# Time-averaged flow field reconstruction based on a multifidelity model using physics-informed neural network (PINN) and nonlinear information fusion

Received 3 May 2023
Revised 1 September 2023
Accepted 21 September 2023

En-Ze Rui, Guang-Zhi Zeng, Yi-Qing Ni, Zheng-Wei Chen and Shuo Hao

*Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, China*

## Abstract

**Purpose** – Current methods for flow field reconstruction mainly rely on data-driven algorithms which require an immense amount of experimental or field-measured data. Physics-informed neural network (PINN), which was proposed to encode physical laws into neural networks, is a less data-demanding approach for flow field reconstruction. However, when the fluid physics is complex, it is tricky to obtain accurate solutions under the PINN framework. This study aims to propose a physics-based data-driven approach for time-averaged flow field reconstruction which can overcome the hurdles of the above methods.

**Design/methodology/approach** – A multifidelity strategy leveraging PINN and a nonlinear information fusion (NIF) algorithm is proposed. Plentiful low-fidelity data are generated from the predictions of a PINN which is constructed purely using Reynold-averaged Navier–Stokes equations, while sparse high-fidelity data are obtained by field or experimental measurements. The NIF algorithm is performed to elicit a multifidelity model, which blends the nonlinear cross-correlation information between low- and high-fidelity data.

**Findings** – Two experimental cases are used to verify the capability and efficacy of the proposed strategy through comparison with other widely used strategies. It is revealed that the missing flow information within the whole computational domain can be favorably recovered by the proposed multifidelity strategy with use

The work was supported by a grant from the Research Grants Council (RGC) of the Hong Kong Special Administrative Region (SAR), China (Grant No. PolyU 152308/22E) and grants from The Hong Kong Polytechnic University (Grant No. 1-WZ0C, 1-BD23). The authors also appreciate the funding support by the Innovation and Technology Commission of Hong Kong SAR Government to the Hong Kong Branch of National Engineering Research Center on Rail Transit Electrification and Automation (Grant No. K-BBY1).

*Data availability statement*: The data that support the findings of this study are available from the corresponding author upon reasonable request.

*Declaration of conflict of interest*: The authors declare that there is no conflict of interest.

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 34 No. 1, 2024
pp. 131-149
Emerald Publishing Limited
0961-5539
DOI 10.1108/HFF-05-2023-0239

of sparse measurement/experimental data. The elicited multifidelity model inherits the underlying physics inherent in low-fidelity PINN predictions and rectifies the low-fidelity predictions over the whole computational domain. The proposed strategy is much superior to other contrastive strategies in terms of the accuracy of reconstruction.

**Originality/value** – In this study, a physics-informed data-driven strategy for time-averaged flow field reconstruction is proposed which extends the applicability of the PINN framework. In addition, embedding physical laws when training the multifidelity model leads to less data demand for model development compared to purely data-driven methods for flow field reconstruction.

**Keywords** Flow field, Reconstruction, Physics-informed neural network, Multifidelity model, Nonlinear information fusion

**Paper type** Research paper

## 1. Introduction

Investigating the flow field around structures has become a major task in wind engineering nowadays, as it directly affects the pressure distribution on structure surfaces (Li *et al.*, 2019; Gao *et al.*, 2021; Liu *et al.*, 2022) and the thermal environment in their surroundings (Tian *et al.*, 2020; Nugroho *et al.*, 2022). Relevant studies have been carried out on investigating the flow fields around civil structures such as buildings (Flaga *et al.*, 2018; Kim *et al.*, 2021), bridges (Chen *et al.*, 2016; Zhou *et al.*, 2018; He *et al.*, 2019) and high-speed rail tunnels (Chen *et al.*, 2017a; Chen *et al.*, 2017b; Li *et al.*, 2022). Numerical simulations based on computational fluid dynamics (CFD) techniques were widely used in previous works to obtain the flow field (Liu *et al.*, 2018; Ntinas *et al.*, 2018; Chen *et al.*, 2022). However, as a side effect of the high-accuracy requirements, the CFD modeling is highly time-consuming and expensive to implement (Ding *et al.*, 2019; Wakes *et al.*, 2021). Also, additional computational resources are needed for parametric analysis and model optimization (Penwarden *et al.*, 2022) during the numerical simulations. Thus, an important issue in constructing flow fields is to seek a balance between accuracy and cost.

To achieve this compromise, numerous investigations have been conducted on reconstructing the flow field within the entire computational domain using a small sampling of high-fidelity data. For example, Abrahamson and Lonnes (1995) adopted the least-squares method to reconstruct the vorticity fields based on direct numerical simulation (DNS) data. Although the result of the least-squares method is close to the averaged field, it ignores details of local flow features, and, therefore, is not conducive to local characteristic analysis of the flow field. In addition, this method requires thousands of high-fidelity data points to reconstruct a flow field (Kim and Moon, 2022), which remains a heavy burden in engineering practice.

Flow field reconstruction can also be achieved by incorporating machine learning algorithms such as various neural network paradigms (Kong *et al.*, 2021; Pruvost *et al.*, 2001) rebuilt the complex flow features and mean velocity components in the flow field based on a radial basis function neural network using sparse experimental data. Nevertheless, as the structure and flow field become more complex, the reconstruction of flow characteristics also requires a huge cost to acquire sufficient amounts of data (Liu *et al.*, 2011; Ladický *et al.*, 2015; Löhner *et al.*, 2021). In addition, as is well known, the internal flow within the flow field is controlled by the Navier–Stokes (NS) equations, which means the flow characteristics are restrained by the underlying physical rules (Sun *et al.*, 2018; Ershkov *et al.*, 2021). However, a purely data-driven approach for flow field reconstruction may yield results that are even contrary to some basic physical principles (Lu *et al.*, 2021).

Machine learning, especially deep learning, has been rapidly evolving over the last couple of decades and has begun to play a predominant role across a variety of scientific disciplines (Hu *et al.*, 2020; Eivazi *et al.*, 2022; Jahromi *et al.*, 2022). Due to its groundbreaking approximation capabilities in solving partial differential equations (PDEs) (Xiang *et al.*, 2022), deep learning has become increasingly popular in the field of fluid dynamics. One of the deep learning methods which enables flow field reconstruction is the physics-informed neural network (PINN) proposed by Raissi *et al.* (2019). PINN recognizes the laws of physics when solving PDEs, which breaks the current deadlock of conventional machine learning methods that are confined to data-driven modeling (Lu *et al.*, 2021; Sun *et al.*, 2021). In other words, PINN frees machine learning methods from their dependence on a large demand for experimental or field-measured data (Rao *et al.*, 2020), thus, offering a promising alternative to solving real-world problems (Rui *et al.*, 2023; Hasanuzzaman *et al.*, 2023). In addition, as aforementioned, extrapolation or observation bias may lead to poor generalization performance for purely data-driven models, and as a result, predictions by such models may not be physically consistent. PINN, on the other hand, uses the residuals of the physical governing equations to form a loss function for neural network training, which serves as a penalty to restrict the space of feasible solutions. PINN can also combine traditional physical models with sparse high-fidelity measurements to reconstruct flow fields (Karpatne *et al.*, 2017; Eivazi and Vinuesa, 2022), which has been a research hotspot and attracted tremendous attention in recent years (Wang *et al.*, 2021; Wang *et al.*, 2022). For instance, Arzani *et al.* (2021) applied the PINN strategy to near-wall blood flow reconstruction, which incorporated the fluid governing functions and sparse internal data into the loss function. The numerical results demonstrated the consistency of PINN predictions with the results from conventional CFD methods. Jin *et al.* (2021) established PINN models to reconstruct laminar and turbulent flows based on NS equations and sparse velocity data on domain boundaries. These PINN-related investigations demonstrate its ability in flow field reconstruction, as well as its generalization capability with fewer training data while remaining accurate. Nevertheless, the limitations of this approach also result in an impediment to the development of its engineering applications, including computational cost and difficulty in dealing with complex problems (Goswami *et al.*, 2020).

To overcome the disadvantages of the above methods, a multifidelity physics-informed data-driven strategy for time-averaged flow field reconstruction is proposed in this article. First, an approximate low-fidelity flow field is obtained from the PINN prediction, which uses Reynold-averaged NS (RANS) equation as the governing function. The RANS equation is the time-averaged form of the NS equation. Central to it is Reynolds decomposition, which separates a transient flow quantity into a time-averaged component and a fluctuating component. It transforms the unsteady turbulent problem into a steady problem, which greatly reduces the computational cost. In engineering practice, the RANS equation has already become the most commonly used fluid governing equation. Second, sparse field or experimental measurements are used as high-fidelity observations. The nonlinear information fusion (NIF) algorithm proposed by Perdikaris *et al.* (2017) is then adopted to establish a multifidelity Gaussian process (GP) model for flow field reconstruction. The nonlinear cross-correlations between low-fidelity approximations and high-fidelity observations are extracted to train the multifidelity GP model and make high-fidelity predictions using the NIF algorithm. Two case studies regarding time-averaged flow field reconstruction are presented to verify the feasibility of the proposed method, including a flow past a hill and a flow past a square cylinder. We also compare the performance of the proposed strategy with other commonly used methods. The results demonstrate that the multifidelity model has superior accuracy in approximating measurement data in the two cases.

The proposed strategy greatly extends the applicability of the physics-based PINN framework, where the PINN model is only used for low-fidelity modeling which requires less accuracy. Furthermore, the embedded physical laws provide essential guidance in multifidelity modeling which results in less data demand in flow field reconstruction compared with purely data-driven methods. The remainder of this article is organized as follows. Section 2 presents the PINN methodology for solving RANS equations and the NIF algorithm for multifidelity modeling. In Section 3, we demonstrate the performance of our proposed strategy for flow field reconstruction in two two-dimensional (2 D) turbulent flow cases. Main conclusions are drawn in Section 4.

## 2. Methodology

### *2.1 Physics-informed neural network for solving Reynold-averaged Navier–Stokes equations*

In this study, PINN is constructed to yield approximate solutions to the 2D RANS equations. The RANS equations for simulating turbulent flows are expressed as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u_i) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + \frac{\partial}{\partial x_j}\left(-\rho\overline{u'_i u'_j}\right) \tag{2}$$

where $\rho$ is the density of the fluid, $\mu$ is the laminar viscosity, $u_i$ is the time-averaged velocity component in the $x_i$ direction, $p$ is the time-averaged pressure and $-\rho\overline{u'_i u'_j}$ is the Reynolds stress.

PINN is a kind of deep neural network that takes the residual of physical constraints as the loss function. During the calculation, there will be thousands of collocation points scattered inside or on the boundaries of the computational domain of the governing equations, which are used to compute the residuals of the governing equations and boundary conditions. These residuals are further embedded in the loss function of the neural network, and they tend to converge toward zero during the training process with the aid of an optimizer. The schematic diagram of PINN that we construct for flow field reconstruction is depicted in Figure 1. The left part of the PINN is a fully connected neural network that

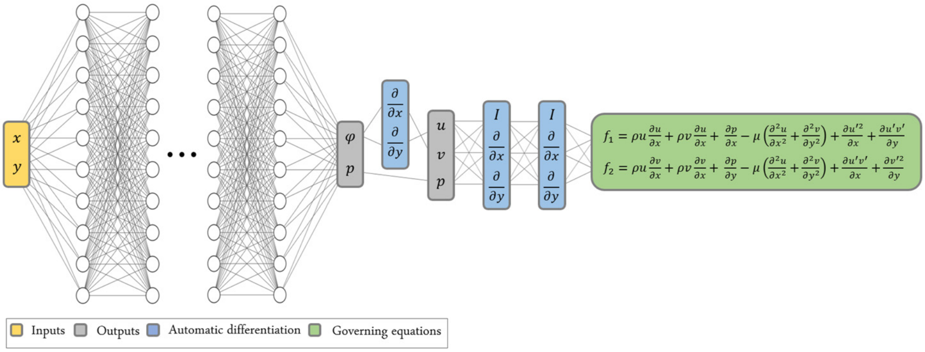

x
y
...
φ
p
∂/∂x
∂/∂y
u
v
p
I
∂/∂x
∂/∂y
I
∂/∂x
∂/∂y
$f_1 = \rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial x} + \frac{\partial p}{\partial x} - \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \frac{\partial u'^2}{\partial x} + \frac{\partial u'v'}{\partial y}$
$f_2 = \rho u\frac{\partial v}{\partial x} + \rho v\frac{\partial v}{\partial x} + \frac{\partial p}{\partial y} - \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \frac{\partial u'v'}{\partial x} + \frac{\partial v'^2}{\partial y}$
Inputs
Outputs
Automatic differentiation
Governing equations

**Figure 1.**
PINN framework for flow field reconstruction

**Source:** Figure by authors

maps the relationship between spatial coordinates ($x$, $y$) and the characteristics of the flow field ($\varphi$, $p$), here $\varphi$ represents the stream function which contains the velocity information, while $p$ represents the pressure.

It is noteworthy that here we adopt stream function instead of velocity to force the fluid continuity equation as a hard constraint. The relationship between the stream function and fluid velocity components is described as:

$$u = \frac{\partial \varphi}{\partial y} \tag{3}$$

$$v = -\frac{\partial \varphi}{\partial x} \tag{4}$$

where $u$ and $v$ are the velocity components in the $x$- and $y$-directions, respectively. Using the stream function as the output of a PINN in 2D scenarios can make the fluid continuity equation a hard constraint that is compelled to satisfy in flow simulations. In the middle part of the PINN framework, *automatic differentiation* (AD) is applied to calculate the gradients of the outputs with respect to the inputs, which plays a significant role in the neural network training process (Baydin *et al.*, 2018). The right part of the PINN framework is the loss function, which is calculated as:

$$L = w_f L_f + w_b L_b \tag{5}$$

where:

$$L_f = \frac{1}{N_f} \sum_{n=1}^{N_f} \sum_{i=1}^{2} |f_i^n|^2 \tag{6}$$

$$L_b = \frac{1}{N_{nb}} \sum_{i=1}^{N_{nb}} |r_{nb}^i|^2 + \frac{1}{N_{db}} \sum_{i=1}^{N_{db}} |r_{db}^i|^2 \tag{7}$$

In the above expressions, $L_f$ and $L_b$ denote the loss components corresponding to the residuals of the governing equations and boundary conditions, respectively; $w_f$ and $w_b$ denote the weighting coefficients of the corresponding loss terms; $f_i^n$ is the residual of the $i$th governing equation as shown in Figure 1; $r_{nb}^i$ and $r_{db}^i$ are the residuals for the Neumann boundary and Dirichlet boundary, respectively; $N_f$ is the number of collocation points used to calculate the residuals of the governing equations, while $N_{nb}$ and $N_{db}$ are the numbers of collocation points used to calculate the residuals for the Neumann boundary and Dirichlet boundary, respectively. It is worth mentioning that training the PINN model requires neither labeled data nor numerical solution to the RANS equations.

### *2.2 Nonlinear information fusion algorithm for multifidelity modeling*

The NIF algorithm proposed by Perdikaris *et al.* (2017) enables us to combine low-fidelity models with small amounts of high-fidelity observations for multifidelity modeling, which mainly relies on the principled framework of GP regression. Here we define $f_h$ and $f_l$ as the

GPs that model the data on the high and low-fidelity levels, respectively. In the NIF algorithm, $f_h$ is expressed in the following form:

$$f_h(\boldsymbol{x}) = g_h\left(\boldsymbol{x}, f_{*l}(\boldsymbol{x})\right) \tag{8}$$

where $g_h \sim \mathcal{GP}\left(\boldsymbol{f_h}|0, k_h\left(\left(\boldsymbol{x}, f_{*l}(\boldsymbol{x})\right), \left(\boldsymbol{x}', f_{*l}(\boldsymbol{x}')\right); \theta_h\right)\right)$; $f_{*l}(\boldsymbol{x})$ is the GP posterior from the low-fidelity level; and $\theta_h$ is the hyperparameter. The kernel $k_h$ is a covariance kernel which can be decomposed as:

$$k_h = k_{h_\rho}\left(\boldsymbol{x}, \boldsymbol{x}'; \theta_{h_\rho}\right) \times k_{h_f}\left(f_{*l}(\boldsymbol{x}), f_{*l}(\boldsymbol{x}'); \theta_{h_f}\right) + k_{h_\delta}\left(\boldsymbol{x}, \boldsymbol{x}'; \theta_{h_\delta}\right) \tag{9}$$

Here, $k_{h_\rho}$, $k_{h_f}$ and $k_{h_\delta}$ are covariance functions which take the squared exponential form with automatic relevance determination weights (Rasmussen, 2004). $\theta_{h_\rho}$, $\theta_{h_f}$ and $\theta_{h_\delta}$ are hyperparameters. We can observe that the NIF algorithm generates the high-fidelity model as a function of the input coordinates $\boldsymbol{x}$ and the output of the low-fidelity model $f_{*l}(\boldsymbol{x})$. In other words, it jointly relates the input space and the posterior prediction of the low-fidelity model to the output of the high-fidelity model. Also, the covariance kernel in equation (9) blends the contributions of both $\boldsymbol{x}$ and $f_{*l}(\boldsymbol{x})$, which helps to capture the nonlinear nonfunctional space-dependent cross-correlations between the low-fidelity and high-fidelity models.

The hyperparameters are optimized by minimizing the negative log marginal likelihood (NLML) of the GP model, which is described as:

$$NLML = \frac{1}{2} log|\boldsymbol{K}| + \frac{1}{2}\boldsymbol{y}^T\boldsymbol{K}^{-1}y + \frac{n}{2} log 2\pi \tag{10}$$

where $\boldsymbol{K}$ is the kernel function, $\boldsymbol{y}$ is the training target and $n$ is the dimension of input space. After the hyperparameters are elicited, the posterior prediction of the high-fidelity model at a test point $(\boldsymbol{x}_*, f_{*l})(\boldsymbol{x}_*)$ is given by:

$$p\left(f_{*h}(\boldsymbol{x}_*)\right) = \int p\left(f_h\left(\boldsymbol{x}_*, f_{*l}(\boldsymbol{x}_*)\right)|\boldsymbol{y}_h, \boldsymbol{x}_h, \boldsymbol{x}_*\right) p\left(f_{*l}(\boldsymbol{x}_*)\right) \mathrm{d}\boldsymbol{x}_* \tag{11}$$

Notice that Monte Carlo simulation is used here to obtain the posterior distribution of the high-fidelity model. This is because only the low-fidelity model is a standard GP regression with parametric input data points, and its posterior prediction follows a Gaussian distribution. Whereas the high-fidelity model is a GP regression model with the input of the posterior prediction from the low-fidelity model. As a result, the posterior distribution of the high-fidelity model is no longer Gaussian. In view of this, Monte Carlo integration is pursued on equation (11) to calculate the posterior mean and variance of the high-fidelity model.

### *2.3 Workflow of the proposed strategy*

The workflow of the proposed multifidelity flow field reconstruction method is summarized as follows:

*Step 1*: A PINN is trained using the RANS equations. The residuals of the governing equations and boundary conditions at sampled collocation points are embedded in the loss function to train the PINN model.

*Step 2*: An immense amount of labeled data, needed for low-fidelity GP modeling, is generated from the PINN model within the computational domain. The hyperparameters in the low-fidelity GP model are optimized by minimizing the NLML of equation (10). At last, the posterior mean and variance of the low-fidelity standard GP regression model are calculated.

*Step 3*: The high-fidelity GP regression model in equation (8) is established based on the posterior prediction of the low-fidelity model and small amounts of high-fidelity observations. The hyperparameters of the high-fidelity GP model are optimized by minimizing the NLML of equation (10), which uses the kernel function of equation (9).

*Step 4*: The posterior mean and variance of the high-fidelity GP model are evaluated by Monte Carlo integration on equation (11), which uses the posterior mean and variance of the low-fidelity standard GP regression model obtained in *Step 2*.

## 3. Results and discussions

### *3.1 Case 1: flow past a hill (Reynolds number: 6 × 10$^4$)*

The data used in the first case study, which are open to all fluid dynamicists around the world, were obtained by Almeida *et al.* (1993) in an experimental study. In the experiment, a fully developed channel flow passed through a single hill at the location of 6 m along the flow direction from the tunnel inlet. Meanwhile, the time-averaged flow velocities in both horizontal and vertical directions around the hill were measured, which will serve as a test database to verify the feasibility of our proposed multifidelity flow field reconstruction method. In this case study, we aim to reconstruct the mainstream velocity of the channel flow by using the low-fidelity PINN predictions and the high-fidelity experimental measurements under the multifidelity modeling strategy.

Making use of the RANS equations and boundary conditions, we first formulate a PINN which can offer solution to the two-dimensional (2D) time-averaged flow field around the hill. It can be seen in equation (2) that the introduction of the Reynolds stress terms makes the RANS equations no longer a closed-form system of equations. To close the RANS equations, Chen's model (Chen and Xu, 1998) is adopted which assumes the Reynolds stress terms can be expressed as:

$$-\rho\overline{u'_i u'_j} = \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) = 0.03874\rho\bar{v}l\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{12}$$

where $\bar{v}$ is the local velocity and $l$ is the distance from the nearest wall. Then, PINN is formulated to solve the Chen's model-based RANS equations. We adopt the same computational domain configurations recommended by Casey and Wintergerste (2000) except that the downstream boundary is defined as a zero-pressure outlet. More details about the computational domain for PINN formulation are shown in Figure 2. A deep neural network containing 6 hidden layers and 40 neurons per layer is adopted to map the relationship between spatial coordinates ($x$ and $y$) and flow characteristics ($\varphi$ and $p$). *Tanh* and *Adam* with a learning rate of $3 \times 10^{-4}$ are used as the activation function and optimizer in training the neural network. One hundred equally spaced collocation points inside the domain are sampled along the $x$-axis and $y$-axis, respectively. A lattice of collocation points with a 100 × 100 size is, thus, generated. Among these points, 254 points are located inside the 2D hill and, hence, they are excluded, thus, resulting in the number of collocation points being 9,766 inside the domain. Meanwhile, there are four distinct boundaries in this case, which are an inlet boundary, an outlet boundary, a symmetry boundary and a wall boundary (2D hill surface and the ground). On each boundary, 500 equally spaced collocation points are sampled (on the wall boundary, the projections of the distances between collocation points along the $x$-axis, instead of the distances themselves, are equal).

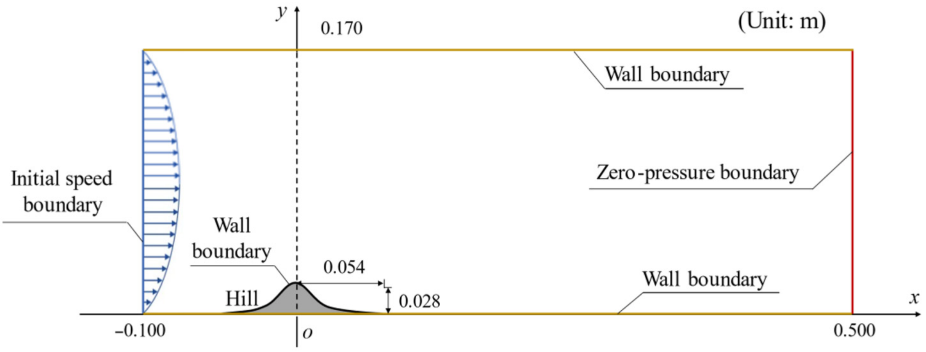


**Figure 2.** Computational domain of the flow past a 2D hill

**Source:** Figure by authors

Thus, there are 2,000 collocation points on the domain boundaries to calculate the residuals of boundary conditions.

More specifically, the residuals of the Chen's model-based RANS equations calculated at the 9,766 domain collocation points form the loss term $L_f$ defined in equation (6), and the residuals of the boundary conditions calculated at the 2,000 boundary collocation points form the loss term $L_b$ defined in equation (7). By minimizing these physical constraints, the configured PINN realizes its function of offering approximate solutions to this flow problem. The PINN predictions of the velocity component $u$ after $1 \times 10^5$ training iterations are shown in Figure 3(a), which are depicted as an orange-curved surface, compared with the red dots which represent the

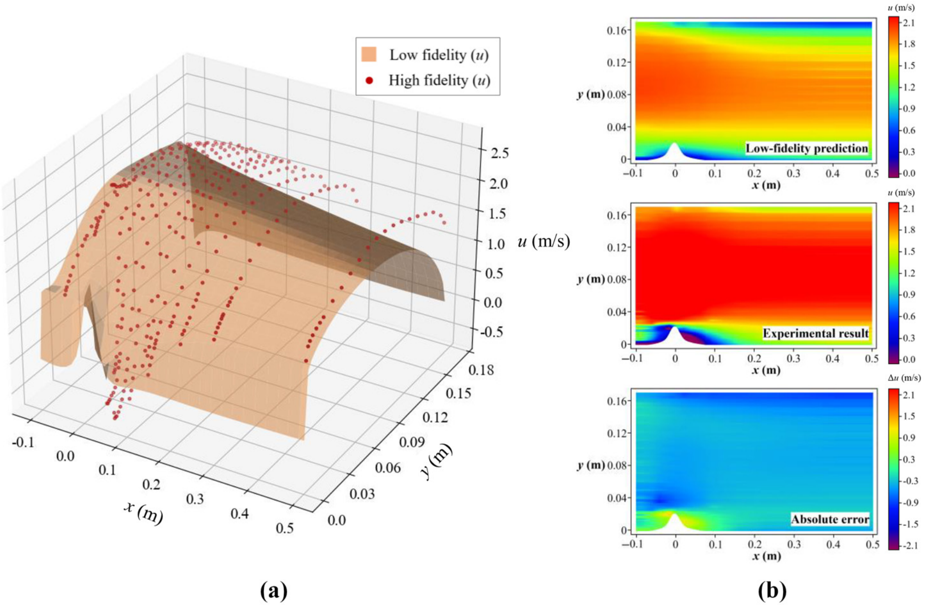


**Figure 3.** Prediction of $u$ using the low-fidelity model in Case 1: (a) general view; and (b) the velocity contour compared with the experimental counterpart (absolute error = prediction – experimental result)

**Source:** Figure by authors

experiment measurements. Figure 3(b) depicts the contour of the velocity component $u$ based on the PINN prediction, which is also compared with the reference data from the experiment. As can be seen in Figure 3, there exists a significant difference between the PINN predictions and the experimental results.

We can easily draw a conclusion from Figure 3 that, without incorporating measurement data to train the PINN, its solution can only be viewed as low-fidelity approximation. To establish the proposed multifidelity model for predicting flow field around the 2D hill, we then select 900 uniformly distributed low-fidelity sampling points (labeled data) generated by the PINN prediction. Meanwhile, among the total 325 experimental measurement points scattered in the computational domain, we pick up 35 points and consider the measured mainstream velocities at these points as high-fidelity training data. The spatial coordinates of the 35 high-fidelity training points are shown in Table 1. In selecting the training points, we abide by the principle of distributing the training points over the whole computational domain as evenly as possible.

Then, the NIF algorithm is implemented to establish the multifidelity model. We first train a GP regression model by using the low-fidelity data to acquire the Gaussian predictive posterior distribution of the mainstream velocity component $u$ on the low-fidelity level. We maximize the marginal log-likelihood to seek optimal hyperparameters by using *L-BFGS* optimizer with the randomized restart strategy. Once acquiring the Gaussian posterior distribution on the low-fidelity level, we proceed to formulating the high-fidelity GP regression model according to equation (8). The negative marginal log-likelihood defined in equation (10) on the high-fidelity level is maximized to optimize the hyperparameters. After the model is fully trained, we get the posterior distribution of the mainstream velocity component $u$ on the high-fidelity level using Monte Carlo integration. The prediction results of the velocity component $u$ using the proposed multifidelity model are first shown in Figure 4(a). In this figure, the green-curved surface represents the prediction of the mainstream velocity component $u$ obtained by the multifidelity model. The yellow curved surface denotes the low-fidelity data generated by the PINN prediction. The red dots are the experimental results of the mainstream velocity component $u$ at all measurement points. In general, the low-fidelity data (PINN predictions) and the predictions by the multifidelity model show a similar trend within the whole computational domain; however, the latter is much closer to the experimental results. This is because the NIF algorithm can precisely capture the nonlinear nonfunctional space-dependent cross-correlations between the low-fidelity and high-fidelity data. As a result, the multifidelity model can learn from the trend of the low-fidelity PINN predictions to fit the scattered data points on the high-fidelity level. In other words, scattered (sparse) data points on the high-fidelity level are used to correct the low-fidelity prediction surface, while the trend of the low-fidelity surface stemming from the physical law is preserved to the greatest extent. The velocity contours of $u$ from the multifidelity model predictions and from the experimental results are compared in Figure 4(b).

| $x$ Coordinate (m) | −0.050 | 0.050 | 0.150 | 0.300 | 0.500 |
|---|---|---|---|---|---|
| $y$ Coordinate (m) | 0.006 | 0.002 | 0.001 | 0.001 | 0.001 |
| | 0.015 | 0.015 | 0.015 | 0.016 | 0.016 |
| | 0.030 | 0.030 | 0.030 | 0.030 | 0.030 |
| | 0.070 | 0.070 | 0.070 | 0.070 | 0.070 |
| | 0.100 | 0.100 | 0.100 | 0.100 | 0.100 |
| | 0.130 | 0.130 | 0.130 | 0.130 | 0.130 |
| | 0.160 | 0.165 | 0.165 | 0.165 | 0.165 |

**Source:** Table by authors

**Table 1.** Spatial coordinates of the high-fidelity points used to formulate the multifidelity model

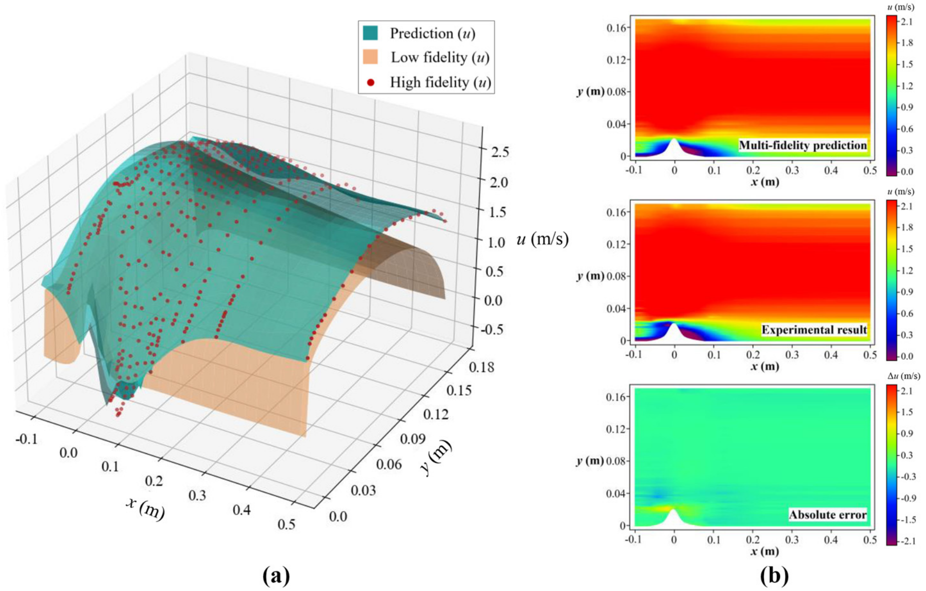


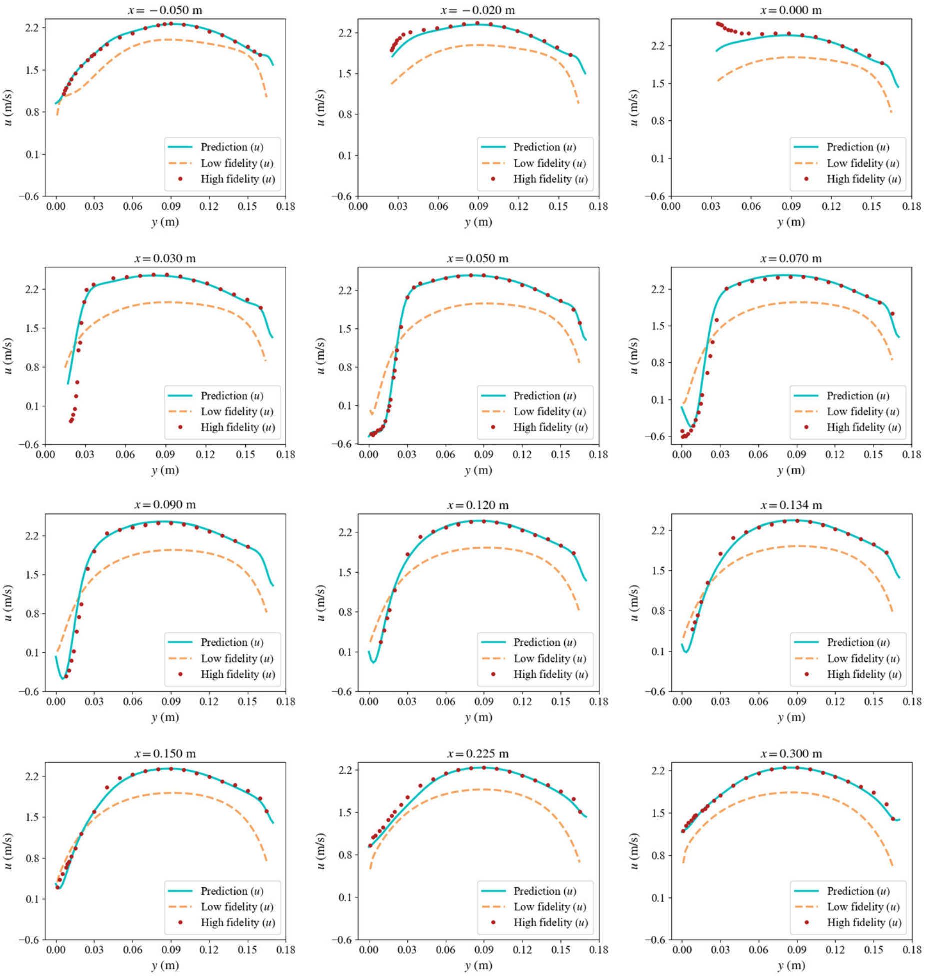
**Figure 4.**
Prediction of $u$ using the multifidelity model in Case 1: (a) general view; and (b) the velocity contour compared with the experimental counterpart (absolute error = prediction – experimental result)

**Source:** Figure by authors

The absolute error is much lower than that of the low-fidelity model which is illustrated in Figure 3(b).

Figure 5 provides a more detailed comparison of the results between the multifidelity model predictions and the high/low-fidelity data on 12 vertical lines within the computational domain. We can easily discover that the multifidelity model predictions show good consistency with the experimental measurements in the whole computational domain. Compared with other lines, better results are achieved on the lines $x = -0.050$ m, $x = 0.050$ m, $x = 0.150$ m, $x = 0.300$ m and $x = 0.500$ m because the 35 high-fidelity training data points are positioned there. On the other lines, the multifidelity model yet demonstrates competitive results, especially when being compared with the PINN predictions. We further quantitatively evaluate the performance of the proposed multifidelity model in Table 2, through comparison with other two widely used strategies for flow field reconstruction, i.e. data-driven PINN and conventional perceptron neural network (PNN). In the data-driven PINN strategy, we not only embed the physical governing equations and boundary conditions into the total loss but also concurrently embed the 35 high-fidelity training data (Table 1) into the PINN training process. For consistency, the configuration of the data-driven PINN is the same as that of the PINN we used to train the low-fidelity model. In the PNN paradigm, we simply conduct a regression task using only 35 high-fidelity training data points. A PNN configuration with only one hidden layer which contains 10 neurons is adopted, while the *L-BFGS* optimizer with a learning rate of $5 \times 10^{-4}$ is adopted to train the neural network. As shown in Table 2, we use the $\ell_2$ error to quantify the prediction accuracy, which is defined as:

$$l_2 \ error = \frac{\| \boldsymbol{U}_i - \tilde{\boldsymbol{U}}_i \|_2}{\| \boldsymbol{U}_i \|_2} \times 100\% \tag{13}$$



**Source:** Figure by authors

**Figure 5.** Comparison of the results between the multifidelity model predictions and the high/low-fidelity data on 12 vertical lines in Case 1

**Table 2.** Performance of different flow field reconstruction strategies

| | Multifidelity model | Data-driven PINN | PNN |
|---|---|---|---|
| $\ell_2$ Error | 9.8% | 24.6% | 55.3% |
| Computing time (s) | $2.3 \times 10^4$ | $2.3 \times 10^4$ | $4.1 \times 10^1$ |

**Source:** Table by authors

where $\|\cdot\|_2$ denotes the $\ell_2$-norm, $\boldsymbol{U_i}$ denotes a vector of the reference data and $\tilde{\mathbf{U}}_\mathbf{i}$ denotes a vector of the predictions. From Table 2, we can see that our proposed multifidelity flow field reconstruction model offers the minimum $\ell_2$ error among the three strategies, which is only 9.8%. In comparison, the results obtained from the data-driven PINN strategy are much worse, which leads to an $\ell_2$ error of 24.6%. Using a PNN to reconstruct the flow field is the fastest strategy in terms of computing time, the relative error is unbearable when comparing its predictions with the experimental measurements. With a computing time of 41 s, the $\ell_2$ error reaches up to 55.3% when the PNN strategy is adopted to reconstruct the flow field using the 35 high-fidelity training data points. We have to admit that the computational cost of the PINN-related strategy may become a stumbling block to its wide application. Considering a PINN with 6 hidden layers, each with 40 neurons, it usually takes around $2.3 \times 10^4$ s for its training process with $1 \times 10^5$ iterations when using the *Adam* optimizer, which would be a heavy burden in engineering applications. For the multifidelity model, it takes additional 53 s for multifidelity modeling in considering that the low-fidelity PINN model can be fully trained off-line because no measurement data is needed in this process. Based on the above comparisons, it can be concluded that our proposed multifidelity model demonstrates the most competitive performance for reconstructing the flow field around the 2D single hill without considering computing resources. It is worth noting that the experimental data of $v$ are unevenly distributed and insufficient to support multifidelity modeling in this case, so we did not address the velocity component in $y$-direction.

### *3.2 Case 2: flow past a square cylinder (Reynolds number: $2 \times 10^4$)*

The data used in the second case study are from the experiment conducted by Lyn and Rodi (1994), which describe a turbulent flow around a 2D square cylinder, as shown in Figure 6. The computational domain is 0.44 m in length and 0.32 m in width, while a 0.04 m × 0.04 m square cylinder is located at the left center of the computational domain. Point A is the bottom left corner of the square cylinder, whose spatial coordinate is (0.10, 0.14) as shown in Figure 6. The left boundary of the computational domain is defined as an initial speed boundary, where the fluid velocity stabilizes at 0.535 m/s. The upper and lower boundaries are defined as symmetry boundaries, while the right boundary of the computational domain is assigned a zero-pressure outlet. The surfaces of the square cylinder are considered as wall boundaries, where the fluid velocity equals zero. In the experiment, there were 517

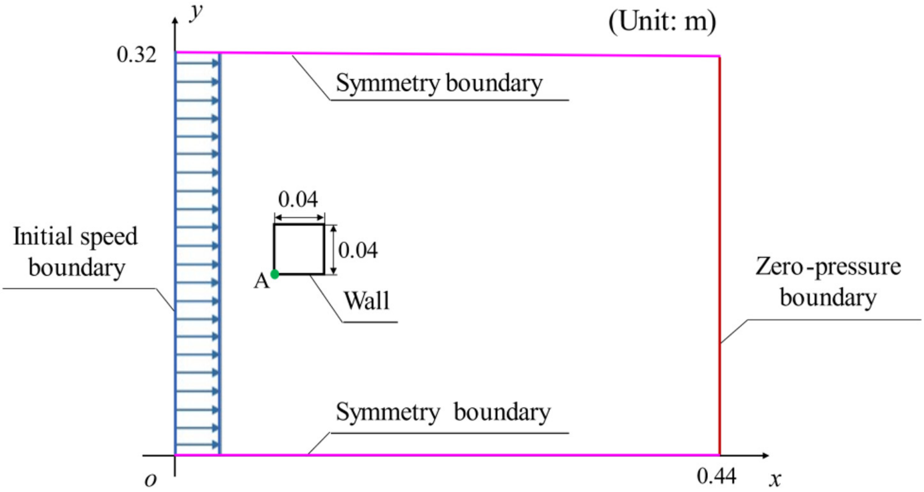


**Figure 6.**
Computational domain of the flow passing a 2D square cylinder

**Source:** Figure by authors

measurement points that captured the time-averaged flow velocity components $u$ and $v$ inside the flow field.

Case 2 differs from Case 1 in two aspects. First, the most intuitive aspect is the different geometric appearances. Second, for comparison with Case 1, both the components $v$ and $u$ at selected measurement points are used to formulate the multifidelity model in Case 2. Among the 517 experimental measurement points, we pick up 36 points scattered on 9 individual vertical lines to train the multifidelity model for time-averaged flow field reconstruction. Their spatial coordinates are shown in Table 3. In selecting training points, we still abide by the principle of distributing the training points over the whole computational domain as evenly as possible.

The objective of Case 2 remains to apply all available physical restrictions and sparse measurement information to reconstruct high-fidelity mainstream velocity $u$ over the entire computational domain. In Case 2, we compare the performances of five schemes to achieve this objective. In the first three schemes, the NIF algorithm and sparse $u$ measurements are used in training the multifidelity model. The only difference between these three schemes is the low-fidelity data source. A PINN without training data (purely physics-based), a PINN with $v$ embedded in its training process and CFD are used as the low-fidelity data sources, respectively. The fourth scheme is a data-driven PINN where both $u$ and $v$ are used to train the neural network, while the NIF algorithm is not engaged in this scheme. The fifth scheme is a PNN like what was built in Case 1. The five schemes are described in Table 4.

Considering the multiple sources of low-fidelity data in this case study, we delve further into the details of the PINN and CFD frameworks separately. For the PINN framework, its configuration remains the same as that in the previous case except that only 108 collocation points inside the 2 D square cylinder are excluded, which ultimately leads to a total of 9,892 collocation points inside the domain. In addition, there are 50 equally-spaced boundary collocation points on the initial speed boundary, zero-pressure outlet, upper symmetry boundary, lower symmetry boundary and each of the four side surfaces of the 2D square cylinder, respectively. Thus, a total of 400 boundary collocation points are used to calculate the physical residuals in this case. For the CFD framework, the simulation of the time-averaged flow field is performed based on the commercial software Star CCM+ in this

**Table 3.** Spatial coordinates of the high-fidelity points used to train the multifidelity model

| $x$ Coordinate (m) | 0.000 | 0.100 | 0.155 | 0.200 | 0.250 | 0.300 | 0.350 | 0.392 | 0.400 |
|---|---|---|---|---|---|---|---|---|---|
| $y$ Coordinate (m) | 0.200 | 0.200 | 0.160 | 0.160 | 0.160 | 0.160 | 0.160 | 0.160 | 0.160 |
| | 0.240 | 0.240 | 0.180 | 0.200 | 0.200 | 0.200 | 0.200 | 0.200 | 0.200 |
| | 0.280 | 0.280 | 0.200 | 0.240 | 0.240 | 0.240 | 0.240 | 0.240 | 0.240 |
| | 0.320 | 0.320 | 0.220 | – | – | 0.280 | – | 0.280 | 0.280 |
| | – | – | – | – | – | 0.320 | – | 0.320 | 0.320 |

**Source:** Table by authors

**Table 4.** Five schemes for flow field reconstruction in Case 2

| Scheme 1 | Scheme 2 | Scheme 3 | Scheme 4 | Scheme 5 |
|---|---|---|---|---|
| PINN + NIF | $v$-embedded PINN + NIF | CFD + NIF | $u$- and $v$-embedded PINN | PNN |

**Source:** Table by authors

study, and the mesh inside the computational domain consists of more than 36,484 cells. For both PINN and CFD frameworks, the standard $k$–$\varepsilon$ model is adopted in RANS turbulence modeling. In the standard $k$–$\varepsilon$ model, the Reynolds stress is described as follows according to the Boussinesq assumption in 2 D cases:

$$-\rho\overline{u'_i u'_j} = \mu_t\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) - k\delta_{ij} \tag{14}$$

where $\delta_{ij}$ is the Kronecker delta function. Apart from the continuity equation and the momentum equation, the kinetic energy equation and the dissipation equation (i.e. $k$-equation and $\varepsilon$-equation) are additionally introduced in the standard $k$–$\varepsilon$ turbulence model to simulate turbulent behaviors. The $k$-equation and the $\varepsilon$-equation can be described as follows (buoyancy is neglected):

$$\frac{\partial}{\partial t}(\rho k)+\frac{\partial}{\partial x_i}(\rho k u_i)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]-\rho\overline{u'_i u'_j}\frac{\partial u_j}{\partial x_i}-\rho\varepsilon \tag{15}$$

$$\frac{\partial}{\partial t}(\rho\varepsilon)+\frac{\partial}{\partial x_i}(\rho\varepsilon u_i)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right]-C_{1\varepsilon}\frac{\varepsilon}{k}\rho\overline{u'_i u'_j}\frac{\partial u_j}{\partial x_i}-C_{2\varepsilon}\rho\frac{\varepsilon^2}{k} \tag{16}$$

where $k$ is the turbulence kinetic energy, and $\varepsilon$ is the turbulent dissipation rate. Also, $k$, $\varepsilon$ and $\mu_t$ satisfy the following relationship:

$$\mu_t = \rho C_\mu\frac{k^2}{\varepsilon} \tag{17}$$

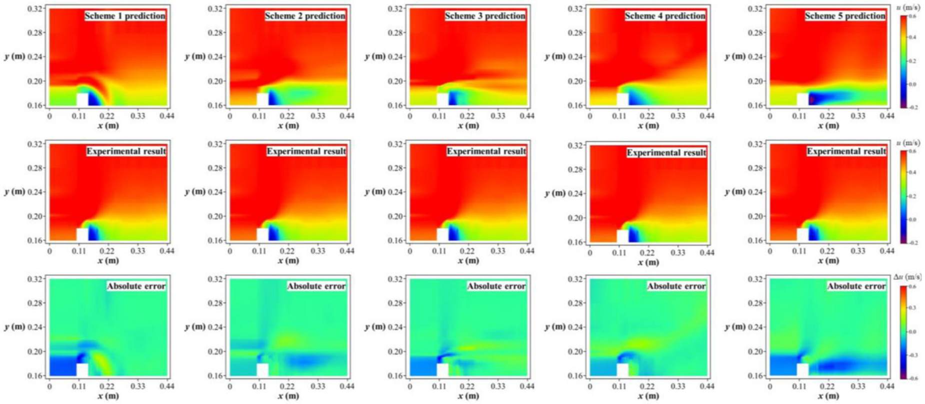
and the values of the coefficients in the standard $k$–$\varepsilon$ model are provided in Table 5.

The prediction results of the mainstream velocity $u$ from different schemes are depicted in Figure 7, which are also compared with the experiment results. These velocity contours show that Scheme 2, notably in the upstream region of the square cylinder, reconstructs the fluid features most accurately and efficiently. This may be due to the fusion of measured fluid features from the upstream regions in low-fidelity modeling in this scheme. Again, $\ell_2$ error is used to quantitatively evaluate the accuracy of predictions from different models. As shown in Table 6, the $\ell_2$ error of Scheme 2 is the lowest among all the schemes, which is only 8.8% compared with the experimental results. In particular, when $v$-embedded PINN, instead of CFD, is used as the low-fidelity data source, the $\ell_2$ error decreases by 6.6%. This is because measurement information has been included; compared to CFD, the simulation results from $v$-embedded PINN are more accurate. However, when field information is unavailable, the precision of the PINN's prediction drops dramatically. This results in the $\ell_2$ error of Scheme 1 being 18.1%, which is the worst among the multifidelity models. We

**Table 5.**
The values of the coefficients in the standard $k$–$\varepsilon$ model

| Coefficient | $C_\mu$ | $C_{1\varepsilon}$ | $C_{2\varepsilon}$ | $\sigma_k$ | $\sigma_\varepsilon$ |
|---|---|---|---|---|---|
| Value | 0.09 | 1.44 | 1.92 | 1.0 | 1.3 |

**Source:** Launder and Sharma (1974)



**Source:** Figure by authors

**Figure 7.** Prediction of $u$ using five schemes: (top) prediction; (middle) experimental result; and (bottom) absolute error

**Table 6.** Performance of different flow field reconstruction strategies

| | Scheme 1 | Scheme 2 | Scheme 3 | Scheme 4 | Scheme 5 |
|---|---|---|---|---|---|
| $\ell_2$ Error | 18.1% | 8.8% | 15.4% | 15.5% | 29.1% |
| Computing time (s) | $2.8 \times 10^4$ | $2.8 \times 10^4$ | $1.8 \times 10^3$ | $2.8 \times 10^4$ | $2.1 \times 10^1$ |

**Source:** Table by authors

observe the significance of physical information fusion in neural network modeling, which results in an improved accuracy by 13.6% for Scheme 4 over Scheme 5. Meanwhile, it is worth mentioning that the modeling process in all PINN-involved schemes (i.e. Schemes 1, 2 and 4) is time-consuming, compared to CFD, because of involving the training of PINN. However, once the model has been fully trained, it can quickly predict wind velocity at any point within the computational domain. In the multifidelity strategy, PINN-based low-fidelity models can be fully trained off-line. Thus, its training cost is not a critical concern. Once we obtain measurement data, the multifidelity modeling can be completed in just a few seconds.

## 4. Conclusions

In this paper, we proposed a novel time-averaged flow field reconstruction strategy in the framework of multifidelity modeling using PINN and the NIF algorithm. It can be viewed as a two-step approach in which the low-fidelity data are generated by a PINN, while sparse experimental or field measurement data are thought of as high-fidelity data which, in conjunction with PINN-generated low-fidelity data, are used to formulate a multifidelity model by means of the NIF algorithm. A flow past a hill and a flow past a square cylinder were used to verify the capability of the proposed multifidelity strategy, and the results demonstrated its efficacy for time-averaged flow field reconstruction. This study comes to the following conclusions:

- Although only small amounts of measurement/experimental data are accessible for reconstructing the time-averaged flow fields in both cases, the missing flow information within the whole computational domain can be favorably recovered by the multifidelity model.

- The predictions of PINN constructed purely using the RANS equations can only be considered as low-fidelity data due to low prediction accuracy in both cases. However, its trend (underlying physics) within the whole computational domain can be inherited through PINN, while the high-fidelity yet sparse measurement data can be used to rectify the low-fidelity prediction surface by implementing the NIF algorithm. The multifidelity model elicited by the NIF algorithm can learn from the trend of the low-fidelity PINN predictions to fit the scattered data points on the high-fidelity level, thereby enabling the reconstruction of time-averaged flow fields with high accuracy.
- Compared with other flow field reconstruction strategies, the proposed strategy demonstrates the most competitive results. The relative errors of the mainstream velocity component from the multifidelity prediction are less than 10% in both cases relative to the experimental measurements. However, the modeling process in all PINN-involved strategies is time-consuming because of involving the training of PINN. Once the model has been trained, it can make predictions quickly.

## References

Abrahamson, S. and Lonnes, S. (1995), "Uncertainty in calculating vorticity from 2D velocity fields using circulation and least-squares approaches", *Experiments in Fluids*, Vol. 20 No. 1, pp. 10-20.

Almeida, G., Durao, D. and Heitor, M. (1993), "Wake flows behind two-dimensional model hills", *Experimental Thermal and Fluid Science*, Vol. 7 No. 1, pp. 87-101.

Arzani, A., Wang, J.-X. and D'Souza, R.M. (2021), "Uncovering near-wall blood flow from sparse data with physics-informed neural networks", *Physics of Fluids*, Vol. 33 No. 7, p. 71905.

Baydin, A.G., Pearlmutter, B.A., Radul, A.A. and Siskind, J.M. (2018), "Automatic differentiation in machine learning: a survey", *Journal of Machine Learning Research*, Vol. 18, pp. 1-43.

Casey, M. and Wintergerste, T. (2000), "European research community on flow, turbulence and combustion", *ERCOFTAC Best Practice Guidelines: ERCOFTAC Special Interest Group on 'Quality and Trust in Industrial CFD'*.

Chen, J., Gao, G. and Zhu, C. (2016), "Detached-eddy simulation of flow around high-speed train on a bridge under cross winds", *Journal of Central South University*, Vol. 23 No. 10, pp. 2735-2746.

Chen, X., Liu, T., Zhou, X., Li, W., Xie, T. and Chen, Z. (2017a), "Analysis of the aerodynamic effects of different nose lengths on two trains intersecting in a tunnel at 350 km/h", *Tunnelling and Underground Space Technology*, Vol. 66, pp. 77-90.

Chen, Z., Liu, T., Zhou, X. and Niu, J. (2017b), "Impact of ambient wind on aerodynamic performance when two trains intersect inside a tunnel", *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 169, pp. 139-155.

Chen, Z., Rui, E., Liu, T., Ni, Y., Huo, X., Xia, Y., Li, W., Guo, Z. and Zhou, L. (2022), "Unsteady aerodynamic characteristics of a high-speed train induced by the sudden change of windbreak wall structure: a case study of the Xinjiang railway", *Applied Sciences*, Vol. 12 No. 14, p. 7217.

Chen, Q. and Xu, W. (1998), "A zero-equation turbulence model for indoor airflow simulation", *Energy and Buildings*, Vol. 28 No. 2, pp. 137-144.

Ding, Y., Zhang, Y., Ren, Y.M., Orkoulas, G. and Christofides, P.D. (2019), "Machine learning-based modeling and operation for ALD of SiO2 thin-films using data from a multiscale CFD simulation", *Chemical Engineering Research and Design*, Vol. 151, pp. 131-145.

Eivazi, H. and Vinuesa, R. (2022), “Physics-informed deep-learning applications to experimental fluid mechanics”, arXiv preprint arXiv:2203.15402.

Eivazi, H., Tahani, M., Schlatter, P. and Vinuesa, R. (2022), “Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations”, *Physics of Fluids*, Vol. 34 No. 7, p. 75117.

Ershkov, S.V., Prosviryakov, E.Y., Burmasheva, N.V. and Christianto, V. (2021), “Towards understanding the algorithms for solving the Navier–Stokes equations”, *Fluid Dynamics Research*, Vol. 53 No. 4, p. 44501.

Flaga, A., Kocoń, A., Kłaput, R. and Bosak, G. (2018), “The environmental effects of aerodynamic interference between two closely positioned irregular high buildings”, *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 180, pp. 276-287.

Gao, H., Liu, T., Gu, H., Jiang, Z., Huo, X., Xia, Y. and Chen, Z. (2021), “Full-scale tests of unsteady aerodynamic loads and pressure distribution on fast trains in crosswinds”, *Measurement*, Vol. 186, p. 110152.

Goswami, S., Anitescu, C., Chakraborty, S. and Rabczuk, T. (2020), “Transfer learning enhanced physics informed neural network for phase-field modeling of fracture”, *Theoretical and Applied Fracture Mechanics*, Vol. 106, p. 102447.

Hasanuzzaman, G., Eivazi, H., Merbold, S., Egbers, C. and Vinuesa, R. (2023), “Enhancement of PIV measurements via physics-informed neural networks”, *Measurement Science and Technology*, Vol. 34 No. 4, p. 44002.

He, X., Zhou, L., Chen, Z., Jing, H., Zou, Y. and Wu, T. (2019), “Effect of wind barriers on the flow field and aerodynamic forces of a train–bridge system”, *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, Vol. 233 No. 3, pp. 283-297.

Hu, G., Liu, L., Tao, D., Song, J., Tse, K.T. and Kwok, K.C. (2020), “Deep learning-based investigation of wind pressures on tall building under interference effects”, *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 201, p. 104138.

Jahromi, H.R.T., Sazonov, I., Jones, J., Coccarelli, A., Rolland, S., Chakshu, N.K., Thomas, H. and Nithiarasu, P. (2022), “Predicting the airborne microbial transmission via human breath particles using a gated recurrent units neural network”, *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 32 No. 9, pp. 2964-2981.

Jin, X., Cai, S., Li, H. and Karniadakis, G.E. (2021), “NSFnets (Navier–Stokes flow nets): physics-informed neural networks for the incompressible Navier–Stokes equations”, *Journal of Computational Physics*, Vol. 426, p. 109951.

Karpatne, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N. and Kumar, V. (2017), “Theory-guided data science: a new paradigm for scientific discovery from data”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 29 No. 10, pp. 2318-2331.

Kim, B., Lee, D., Preethaa, K.S., Hu, G., Natarajan, Y. and Kwok, K.C. (2021), “Predicting wind flow around buildings using deep learning”, *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 219, p. 104820.

Kim, M. and Moon, J.H. (2022), “Deep neural network prediction for effective thermal conductivity and spreading thermal resistance for flat heat pipe”, *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 33 No. 2.

Kong, C., Chang, J., Wang, Z., Li, Y. and Bao, W. (2021), “Data-driven super-resolution reconstruction of supersonic flow field by convolutional neural networks”, *AIP Advances*, Vol. 11 No. 6, p. 65321.

Ladický, L., Jeong, S., Solenthaler, B., Pollefeys, M. and Gross, M. (2015), “Data-driven fluid simulations using regression forests”, *ACM Transactions on Graphics (TOG)*, Vol. 34 No. 6, pp. 1-9.

Launder, B.E. and Sharma, B.I. (1974), “Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc”, *Letters in Heat and Mass Transfer*, Vol. 1 No. 2, pp. 131-137.

Li, X., Chen, G., Zhou, D. and Chen, Z. (2019), “Impact of different nose lengths on flow-field structure around a high-speed train”, *Applied Sciences*, Vol. 9 No. 21, p. 4573.

Li, W., Liu, T., Martinez-Vazquez, P., Guo, Z., Huo, X., Xia, Y. and Chen, Z. (2022), “Effects of embankment layouts on train aerodynamics in a wind tunnel configuration”, *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 220, p. 104830.

Liu, Y., Huang, H. and Zhang, X. (2011), “A data-driven approach to selecting imperfect maintenance models”, *IEEE Transactions on Reliability*, Vol. 61 No. 1, pp. 101-112.

Liu, M., Li, Q., Huang, S., Shi, F. and Chen, F. (2018), “Evaluation of wind effects on a large span retractable roof stadium by wind tunnel experiment and numerical simulation”, *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 179, pp. 39-57.

Liu, T., Wang, L., Chen, Z., Gao, H., Li, W., Guo, Z., Xia, Y., Huo, X. and Wang, Y. (2022), “Study on the pressure pipe length in train aerodynamic tests and its applications in crosswinds”, *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 220, p. 104880.

Löhner, R., Antil, H., Tamaddon-Jahromi, H., Chakshu, N.K. and Nithiarasu, P. (2021), “Deep learning or interpolation for inverse modelling of heat and fluid flow problems?”, *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 31 No. 9, pp. 3036-3046.

Lu, Z., Qu, J., Liu, H., He, C., Zhang, B. and Chen, Q. (2021), “Surrogate modeling for physical fields of heat transfer processes based on physics-informed neural network”, *CIESC Journal*, In Chinese, Vol. 72 No. 3, pp. 1496-1503.

Lyn, D. and Rodi, W. (1994), “The flapping shear layer formed by flow separation from the forward corner of a square cylinder”, *Journal of Fluid Mechanics*, Vol. 267, pp. 353-376.

Ntinas, G.K., Shen, X., Wang, Y. and Zhang, G. (2018), “Evaluation of CFD turbulence models for simulating external airflow around varied building roof with wind tunnel experiment”, *Building Simulation*, Vol. 11 No. 1, pp. 115-123.

Nugroho, N.Y., Triyadi, S. and Wonorahardjo, S. (2022), “Effect of high-rise buildings on the surrounding thermal environment”, *Building and Environment*, Vol. 207, p. 108393.

Penwarden, M., Zhe, S., Narayan, A. and Kirby, R.M. (2022), “Multifidelity modeling for physics-informed neural networks (PINNs)”, *Journal of Computational Physics*, Vol. 451, p. 110844.

Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N.D. and Karniadakis, G.E. (2017), “Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling”, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 473 No. 2198, p. 20160751.

Pruvost, J., Legrand, J. and Legentilhomme, P. (2001), “Three-dimensional swirl flow velocity-field reconstruction using a neural network with radial basis functions”, *Journal of Fluids Engineering*, Vol. 123 No. 4, pp. 920-927.

Raissi, M., Perdikaris, P. and Karniadakis, G.E. (2019), “Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”, *Journal of Computational Physics*, Vol. 378, pp. 686-707.

Rao, C., Sun, H. and Liu, Y. (2020), “Physics-informed deep learning for incompressible laminar flows”, *Theoretical and Applied Mechanics Letters*, Vol. 10 No. 3, pp. 207-212.

Rasmussen, C.E. (2004), “Gaussian processes in machine learning”, in Bousquet, O., von Luxburg, U. and Rätsch, G.S. (Eds), *Advanced Lectures on Machine Learning: ML Summer Schools 2003*, Springer, Berlin Heidelberg, pp. 63-71.

Rui, E., Chen, Z., Ni, Y., Yuan, L. and Zeng, G. (2023), “Reconstruction of 3D flow field around a building model in wind tunnel: a novel physics-informed neural network framework adopting dynamic prioritization self-adaptive loss balance strategy", *Engineering Applications of Computational Fluid Mechanics*, Vol. 17 No. 1, p. 2238849.

Sun, S., Liu, S., Liu, J. and Schlaberg, H.I. (2018), "Wind field reconstruction using inverse process with optimal sensor placement", *IEEE Transactions on Sustainable Energy*, Vol. 10 No. 3, pp. 1290-1299.

Sun, Y., Sun, Q. and Qin, K. (2021), "Physics-based deep learning for flow problems", *Energies*, Vol. 14 No. 22, p. 7760.

Tian, G., Fan, Y., Wang, H., Peng, K., Zhang, X. and Zheng, H. (2020), "Studies on the thermal environment and natural ventilation in the industrial building spaces enclosed by fabric membranes: a case study", *Journal of Building Engineering*, Vol. 32, p. 101651.

Wakes, S.J., Bauer, B.O. and Mayo, M. (2021), "A preliminary assessment of machine learning algorithms for predicting CFD-simulated wind flow patterns over idealised foredunes", *Journal of the Royal Society of New Zealand*, Vol. 51 No. 2, pp. 290-306.

Wang, H., Liu, Y. and Wang, S. (2022), "Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network", *Physics of Fluids*, Vol. 34 No. 1, p. 17116.

Wang, L., Luo, Z., Xu, J., Luo, W. and Yuan, J. (2021), "A novel framework for cost-effectively reconstructing the global flow field by super-resolution", *Physics of Fluids*, Vol. 33 No. 9, p. 95105.

Xiang, Z., Peng, W., Liu, X. and Yao, W. (2022), "Self-adaptive loss balanced physics-informed neural networks", *Neurocomputing*, Vol. 496, pp. 11-34.

Zhou, L., He, X., Chen, Z., Xie, T. and Jing, H. (2018), "Numerical study of effect of wind barrier on aerodynamic of bridge and train-bridge system", *Journal of Central South University (Science and Technology)*, Vol. 49 No. 7, pp. 1742-1752.

**Corresponding author**
Yi-Qing Ni can be contacted at: ceyqni@polyu.edu.hk