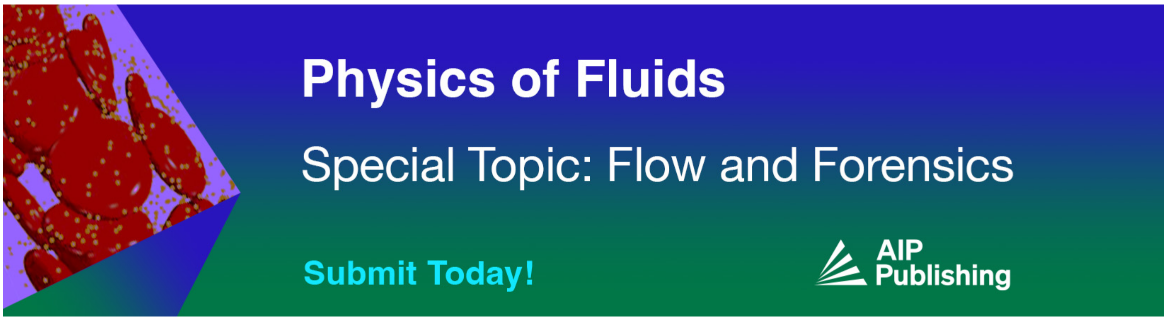
Physics of Fluids

RESEARCH ARTICLE | APRIL 05 2022

A measure–correlate–predict model based on neural networks and frozen flow hypothesis for wind resource assessment

Danyang Chen (陈丹阳) ; Zhideng Zhou (周志登) ; Xiaolei Yang (杨晓雷)

Physics of Fluids 34, 045107 (2022)
https://doi.org/10.1063/5.0086354



AIP Publishing

A measure–correlate–predict model based on neural networks and frozen flow hypothesis for wind resource assessment

Cite as: Phys. Fluids **34**, 045107 (2022); doi: 10.1063/5.0086354
Submitted: 25 January 2022 · Accepted: 19 March 2022 ·
Published Online: 5 April 2022

View Online Export Citation CrossMark

Danyang Chen (陈丹阳),[1,2] Zhideng Zhou (周志登),[1,2] and Xiaolei Yang (杨晓雷)[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 Sciences, 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**: xyang@imech.ac.cn

ABSTRACT

In this paper, a measure–correlate–predict (MCP) model based on neural networks (NN) and frozen flow hypothesis, which is abbreviated as the MCPNN-frozen model, is proposed for wind resource assessment and tested using turbulent channel flows with three different surface roughness lengths, i.e., $k_0 = 0.001$, 0.01, and 0.1 m. The predictions from the MCPNN-frozen model are compared with the real data for different separations (s) between the reference point and the target point. The results show that the correlation coefficients *C.C.* between the model predictions and real data are roughly higher than 0.5 for small separations $s/\delta \leq 3$ (where δ is the boundary layer thickness), and the coefficients of determination (R^2) are approximately higher than 0.3 when $s/\delta \leq 2$. The generalization capacity of the MCPNN-frozen model is tested for different roughness lengths and different velocity components. Further analyses show that, even though *C.C.* and R^2 decrease when increasing s, the large-scale variations of velocity fluctuations are well captured by the MCPNN-frozen model especially for the one trained using the data filtered in time. Furthermore, it is found that the model trained using the filtered data without a spanwise offset can well predict the large-scale variations at the target point when the spanwise offsets between the target point and the reference point are small (e.g., 0.1δ and 0.2δ). The proposed model leverages the power of neural networks and physical understanding. Further development of the model for complex scenarios will be carried out in the future work.

I. INTRODUCTION

Wind resource assessment is an important prerequisite for the planning and operation of wind farms to achieve the maximal performance.[1] With further penetration of wind into the world's energy portfolio to achieve the goal of net zero emission, the levelized cost of energy (LCOE) needs to be further reduced.[2,3] Wind resource assessment at high spatial and temporal resolutions plays an important role in the reduction of LCOE.[4,5] This work is devoted to developing a measure–correlate–predict (MCP) model based on neural networks and Taylor's frozen flow hypothesis for predicting wind speed variations at a location using wind data from its nearby locations.

The MCP method estimates the long-term wind variations of the target site using the short-term on-site data and concurrent data at nearby meteorological stations (which have long-term data) with similar topographic and climatic conditions.[6] Building a correlation between the reference site and the target site is at the center of different MCP methods, which is often in the form of a statistical model obtained from the short-term data. A general description of the MCP methodology can be found in the papers by Burton[7] and Letcher.[8] A large variety of MCP methodologies have been proposed and summarized in the literature,[6] such as (i) linear regression, (ii) variance ratio, (iii) Weibull scale, (iv) artificial neural networks (ANNs), (v) support vector regression (SVR), and (vi) Mortime. Wind varies at different time scales, e.g., diurnal, seasonal, and inter-annual variation. The MCP method is often applied to predict the long-term wind behavior at the target point (TP) for more than ten years based on short-term measurements (longer than a year is, in general, required) at the reference point. In later work, the MCP methods for predicting wind characteristics at smaller time scales (an hour or less) were developed.[6,9–11] Carta *et al.*[12] reviewed a wide range of MCP methods using data at a single reference site and multiple reference sites based on machine learning. In addition, the authors commented on the limitations when using the MCP models, the uncertainties associated with them, and the different reference data sources that have been used. Liu *et al.*[13] proposed a temporal interpolation extrapolation method for missing wind speed data, which considers all mixing uncertainties, to address the problem of ignoring wind speed mixing uncertainties by the traditional method. Sharma *et al.*[11] used remote sensing techniques to measure wind data, and the MCP methods of linear regression and variance ratios were employed to predict long-term wind index distributions. Diaz *et al.*[14] applied for the first time an MCP model with support vector machines and multiple linear regression to predict wind energy density, the model considered not only the variability of wind speed at the reference sites, but also the wind direction and air density. Addison *et al.*[15] applied linear neural networks (multi-layer perceptrons) to long-term energy prediction of wind direction and speed data at the target site. They found that considering the time-lagged wind speed improves the accuracy of the prediction.

The machine learning method, which has become an increasingly powerful tool in scientific computing,[16] provides a new opportunity for developing MCP models for predicting temporal fluctuations of wind turbulence. The machine learning method has been applied to various turbulent flow problems,[17] e.g., the development of turbulence models in Reynolds-averaged Navier–Stokes simulation[18–20] and large-eddy simulation (LES),[21–25] reconstruction of the turbulent flow fields,[26,27] turbulent flow control,[28–30] and temporal prediction of turbulence.[31–33] Qin *et al.*[34] proposed a training-based method for wind turbine signal forecasting by utilizing the mixed convolutional neural networks (CNNs) and long short-term memory (LSTM) network and a multi-task learning ideas. Lee and You[31] compared the performance of generative adversarial networks (GANs) and multi-scale CNNs with and without physical loss functions in the training and prediction of unsteady flow fields over a circular cylinder. They found that all the four techniques well predict the flow fields in the near future, but for the long-term flow predictions, the techniques with the physical loss function perform better. The GAN-based networks can be applied to problems without prior knowledge on the underlying flow physics. Deng *et al.*[35] proposed an LSTM-based proper orthogonal decomposition (POD) model to reconstruct the time-resolved turbulent flow fields from non-time-resolved particle image velocimetry measurements. The results demonstrated that the model can successfully recover the temporal revolution of POD coefficients. Mohan *et al.*[32] evaluated the convolutional GAN (C-GAN) and compressed convolutional LSTM (CC-LSTM) network in the temporal prediction of homogeneous isotropic turbulence and scalar turbulence. Their results showed that the CC-LSTM is able to predict the flow dynamics, while the C-GAN cannot. Nakamura *et al.*[36] combined the CNN autoencoder and LSTM to represent the spatiotemporal high-dimensional dynamics of turbulent channel flow by integrating the temporal evolution of the low-dimensional latent dynamics. Eivazi *et al.*[37] assessed the recurrent neural networks and newly developed Koopman with non-linear forcing (KNF) in the prediction of temporal dynamics of the low-order model of near-wall turbulence and showed that the KNF framework outperforms the LSTM in the short-term predictions.

How velocity fluctuations at different locations and temporal instants are correlated with each other is an active research area in turbulence.[38–40] Taylor's frozen flow model[41] is the most classic one describing the space-time correlations of turbulence, which suggests that the velocity fluctuations at a fixed point is simply due to the unchanging turbulent flow structures advected from upstream due to the mean flow. Advanced space-time correlations models including the elliptic approximation model[42,43] can be found in the review by He *et al.*[44] To leverage the power of the machine learning method and our prior knowledge on the space-time correlation of turbulence, in this work we propose to develop a MCP model based on neural networks (NN) and frozen flow hypothesis, which is abbreviated as the MCPNN-frozen model, for predicting the temporal fluctuations of wind turbulence at the target site using the short-term on-site data and the long-term data at the reference sites. The data from turbulent channel flows with three different surface roughness lengths are employed to train and test the model.

The rest of the paper is organized as follows. The proposed MCPNN-frozen model is first introduced in Sec. II. Then the training of the model is presented in Sec. III. The predictions from the model are analyzed in Sec. IV. At last, conclusions are drawn in Sec. V.

II. THE MCPNN-FROZEN MODEL

In this section, we present the proposed MCPNN-frozen model, which leverages the power of Taylor's frozen flow hypothesis and neural networks. In Taylor's frozen flow hypothesis, it is assumed that the spatial patterns of turbulent motions are carried past a fixed point at a convection velocity without any changes, such that the velocity fluctuation at one location can be computed using the velocity at the other location as follows:

$$\mathbf{u}'(x, t) = \mathbf{u}'(x_1, t - (x - x_1)/U), \tag{1}$$

where $\mathbf{u}'$ is the velocity fluctuation, x and x_1 are the streamwise coordinates, and U is the convection velocity. Neural networks are capable of recognizing patterns in complex datasets. The idea of the MCPNN-frozen model is to utilize neural networks to bridge the knowledge gap between Taylor's frozen flow hypothesis and the actual spatial and temporal variations of turbulent flows.

The idea of the MCP method is described in Fig. 1. As seen in Fig. 1, the MCP procedure includes three steps, i.e., step (1) measure the short-term data at both reference points and target points, step (2) correlate the velocity at the target point with that at the reference

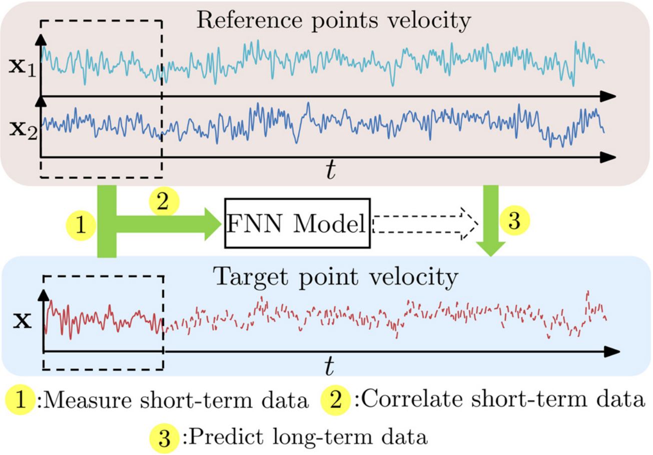

Reference points velocity
$\mathbf{x}_1$
$\mathbf{x}_2$
t
1
2
FNN Model
3
Target point velocity
$\mathbf{x}$
t
1 :Measure short-term data
2 :Correlate short-term data
3 :Predict long-term data

FIG. 1. Procedure of the measure–correlate–predict method.

points, and step (3) predict the long-term velocity variations at the target point using the obtained correlation. In this work, the following situation is considered: (1) the time series of velocity at the two reference points $\mathbf{x}_1$ and $\mathbf{x}_2$ are known for both short-term and long-term; (2) at the target point $\mathbf{x}$, only the short-term time series of velocity exist; and (3) the objective is to predict the long-term velocity variations at the target point using the available short-term data. In almost all the cases, the reference points $\mathbf{x}_1$ and $\mathbf{x}_2$ are located directly in the upstream or downstream of the target point $\mathbf{x}$, except for several cases, in which the model is evaluated for cases with spanwise offsets.

We propose to employ neural networks and Taylor's frozen flow hypothesis to build the correlation between the reference points and the target point to predict the velocity at the target point. Specifically, the neural networks are employed to build the correlation between the velocity at the target point and the input features, and the Taylor's frozen flow hypothesis is employed for constructing the input features. A schematic of the employed neural network is given in Fig. 2. In this method, the velocity data at two reference points are employed as input features. At each reference point, the velocity data at five equally spaced time instants, i.e., t_0, …, t_4, are set as input features. Let t denotes the time at which the velocity needs to be predicted. The time instants for the input features are determined based on Taylor's frozen flow hypothesis as follows:

$$\begin{aligned} t_0 &= t_2 - 2\Delta t, \\ t_1 &= t_2 - \Delta t, \\ t_2 &= t - (x - x_i)/U, \\ t_3 &= t_2 + \Delta t, \\ t_4 &= t_2 + 2\Delta t, \end{aligned} \tag{2}$$

where x_i ($i = 1, 2$) and x denote the streamwise coordinates of the reference point and target point, respectively, and U is convection velocity (which is taken as the mean streamwise velocity at 100 m above the ground for the considered cases). It is noted that the number of reference points and the number of time instants can be easily changed if needed. In Appendix A, how different parameters of the MCPNN-frozen model affect the model performance is examined. In Appendix B, the effect of the convection velocity computed using different ways on the performance of the model is tested.

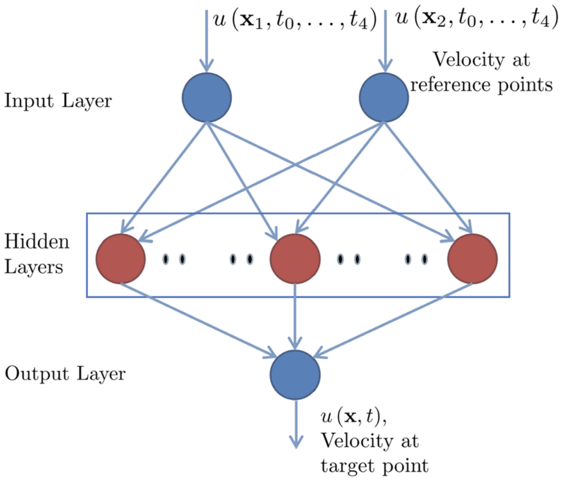

$u(\mathbf{x}_1, t_0, \ldots, t_4)$
$u(\mathbf{x}_2, t_0, \ldots, t_4)$
Velocity at reference points
Input Layer
Hidden Layers
Output Layer
$u(\mathbf{x}, t)$, Velocity at target point

FIG. 2. Schematic of the neural network employed for building the correlation between the velocities at the target point (the output layer) and the reference points (the input layer).

III. TRAINING OF THE MCPNN-FRONZEN MODEL

In this section, we describe the data generation and preparation, and the training of the MCPNN-frozen model.

A. Data generation and preparation

The data employed for training and testing the MCPNN-frozen model are from wall-modelled large-eddy simulation of turbulent channel flows. The virtual flow simulator (VFS-Wind) code,[45,46] which has been successfully applied to different flow problems,[47–53] is employed to carry out simulations of turbulent channel flows. In the LES module of the VFS-Wind code, the spatially filtered Navier–Stokes equations are solved. The governing equations are discretized in space using a second-order central difference scheme, and advanced in time using a second-order fractional step method.[54] The dynamic Smagorinsky model is employed for modeling the subgrid scales.[55] More details on the numerical methods of the VFS-Wind code can be found in the literature.[45]

In the simulated cases, three different surface roughness lengths, i.e., $k_0 = 0.001$, 0.01, 0.1 m, are considered. The Reynolds numbers based on the mean streamwise velocity U_h at $z = z_h$ and the thickness of the atmospheric boundary layer (δ) are $Re = U_h\delta/\nu = 4.89$, 4.71, 4.44×10^8 for the cases with $k_0 = 0.001$, 0.01, 0.1 m, respectively, where $z_h = 100$ m denotes the typical hub height of the wind turbine and ν is the viscosity of the air. The streamwise length and spanwise length of the computational domain are 22.50δ and 14.87δ, respectively, where $\delta = 1000$ m is the thickness of the atmospheric boundary, as shown in Fig. 3. The number of grids nodes are 1126 × 1488 × 152 in the streamwise, spanwise and vertical directions, respectively. Periodic boundary conditions are applied in the horizontal directions. The free-slip boundary condition is applied at the top boundary. At the bottom boundary, a wall model for rough wall, which provides the outer flow simulation with the wall shear stress boundary condition for the wall-parallel velocity component, is employed (Fig. 3). For the wall-normal velocity component, the non-penetration boundary condition is enforced at the wall. The wall shear stress employed in the wall model is computed using the logarithmic law, i.e., $\frac{U(z)}{u_\tau} = \frac{1}{\kappa}\ln\left(\frac{z}{k_0}\right)$, where u_τ is the friction velocity. Specifically, in the present implementation, the wall-parallel velocity component at the second off-wall point is employed to compute the wall shear stress with the logarithmic law in an iterative way. The obtained wall shear stress is then enforced when solving the momentum equation. The size of time step

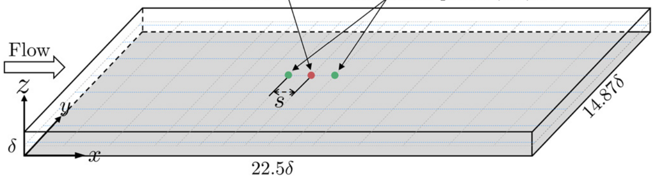

Target point, $\mathbf{x}$
Reference points , $\mathbf{x}_1, \mathbf{x}_2$
Flow
z
y
x
s
δ
22.5δ
14.87δ

FIG. 3. Schematic of the computational domain for the turbulent channel flow and the reference and target points for testing the MCPNN-frozen model.

is 0.01 U_b/δ, where U_b is the bulk velocity maintained as a constant in the simulation by enforcing a mean pressure gradient term in the streamwise direction. The total number of time steps is 30 000. The vertical profiles of the flow statistics are shown in Fig. 4 for different surface roughness lengths, where σ_u, σ_v, and σ_w are the standard deviations of velocity fluctuations in the streamwise, spanwise, and vertical directions, respectively. Significant differences are observed in terms of the mean streamwise velocity U and the turbulence statistics σ_u, σ_v, and σ_w.

From the simulations, the instantaneous velocity fields on the horizontal plane located at $z = z_h$ are saved for all time instants. Employing all the data for training and testing the model will be extremely time-consuming. In this work, the time series of data at 15 streamwise lines, which are located at 15 different spanwise locations, i.e., $y/\delta = 0.4$, 1.4, …, 13.4, 14.4, are employed for the task of this work. For the cases with the spanwise offset, additional time series of data are extracted for different offsets. There are 1125 grid points on each line for 30 000 instantaneous snapshots. In total, we have 15 × 1125 × 30 000 sample points. With these data, pairs of input and output are prepared for 20 target points with their streamwise location $x/\delta = 8.5$, 9, …, 17.5, 18, with the corresponding reference points determined by the distance from the target point (s). With the use of Taylor's frozen flow hypothesis, to predict the velocity at a time instant at the target point requires the velocity at the reference point at instants some time earlier or later based on the relative location (downstream or upstream) with the target point, such that not all of the time instants can be employed for the training and testing of the model. After the whole process, a dataset (pairs of input and output) with the size of 15 × 20 × 28 000 is obtained and is divided based on the locations into three sets of 60%, 20%, and 20% for model training, validation, and testing, respectively. It should be noted that the above procedure for data partition does not exactly follow the way of MCP, where the model should be trained based on the short-term data and applied to predict the long-term variations. This is mainly caused by the nature of the data we have in this work, which is statistically steady as a result of the relatively simple LES cases for data generation, that the data partition in time does not have a clear benefit. On the other hand, the employed data partition also brings a challenge to the model, as it will be trained using the data at one location and tested at the other different location. It is noted that the models for different roughness lengths and different separations are trained separately. The generalization capability of the trained model for different roughness lengths, different velocity component, and cases with different spanwise offsets between the target and the reference point will be tested.

B. Training of the MCPNN-Frozen model

A neural network contains an input layer, one or more hidden layers of neurons and an output layer. In this work, a simple neural

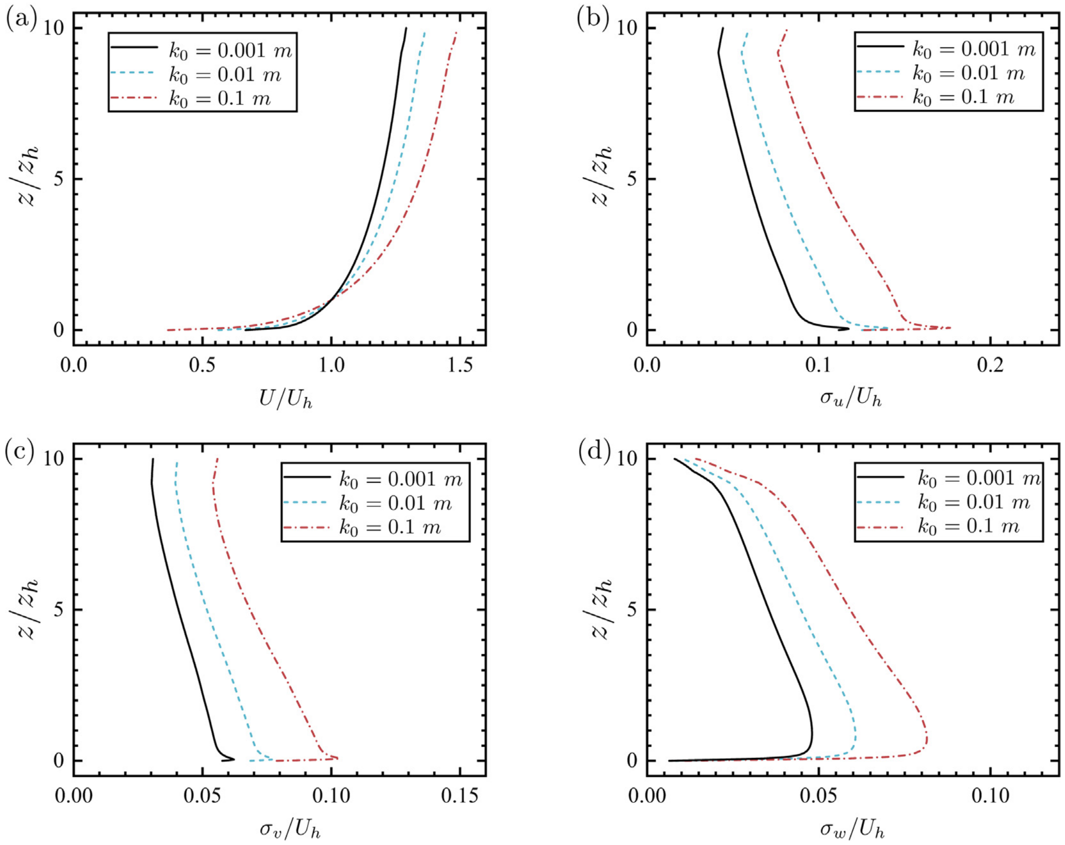


FIG. 4. Flow statistics of the channel flow with different ground roughness lengths.

TABLE I. Details of the employed neural network.

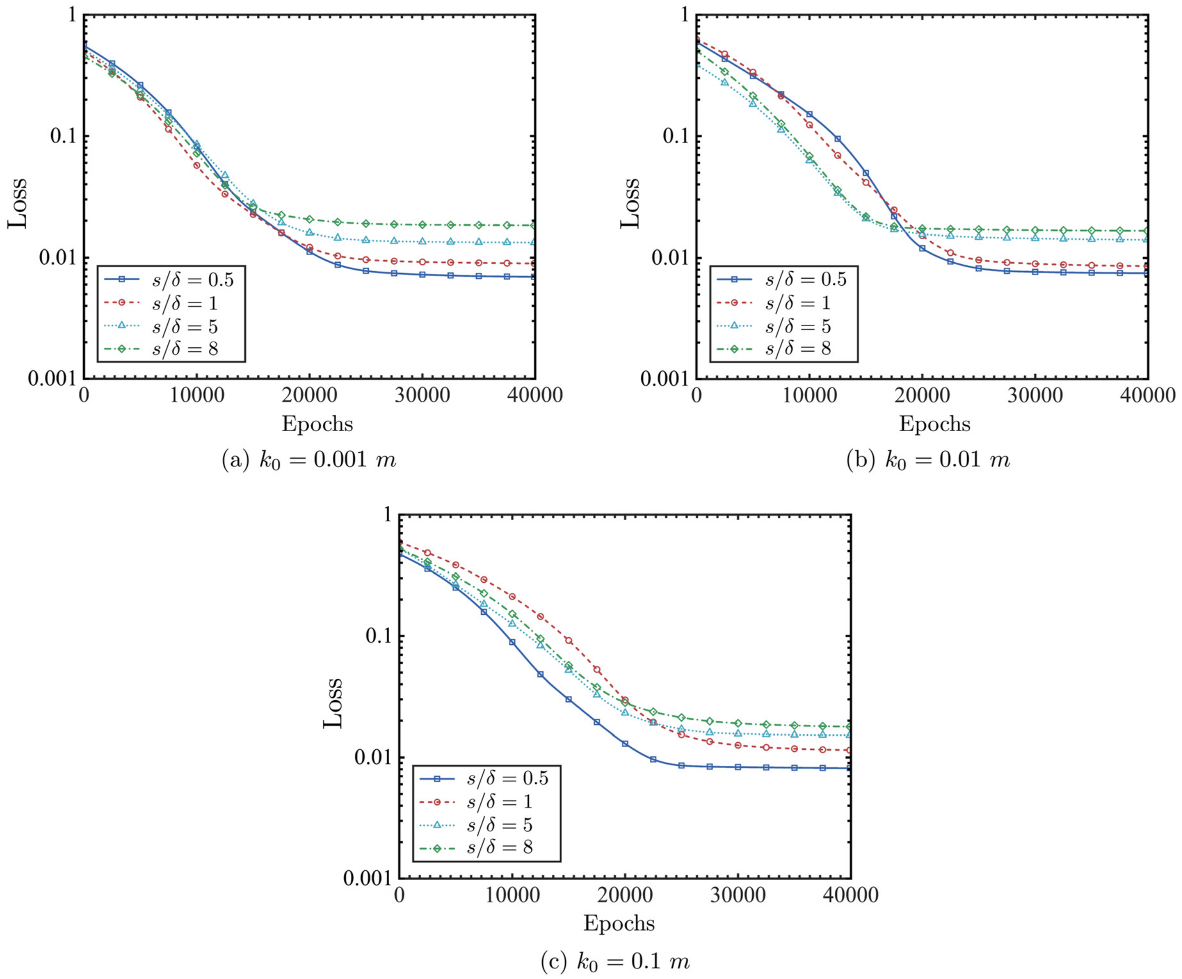
Number of inputs	10
Number of outputs	1
Number of layers	1
Number of neurons in layer	15
Training methodology	Error backpropagation (BP) scheme[56]
Batch size	1000
Learning rate	0.000 01
Epochs	2000
Loss function	Mean squared error (MSE)
Activation function	ReLU
Optimizer	Adam
Percentage of training dataset	75%
Percentage of verification dataset	25%

network with one hidden layer is employed with details shown in Table I. Different numbers of hidden layers as well as different number of neurons in each layer have been tested without seeing significant differences for the considered cases. The activation function used in this paper is the rectified linear unit (ReLU). The details of the employed neural network are shown in Table I.

In this work, we train the MCPNN-frozen model for different separations between the reference point and the target point separately. The loss functions for different cases are shown in Fig. 5. As seen, the loss decreases for about two orders of magnitude at about 15 000 to 25 000 epochs being different for cases with different k_0 and s values. For cases with different k_0, it is seen that the loss from the cases with smaller s are smaller when the training is converged.

IV. RESULTS AND ANALYSIS

In this section, we evaluate the prediction accuracy and generalization capacity of the MCPNN-frozen model using the test data from the same roughness length and those from different roughness lengths



(a) $k_0 = 0.001\ m$

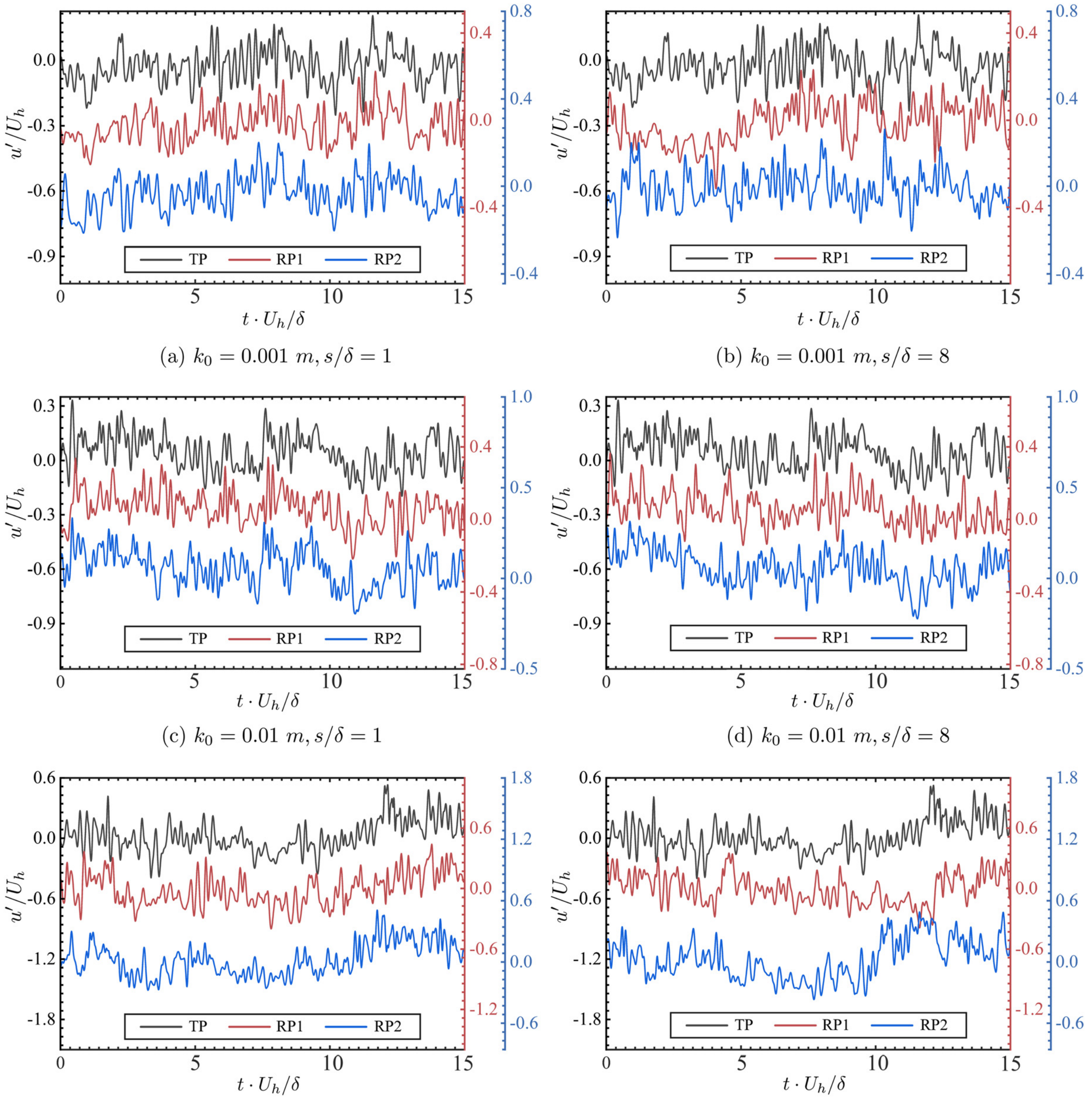
(b) $k_0 = 0.01\ m$

(c) $k_0 = 0.1\ m$

FIG. 5. Loss functions for training the MCPNN-frozen model for different roughness lengths and different separations between the reference point and the target point.

for ten different spacings between the reference point and the target point, i.e., $s/\delta = 0.2, 0.5, 1, 2, 3, 4, 5, 6, 7, 8$. Particularly, the predictions from the MCPNN-frozen model are compared with the predictions using only Taylor's frozen flow hypothesis.

First, we show in Fig. 6 the velocity fluctuations at the target point with those constructed from the reference points based on Taylor's frozen flow hypothesis. It is seen that the flow patterns change over time. For the cases with small separations between the reference point and target point, i.e., $s/\delta = 1$, it is observed from Figs. 6(a), 6(c), and 6(e) that the temporal variations at the target point are similar with those predicted based on Taylor's frozen flow hypothesis especially for the large scale fluctuations for different surface roughness lengths. Whereas for the cases with large values of s (i.e., $s/\delta = 8$) as shown in Figs. 6(b), 6(d), and 6(f), the temporal patterns at the target point are different from those predicted based on Taylor's frozen flow hypothesis, with less similarities in both large and small scale fluctuations.



(a) $k_0 = 0.001\ m, s/\delta = 1$

(b) $k_0 = 0.001\ m, s/\delta = 8$

(c) $k_0 = 0.01\ m, s/\delta = 1$

(d) $k_0 = 0.01\ m, s/\delta = 8$

(e) $k_0 = 0.1\ m, s/\delta = 1$

(f) $k_0 = 0.1\ m, s/\delta = 8$

FIG. 6. Comparison of the velocity fluctuations at the target point with those predicted from the reference points using Taylor's frozen flow hypothesis for (a) $k_0 = 0.001$ m, $s/\delta = 1$, (b) $k_0 = 0.001$ m, $s/\delta = 8$, (c) $k_0 = 0.01$ m, $s/\delta = 1$, (d) $k_0 = 0.01$ m, $s/\delta = 8$, (e) $k_0 = 0.1$ m, $s/\delta = 1$, and (f) $k_0 = 0.1$ m, $s/\delta = 8$. TP: the target point datasets, RP1: predictions using the data at the upstream reference point, and RP2: predictions using the data at the downstream reference point.

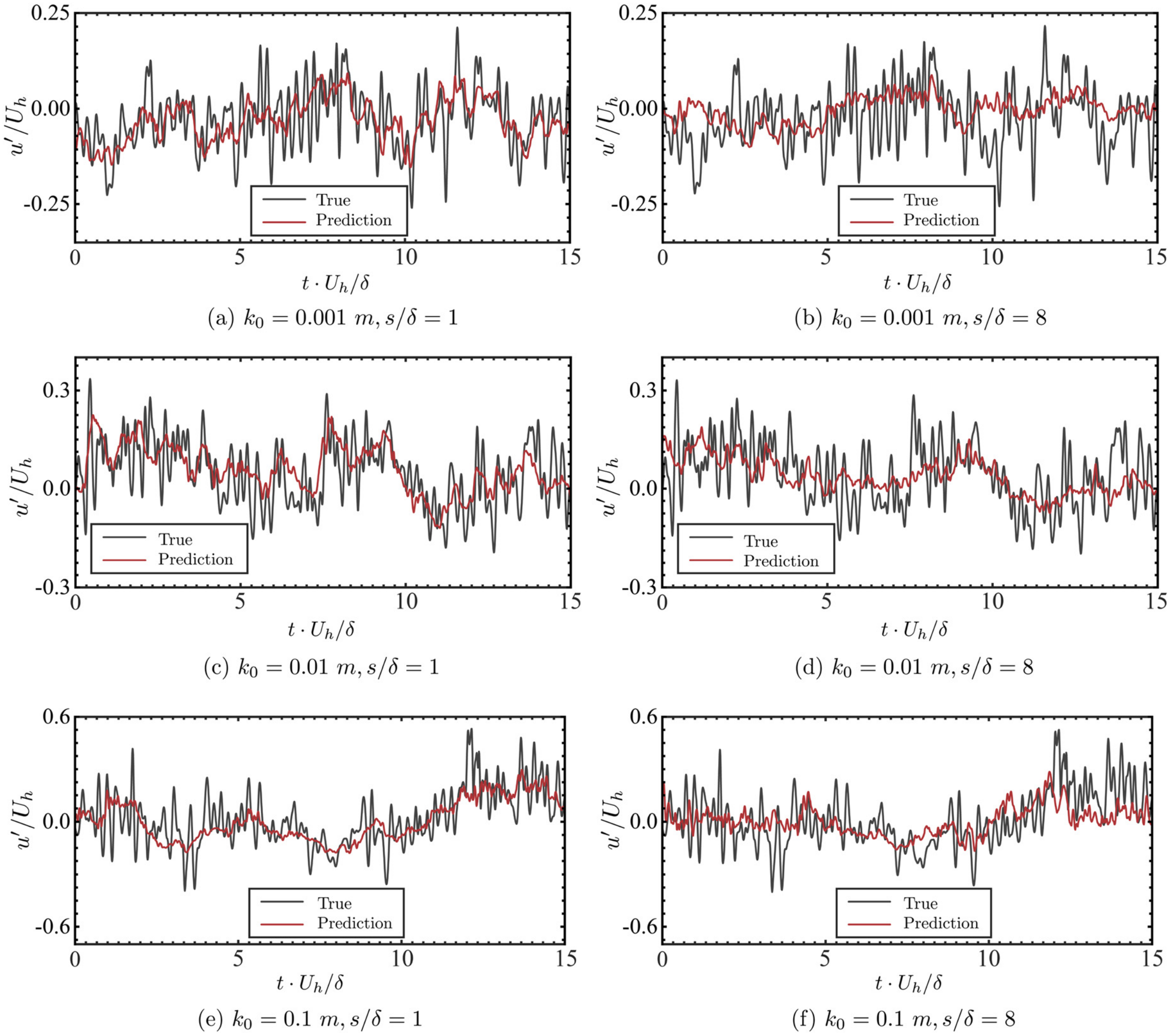
We then compare the time series of velocity fluctuations predicted from the MCPNN-frozen model with the real data in Fig. 7 for two different separations between the reference point and target point (i.e., $s/\delta = 1, 8$). It is seen that the MCP model in general predicts the large-scale variations of the velocity fluctuations for both values of s. The peaks of the unresolved scales, on the other hand, are significantly underpredicted. When increasing s/δ from 1 to 8, the scale resolved by the MCPNN-frozen model decreases as expected. After comparing the temporal variations of the model predictions and real data, the accuracy of the MCPNN-frozen model is examined via three performance metrics: the mean square error (MSE), the correlation coefficient ($C.C.$), and the coefficient of determination (R^2). Their definitions are given as follows. The MSE, which is one of the commonly used metrics to detect any departure from the real datasets in residual analysis, is defined as

$$\mathrm{MSE} = \left\langle \frac{1}{N}\sum_{i=1}^{N}\left(u'_P - u'_R\right)^2 \right\rangle, \tag{3}$$

where u'_R and u'_P are velocity fluctuations from real data and model predictions, respectively, $\langle\cdot\rangle$ represents the ensemble averaging, and N is the length of the dataset. The correlation coefficient between the model predictions and the real data $C.C.$, which is often used to assess the success rate of model learning,[24,57] is computed via the following expression:

$$C.C.(u') = \frac{\langle u'_R u'_P\rangle}{\sqrt{\langle (u'_R)^2\rangle}\sqrt{\langle (u'_P)^2\rangle}}. \tag{4}$$

The coefficient of determination, which is frequently used to measure the prediction accuracy of the model, is formulated as



(a) $k_0 = 0.001\ m, s/\delta = 1$

(b) $k_0 = 0.001\ m, s/\delta = 8$

(c) $k_0 = 0.01\ m, s/\delta = 1$

(d) $k_0 = 0.01\ m, s/\delta = 8$

(e) $k_0 = 0.1\ m, s/\delta = 1$

(f) $k_0 = 0.1\ m, s/\delta = 8$

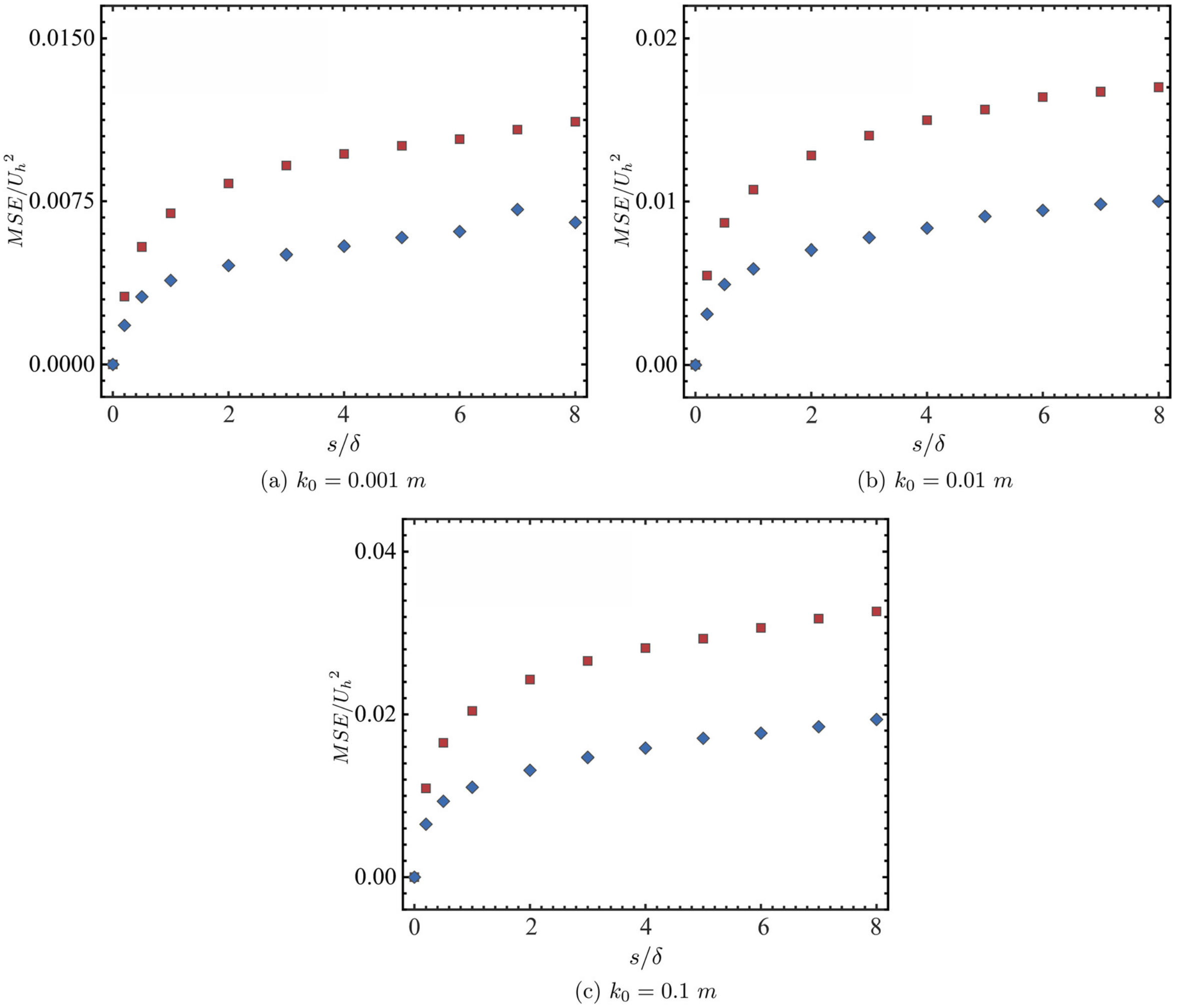
FIG. 7. Comparison of the predictions from the MCPNN-frozen model with the real data at the target point for (a) $k_0 = 0.001$ m, $s/\delta = 1$, (b) $k_0 = 0.001$ m, $s/\delta = 8$, (c) $k_0 = 0.01$ m, $s/\delta = 1$, (d) $k_0 = 0.01$ m, $s/\delta = 8$, (e) $k_0 = 0.1$ m, $s/\delta = 1$, and (f) $k_0 = 0.1$ m, $s/\delta = 8$.

$$R^2 = 1 - \frac{\langle (u'_P - u'_R)^2 \rangle}{\langle (u'_R - \langle u'_R \rangle)^2 \rangle}. \tag{5}$$

The MSEs of the MCPNN-frozen predictions are shown in Fig. 8 and compared with those of the predictions based on Taylor's frozen flow hypothesis. First, it is seen that the MCP-frozen model reduces the MSE for more than 30% when compared with the predictions based on Taylor's frozen flow hypothesis. With the increase in the separation between the reference point and the target point s, the MSE increases at a high rate for $s/\delta < 1$, while increases gradually at a low rate when further increasing s/δ. Furthermore, it is seen that the MSE is higher for higher surface roughness lengths.

The prediction accuracy is further examined in Fig. 9 by plotting the correlation coefficients ($C.C$) between the predicted velocity fluctuations and the real data. For all the three cases with different ground roughness lengths, it is observed that the values of $C.C.$ from the MCPNN-frozen predictions are higher than those predicted based on Taylor's frozen flow hypothesis, indicating that neural networks' capability in compensating the physics beyond Taylor's frozen flow hypothesis. For different separations between the reference point and the target point (s), the correlation coefficient $C.C.$ gradually decreases with the increase in s as expected. It is seen that the value of $C.C.$ is higher than 0.5 for $s/\delta < 3$ for $k_0 = 0.001, 0.01$ m, which is somewhat lower than 0.5 at $s/\delta = 3$ for $k_0 = 0.1$ m. When further increasing the value of s, $C.C.$ reduces to approximately 0.3 and 0.25 at $s/\delta = 8$ for $k_0 = 0.001, 0.01$ m and $k_0 = 0.1$ m, respectively.

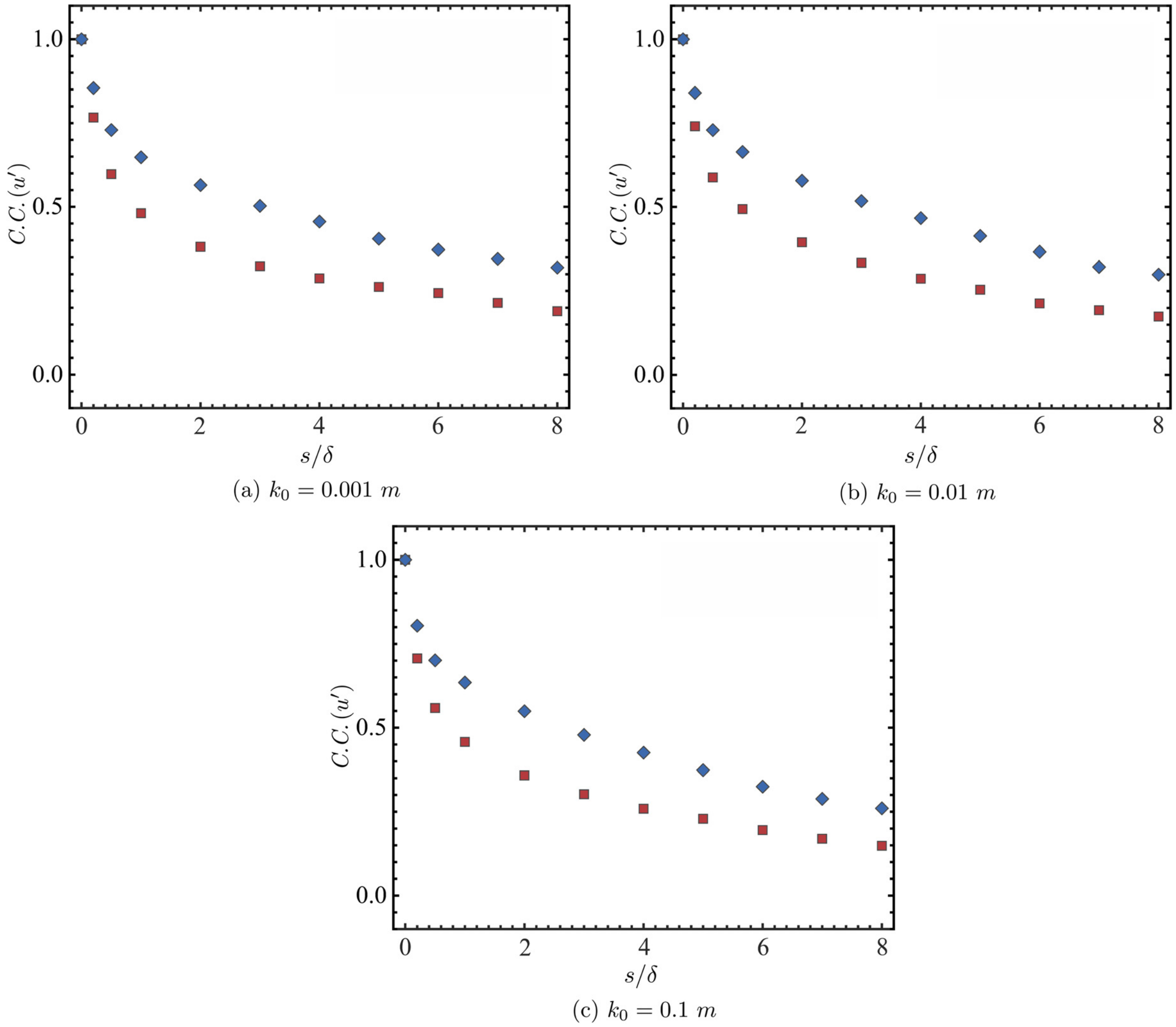
Figure 10 shows the coefficient of determination (R^2) computed using Eq. (5). It is clearly seen that in all cases, the R^2 values for the MCPNN-frozen model are greater than those based on Taylor's frozen flow hypothesis, demonstrating that the capability of the neural network in compensating for the turbulence physics, which is not described by Taylor's frozen flow hypothesis. As expected, the coefficient of determination gradually decreases when increasing the separation (s) between the reference point and the target point. When



(a) $k_0 = 0.001\ m$ (b) $k_0 = 0.01\ m$

(c) $k_0 = 0.1\ m$

FIG. 8. MSE of the predictions from the MCPNN-frozen model (blue diamond symbols) and those based on Taylor's frozen flow hypothesis (red square symbols) for different separations between the reference point and the target point for (a) $k_0 = 0.001$ m, (b) $k_0 = 0.01$ m, and (c) $k_0 = 0.1$ m.



(a) $k_0 = 0.001\ m$

(b) $k_0 = 0.01\ m$

(c) $k_0 = 0.1\ m$

FIG. 9. Correlation coefficients *C.C.* between the predictions and the real data for the MCPNN-frozen model (blue diamond symbols) and the model only using Taylor's frozen flow hypothesis (red square symbols) for different separations between the reference point and the target point for (a) $k_0 = 0.001$ m, (b) $k_0 = 0.01$ m, and (c) $k_0 = 0.1$ m.

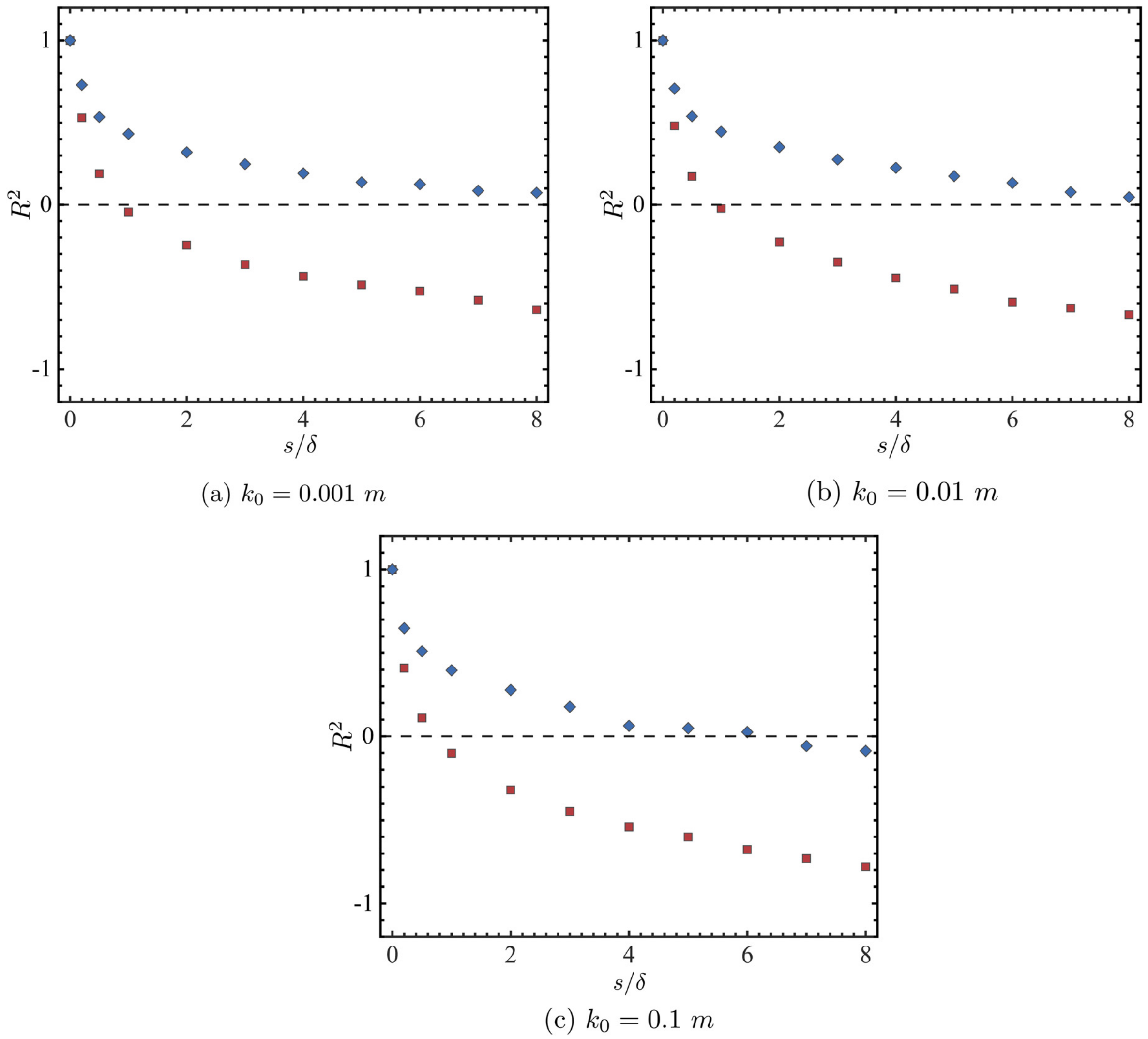
$s/\delta < 2$, the value of R^2 is higher than 0.3 for $k_0 = 0.001, 0.01$ m, and it is slightly lower than 0.3 for $k_0 = 0.1$ m. Finally, with the increase in s, the value of R^2 decreases to about 0.05 for $k_0 = 0.001, 0.01$ m, and about −0.1 for $k_0 = 0.1$ m when $s/\delta = 8$.

To further examine the characteristics of the velocity fluctuations predicted by the MCPNN-frozen model, we plot the time correlation function $R_{u'u'}$ in Fig. 11, which indicates the degree of correlation of velocity fluctuations for different time separations and is defined as follows:

$$R_{u'u'}(t, t+\tau) = \frac{\langle u'(t)u'(t+\tau)\rangle}{\sqrt{\langle (u'(t))^2\rangle}\sqrt{\langle (u'(t+\tau))^2\rangle}}, \qquad (6)$$

where τ represents the separation between two temporal instants. It is observed in Fig. 11 that the overall variations of $R_{u'u'}$ are well predicted by the MCPNN-frozen model. For small temporal separations (τ), it is seen that the values of $R_{u'u'}$ from the model predictions are higher than those of the real data before $R_{u'u'}$ reaches zero for all three roughness lengths, which indicates the predicted small scales do not possess the structures of the real data. Decreasing the separation between the reference point and the target point (s) does not improve the prediction of $R_{u'u'}$ for small temporal separations (τ). The most important and interesting observation is that the velocity fluctuations predicted by the MCPNN-frozen model decorrelates at approximately the same temporal separation τ (where $R_{u'u'}$ reaches zero), which implies that the large-scale patterns are well captured by the MCPNN-frozen model.

To further examine the generalization capacity of the MCPNN-frozen model. We apply the MCPNN-frozen model trained using the data with $k_0 = 0.001$ m to the cases with $k_0 = 0.01, 0.1$ m. The performance metrics *MSE* and *C.C.* and R^2, and the temporal correlation function $R_{u'u'}$ are examined in Fig. 12. It is seen in Figs. 12(a) and 12(b) that



(a) $k_0 = 0.001\ m$

(b) $k_0 = 0.01\ m$

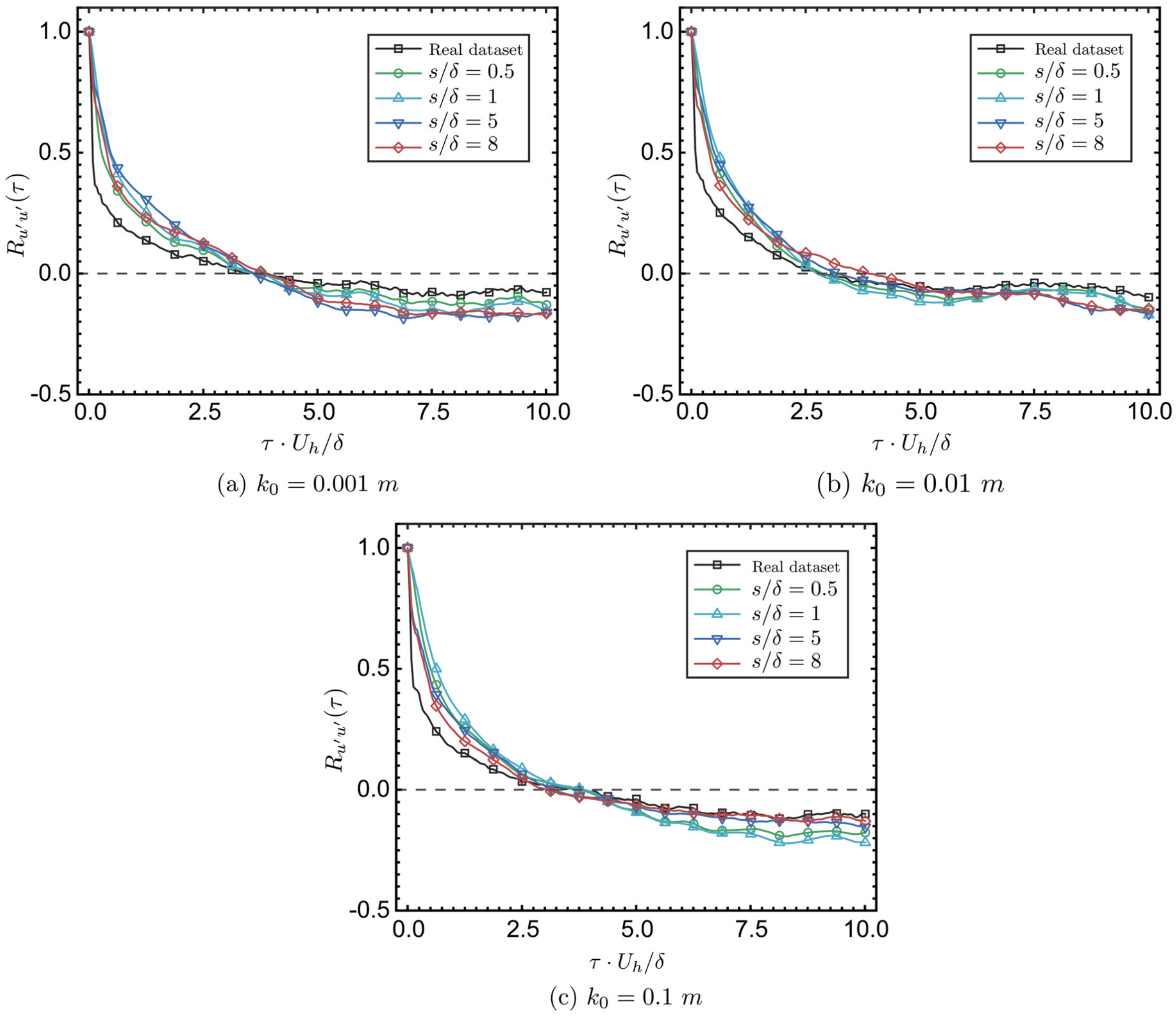
(c) $k_0 = 0.1\ m$

FIG. 10. Coefficients of determination (R^2) for the MCPNN-frozen model (blue diamond symbols) and the model only using Taylor's frozen flow hypothesis (red square symbols) for different separations between the reference point and target point for (a) $k_0 = 0.001$ m, (b) $k_0 = 0.01$ m, and (c) $k_0 = 0.1$ m.

the MSE values predicted by the MCPNN-frozen model trained using the data with $k_0 = 0.001$ m are slightly higher than those predicted by the model trained using the data with the same k_0. As shown in Figs. 12(c) and 12(d), the *C.C.* values from the MCPNN-frozen model trained using a k_0 different from the test data, on the other hand, are collapsed with the model trained using k_0 the same as the test data. In Figs. 12(e) and 12(f), the coefficients of determination R^2 for the MCPNN-frozen model trained using the data with $k_0 = 0.001$ m are compared with the one trained using the data with the same k_0, showing that the latter performs better, even though no significant differences are observed for the correlation coefficients *C.C.* The time correlation functions $R_{u'u'}$ from cases with different separations between the reference point and the target point are further examined in Figs. 12(g) and 12(h). It is seen that $R_{u'u'}$ predicted by the MCPNN-frozen model with k_0 different from the test data reaches zero at approximately the same value of τ when compared with the real data, indicating that the employed model is able to predict the large-scale variations of the velocity fluctuations.

Here, the generalization capacity of the MCPNN-frozen model is tested by applying the model trained using the streamwise velocity fluctuations to the spanwise velocity fluctuations ($k_0 = 0.001$ m). The temporal variations of the predictions are shown in Fig. 13 and compared with the real data. It is seen that large-scale variations in time are well predicted by the MCPNN-frozen model, while the small-scale fluctuations are underpredicted, which is also observed when predicting the streamwise velocity fluctuations.

The MSE, *C.C.*, R^2, and $R_{v'v'}$ are further examined in Fig. 14 to quantitatively examine the performance of the model trained using the streamwise velocity fluctuations in predicting the spanwise velocity fluctuations. It is observed in Figs. 14(a)–14(c) that the MCPNN-frozen



(a) $k_0 = 0.001\ m$

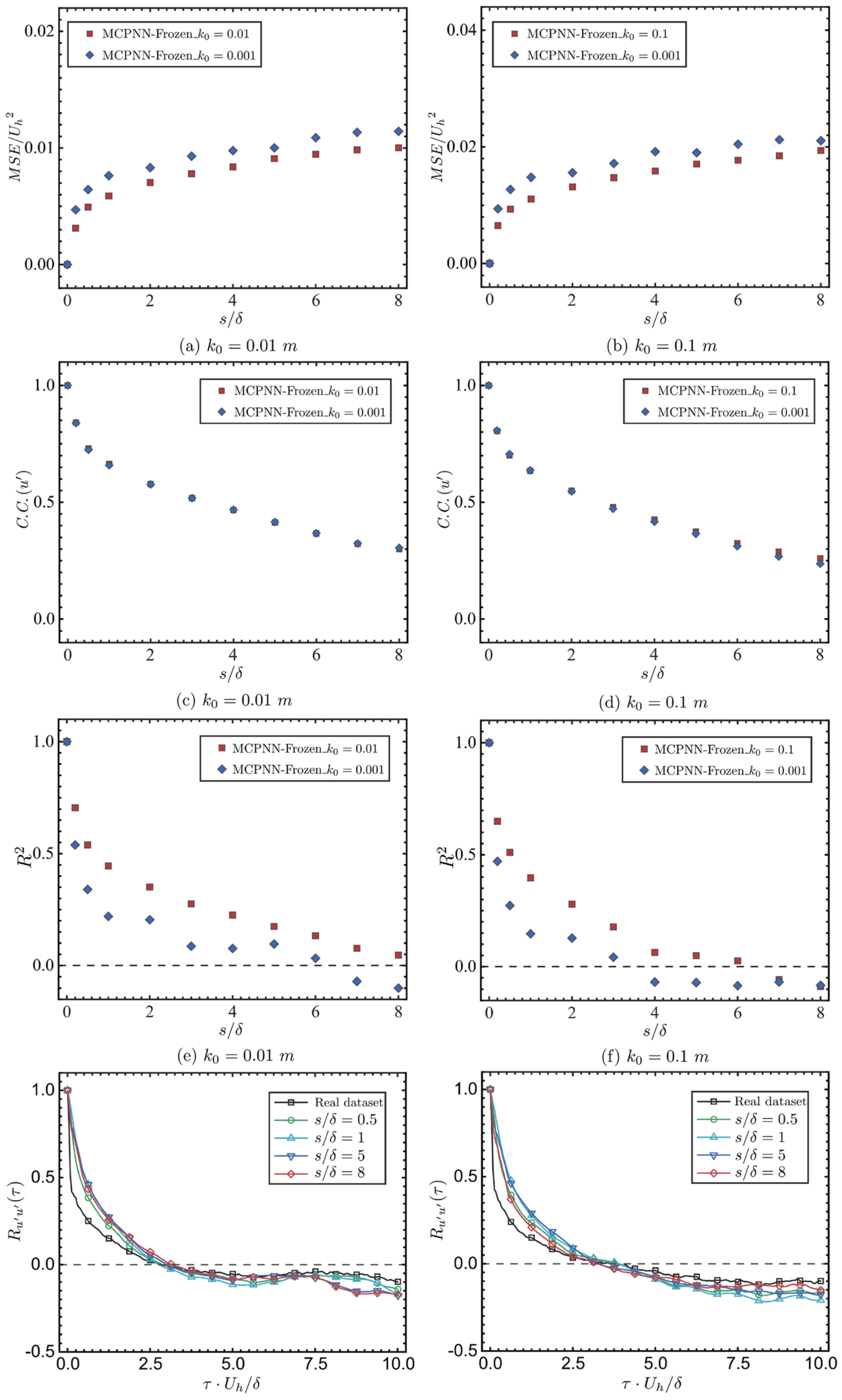
(b) $k_0 = 0.01\ m$

(c) $k_0 = 0.1\ m$

FIG. 11. Time correlation functions of the real data and the model predictions for different separations between the reference point and the target point for (a) $k_0 = 0.001$ m, (b) $k_0 = 0.01$ m, and (c) $k_0 = 0.1$ m.

model, even though it is not trained using the spanwise velocity fluctuations, outperforms the model based on Taylor's frozen hypothesis. As for the time correlation function $R_{v'v'}$, it is observed in Fig. 14(d) that the values of $R_{v'v'}$ of the predictions are higher than that of the real data, which is also observed for the streamwise velocity. The decorrelation time, when $R_{v'v'}$ reaches zero, is observed being higher for the predicted velocity fluctuations, which is improved for smaller separation s between the reference point and the target point.

To confirm that the MCPNN-frozen model is capable of predicting the large-scale patterns of the temporal variations of velocity fluctuations, two scenarios are considered: (1) we compare the time-filtered velocity fluctuations from the model predictions and the real dataset; and (2) we train a new model using the time-filtered data and test the new model. In Fig. 15, the results from the first scenario are presented. One additional case [case 2 as shown in Fig. 15(a)] is carried out, which employs the filtered velocity as the input features, in comparison with case 1, where the original velocity is employed as the input features (as done in other cases). For both case 1 and case 2, the MCPNN-frozen model is trained using the original velocity data without additional filtering. For the filtered velocity shown in Fig. 15(a), the filter width is $t_f U_h/\delta = 5.650$. It is seen that MCPNN-frozen model captures the large-scale variations in an acceptable way for $s/\delta = 8$, even though the correlation coefficient *C.C.* without filtering is approximately 0.3. The values of *C.C.* for different filter widths are shown in Fig. 15(b). As seen, the correlation *C.C.* is significantly increased from approximately 0.3 to 0.55 for filter width $t_f U_h/\delta \geq 5.650$. The results from the second scenario, where the model is trained and tested using the filtered datasets, are presented in Fig. 16. The MSE, *C.C.*, and R^2 are examined to quantify the performance of the model. Compared with the performance of the model trained using the unfiltered data, it is seen that the performance is greatly improved especially for the MCP-frozen model in terms of all the three metrics. Specifically, the figure shows that the correlation coefficient *C.C.* is higher than 0.7, and the coefficient of determination



(a) $k_0 = 0.01\ m$

(b) $k_0 = 0.1\ m$

(c) $k_0 = 0.01\ m$

(d) $k_0 = 0.1\ m$

(e) $k_0 = 0.01\ m$

(f) $k_0 = 0.1\ m$

(g) MCPNN-Frozen_$k_0 = 0.001$,$k_0 = 0.01\ m$

(h) MCPNN-Frozen1_$k_0 = 0.001$,$k_0 = 0.1\ m$

FIG. 12. Performance test of the MCPNN-frozen model trained using the data with the k_0 (i.e., $k_0 = 0.001$ m) different from the test data (i.e., $k_0 = 0.01, 0.1$ m) for (a) and (b) MSE, (c) and (d) *C.C.*, (e) and (f) R^2, and (g) and (h) $R_{u'u'}$.

is close to 0.5 when the size of filter $t_f U_h/\delta$ is approximately higher than 2, indicating that the MCPNN-frozen model captures well the large-scale variations for a large separation between the reference point and the target point, e.g., $s/\delta = 8$.

So far, the target point is located on the same line as the reference points. It is very likely that a spanwise offset (d) exists in real-life applications either because of the relative locations of the reference point and the target point or the difference of wind directions between the

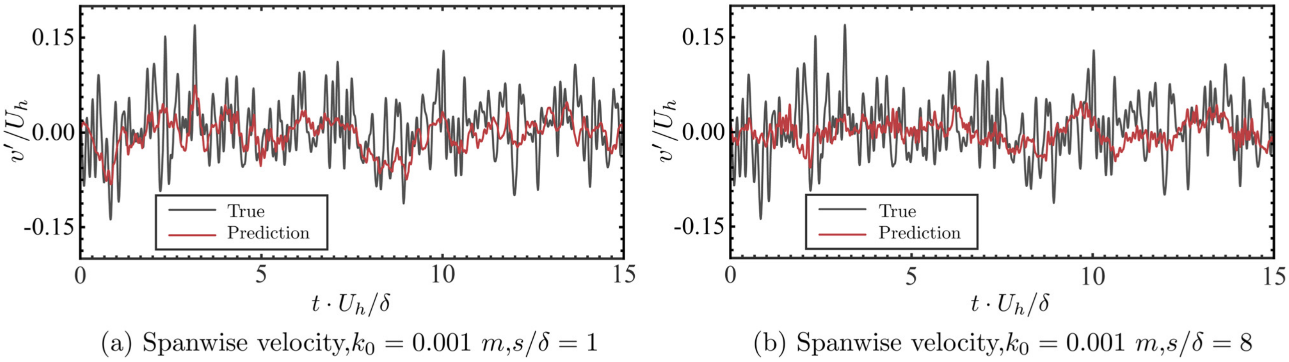


FIG. 13. Temporal variations of the spanwise velocity fluctuations predicted by the MCPNN-frozen model trained using the data of the streamwise velocity fluctuations for (a) $s/\delta = 1$ and (b) $s/\delta = 8$.

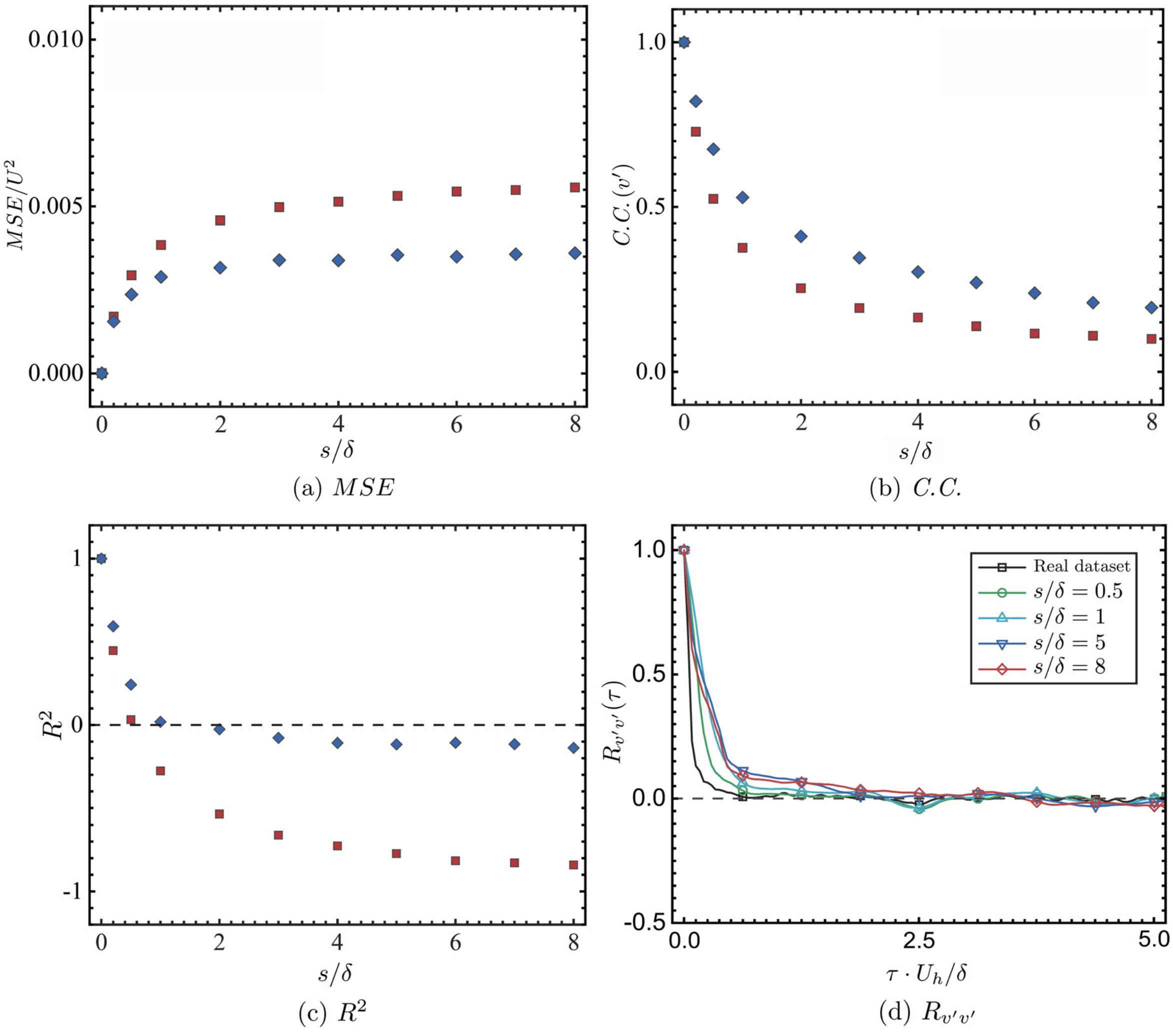
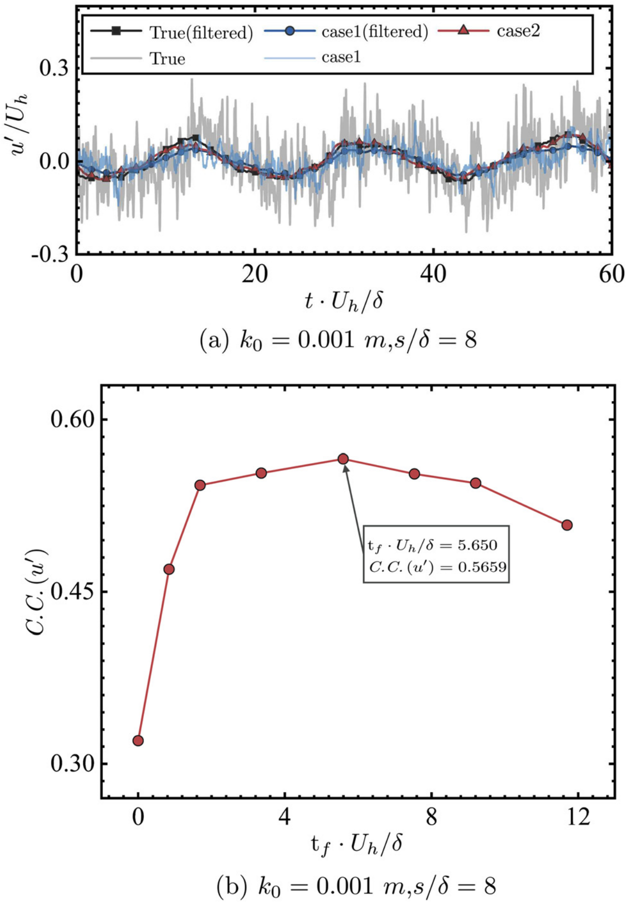


FIG. 14. Performance test in predicting the spanwise velocity fluctuations using the MCPNN-frozen model trained using the streamwise velocity fluctuations for (a) *MSE*, (b) *C.C.*, (c) R^2, and (d) $R_{v'v'}$. In figures (a)–(c), blue diamond symbols and red square symbols are for the predictions from the MCPNN-frozen model and the predictions based on Taylor's frozen flow hypothesis, respectively.



(a) $k_0 = 0.001\ m, s/\delta = 8$

(b) $k_0 = 0.001\ m, s/\delta = 8$

FIG. 15. (a) Comparison of the filtered velocity fluctuations (width of the temporal filter $t_f U_h/\delta = 5.650$) between the predictions from the MCPNN-frozen model and real data; (b) correlation coefficient *C.C.* of the filtered velocity fluctuations for different filter widths. In figure (a): in case 1, the original velocity fluctuations are employed as the input; in case 2, the filtered velocity fluctuations are employed as the input. In both case 1 and case 2, the MCPNN-frozen model is trained using the original data without filtering.

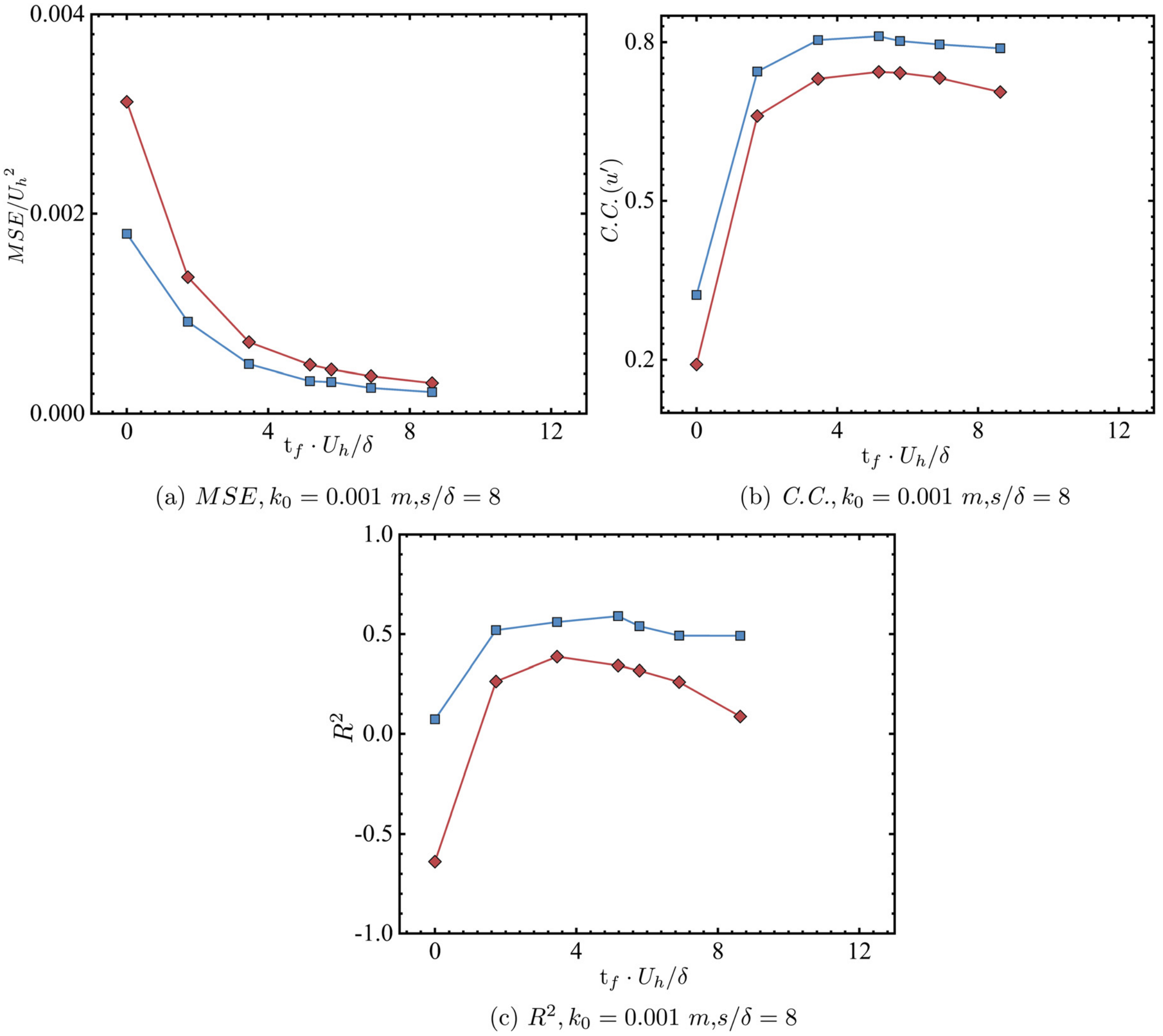
training data and the testing data (or actual application). It has been shown in Fig. 16 that the proposed MCPNN-frozen model can predict well the large-scale temporal variations. Here, we examine the performance of MCPNN-frozen model trained using the filtered data without a spanwise offset for cases with different spanwise offsets. Specifically, the model, which is trained using the $k_0 = 0.001$ m data filtered with $t_f U_h/\delta = 5.180$ for $s/\delta = 8$ without a spanwise offset, is evaluated for cases with different spanwise offsets. In these cases, the two reference points are located at *d* from the target point in the opposite spanwise direction. Four different values of *d*, i.e., $d/\delta = 0.0$, 0.1, 0.2, 0.5, 0.8, are considered. The results from the case with $d/\delta = 0.1$ are shown in Fig. 17. As seen in Fig. 17(a), the large-scale variations are well predicted by the MCPNN-frozen model. It is shown in Fig. 17(b) that the correlation coefficient *C.C.* is higher than 0.65 for $d/\delta = 0.1$, 0.2. For higher values of spanwise offsets, the values of *C. C.* are approximately higher than 0.45 even for $d/\delta = 0.8$. Compared with the model using only Taylor's frozen flow hypothesis, the performance of the MCP-frozen model is observed being significantly better.

V. CONCLUSIONS

In this work, a measure–correlate–predict model, which is based on neural networks and Taylor's frozen flow hypothesis and dubbed as MCPNN-frozen model, was proposed for wind resource assessment and tested using turbulent channel flows with different surface roughness lengths.

In the proposed MCPNN-frozen model, the velocity fluctuation data at the two reference points, which are located upstream and downstream of the target point, respectively, are employed to predict the velocity fluctuation at the target point. A neural network is trained to correlate the velocity fluctuations at the reference point and the target point. In the employed neural network, the velocity fluctuations from the reference point at five successive time instants ($N_i = 5$) with the temporal separation Δt are employed as the input features. Taylor's frozen flow hypothesis is employed to process the velocity fluctuations at the reference points before they are employed as input features in the neural network. In the employed neural network, one hidden layer with fifteen neurons is employed. The effects of the number of hidden layers (N_L), the number of temporal instants (N_i), the size of Δt, and the number of reference locations on the performance of the model are tested (in Appendix A). The test results confirm that the employed parameters of the MCPNN-frozen model are appropriate for the considered cases.

The proposed MCPNN-frozen model was tested using the test data from turbulent channel flows with different surface roughness lengths, i.e., $k_0 = 0.001, 0.01,\ 0.1\,\mathrm{m}$, for different separations (*s*) between the reference point and the target point. The prediction accuracy and generalization capacity were evaluated with the mean square error (MSE), the correlation coefficient between the predictions and the real data (*C.C.*), the coefficient of determination (R^2), and the time correlation function ($R_{u'u'}$). The results show that the performance of the MCPNN-frozen model is significantly higher than the model based on Taylor's frozen flow hypothesis. For small separations between the reference point and the target point (roughly $s/\delta \leq 3$, where δ is the boundary layer thickness), the correlation coefficient *C. C.* is higher than 0.5, and the coefficient of determination R^2 is approximately higher than 0.3 when $s/\delta \leq 2$. When increasing the values of *s*, *C.C.* gradually decreases and reaches approximately 0.3 and 0.25 at $s/\delta = 8$ for $k_0 = 0.001$, 0.01 m, and $k_0 = 0.1$ m, respectively, and the value of R^2 decreases to 0.05 for $k_0 = 0.001, 0.01$ m and approximately -0.1 for $k_0 = 0.1$ m. The generalization capacity of the MCPNN-frozen model was tested using the test data with roughness lengths different from the training data, showing performance similar with the model trained using the data with the same roughness length as the test data. The generalization capacity of the MCPNN-frozen model was further tested by applying the model trained using the streamwise velocity fluctuations to predict the spanwise velocity fluctuations. Performance similar to that of the streamwise velocity fluctuations is observed. Further analyses show that the MCPNN-frozen model is able to capture the large-scale patterns of velocity fluctuations although the correlation coefficient *C.C.* is low. The performance of the MCPNN-frozen model trained using the filtered data is also examined. The results show that the correlation coefficient *C.C.* is higher



(a) $MSE, k_0 = 0.001\ m, s/\delta = 8$

(b) $C.C., k_0 = 0.001\ m, s/\delta = 8$

(c) $R^2, k_0 = 0.001\ m, s/\delta = 8$

FIG. 16. Performance test of the MCPNN-frozen model trained using the filtered streamwise velocity fluctuations for (a) *MSE*, (b) *C.C.*, and (c) R^2, for different filter sizes t_f, for the cases with $s/\delta = 8$ and $k_0 = 0.001$. Blue square symbols: the MCPNN-frozen model; red square diamond symbols: the model only using Taylor's frozen flow hypothesis.

than 0.7, and the coefficient of determination R^2 is higher than 0.5 when the size of the filter $t_f U_h/\delta$ is approximately higher than 2. Furthermore, it is found that the MCPNN-frozen model trained using the filtered data without a spanwise offset can well predict the large-scale temporal variations for different spanwise offsets between the target point and the reference points.

The idea of the proposed MCPNN-frozen model is to leverage the power of Taylor's frozen flow hypothesis and neural networks. From the neural network side, Taylor's frozen flow hypothesis provides a way for selecting the most related data as the input features. From the side of Taylor's frozen flow hypothesis, the neural network supplements the missing flow physics in a black box way. Obviously, more needs to be done especially for those high frequency, small-scale fluctuations. Furthermore, the proposed MCPNN-frozen model was focused on relatively simple cases, in which the terrain is flat, and the thermal stratification is neutral. Developing models accounting for more effects in atmospheric turbulence will be carried out in our future work.

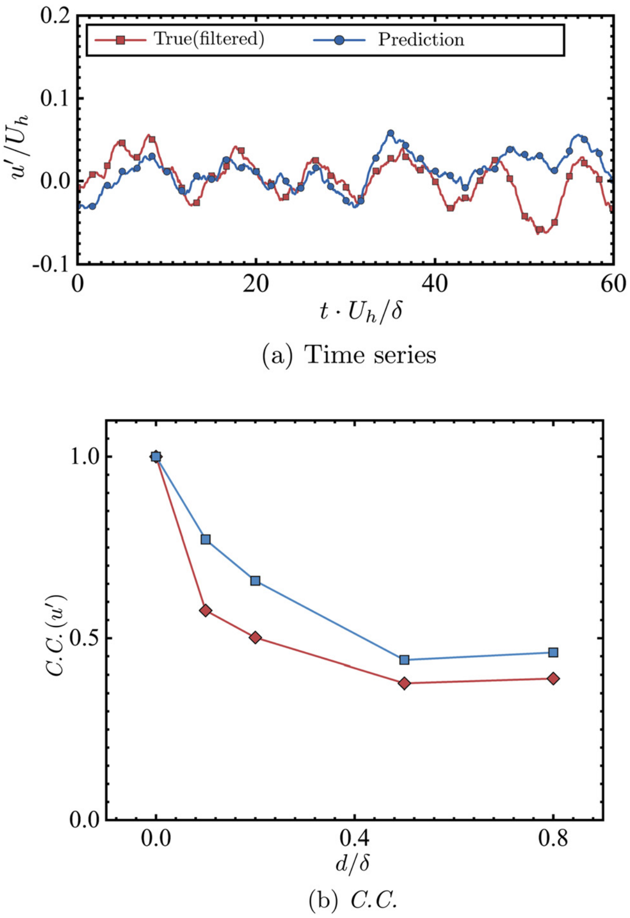
ACKNOWLEDGMENTS

This work was supported by NSFC Basic Science Center Program for "Multiscale Problems in Nonlinear Mechanics" (No. 11988102), the National Natural Science Foundation of China (No. 12172360), Institute of Mechanics CAS, and Chinese Academy of Sciences.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.



(a) Time series

(b) $C.C.$

FIG. 17. Performance test of the MCPNN-frozen model for different spanwise offsets between the reference point and target point for (a) time series for $d/\delta = 0.1$, and (b) correlation coefficient $C.C.$. The MCPNN-frozen model is trained using the filtered data for $s/\delta = 8$ without a spanwise offset. The ground roughness length $k_0 = 0.001$ m. The testing data are filtered with $t_f U_h/\delta = 5.180$ the same as the training data. In (b), blue square symbols and red diamond symbols are for the MCPNN-frozen model and the model using only Taylor's frozen flow hypothesis, respectively.

DATA AVAILABILITY

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

APPENDIX A: EVALUATION OF DIFFERENT MODEL PARAMETERS

In this appendix, we test the effects of the number of hidden layers employed in the neural network, the number of temporal instants (N_i), the time increment Δt between two successive time steps for the data employed for prediction at the reference point, and the number of reference points on the performance of the MCPNN-frozen model. The data from the case with $k_0 = 0.001$ m (if not otherwise specified) are employed for the test carried out in this appendix. When evaluating the effect of one particular parameter, other parameters are kept the same as those of the model employed throughout the paper. The performance of the MCPNN-frozen model with different numbers of hidden layers is compared in Fig. 18. It is observed that although the MSE slightly decreases when N_L increases from 1 to 2, the correlation between the predictions and real data also slightly decreases. When further increasing N_L, improvements are not observed either.

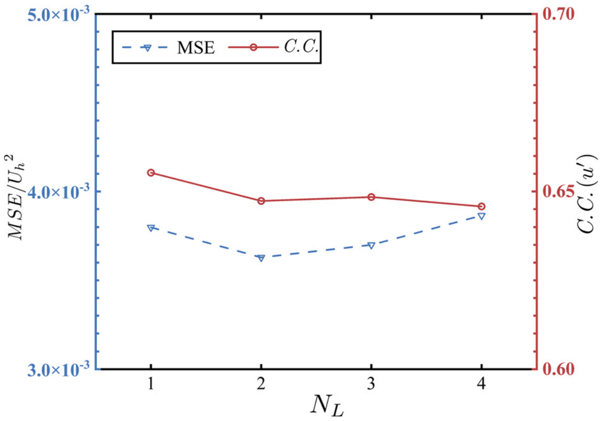


FIG. 18. Effects of different numbers of hidden layers on the performance of the MCPNN-frozen model.

How the number of temporal instants (N_i) selected at the reference point affects the model performance is examined in Fig. 19. It is seen that MSE gradually decreases and $C.C.$ gradually increases with the increase in the number of temporal instants until $N_i = 5$, after which no significant improvement is observed. In the rest of paper, $N_i = 5$ is employed.

The effects of Δt on the performance of the MCPNN-frozen model are evaluated in Fig. 20 by plotting the MSE and correlation coefficients ($C.C.$). It is seen that the MSE first decreases and then increases when increasing the value of Δt, that there is an optimal Δt for the maximal performance. The values of the optimal Δt are slightly different for different surface roughness lengths, which,

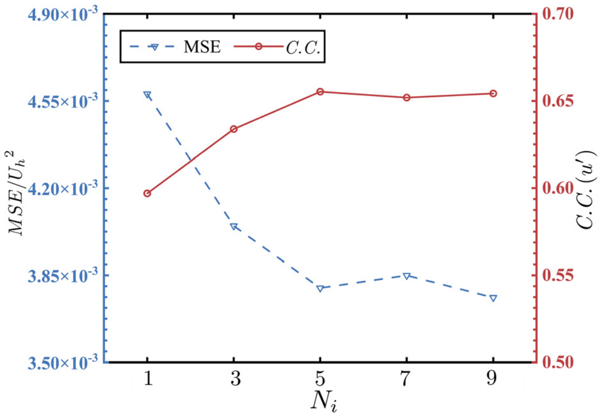
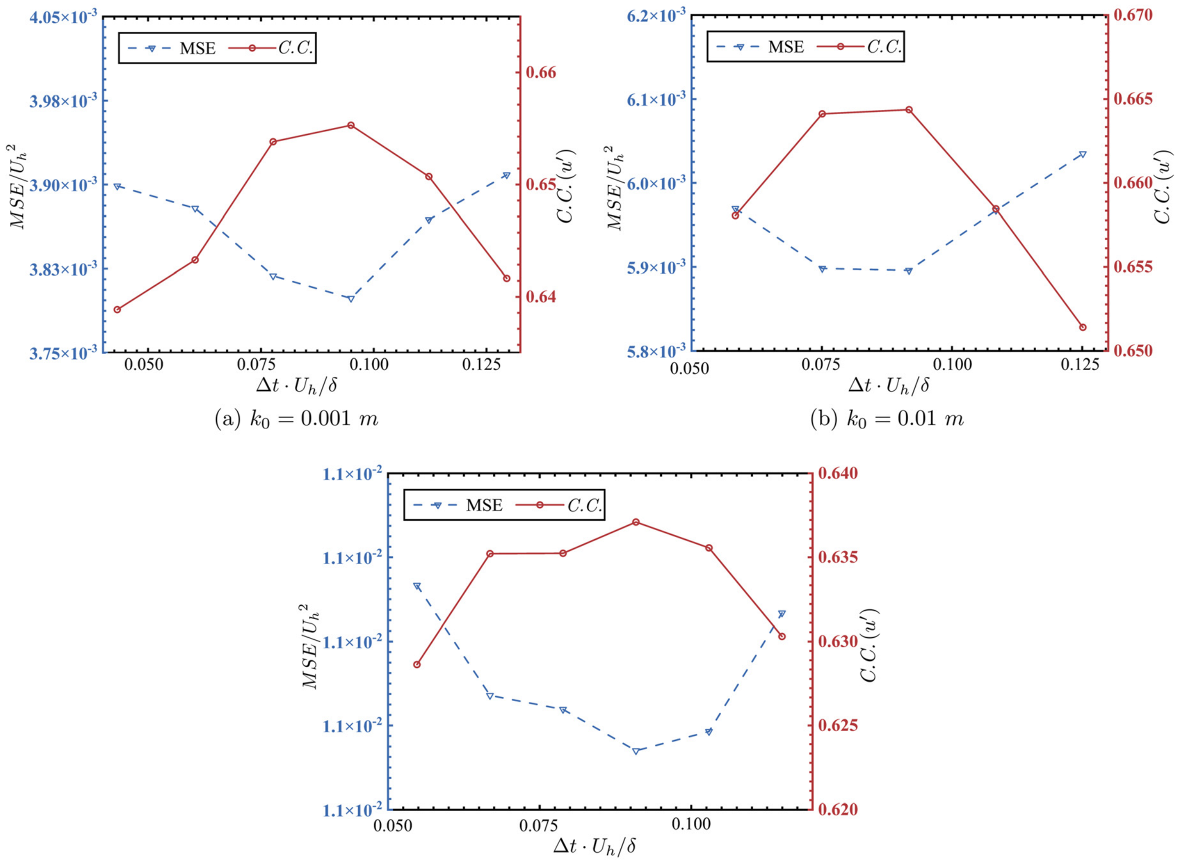


FIG. 19. Effects of different numbers of temporal instants at the reference point on the performance of the MCPNN-frozen model.



(a) $k_0 = 0.001\ m$

(b) $k_0 = 0.01\ m$

(c) $k_0 = 0.1\ m$

FIG. 20. Effects of different time intervals Δ on the performance of the MCPNN-frozen model.

however, nearly fall into the same range (i.e., $\Delta tU/\delta \in [0.075, 0.10]$) for different cases. In the test of the model throughout the paper, the corresponding optimal value of Δt is employed for each roughness length.

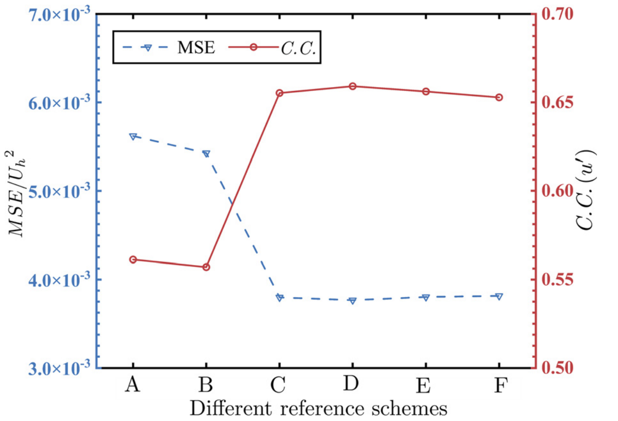


FIG. 21. Effects of different choices of reference points on the performance of the MCPNN-frozen model using the test data with $k_s = 0.001$ m.

Figure 21 shows how different schemes for choosing the reference points affect the model performance. Six different schemes as listed in Table II are examined. As shown in Fig. 21, the performance of the model with reference points at both upstream and downstream locations (i.e., schemes C, D, E, and F) is significantly better than that with reference points located on one side of the

TABLE II. Different schemes for choosing the reference points.

Scheme	Different numbers and locations of the reference points
A	Downstream two at $x+\delta$ and $x+2\delta$
B	Upstream two at $x-\delta$ and $x-2\delta$
C	Upstream one at $x-\delta$, downstream one at $x+\delta$
D	Upstream two at $x-\delta$ and $x-2\delta$, downstream one at $x+\delta$
E	Upstream two at $x-\delta$ and $x-2\delta$, downstream two at $x+\delta$ and $x+2\delta$
F	Upstream three at $x-\delta$, $x-2\delta$, and $x-3\delta$, downstream two at $x+\delta$ and $x+2\delta$

target point (i.e., schemes A and B), and that MSE is about 30% smaller and *C.C.* is approximately 1.2 times higher. The differences between schemes C, D, E, and F, on the other hand, are minor, indicating that further increasing the number of reference points does not improve the model performance. In the rest of paper, the scheme *C* for is employed for choosing the reference points.

APPENDIX B: TEST OF DIFFERENT WAYS FOR COMPUTING THE CONVECTION VELOCITY

To train the MCPNN-frozen model, the convection velocity has to be given in order to determine the time instants [Eq. (2)] employed as the input features at the reference point. For the model trained in this paper, the mean streamwise velocity at $z = z_h$ is employed as the convection velocity. For real-life applications, the convection velocity cannot be known in advance. In this appendix, we examine the effect of different ways for computing the convection velocity on the performance of the MCPNN-frozen model. Two approaches for computing the convection velocity *U* are tested and compared, i.e., (i) *U* is taken as the streamwise velocity averaged in time and in the horizontal directions (the approach employed in the main text); and (ii) *U* is computed as the streamwise velocity averaged over the time instants employed as the input.

Figure 22 compares the performance of the MCPNN-frozen model trained using the convection velocity *U* computed using approach (i) and approach (ii). It is observed that the correlation coefficients *C.C.* are fairly close to each other for the two different ways of computing the convection velocity. The approach (ii) will work well in real-life applications as the data at those time instants for input are readily available. The good agreement between the two approaches indicates the proposed MCPNN-frozen model is not sensitive to the convective velocity. As more than 1 (which is 5 in this work) time instants are employed as the input features, it is expected the most appropriate input feature can be constructed with the neural network.

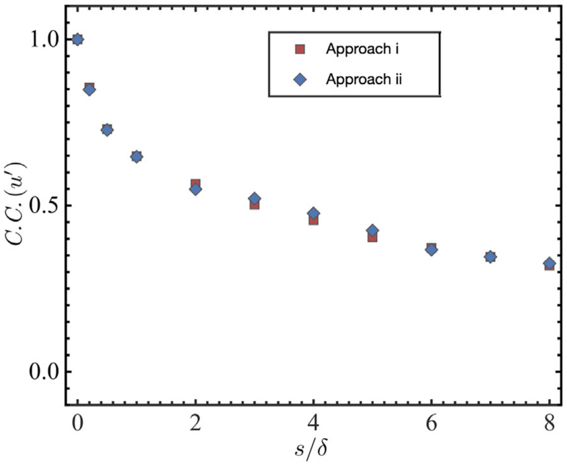


FIG. 22. Effect of different approaches for computing the convection velocity on the performance of the MCPNN-frozen model using the test data with $k_s = 0.001$ m.

REFERENCES

[1] P. Veers, K. Dykes, E. Lantz, S. Barth, C. L. Bottasso, O. Carlson, A. Clifton, J. Green, P. Green, H. Holttinen *et al.*, "Grand challenges in the science of wind energy," Science **366**, eaau2027 (2019).

[2] S. Bouckaert, A. F. Pales, C. McGlade, U. Remme, B. Wanner, L. Varro, D. D'Ambrosio, and T. Spencer, *Net Zero by 2050: A Roadmap for the Global Energy Sector* (The National Academics of Sciences Engineering Medicine, 2021).

[3] G. He, J. Lin, F. Sifuentes, X. Liu, N. Abhyankar, and A. Phadke, "Rapid cost decrease of renewables and storage accelerates the decarbonization of China's power system," Nat. Commun. **11**, 2486 (2020).

[4] X. Yang, C. Milliren, M. Kistner, C. Hogg, J. Marr, L. Shen, and F. Sotiropoulos, "High-fidelity simulations and field measurements for characterizing wind fields in a utility-scale wind farm," Appl. Energy **281**, 116115 (2021).

[5] F. Daniel, "Coherent vorticity dynamics and dissipation in a utility-scale wind turbine wake with uniform inflow," Theor. Appl. Mech. Lett. **11**, 100292 (2021).

[6] J. Zhang, S. Chowdhury, A. Messac, and B.-M. Hodge, "A hybrid measure-correlate-predict method for long-term wind condition assessment," Energy Convers. Manage. **87**, 697–710 (2014).

[7] T. Burton, *Wind Energy Handbook* (Wiley, 2011).

[8] T. Letcher, *Wind Energy Engineering* (Academic Press, 2017).

[9] S. M. Weekes and A. S. Tomlin, "Data efficient measure-correlate-predict approaches to wind resource assessment for small-scale wind energy," Renewable Energy **63**, 162–171 (2014).

[10] M. D. Mifsud, T. Sant, and R. N. Farrugia, "A comparison of measure-correlate-predict methodologies using LiDAR as a candidate site measurement device for the Mediterranean Island of Malta," Renewable Energy **127**, 947–959 (2018).

[11] P. K. Sharma, V. Warudkar, and S. Ahmed, "Application of lidar and measure correlate predict method in offshore wind resource assessments," J. Cleaner Prod. **215**, 534–543 (2019).

[12] J. A. Carta, S. Velazquez, and P. Cabrera, "A review of measure-correlate-predict (MCP) methods used to estimate long-term wind characteristics at a target site," Renewable Sustainable Energy Rev. **27**, 362–400 (2013).

[13] X. Liu, X. Lai, and J. Zou, "A new MCP method of wind speed temporal interpolation and extrapolation considering wind speed mixed uncertainty," Energies **10**, 1231 (2017).

[14] S. Diaz, J. A. Carta, and J. M. Matias, "Comparison of several measure-correlate-predict models using support vector regression techniques to estimate wind power densities. A case study," Energy Convers. Manage. **140**, 334–354 (2017).

[15] J. Addison, A. Hunter, J. Bass, and M. Rebbeck, "A neural network version of the measure correlate predict algorithm for estimating wind energy yield," in G730 Neural Computing, 2000.

[16] S. Tang and Y. Yang, "Why neural networks apply to scientific computing?," Theor. Appl. Mech. Lett. **11**, 100242 (2021).

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

[18] J.-X. Wang, J.-L. Wu, and H. Xiao, "Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data," Phys. Rev. Fluids **2**, 034603 (2017).

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

[20] L. Zhu, W. Zhang, J. Kou, and Y. Liu, "Machine learning methods for turbulence modeling in subsonic flows around airfoils," Phys. Fluids **31**, 015105 (2019).

[21] X. I. A. Yang, S. Zafar, J.-X. Wang, and H. Xiao, "Predictive large-eddy-simulation wall modeling via physics-informed neural networks," Phys. Rev. Fluids **4**, 034602 (2019).

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

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

[24] Z. Wang, K. Luo, D. Li, J. Tan, and J. Fan, "Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation," Phys. Fluids **30**, 125101 (2018).

[25] C. Xie, Z. Yuan, and J. Wang, "Artificial neural network-based nonlinear algebraic models for large eddy simulation of turbulence," Phys. Fluids **32**, 115101 (2020).

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

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

[28] Y. Zhou, D. W. Fan, B. F. Zhang, R. Y. Li, and B. R. Noack, "Artificial intelligence control of a turbulent jet," J. Fluid Mech. **897**, A27 (2020).

[29] F. Ren, J. Rabault, and H. Tang, "Applying deep reinforcement learning to active flow control in weakly turbulent conditions," Phys. Fluids **33**, 037121 (2021).

[30] Y. Yin, P. Yang, Y. Zhang, H. Chen, and S. Fu, "Feature selection and processing of turbulence modeling based on an artificial neural network," Phys. Fluids **32**, 105117 (2020).

[31] S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," J. Fluid Mech. **879**, 217–254 (2019).

[32] A. T. Mohan, D. Tretiak, M. Chertkov, and D. Livescu, "Spatio-temporal deep learning models of 3D turbulence with physics informed diagnostics," J. Turbul. **21**, 484–524 (2020).

[33] C. Jiang, R. Vinuesa, R. Chen, J. Mi, S. Laima, and H. Li, "An interpretable framework of data-driven turbulence modeling using deep neural networks," Phys. Fluids **33**, 055133 (2021).

[34] Y. Qin, K. Li, Z. Liang, B. Lee, F. Zhang, Y. Gu, L. Zhang, F. Wu, and D. Rodriguez, "Hybrid forecasting model based on long short term memory network and deep learning neural network for wind signal," Appl. Energy **236**, 262–272 (2019).

[35] Z. W. Deng, Y. J. Chen, Y. Z. Liu, and K. C. Kim, "Time-resolved turbulent velocity field reconstruction using a long short-term memory (LSTM)-based artificial intelligence framework," Phys. Fluids **31**, 075108 (2019).

[36] T. Nakamura, K. Fukami, K. Hasegawa, Y. Nabae, and K. Fukagata, "Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow," Phys. Fluids **33**, 025116 (2021).

[37] H. Eivazi, L. Guastoni, P. Schlatter, H. Azizpour, and R. Vinuesa, "Recurrent neural networks and Koopman-based frameworks for temporal predictions in a low-order model of turbulence," Int. J. Heat Fluid Flow **90**, 108816 (2021).

[38] H. Wang and Q. Gao, "A study of inner-outer interactions in turbulent channel flows by interactive pod," Theor. Appl. Mech. Lett. **11**, 100222 (2021).

[39] T. Wu and G.-W. He, "Stochastic dynamical model for space-time energy spectra in turbulent shear flows," Phys. Rev. Fluids **6**, 054602 (2021).

[40] T. Wu and G. He, "Space-time energy spectra in turbulent shear flows," Phys. Rev. Fluids **6**, 100504 (2021).

[41] G. I. Taylor, "The spectrum of turbulence," Proc. R. Soc. London, Ser. A **164**, 476–490 (1938).

[42] G.-W. He and J.-B. Zhang, "Elliptic model for space-time correlations in turbulent shear flows," Phys. Rev. E **73**, 055303 (2006).

[43] X. Zhao and G.-W. He, "Space-time correlations of fluctuating velocities in turbulent shear flows," Phys. Rev. E **79**, 046316 (2009).

[44] G. He, G. Jin, and Y. Yang, "Space-time correlations and dynamic coupling in turbulent flows," Annu. Rev. Fluid Mech. **49**, 51–70 (2017).

[45] X. Yang, F. Sotiropoulos, R. J. Conzemius, J. N. Wachtler, and M. B. Strong, "Large-eddy simulation of turbulent flow past wind turbines/farms: The virtual wind simulator (VWiS)," Wind Energy **18**, 2025–2045 (2015).

[46] X. Yang and F. Sotiropoulos, "A new class of actuator surface models for wind turbines," Wind Energy **21**, 285–302 (2018).

[47] X. Yang, A. Khosronejad, and F. Sotiropoulos, "Large-eddy simulation of a hydrokinetic turbine mounted on an erodible bed," Renewable Energy **113**, 1419–1433 (2017).

[48] X. Yang and F. Sotiropoulos, "On the dispersion of contaminants released far upwind of a cubical building for different turbulent inflows," Build. Environ. **154**, 324–335 (2019).

[49] Y. Chen, X. Yang, A. J. Iskander, and P. Wang, "On the flow characteristics in different carotid arteries," Phys. Fluids **32**, 101902 (2020).

[50] Z. Li and X. Yang, "Large-eddy simulation on the similarity between wakes of wind turbines with different yaw angles," J. Fluid Mech. **921**, A11 (2021).

[51] S. Li, X. Yang, G. Jin, and G. He, "Wall-resolved large-eddy simulation of turbulent channel flows with rough walls," Theor. Appl. Mech. Lett. **11**, 100228 (2021).

[52] Z. D. Zhou, Z. B. Li, G. W. He, and X. L. Yang, "Towards multi-fidelity simulation of flows around an underwater vehicle with appendages and propeller," Theor. Appl. Mech. Lett. **2021**, 100318.

[53] G. Dong, Z. Li, J. Qin, and X. Yang, "How far the wake of a wind farm can persist for?," Theor. Appl. Mech. Lett. **2021**, 100314.

[54] L. Ge and F. Sotiropoulos, "A numerical method for solving the 3D unsteady incompressible Navier-Stokes equations in curvilinear domains with complex immersed boundaries," J. Comput. Phys. **225**(2), 1782–1809 (2007).

[55] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, "Erratum: A dynamic subgrid-scale eddy viscosity model," Phys. Fluids A **3**, 1760 (1991).

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

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