rangle$ and $\langle w' \rangle$ underestimate the DNS profiles. The velocity component of u presents a good approximation compared to v and w. This is because the streamwise velocity is much larger in magnitude than the spanwise and wall-normal velocities in wall-bounded turbulence. When considering the spatially correlated noise ($S3$), the statistical profiles obtained by the PINN present larger deviations from the DNS. The mean streamwise velocity is overestimated by the PINN at $y^+ \leq 15$, and the velocity fluctuations are still underestimated. The PINN with $S2$ performs slightly better than that with $S1$, which means that the PTV data are more suitable for the PINN.

Based on the numerical assessments of wall-bounded turbulence, we would like to offer two remarks on the performance of the PINN.

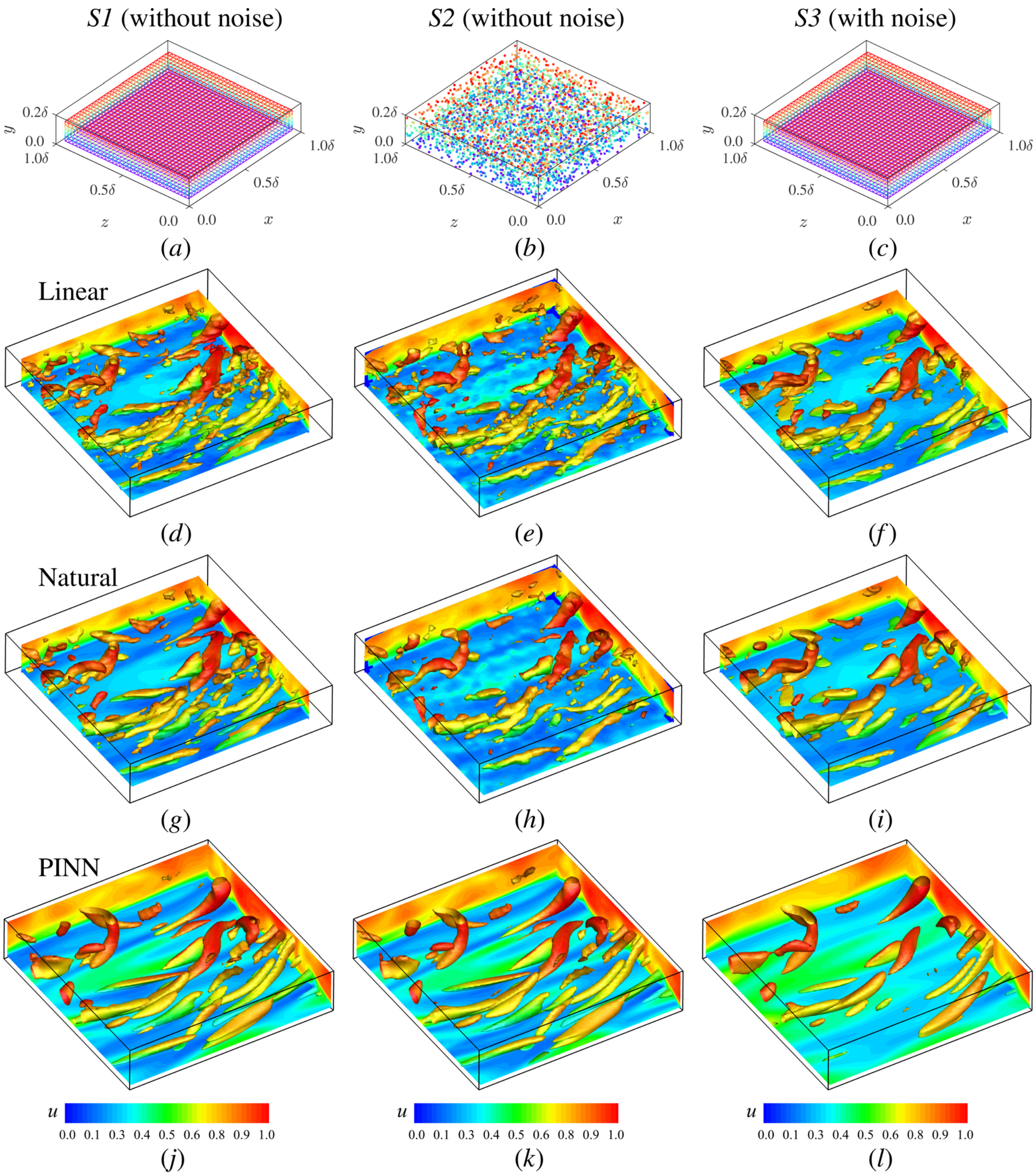


FIG. 10. Instantaneous flow field reconstructed by different methods; the time instant is the same as the DNS result shown in Fig. 9. The isosurfaces colored by the streamwise velocity u represent vortical structures identified by the Q-criterion. The contour map of the streamwise velocity is also presented in the $x-z$ plane at $y = 0.01$ ($y^+ = 5.5$). From top to bottom, the results of linear interpolation (d)–(f), natural interpolation (g)–(i) and the PINN (j)–(l). From left to right, the results for $S1$ (d), (g), and (j), $S2$ (e), (h), and (k) and $S3$ (f), (i), and (l). The value of Q for $S3$ ($Q = 3$) is lower than that for $S1$ and $S2$ ($Q = 6$) since the velocity for $S3$ is filtered by a Gaussian kernel. Schematic diagrams of the distribution of the data points for $S1$, $S2$, and $S3$ are displayed in figures (a)–(c), respectively.

First, the capacity of the PINN to reconstruct complex flows such as turbulence is of great significance. However, large scales can be more easily recovered than small scales for the PINN. Second, how to reduce the measurement error and increase the data resolution must be considered. High-quality data such as PTV data can improve the performance of the PINN. Overall, when applying the PINN to tackle complex flows, a network of an appropriate scale and high-quality data should be considered.

TABLE II. Summary of the parameters of the PINN for wall-bounded turbulence.

N_{eqns}	N_{data}	N_{cell}	N_{layer}	Trainable params
5×10^6	2.2×10^6	300	15	1 266 919

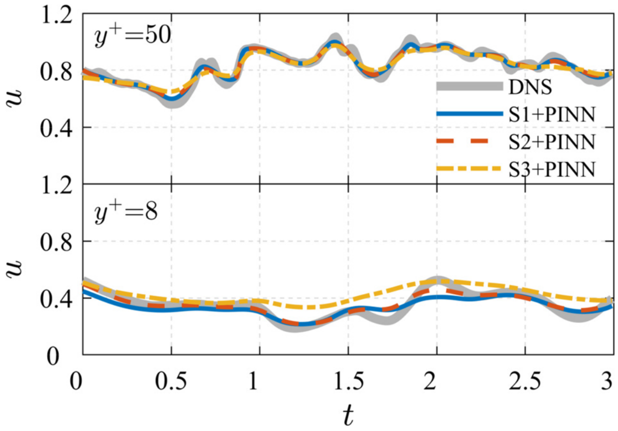


FIG. 11. Time series of streamwise velocity u reconstructed by the PINN with data points of $S1$, $S2$, and $S3$. The velocity is sampled at the center position of the wall-parallel plane, and the wall-normal location y^+ is 50 and 8.

IV. EXPERIMENTAL ASSESSMENTS ON THE 3D WAKE FLOW OF A HEMISPHERE

In this section, the PINN is applied to improve the quality of experimental data of the 3D wake flow of a hemisphere measured by adopting Tomo-PIV. The experimental setup is shown in Fig. 13, which was described in the doctoral thesis by Wang (2017) and the paper by Wang *et al.* (2018) in detail. A short and brief description of the experimental setup will be provided in this section. This experiment was carried out in a water tunnel at Beihang University, China. A hemisphere with a diameter $D = 20$ mm was mounted on a flat plexiglass plate, and this plate was vertically posited in the test section of the water tunnel. The distance between the leading edge of the plate and the hemisphere was 700 mm. The freestream velocity U_∞ in the present case is approximately 145 mm/s. Therefore, the Reynolds number based on the hemisphere diameter $Re_D = U_\infty D/\nu$ is approximately 2750 at a temperature of 18°C, where ν is the kinematic viscosity of water. The thickness of the boundary layer at the location of the hemisphere is approximately 11.65 mm, which was measured by planar PIV. Note that the base flow without the hemisphere is a laminar boundary layer. The x direction is defined as the freestream velocity direction, and the z direction is defined as the wall-normal direction. A time-resolved Tomo-PIV system was utilized to record and analyze particle images. The Tomo-PIV parameters are summarized in Table III. A high-frequency double-pulse laser with a wavelength of 532 nm was used to illuminate hollow glass beads with diameters ranging from 5 to 20 μm, and the energy of the laser was set to 30 mJ/pulse. The minimum response frequency of the tracers is approximately 45 kHz. According to a spectral analysis (not shown in this paper),

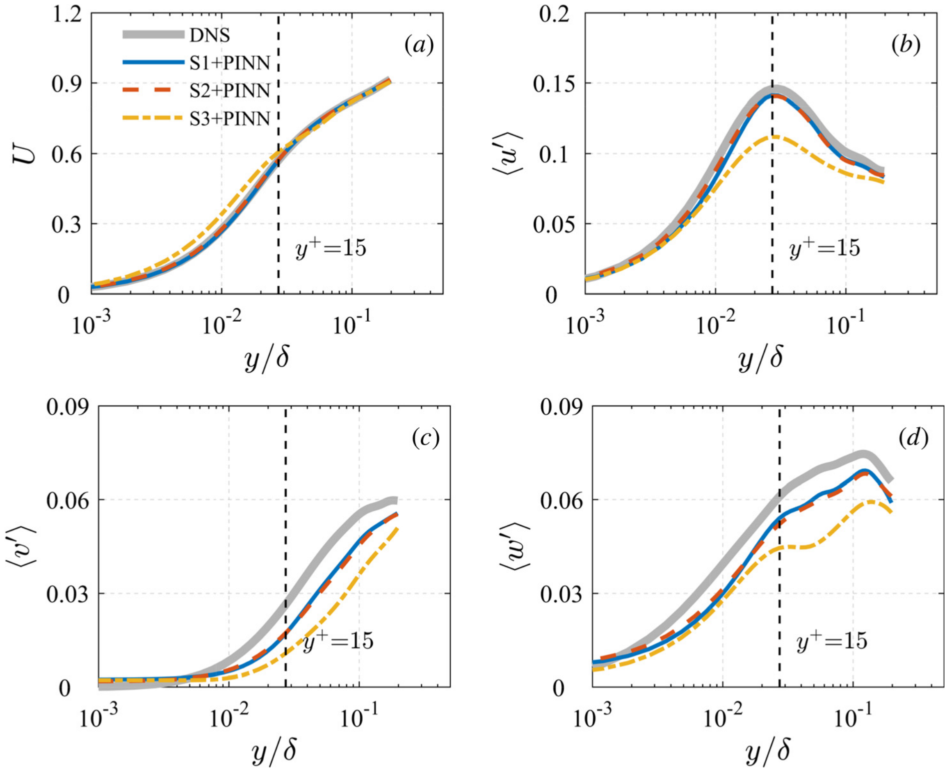


FIG. 12. Statistical profiles obtained by the PINN with different kinds of data points. (a) Mean streamwise velocity U, (b) streamwise velocity fluctuation $\langle u' \rangle$, (c) wall-normal velocity fluctuation $\langle v' \rangle$, and (d) spanwise velocity fluctuation $\langle w' \rangle$. The black dashed lines indicate the location of $y^+ = 15$.

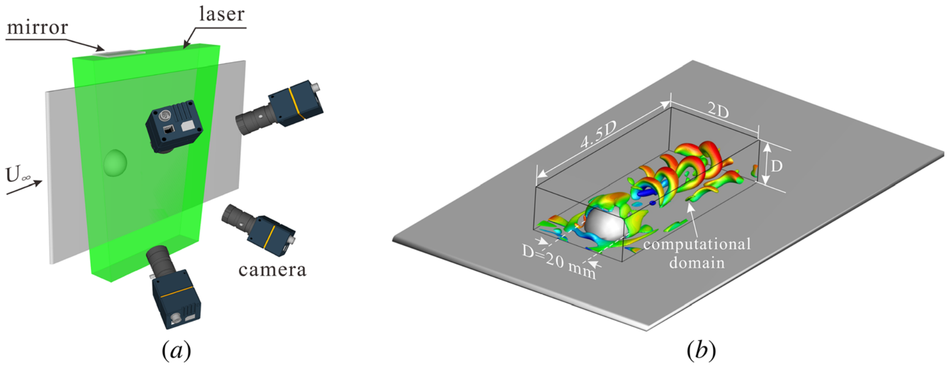


FIG. 13. (a) Experimental setup for applying Tomo-PIV to the 3D wake flow of a hemisphere. A mirror is arranged ahead of the laser to illustrate the shadow region behind the hemisphere. (b) Dimensions of the computational domain, which is smaller than the measurement region.

the characteristic frequency of the wake is ~10 Hz. The Stokes number is 2.2×10^{-4}, which indicates that the tracer particles can fully follow the main flow motion. The motion of the tracers was recorded by four Photron high-speed CMOS cameras (Fastcam SA2) with a resolution of 2048 × 1024 pixels at a frequency of 250 Hz. The cameras were laterally placed to form a "×" shape, and the maximum view angle between two cameras was approximately 45°. The 3 D particle distribution was reconstructed using the intensity-enhanced multiplicative algebraic reconstruction technique (MART, Wang *et al.*, 2016). Keane and Adrian (1992) showed that for 8 particle image pairs, the valid detection probability in PIV measurements exceeds 95%. Considering the measurement noise, therefore, a final interrogation volume size of 48 × 48 × 48 voxels at 75% overlap was used to estimate the displacement. This size yields approximately 17 particles in the interrogation volume based on the particle concentration of 0.05 ppp. The particle image size and the histogram of the estimated displacement were also checked. We are certain that no peak locking is observed. The velocity fields were validated using a normalized median test and smoothed using a Gaussian filter. The vector spacing was 0.768 mm = 0.0384D along with all three directions.

TABLE III. Summary of the flow parameters and Tomo-PIV parameters.

Flow parameters	Freestream velocity U_∞	145 mm/s
	Hemisphere diameter D	20 mm
	Water temperature T	18 °C
	Reynolds number Re_D	2750
Tomo-PIV parameters	Laser energy	30 mJ/pulse
	Laser thickness	20 mm
	Laser wavelength	532 nm
	Number of cameras	4
	Image size	2048 × 1024 pixels
	Digital resolution	16 pixels/mm
	Particle diameter	5–20 μm
	Particle concentration	~0.05 ppp
	Acquisition frequency	250 Hz
	Reconstruction algorithm	Intensity-enhanced MART (Wang *et al.*, 2016)
	Reconstruction volume size	110 × 46 × 20 mm^3
	Interrogation volume size	48 × 48 × 48 voxels
	Interrogation volume overlap	75%
	Vector spacing	0.768 mm = 0.0384D

The volume with a size of $4.5D \times 2.0D \times 1.0D$ was tackled by the PINN, as shown in Fig. 13(b); this computational domain is slightly smaller than the reconstruction volume size. Notably, the hemisphere is included in the computational domain. This implies that the no-slip boundary condition is considered in the PINN. Table IV lists the parameters of the PINN for Tomo-PIV. This configuration yields a total of 923 411 trainable variables. The weight coefficient α was set to 1. Ten million Tomo-PIV data points, five million boundary conditions, and twelve million equation points were randomly extracted from 500 sequential snapshots. These data were used to train the network for 10 000 epochs. The batch size and the iteration steps per epoch were set to 5000 and 10, respectively. After training, the network can be used to predict the velocity at any location within the volume. In the present case, the wall-normal location of the predicted velocity starts from 0, and the vector spacing is set to half that of the original Tomo-PIV data. This implies that the resolution is improved by a factor of 2 in all directions.

Figure 14 shows the comparison between the original Tomo-PIV data (top), the PINN results (middle) and the DNS results (bottom) at two time instants. The $Q = 0.2$ isosurfaces are colored based on the streamwise velocity. The semitransparent isosurfaces indicate a streamwise velocity of u/U_∞=0.5. The velocity and coordinates are normalized by the freestream velocity U_∞ and the hemisphere diameter D, and the time is normalized by $t^* = D/U_\infty$. DNS was carried out to simulate the experimental situation. The computational domain size was large, with dimensions of $L_x \times L_y \times L_z = 120D \times 40D \times 35D$, where approximately one hundred million nodes were employed to resolve the flow motion. The points were closely concentrated near the wall. An in-house computational fluid dynamics solver (Liu *et al.*, 2018; Liu, 2020; Liu *et al.*, 2021) was used to compute the velocity and pressure. For the raw experimental data, the flow structures are contaminated by strong measurement noise, and only the

TABLE IV. Summary of the parameters of the PINN for Tomo-PIV.

N_{eqns}	N_{data}	N_{cell}	N_{layer}	Trainable params
1.2×10^7	1×10^7	256	15	923 411

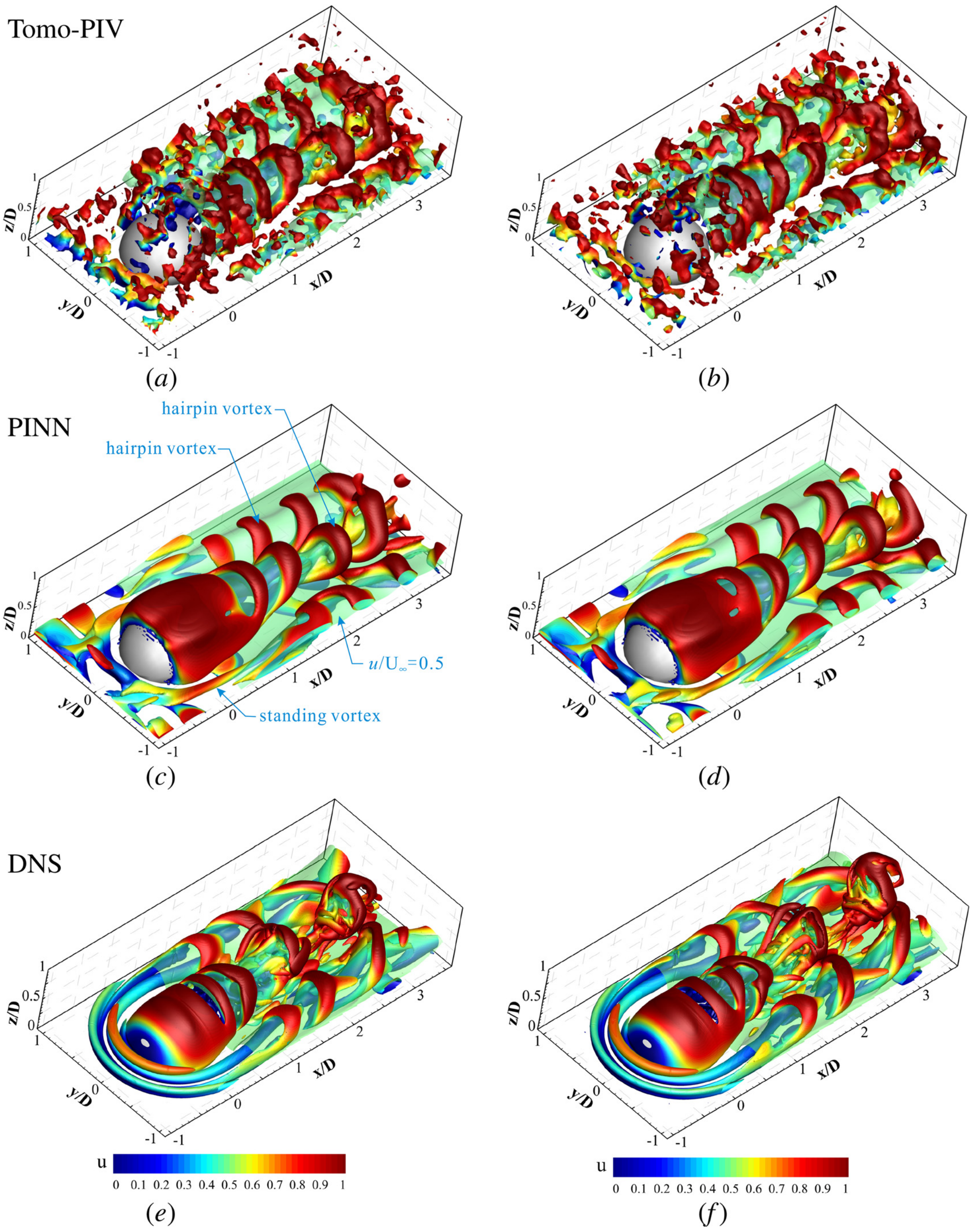


FIG. 14. Instantaneous flow fields of the raw Tomo-PIV data (a) and (b), PINN results (c) and (d), and DNS results (e) and (f). For Tomo-PIV and the PINN, the time instants of the left plane (a, c) and right plane (b) and (d) are 2.90t^* and 3.48t^*, respectively. The time instants of DNS are different from those of Tomo-PIV and the PINN, and the time interval between (f) and (e) is 0.5t^*. The isosurfaces colored by the streamwise velocity u represent vortical structures identified by the Q-criterion with $Q = 0.2$. The semi-transparent isosurfaces indicate a streamwise velocity of $u/U_\infty = 0.5$.

primary hairpin vortices shedding from the hemisphere can be clearly identified. After being processed by the PINN, the quality of the flow fields is improved to the level of the numerical simulation, although the small-scale vortical structures in the DNS data are not reconstructed by the PINN. The standing vortex (also called horseshoe vortex) around the hemisphere is distinctly reconstructed by the PINN, while it is missing in the raw Tomo-PIV data. The hairpin vortices also present a perfect arch shape instead of coarse and broken

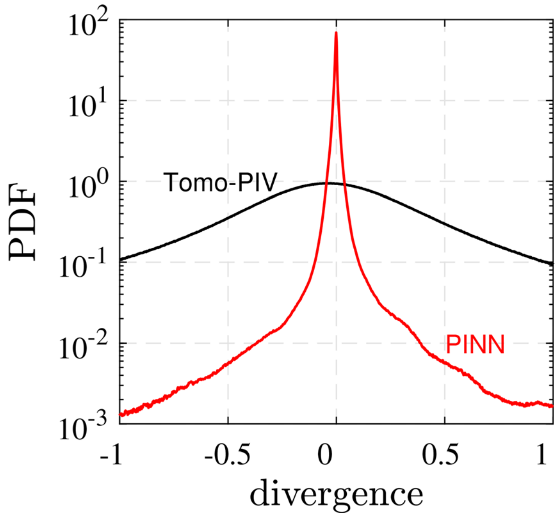


FIG. 15. PDFs of the divergence for the original Tomo-PIV data and the fields reconstructed by the PINN.

vortices, as shown in Figs. 14(a) and 14(b). The primary hairpin vortices behind the hemisphere are created by the 3D separation of the flow over the hemisphere, while the formation of the secondary hairpin vortices aside from the hemisphere is closely related to the horseshoe vortex. The dynamics of vortex formation were described by Acarlar and Smith (1987) and Cao and Tamura (2020). We also plot the probability density functions (PDFs) of the divergence for the original Tomo-PIV data and the fields reconstructed by the PINN in Fig. 15. The maximum PDF of the PINN is almost two orders of magnitude higher than that of the raw data.

In addition to the ability of the PINN to interpolate sparse data into a structured mesh with a high resolution, another advantage of the PINN is that it can automatically solve for the pressure fields without being provided with any information about the pressure. Figure 16 presents the contours of pressure at the slice of $y = 0$. The green isosurfaces represent vortical structures of $Q = 1$. In a vortex, the centrifugal force is balanced by the pressure force (Jeong and Hussain, 1995); therefore, the low-pressure region should be consistent with the vortex core, as shown in Fig. 16. Additionally, a high-pressure region exists before the hemisphere since the flow is stopped by the hemisphere, and a low-pressure region appears behind the hemisphere due to flow separation. We also computed the pressure from the raw Tomo-PIV data by using the Poisson equation (not shown in this paper), and the pressure is totally deteriorated by the high-level measurement noise, which should be suppressed by a spatial–temporal filter (Wang *et al.*, 2018; He *et al.*, 2020). From visual inspection of the results, we can conclude that the velocity field reconstructed by the PINN has a great improvement over the original Tomo-PIV data, and the PINN has the ability to reconstruct the pressure field.

V. CONCLUSIONS

As a novel DA method, the PINN can reconstruct a dense velocity field from sparse experimental data with the constraint of N–S equations. Within the PINN, the computational fluid dynamics solver is replaced by a network, and the loss consisting of the residual of data

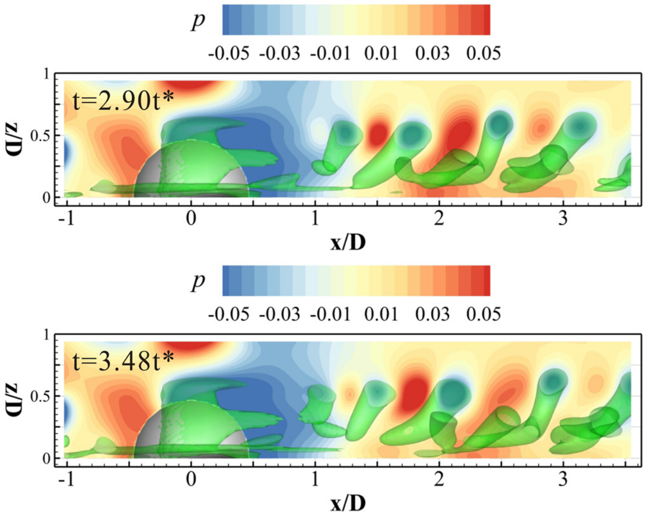


FIG. 16. Contour maps of pressure estimated by the PINN at the slice of $y = 0$. The time instants of the top plane and bottom plane are 2.90t^* and 3.48t^*, respectively. The green isosurfaces represent vortical structures identified by the Q-criterion with $Q = 1$.

$\mathscr{L}_{data}$ and the residual of the equations $\mathscr{L}_{eqns}$ is minimized by an optimizer. In the present work, the performance of the PINN is investigated using 2D Taylor's decaying vortices, turbulent channel flow, and an experiment on the 3D wake flow of a hemisphere to address the question of whether the PINN is suitable for noisy PIV/PTV data. The relevant discussions and conclusions are summarized as follows.

First, the basic properties of the proposed approach are explored using 2D Taylor's decaying vortices with and without noise. The PINN uses a network to approximate or fit the complex flow fields; therefore, the activation function plays a critical role in the nonlinear approximation. The adaptive swish function is adopted since it has outstanding performance. To minimize the loss function, two optimizers, Adam and BFGS, are investigated. The BFGS optimizer is not adopted in the present work since it is only practical for small-scale ML applications, although it converges faster than Adam. We also consider the influences of the number of data points N_{data}, the number of neurons in each layer N_{cell} and weight coefficient α on the performance of the PINN. In general, increasing N_{data} and N_{cell} can improve the accuracy of the prediction by the PINN. With noisy data, the optimal α is 1 according to our test. Only when the noise level is very high, it is valid to suppress the noise by increasing α.

Second, the velocity field of wall turbulence is reconstructed by the present method. Wall turbulence is used in this case because the PINN has difficulty reconstructing inhomogeneous and anisotropic fields. Three kinds of data points are generated to simulate experimental data, including the regular velocity for PIV, random velocity for PTV, and regular velocity with spatially correlated Gaussian random noise. The ability of the proposed approach to reconstruct complex flows such as turbulence is of great significance, although large scales are much easier to reconstruct than small scales. We also found that the spatially correlated noise introduced by the cross-correlation algorithm in PIV can result in a large deviation for the PINN results. Therefore, PTV data with high quality and resolution are more suitable for the PINN.

Third, one great advantage is that the PINN can provide pressure fields. Numerical assessment on 2D Taylor's decaying vortices shows that the error in pressure is approximately one order of magnitude larger than that in velocity. In this case, the relative streamwise velocity L_2 error can be reduced to less than 1% by the PINN, and the corresponding pressure error is approximately 10%. How to further improve the pressure accuracy for the PINN is worth much deeper research. The experimental Tomo-PIV velocity in the 3D wake flow of a hemisphere is optimized by the PINN. The primary hairpin vortices, secondary hairpin vortices, and standing vortices can be clearly identified after being processed by the PINN. The pressure around the hemisphere is simultaneously obtained and is qualitatively correct. With this approach, the force on an object can be estimated.

Dense velocity reconstruction from PIV/PTV data can be implemented by using the PINN. One of the most important limitations of the PINN is that the loss function converges slowly, especially for a large-scale network. Further accelerating the training process and improving the accuracy of the PINN are critical for practical applications.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (NSFC) Basic Science Center Program for "Multiscale Problems in Nonlinear Mechanics" Grant No. 11988102 and by NSFC Grant No. 12072348.

AUTHOR DECLARATIONS

Conflict of Interest

We declare that we have no conflicts of interest to disclose.

DATA AVAILABILITY

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

REFERENCES

Acarlar, M. S. and Smith, C. R., "A study of hairpin vortices in a laminar boundary layer. Part 1. Hairpin vortices generated by a hemisphere protuberance," J. Fluid Mech. **175**, 1–41 (1987).

Adamczyk, A. A. and Rimai, L., "2-dimensional particle tracking velocimetry (PTV): Technique and image processing algorithms," Exp. Fluids **6**, 373–380 (1988).

Adrian, R. J., "Scattering particle characteristics and their effect on pulsed laser measurements of fluid flow: Speckle velocimetry vs particle image velocimetry," Appl. Opt. **23**, 1690–1691 (1984).

Adrian, R. J., "Twenty years of particle image velocimetry," Exp. Fluids **39**, 159–169 (2005).

Adrian, R. J., Meinhart, C. D., and Tomkins, C. D., "Vortex organization in the outer region of the turbulent boundary layer," J. Fluid Mech. **422**, 1–54 (2000).

Arzani, A., Wang, J.-X., and D'Souza, R. M., "Uncovering near-wall blood flow from sparse data with physics-informed neural networks," Phys. Fluids **33**, 071905 (2021).

Azijli, I. and Dwight, R. P., "Solenoidal filtering of volumetric velocity measurements using Gaussian process regression," Exp. Fluids **56**, 198 (2015).

Bollapragada, R., Mudigere, D., Nocedal, J., Shi, H.-J. M., and Tang, P. T. P., "A progressive batching L-BFGS method for machine learning," arXiv:1802.05374 (2018).

Brücker, C., "Digital-Particle-Image-Velocimetry (DPIV) in a scanning light-sheet: 3D starting flow around a short cylinder," Exp. Fluids **19**, 255–263 (1995).

Cai, S., Mao, Z., Wang, Z., Yin, M., and Karniadakis, G. E., "Physics-informed neural networks (PINNs) for fluid mechanics: A review," arXiv:2105.09506 (2021a).

Cai, S., Wang, Z., Fuest, F., Jeon, Y. J., Gray, C., and Karniadakis, G. E., "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 (2021b).

Cao, Y. and Tamura, T., "Large-eddy simulation study of Reynolds number effects on the flow around a wall-mounted hemisphere in a boundary layer," Phys. Fluids **32**, 025109 (2020).

Chandramouli, P., Memin, E., Heitz, D., and Fiabane, L., "Fast 3D flow reconstructions from 2D cross-plane observations," Exp. Fluids **60**, 30 (2019).

Cierpka, C. and Kähler, C. J., "Particle imaging techniques for volumetric three-component (3D3C) velocity measurements in microfluidics," J. Visualization **15**, 1–31 (2012).

de Kat, R. and van Oudheusden, B. W., "Instantaneous planar pressure determination from PIV in turbulent flow," Exp. Fluids **52**, 1089–1106 (2012).

de Silva, C. M., Baidya, R., and Marusic, I., "Enhancing Tomo-PIV reconstruction quality by reducing ghost particles," Meas. Sci. Technol. **24**, 024010 (2013).

Deng, Z. W., He, C. X., Liu, Y. Z., and Kim, C. K., "Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework," Phys. Fluids **31**, 125111 (2019).

Discetti, S. and Coletti, F., "Volumetric velocimetry for fluid flows," Meas. Sci. Technol. **29**, 042001 (2018).

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

Elsinga, G. E., Scarano, F., Wieneke, B., and van Oudheusden, B. W., "Tomographic particle image velocimetry," Exp. Fluids **41**, 933–947 (2006).

Elsinga, G. E. and Tokgoz, S., "Ghost hunting-an assessment of ghost particle detection and removal methods for tomographic-PIV," Meas. Sci. Technol. **25**, 084004 (2014).

Elsinga, G. E., Westerweel, J., Scarano, F., and Novara, M., "On the velocity of ghost particles and the bias errors in tomographic-PIV," Exp. Fluids **50**, 825–838 (2011).

Ethier, C. R. and Steinman, D. A., "Exact fully 3D Navier-Stokes solutions for benchmarking," Int. J. Numer. Methods Fluids **19**, 369–375 (1994).

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

Ganapathisubramani, B., Longmire, E. K., Marusic, I., and Pothos, S., "Dual-plane PIV technique to determine the complete velocity gradient tensor in a turbulent boundary layer," Exp. Fluids **39**, 222–231 (2005).

Gao, H., Sun, L., and Wang, J.-X., "Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels," Phys. Fluids **33**, 073603 (2021).

Gao, Q., Wang, H. P., and Shen, G. X., "Review on development of volumetric particle image velocimetry," Chin. Sci. Bull. **58**, 4541–4556 (2013).

Gesemann, S., Huhn, F., Schanz, D., and Schrder, A., "From noisy particle tracks to velocity, acceleration and pressure fields using B-splines and penalties," in International Symposium on Applications of Laser Techniques to Fluid Mechanics (2016).

Gillissen, J. J. J., Bouffanais, R., and Yue, D. K. P., "Data assimilation method to de-noise and de-filter particle image velocimetry data," J. Fluid Mech. **877**, 196–213 (2019).

He, C. X., Liu, Y. Z., and Gan, L., "Instantaneous pressure determination from unsteady velocity fields using adjoint-based sequential data assimilation," Phys. Fluids **32**, 035101 (2020).

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

Hinsch, K. D., "Holographic particle image velocimetry," Meas. Sci. Technol. **13**, R61–R72 (2002).

Hornik, K., Stinchcombe, M., and White, H., "Multilayer feedforward networks are universal approximators," Neural Networks **2**, 359–366 (1989).

Humble, R. A., Elsinga, G. E., Scarano, F., and van Oudheusden, B. W., "Three-dimensional instantaneous structure of a shock wave/turbulent boundary layer interaction," J. Fluid Mech. **622**, 33–62 (2009).

Hutchins, N. and Marusic, I., "Evidence of very long meandering features in the logarithmic region of turbulent boundary layers," J. Fluid Mech. **579**, 1–28 (2007).

Jagtap, A. D., Kawaguchi, K., and Karniadakis, G. E., "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," J. Comput. Phys. **404**, 109136 (2020).

Jeong, J. and Hussain, F., "On the identification of a vortex," J. Fluid Mech. **285**, 69–94 (1995).

Jin, X., Cai, S., Li, H., and Karniadakis, G. E., "NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations," J. Comput. Phys. **426**, 109951 (2021).

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L., "Physics-informed machine learning," Nat. Rev. Phys. **3**, 422 (2021).

Keane, R. D. and Adrian, R. J., "Theory of cross-correlation analysis of PIV images," Appl. Sci. Res. **49**, 191–215 (1992).

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

Kim, J. and Moin, P., "Application of a fractional-step method to incompressible Navier-Stokes equations," J. Comput. Phys. **59**, 308–323 (1985).

Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," arXiv:1412.6980 (2014).

Ledoux, H. and Gold, C., "An efficient natural neighbour interpolation algorithm for geoscientific modelling," in *Developments in Spatial Data Handling* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005), pp. 97–108.

Lee, S.-K., Lee, H., Back, J., An, K., Yoon, Y., Yum, K., kim, S., and Hwang, S.-U., "Prediction of tire pattern noise in early design stage based on convolutional neural network," Appl. Acoust. **172**, 107617 (2021).

Lemke, M. and Sesterhenn, J., "Adjoint-based pressure determination from PIV data in compressible flows—Validation and assessment based on synthetic data," Eur. J. Mech.-B/Fluids **58**, 29–38 (2016).

Liu, B., Tang, J. P., Huang, H. B., and Lu, X. Y., "Deep learning methods for super-resolution reconstruction of turbulent flows," Phys. Fluids **32**, 025105 (2020).

Liu, D. C. and Nocedal, J., "On the limited memory BFGS method for large scale optimization," Math. Programming **45**, 503–528 (1989).

Liu, Y., "Turbulence modeling and its uncertainty quantification for complex aerodynamic flows," Ph.D. thesis (Northwestern Polytechnical University, Xi'an, China, 2020).

Liu, Y., Wang, G., and Ye, Z., "Dynamic mode extrapolation to improve the efficiency of dual time stepping method," J. Comput. Phys. **352**, 190–212 (2018).

Liu, Y., Zhou, Z., Zhu, L., and Wang, S., "Numerical investigation of flows around an axisymmetric body of revolution by using Reynolds-stress model based hybrid Reynolds-averaged Navier-Stokes/large eddy simulation," Phys. Fluids **33**, 085115 (2021).

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E., "DeepXDE: A deep learning library for solving differential equations," SIAM Rev. **63**, 208–228 (2021).

Maas, H. G., Gruen, A., and Papantoniou, D., "Particle tracking velocimetry in three-dimensional flows," Exp. Fluids **15**, 133–146 (1993).

Malik, N. A., Dracos, T., and Papantoniou, D. A., "Particle tracking velocimetry in three-dimensional flows," Exp. Fluids **1515**, 279–294 (1993).

Markidis, S., "The old and the new: Can physics-informed deep-learning replace traditional linear solvers?," Front. Big Data **4**, 669097 (2021).

Mons, V., Chassaing, J. C., Gomez, T., and Sagaut, P., "Reconstruction of unsteady viscous flows using data assimilation schemes," J. Comput. Phys. **316**, 255–280 (2016).

Pan, Z., Whitehead, J., Thomson, S., and Truscott, T., "Error propagation dynamics of PIV-based pressure field calculations: How well does the pressure Poisson solver perform inherently?," Meas. Sci. Technol. **27**, 084012 (2016).

Raffel, M., Willert, C. E., Scarano, F., Kähler, C. J., Wereley, S. T., and Kompenhans, J., *Particle Image Velocimetry: A Practical Guide* (Springer, 2018).

Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686–707 (2019).

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

Ramachandran, P., Zoph, B., and Le, Q. V., "Searching for activation functions," arXiv:1710.05941 (2017).

Rao, C., Sun, H., and Liu, Y., "Physics-informed deep learning for incompressible laminar flows," Theor. Appl. Mech. Lett. **10**, 207–212 (2020).

Scarano, F., "Tomographic PIV: Principles and practice," Meas. Sci. Technol. **24**, 012001 (2013).

Schanz, D., Gesemann, S., and Schrder, A., "Shake-The-Box: Lagrangian particle tracking at high particle image densities," Exp. Fluids **57**, 1–27 (2016).

Scharnowski, S. and Kähler, C. J., "Particle image velocimetry—Classical operating rules from today's perspective," Opt. Lasers Eng. **135**, 106185 (2020).

Schneiders, J. F. G. and Scarano, F., "Dense velocity reconstruction from tomographic PTV with material derivatives," Exp. Fluids **57**, 1–22 (2016).

Schröder, A., Geisler, R., Staack, K., Elsinga, G. E., Scarano, F., Wieneke, B., Henning, A., Poelma, C., and Westerweel, J., "Eulerian and Lagrangian views of a turbulent boundary layer flow using time-resolved tomographic PIV," Exp. Fluids **50**, 1071–1091 (2011).

Sciacchitano, A., Dwight, R., and Scarano, F., "Navier-Stokes simulations in gappy PIV data," Exp. Fluids **53**, 1421–1435 (2012).

Shao, S., Mallery, K., Kumar, S. S., and Hong, J., "Machine learning holography for 3D particle field imaging," Opt. Express **28**, 2987–2999 (2020).

Shi, S., Ding, J., Atkinson, C., Soria, J., and New, T. H., "A detailed comparison of single-camera light-field PIV and tomographic PIV," Exp. Fluids **59**, 46 (2018).

Shi, S., Ding, J., New, T. H., and Soria, J., "Light-field camera-based 3D volumetric particle image velocimetry with dense ray tracing reconstruction technique," Exp. Fluids **58**, 78 (2017).

Symon, S., Dovetta, N., McKeon, B. J., Sipp, D., and Schmid, P. J., "Data assimilation of mean velocity from 2D PIV measurements of flow over an idealized airfoil," Exp. Fluids **58**, 61 (2017).

Taylor, G. I., "LXXV. On the decay of vortices in a viscous fluid," London, Edinburgh, Dublin Philos. Mag. J. Sci. **46**, 671–674 (1923).

Uddin, G. M., Arafat, S. M., Kazim, A. H., Farhan, M., Niazi, S. G., Hayat, N., Zeid, I., and Kamarthi, S., "Artificial intelligence-based Monte-Carlo numerical simulation of aerodynamics of tire grooves using computational fluid dynamics," Artif. Intell. Eng. Des., Anal. Manuf. **33**, 302–316 (2019).

Wang, C., Gao, Q., Wang, J., Wang, B., and Pan, C., "Experimental study on dominant vortex structures in near-wall region of turbulent boundary layer based on tomographic particle image velocimetry," J. Fluid Mech. **874**, 426–454 (2019).

Wang, H.-P., Gao, Q., Wang, S.-Z., Li, Y.-H., Wang, Z.-Y., and Wang, J.-J., "Error reduction for time-resolved PIV data based on Navier-Stokes equations," Exp. Fluids **59**, 149 (2018).

Wang, H. P., Gao, Q., Wei, R. J., and Wang, J. J., "Intensity-enhanced MART for tomographic PIV," Exp. Fluids **57**, 1–19 (2016).

Wang, Z. Y., "PIV based pressure reconstruction technique and its application," Ph.D. thesis (Beihang University, Beijing, China, 2017).

Westerweel, J., Elsinga, G. E., and Adrian, R. J., "Particle image velocimetry for complex and turbulent flows," Annu. Rev. Fluid Mech. **45**, 409–436 (2013).

Wieneke, B., "Iterative reconstruction of volumetric particle distribution," Meas. Sci. Technol. **24**, 024008 (2013).

Willert, C. E. and Gharib, M., "Three-dimensional particle imaging with a single camera," Exp. Fluids **12**, 353–358 (1992).

Xu, H., Zhang, W., and Wang, Y., "Explore missing flow dynamics by physics-informed deep learning: The parameterized governing systems," Phys. Fluids **33**, 095116 (2021).

Yang, Y., Robinson, C., Heitz, D., and Memin, E., "Enhanced ensemble-based 4DVar scheme for data assimilation," Comput. Fluids **115**, 201–210 (2015).

Ye, Q., Schrijer, F. F. J., and Scarano, F., "Boundary layer transition mechanisms behind a micro-ramp," J. Fluid Mech. **793**, 132–161 (2016).

Zhang, L., Cheng, L., Li, H., Gao, J., Yu, C., Domel, R., Yang, Y., Tang, S., and Liu, W. K., "Hierarchical deep-learning neural networks: Finite elements and beyond," Comput. Mech. **67**, 207–230 (2021).

Zhang, X. L., Xiao, H., Gomez, T., and Coutier-Delgosha, O., "Evaluation of ensemble methods for quantifying uncertainties in steady-state CFD applications with small ensemble sizes," Comput. Fluids **203**, 104530 (2020).

Zhou, Z. D., He, G. W., Wang, S. Z., and Jin, G. D., "Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network," Comput. Fluids **195**, 104319 (2019).

Zhou, Z. D., He, G. W., and Yang, X. L., "Wall model based on neural networks for LES of turbulent flows over periodic hills," Phys. Rev. Fluids **6**, 054610 (2021).