Modeling Non-Stationary Wind-Induced Fluid Motions With Physics-Informed Neural Networks for the Shallow Water Equations in a Polar Coordinate System

Zhou, Zaiyang; Kuai, Yu; Ge, Jianzhong; van Maren, Bas; Wang, Zhenwu; Huang, Kailin; Ding, Pingxing; Wang, Zhengbing

DOI
10.1029/2024WR037490

Publication date
2025

Document Version
Final published version

Published in
Water Resources Research

Citation (APA)
Zhou, Z., Kuai, Y., Ge, J., van Maren, B., Wang, Z., Huang, K., Ding, P., & Wang, Z. (2025). Modeling Non-Stationary Wind-Induced Fluid Motions With Physics-Informed Neural Networks for the Shallow Water Equations in a Polar Coordinate System. *Water Resources Research*, *61*(1), Article e2024WR037490. https://doi.org/10.1029/2024WR037490

Water Resources Research

RESEARCH ARTICLE
10.1029/2024WR037490

Special Collection:
Advances in Machine Learning for Earth Science: Observation, Modeling, and Applications

Modeling Non-Stationary Wind-Induced Fluid Motions With Physics-Informed Neural Networks for the Shallow Water Equations in a Polar Coordinate System

Zaiyang Zhou[1], Yu Kuai[2], Jianzhong Ge[1,3], Bas van Maren[1,2,4], Zhenwu Wang[1], Kailin Huang[5], Pingxing Ding[1], and Zhengbing Wang[2,4]

[1]State Key Laboratory of Estuarine and Coastal Research, East China Normal University, Shanghai, China, [2]Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, The Netherlands, [3]Institute of Eco-Chongming (IEC), Shanghai, China, [4]Unit of Marine and Coastal Systems, Delft, The Netherlands, [5]State Key Laboratory of Water Resources Engineering and Management, Wuhan University, Wuhan, China

Abstract Physics-informed neural networks (PINNs) are increasingly being used in various scientific disciplines. However, dealing with non-stationary physical processes remains a significant challenge in such models, whereas fluid motions are typically non-stationary. In this study, a PINN-based method was designed and optimized to solve non-stationary fluid dynamics with shallow water equations in a polar coordinate system (PINN-SWEP). It was developed and validated with a classic circular basin case that is well-documented in scientific literature. In the validation case, the wind-induced water surface fluctuations are less than 1 cm, posing challenges in modeling. However, our PINN-SWEP model can accurately simulate such tiny water surface fluctuations and resolve complex fluid motions based on limited and sparse data. A boundary discontinuity problem associated with the use of a polar coordinate system is further discussed and improved, thereby enhancing the applicability of PINN in water research. The methodology can provide an alternative solution for numerical or analytical solutions with high accuracy.

Plain Language Summary Winds generate flows and waves over open water, with dimensions ranging from small ripples to large ocean waves. The wind-induced fluid motions can be described using governing equations. These equations are usually difficult to solve mathematically. Therefore, we propose a machine learning (ML) model that combines the governing equations and sparse data to reproduce the wind-induced fluid motions. This data-physics hybrid model is validated by simulating classic wind-induced fluid motions in a circular basin. Our method shows good performance, indicating a promising new approach that can coexist with conventional models. Additionally, machine learning models are prone to encountering a boundary discontinuity issue when solving problems defined in polar coordinate systems. This issue has been solved ingeniously, expanding the applicability of ML methods in geophysical and water research.

Key Points:

- A data-physics hybrid machine learning model is developed to solve shallow water equations in a polar coordinate system
- The validation using a classic circular basin case demonstrates the capability of our model to solve non-stationary hydrodynamics
- A boundary discontinuity problem due to the polar coordinate system is discussed and improved

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

J. Ge,
jzge@sklec.ecnu.edu.cn

Citation:

Zhou, Z., Kuai, Y., Ge, J., van Maren, B., Wang, Z., Huang, K., et al. (2025). Modeling non-stationary wind-induced fluid motions with physics-informed neural networks for the shallow water equations in a polar coordinate system. *Water Resources Research*, *61*, e2024WR037490. https://doi.org/10.1029/2024WR037490

Received 10 MAR 2024
Accepted 6 JAN 2025

Author Contributions:

Conceptualization: Zaiyang Zhou, Jianzhong Ge
Data curation: Yu Kuai
Formal analysis: Zaiyang Zhou
Funding acquisition: Jianzhong Ge, Zhengbing Wang
Investigation: Zhenwu Wang
Methodology: Zaiyang Zhou, Bas van Maren, Kailin Huang
Software: Zaiyang Zhou, Kailin Huang

1. Introduction

Wind-induced fluid motions occur in most open water systems, leading to vertical fluctuations in surface water, horizontal oscillations, and residual circulations. In recent decades, our understanding of wind-induced gravity waves has advanced significantly (Mitsuyasu & Honda, 1982; Phillips, 1957; Zdyrski & Feddersen, 2020). The two-dimensional shallow water equations (SWEs) are often used to describe wind-induced circulations and surface gravity waves. These depth-integrated Navier-Stokes equations have been extensively applied to study fluid dynamics in rivers, coastal areas, and oceans, including, for example, river floods (Seyoum et al., 2012), tsunamis (Arcas & Wei, 2011), and complex flows around structures (Song et al., 2023). With appropriate assumptions and simplifications, analytical solutions (AS) can be derived for the SWEs (Csanady, 1967; Läuter et al., 2005). For example, Csanady (1967, 1968a, 1968b) analytically solved the SWEs to study wind-induced circulations in a large-scale circular basin. Birchfield (1969) revised Csanady's solutions by correcting the omission of a forcing term in the boundary condition. For more complicated problems, SWEs are usually solved numerically.

Recently, machine learning (ML) techniques have been increasingly applied in water and geophysical research (Bertels & Willems, 2023; Karniadakis et al., 2021; Lu et al., 2021; Mulia et al., 2022; Shen, 2018; Verjans & Robel, 2024). For instance, ML techniques are applied to parameterize processes unresolved in coarse resolution

Supervision: Pingxing Ding
Validation: Yu Kuai, Zhenwu Wang
Visualization: Zaiyang Zhou, Yu Kuai
Writing – original draft: Zaiyang Zhou
Writing – review & editing: Yu Kuai, Jianzhong Ge, Bas van Maren, Pingxing Ding, Zhengbing Wang

global models (e.g., Verjans & Robel, 2024); and are frequently used to inversely determine spatially variable parameters in an efficient way (Guo et al., 2023). An important precondition enabling the establishment and application of an ML model is the availability of a large amount of data. However, even though the observational methods have substantially improved, data availability is still insufficient in many cases (especially for conditions where satellite images are not available as input data), preventing a purely data-driven ML technique. Furthermore, complicated dynamics in water systems strengthen the data requirements and enhance the difficulty for ML to capture instantaneous, multi-scale oceanic, estuarine, or riverine processes.

To lower the need for large data sets and improve interpretability, physical constraints (governing equations) are more and more incorporated into ML-based methods. One example is physics-informed neural networks (PINN) (Chen et al., 2022; Feng et al., 2023; Jin et al., 2021; Raissi et al., 2019, 2020). PINN is an ML-based solver for partial differential equations (PDEs). By optimizing a loss function, which includes the deviation of PDEs, a neural network (NN) with trained parameters can be obtained as a solution function. Although purely physics-driven PINN has the potential to deal with various problems, currently they can only solve simple geometries and simple PDEs and struggle with time-dependent problems (Dong et al., 2023; Meng et al., 2020). A solution is to combine PDEs with sparse data to establish hybrid PINN (Chen et al., 2023; Karniadakis et al., 2021; Li et al., 2023; Wang et al., 2022). These hybrid approaches use small data sets to guide the training process, significantly enhancing the practicability of PINN by overcoming the data limitation and improving training efficiency. So far, most PINN-based models are applied to solve a variety of PDEs written in cartesian coordinate systems. For example, a data-involved PINN method is developed to simulate dam-break flows described by SWEs in cartesian coordinates (Li et al., 2023). Moreover, advection-dispersion equations in Cartesian coordinates have also been revisited and solved using PINNs (He & Tartakovsky, 2021). Even though research on using PINN in polar (or spherical) coordinate systems does exist (e.g., Bihlo & Popovych, 2022), the result of a circular (or spherical) domain is unfolded into a rectangular plane, or the computational domain does not cover an entire circle (or sphere). This measure may coincidentally hide a problem of boundary discontinuity, which is obvious if the results are directly shown without unfolding. The insufficient focus on polar coordinates negatively affects the promotion of PINN in solving equations on the sphere, which are quite often applied in hydrological and geophysical research. Well-known analytical models such as the aforementioned circular basin case (Csanady, 1968a, 1968b) provide training data as well as the governing equations in a polar coordinate system and are therefore ideal cases to investigate the suitability of PINN to resolve wind-driven gravity waves.

In this study, we apply and improve PINN to resolve wind-induced fluid motions in open water. Random, sparse, pseudo-observational data were derived from Csanady's analytical model and combined with the governing equations to establish a data-physics hybrid model. Successful application of PINN to this circular basin required the development of methods to deal with the discontinuity caused by the polar coordinate system. A detailed description of our method is given in Section 2. Results and discussion are provided in Section 3. Conclusions are summarized in Section 4.

2. Methods

2.1. Governing Equations

The data-physics hybrid PINN model is established to solve SWEs in polar coordinate systems. A large circular basin case is used to validate the model where the fluid motions in response to a constant wind (Figure 1a) can be described with the dimensionless SWEs in a polar coordinate system (Birchfield, 1969; Chen et al., 2007):

$$\frac{\partial u^*}{\partial t^*} - v^* + \lambda \frac{\partial \zeta^*}{\partial r^*} = 0 \tag{1}$$

$$\frac{\partial v^*}{\partial t^*} + u^* + \lambda \frac{\partial \zeta^*}{r^* \partial \theta} = 0 \tag{2}$$

$$\frac{\partial \zeta^*}{\partial t^*} + \lambda \left(\frac{1}{r^*} \frac{\partial (r^* u^*)}{\partial r^*} + \frac{1}{r^*} \frac{\partial v^*}{\partial \theta} \right) = 0 \tag{3}$$

Although the equations above are written and solved in the polar coordinate system, the Cartesian coordinate system is used to present results (figures) for a clear description of relative position. The direction of the wind is

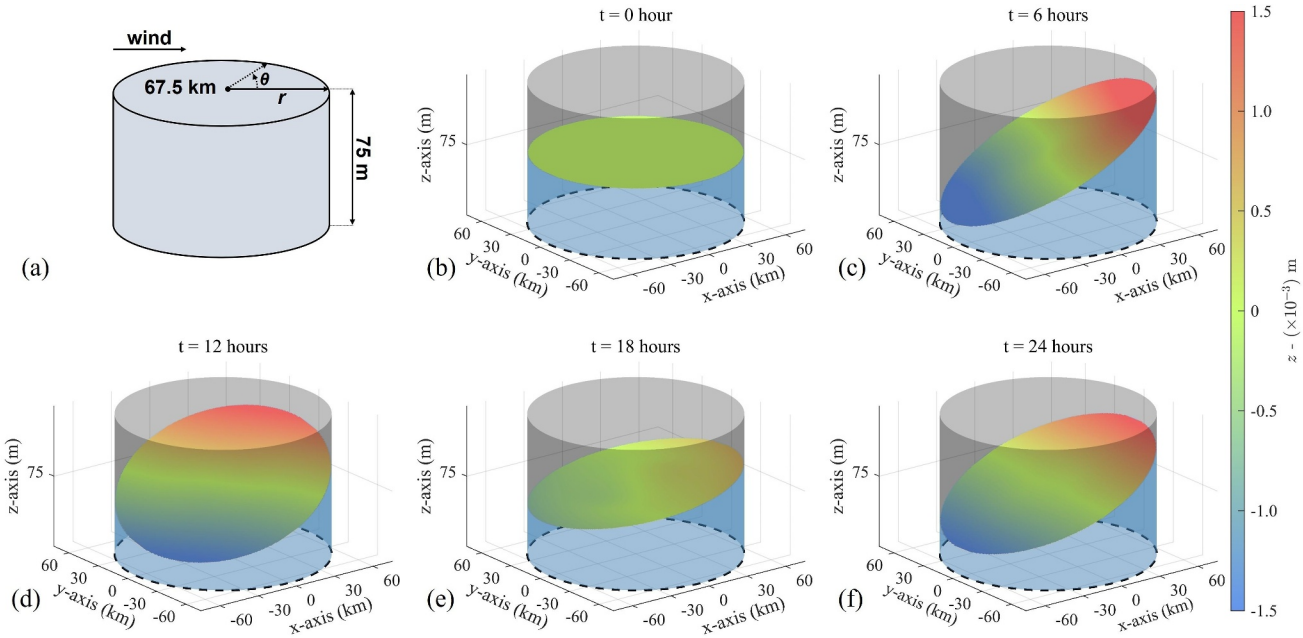

Figure 1. Schematic diagram of the defined large circular basin and the constant wind (a), and surface water elevations derived from the analytical solutions at $t = 0$ (b), 6 (c), 12 (d), 18 (e), and 24 (f) hours. The color indicates surface fluctuations from $z = 75$ m.

west, or toward the positive x-axis in the Cartesian coordinate system. In Equations 1–3, $\theta \in [0, 2\pi]$ (anti-clockwise from the east); $r^* \in [0, 1]$ is the dimensionless radius, with or without the superscript "*" denotes dimensionless or dimensional values, respectively, consistently defined henceforth; $r^* = r/R$, $R = 67.5$ km is the radius of the circular basin; $t^* = t \times f$ is the dimensionless time, and $f = 10^{-4}\text{s}^{-1}$ is the Coriolis parameter; (u^*, v^*) are dimensionless velocities in (r,θ) directions, respectively, $(u^*, v^*) = (u,v)/c$, where $c = \sqrt{g \times H} = 27.125\ m/s$ is the speed of gravity waves, $g = 9.81$ m/s^2, $H = 75$ m is the total water depth (Figure 1a); $\zeta^* = z^* - \hat{z}$ is the combination of dimensionless water depth and dimensionless wind stress term, where $z^* = z/H$ is the dimensionless disturbed height of the free surface, $\hat{z} = \tau_0 r \cos\theta/\lambda^4$, $\tau_0 = \frac{g\tau_*}{R^3 f^4 \rho}$, $\tau_* = 0.013$ N/m^2 is the magnitude of the uniform wind stress (with respect to the wind speed of 3 m/s, consistent in all cases of this study), $\rho = 10^3$ kg/m^3, $\lambda = c/(Rf) = 4.019$. The initial and boundary conditions (IC and BC) are given as:

$$(z,u,v)|_{t=0} = 0 \tag{4}$$

$$u|_{r=R} = 0;\ (z,u,v)|_{r=0} \to finite \tag{5}$$

Analytical solutions for Equations 1–5 are given as (Birchfield, 1969; Chen et al., 2007):

$$z^*(\theta,r^*,t^*) = \frac{\tau_0}{\lambda^4}\left\{A_0(r^*)\cos(\theta) + \sum_{k=1}^{\infty} a_k A_k(r^*)\cos(\theta - \sigma_k t^*)\right\} \tag{6}$$

$$u^*(\theta,r^*,t^*) = \frac{\tau_0}{\lambda^3}\left\{\left[\frac{A_0(r^*)}{r^*} - 1\right]\sin(\theta) - \sum_{k=1}^{\infty} b_k F_k(r^*)\sin(\theta - \sigma_k t^*)\right\} \tag{7}$$

$$v^*(\theta,r^*,t^*) = \frac{\tau_0}{\lambda^3}\left\{\left[\frac{dA_0(r^*)}{dr^*} - 1\right]\cos(\theta) - \sum_{k=1}^{\infty} b_k G_k(r^*)\cos(\theta - \sigma_k t^*)\right\} \tag{8}$$

The detailed calculation of $A_0(r^*)$, $A_k(r^*)$, $F_k(r^*)$, $G_k(r^*)$ can be found in Appendix A of Supporting Information S1. Solutions Equations 6–8 consist of two parts, namely the steady fluid motion (time-independent) and the

Poincare waves from $t = 0$ (time-dependent). Surface water elevations at different times are presented in Figure 1, and obvious water fluctuations and rotations can be found. Both AS (Csanady, 1968b) and numerical simulations using various models (Chen et al., 2007) demonstrate that the water elevation fluctuates within a range of several millimeters, indicating tiny fluid motions induced by gentle wind forces.

The AS are used to generate the pseudo-observational data for model training, and validation data for evaluating model performance. Although the PINN method has been demonstrated for its capability in solving PDEs without any real data, it is currently difficult to train such a model designed for the compound fluid motions. Moreover, due to the high complexity of time dependence, PINN still encounters great challenges in non-stationary problems (Dong et al., 2023; Meng et al., 2020). We have therefore developed a data-physics hybrid ML model (PINN-SWEP) that combines governing equations and sparse, limited, pseudo-observational data.

2.2. PINN for SWEs in a Polar Coordinate System (PINN-SWEP)

Conventional ML techniques learn and determine patterns between the input and output data (Ellenson et al., 2020). By computing and optimizing a loss function that compares the current NN model output with the desired output (namely the ground truth data), ML techniques can capture the non-linear relationships as universal approximators (Chen & Chen, 1993; Mhaskar & Hahm, 1997; Rossi & Conan-Guez, 2005). In contrast to a traditional ML model, which is solely driven by data, PINN incorporates the deviation from PDEs into the loss function. By minimizing the total loss toward zero, the PDEs are also approximately solved. Specifically, a PINN based physics-data hybrid model usually finds a solution for the PDEs through the following steps (taking the SWEs introduced before as an example):

Step (1): Define a NN model and calculate its output

Define the architecture of a NN model. Randomly sample the input variables (θ, r, t) within the computational domain; then output the variables ($\tilde{z}$, $\tilde{u}$, $\tilde{v}$) using the predefined NN (with the tilde representing the output of NN) and exploit auto-differentiation to obtain the corresponding derivatives ($\widetilde{\frac{\partial}{\partial\theta}}$, $\widetilde{\frac{\partial}{\partial r}}$, $\widetilde{\frac{\partial}{\partial t}}$).

Step (2): Calculate the loss terms induced by physics

Rearrange the equations to ensure that the right sides are all zero, then compute the loss (L) of PDEs, IC, and BC according to the equations; for instance, for Equation 1, $L_{PDE_1} = \widetilde{\frac{\partial u^*}{\partial t^*}} - \widetilde{v^*} + \lambda\widetilde{\frac{\partial \zeta^*}{\partial r^*}}$ (with the superscript "*" denoting dimensionless); note that these physics related loss terms exactly follow the left side of the corresponding equations.

Step (3): Calculate the loss terms induced by data

The data-induced loss is defined as $L_{OBS} = F((\tilde{z},\tilde{u},\tilde{v}),(\hat{z},\hat{u},\hat{v}))$. F is a function used to quantify the deviation between the model output and data; the circumflex indicates the ground truth values.

Step (4): Minimize the total loss and update the model

Use the gradient descent algorithms to minimize the total loss function toward zero, therefore each term of the total loss approaches zero; in particular, '$L_{PDEs} \to 0$' means the equations are approximately solved.

For a simple PDE, no real data is needed; Step (3) can be omitted and the model is fully driven by physical constraints. However, for complex equations (sets), it is difficult to design and train a suitable NN model without any guidance of real data, as is the case in this study. Therefore, a data-physics hybrid PINN model was established (Figure 2) to solve SWEs in Polar Coordinate Systems (PINN-SWEP). In addition to physics and data, the model also incorporates extra BCs to alleviate the boundary discontinuity problem due to the polar coordinate system, which will be introduced in detail later in the text.

To clarify the difference of imposing physical constraints, the pseudo-observational data, and the extra boundary conditions, here we define three operators, representing the error arising from the PDEs (Q_1), from the data-model comparison (Q_2), and from self-output comparison (Q_3), respectively:

$$Q_1 : MSE\left(PDE_{(\theta,r,t)}, 0\right) \tag{9}$$

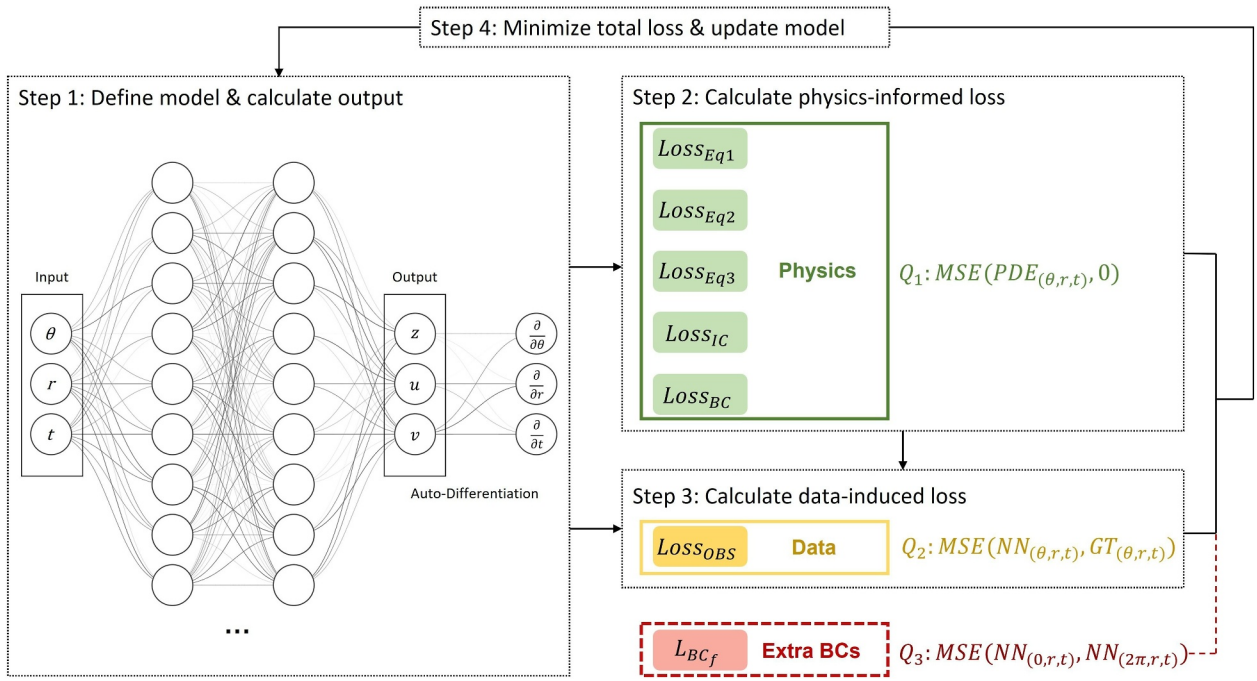

Figure 2. The architecture of PINN-SWEP to solve shallow water equations in a polar coordinate system. θ, r, t are input channels, z, u, v are the outputs of neural network for water elevations (z) and velocities (u,v) in (r, θ) directions. The loss function mainly contains three parts, loss of physics, loss of data, and loss of extra boundary conditions, calculated by three operators (Q_1, Q_2, Q_3), respectively.

$$Q_2 : MSE\left(NN_{(\theta,r,t)}, GT_{(\theta,r,t)}\right) \tag{10}$$

$$Q_3 : MSE\left(NN_{(0,r,t)}, NN_{(2\pi,r,t)}\right) \tag{11}$$

where MSE is the mean squared error function, an option for the function F in Step (3); $PDE_{(\theta,r,t)}$ is the residual of equation at (θ,r,t); $NN_{(\theta,r,t)}$ means the PINN-SWEP output, therefore $NN_{(0,r,t)}$ and $NN_{(2\pi,r,t)}$ in Equation 11 are model output when $\theta = 0$ or 2π; $GT_{(\theta,r,t)}$ denotes the ground truth values, that is, the pseudo-observational data. The NN and GT functions can output all z, u, and v as needed. With these three operators, all terms in the total loss function can be calculated as:

$$\text{Loss} = \underbrace{\lambda_1 L_{PDE_1} + \lambda_2 L_{PDE_2} + \lambda_3 L_{PDE_3} + \lambda_4 L_{IC} + \lambda_5 L_{BC}}_{\text{Physics}} + \underbrace{\lambda_6 L_{OBS}}_{\text{Data}} + \underbrace{\lambda_7 L_{BC_f}}_{\text{Extra } BCs} \tag{12}$$

where λ_i represents the weighting coefficients of each loss term. The weighting coefficients need to balance the contribution of each term, especially the main contributors, that is, the loss of physics and the loss of data. These two loss terms should have a similar order of magnitude, thus avoiding the training process being dominated by only part of the loss function. According to this concept, the model has undergone initial tests to determine the appropriate coefficients.

The total Loss can be classified as three categories (Equation 12), that is, physics including initial and boundary conditions (PDEs + IC + BC), data (assuming we have limited observations), and extra BCs due to the use of a polar coordinate system (referred to as fictitious BCs, BC_f). The detailed calculations for each category are outlined below.

2.2.1. Loss of Physics

The physical constraints are described as $\lambda_1 L_{PDE_1} + \lambda_2 L_{PDE_2} + \lambda_3 L_{PDE_3} + \lambda_4 L_{IC} + \lambda_5 L_{BC}$. Among them, L_{PDE_1}, L_{PDE_2}, L_{PDE_3} are deviations between (the left side of) Equations 1–3 and zero, respectively. These three

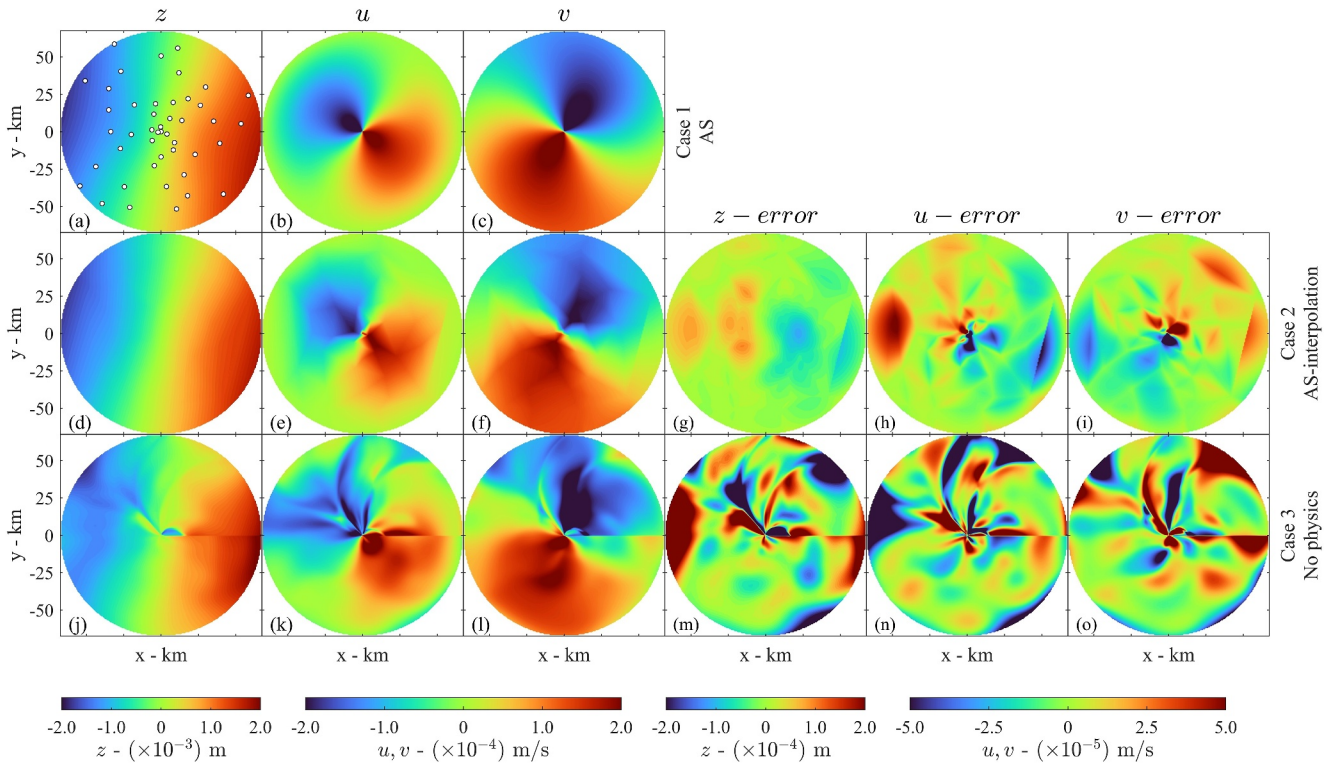

Figure 3. The wind-induced water fluctuation z (first column), velocity u in r-axis (second column), velocity v in θ-axis (third column), and errors (from fourth to sixth columns) of (z, u, v) compared to Case 1 at $t = 195$ min derived from 3 methods, including the analytical solutions (Case 1, first row, also regarded as the benchmark case), direct interpolation based on 45 sites (Case 2, second row), and a neural network method without governing equations (Case 3, third row). The white round dots in panel (a) are the positions of the 45 assumed observational sites.

terms are calculated with the operator Q_1. Usually, in a concrete problem, the initial and boundary conditions are more explicit compared to the dynamics described by PDEs. For example, in this study, the boundary condition "$u|_{r=R} = 0$" indicates the boundary is impermeable, meaning that the along-radius velocity u has a specific value (zero) on the boundary. This information is inherently provided by the problem definition, like real data, but at no observational cost. Consequently, the operators Q_1 (for PDEs) has the same function as Q_2 (for data) in terms of the calculation of L_{IC} and L_{BC}. λ_1, λ_2, λ_3 are specified as 5×10^9, $\lambda_4 = 100$, $\lambda_5 = 1$ according to the initial tests.

The set of 3-D collocation points (θ, r, t) utilized for physical constraints consists of 8,000 interior sampling residual points for PDEs, 100 points for IC, and 50 points for BC. It is important to note that these collocation points are fully randomly sampled and are not relevant to any observational data. Furthermore, they will be resampled every 20,000 epochs. This resampling method follows earlier PINN research (e.g., Iwasaki & Lai, 2023; Li et al., 2023). The method ensures sufficient collocation points to train a NN model without any prior knowledge of the solution to a problem. It should be noted that although this resampling method can enhance model performance (Wu et al., 2023), it may also lead to some randomly sampled collocation points unintentionally appearing in the validation set, negatively affecting the interpretation of the results. In this study, to further minimize this effect, we applied two distinct random point generation functions for resampling collocation points and creating the validation data set, respectively. In addition, we conducted 1,000 repetitive tests for random point generation, and did not observe data overlap between interior sampling and the validation data set caused by the resampling method.

2.2.2. Loss of Data

It is assumed that we have sparse observational data as training data, which are generated by the AS in this study. We randomly generate 45 points (using the Sobol sequence (Bratley & Fox, 1988) to ensure the reproducibility of the experiments) in the computational domain (white dots in Figure 3a), which amounts to 0.55% of the total collocation points. We assume that data of z, u, v at 10-min frequency are available at those 45 pseudo-

observational sites, which is not difficult for present-day ocean bottom-mounted observation systems (Ge et al., 2018; Zhou et al., 2019). The data distribution density (45 sites in 1.43×10^4 km^2) is very low and comparable to other data-physics hybrid PINN model applications (e.g., Chen et al., 2023). In addition, a sensitivity test on model performance using various numbers (20–60, with an interval of 5) of pseudo-observational sites is presented in Section 3.4. The data loss term L_{OBS} is computed using Q_2, that is, comparing the NN model output with the ground truth observations. λ_6 is set as (1, 10, 10) for (z, u, v) to balance their different orders of magnitude, as the magnitude of z is apparently larger than that of the other two.

2.2.3. Loss of Fictitious BC

In a polar coordinate system, $\theta = 0$ and 2π overlap, whereas the NN model treats $\theta = 0$ and $\theta = 2\pi$ as different realizations. The extrapolation toward two boundaries of θ results in a discontinuity of model outputs when $\theta = 0$ or 2π. To mitigate this issue induced by the polar coordinate system, we enforce a fictious boundary condition (BC_f) for θ by incorporating an additional loss term to constrain the difference between $NN_{(0,r,t)}$ and $NN_{(2\pi,r,t)}$, which are values at the same points in space but expressed using different values of θ. It is important to note that this loss term does not rely on any data; it is just a self-constraint of the NN model. The calculation of L_{BC_f} is conducted using the operator Q_3:$MSE(NN_{(0,r,t)},NN_{(2\pi,r,t)})$.

For ML models, additional loss terms present optimization challenges, turning it into a more complex multi-target problem and increasing the difficulty of reducing the loss. The L_{BC_f} as an additional loss term can include all (z, u, v), or consider only one or some combinations of them by adjusting the weighting coefficients λ_7 (depending on which variables are important for a specific problem). Therefore, this BC_f loss term can be incorporated into the PINN-SWEP model in a flexible way. For instance, in this study, the boundary problems are more prominent in variable u and v, so a BC_f for v ($\lambda_7 = (0, 0, 40)$ for (z, u, v)) is introduced and presented as the main results to demonstrate the effectiveness of this method. Another scenario applying a different BC_f is included (Appendix C in Supporting Information S1).

2.2.4. Model Implementation and Scenarios

The PINN-SWEP model is established based on the multi-layer perceptron (MLP) architecture, one of the most widely used artificial neural networks (Gracia et al., 2021). The MLP contains an input layer, an output layer, and hidden layers in between (Figure 2). Each hidden layer consists of some neurons, and each neuron is connected to all neurons from the previous layer. These connections are modeled with weights and biases. The activation function is also operated on these neurons to enable their non-linear expression ability. The PINN-SWEP model contains 4 hidden layers of 100 neurons per layer. A hyperbolic tangent is used as the activation function (Huang et al., 2022; Jin et al., 2021; Tarbiyati & Nemati Saray, 2023). The gradient decrease is performed using an Adam optimizer (Kingma & Ba, 2014). For all scenarios, the model is trained for 600,000 epochs (i.e., iteration steps). The learning rate (lr) is determined by $lr = lr_i/(1 + dr \times epochs/100)$, $lr_i = 8 \times 10^{-3}$ is the initial lr, $dr = 5 \times 10^{-3}$ is the decay rate. During the 600,000 epochs, lr decreases from 8×10^{-3} to 2.58×10^{-4}.

During the training process, the model performance is evaluated synchronously using a validation data set. The validation data are also generated by the AS. Concretely, a $10^4 \times 3$ matrix is generated using the Sobol sequence to provide a reproducible random sampling of (θ, r, t). With these 10^4 scattered points of (θ, r, t), corresponding output (z, u, v) can be generated using the AS, forming a validation data set that randomly distributed in the computational domain and period. The root mean squared errors (RMSE) between the validation data set and the results of the NN model are used to evaluate model performance. For model development, this validation data set is designed to be random, abundant, and widely distributed in the entire computational domain to demonstrate the generalization of the model. In real applications, the model can also be validated using individual observational site, an example of which is given in Section 3.3 (Figure 5). The information on all training data, including methods to obtain or generate them, is summarized in Table 1. Only the validation data set directly samples from (θ, r, t), while others sample from (θ, r) with a default temporal resolution of 10 min. In Table 1, L_{PDE_s} follow Equations 1–3 and include (z, u, v) and their derivatives, L_{IC} specifies (z, u, v) at the initial time, L_{BC} only specify u at the boundary of calculation domain, L_{OBS} involves the comparison of all (z, u, v) between the NN model output and the ground truth data.

Table 1
Summary of Different Types of Sampling Points for Training and Validating

Type	Number of points	Sampling methods	Update frequency
L_{PDE_1}, L_{PDE_2}, L_{PDE_3}	8,000	purely random	20,000 epochs
L_{IC}	100	purely random	20,000 epochs
L_{BC}	50	purely random	20,000 epochs
L_{BC_f}	50	purely random	20,000 epochs
L_{OBS} (observational)	45	Sobol sequence	/
Validation	10,000	Sobol sequence	/

The PINN-SWEP is developed with MATLAB® R2023b, applying GPU acceleration for ML. The training is performed on an NVIDIA RTX 3090 GPU and an Intel Core i9-13900K processor. The training process takes approximately 18 hr to complete 600,000 epochs for 4 cases simultaneously, providing simulated results of wind-induced hydrodynamics over the first 24 hr with acceptable accuracy. Concretely, the RMSEs of (z, u, v) of the validation data set are controlled under 5.0×10^{-5} m, 2.0×10^{-5} m/s, and 2.0×10^{-5} m/s (two orders of magnitude smaller than the variation ranges of the variables). Information on the main scenarios is summarized in Table 2. Case 1 provides the results of the AS, and there is no limit on data resolution in space, which is regarded as a benchmark scenario. Case 2 is the result of data interpolation based on the aforementioned 45 pseudo-observational sites. From Case 3 to Case 6, the results of the basic PINN-SWEP model and the optimized models are presented.

3. Results and Discussion

3.1. Significance of Physics

Taking advantage of the prior knowledge of hydrodynamics (i.e., governing equations) is at the core of the PINN methodology. Therefore, the significance of PDEs is first investigated. In Case 1, patterns of z, u, v at $t = 195$ min generated by the AS are shown in Figures 3a–3c. This moment is not part of the pseudo-observational data, which is generated every 10 min (i.e., at 190 and 200 min). For an area with a radius of 67.5 km, 45 observational sites are insufficient to describe the fluid dynamics, especially for u and v. This has been demonstrated by a direct linear interpolation based on the 45 pseudo-observational sites (Case 2, Figures 3d–3i). Since the water fluctuation z indicates a linear pattern (Figure 3a), it is therefore possible to be decently resolved by direct interpolation using limited data (Figures 3d and 3g). However, simple interpolation is insufficient to resolve the more complex distributions of u and v (Figures 3e, 3f, 3h and 3i). Using the 45 samples to train a fully data-driven NN without physics by omitting loss terms $L_{PDE_1} - L_{PDE_3}$, L_{IC}, and L_{BC} (Case 3, Figures 3j–3o) results in a poorer representation of z, u and v compared to Case 2. As a result, the significance of physical constraints and PDEs is highlighted, and the value of PINN is further demonstrated by Case 4 (in the following section) which significantly improves the model performance by including the physical constraints.

Table 2
Scenario Summary

Case number	Number of samples	With BC_f or not	Descriptions
1	/	/	Analytical solutions
2	45	/	Interpolation (linear) based on 45 observational sites
3	45	/	No physics (equations, IC, and BC excluded)
4	45	N	Basic PINN-SWEP
5	45	Y	PINN-SWEP with BC_f for v only
6	45	N	PINN-SWEP with data arrangement strategy: 5 sites on the θ boundary, 40 from the Sobol sequence

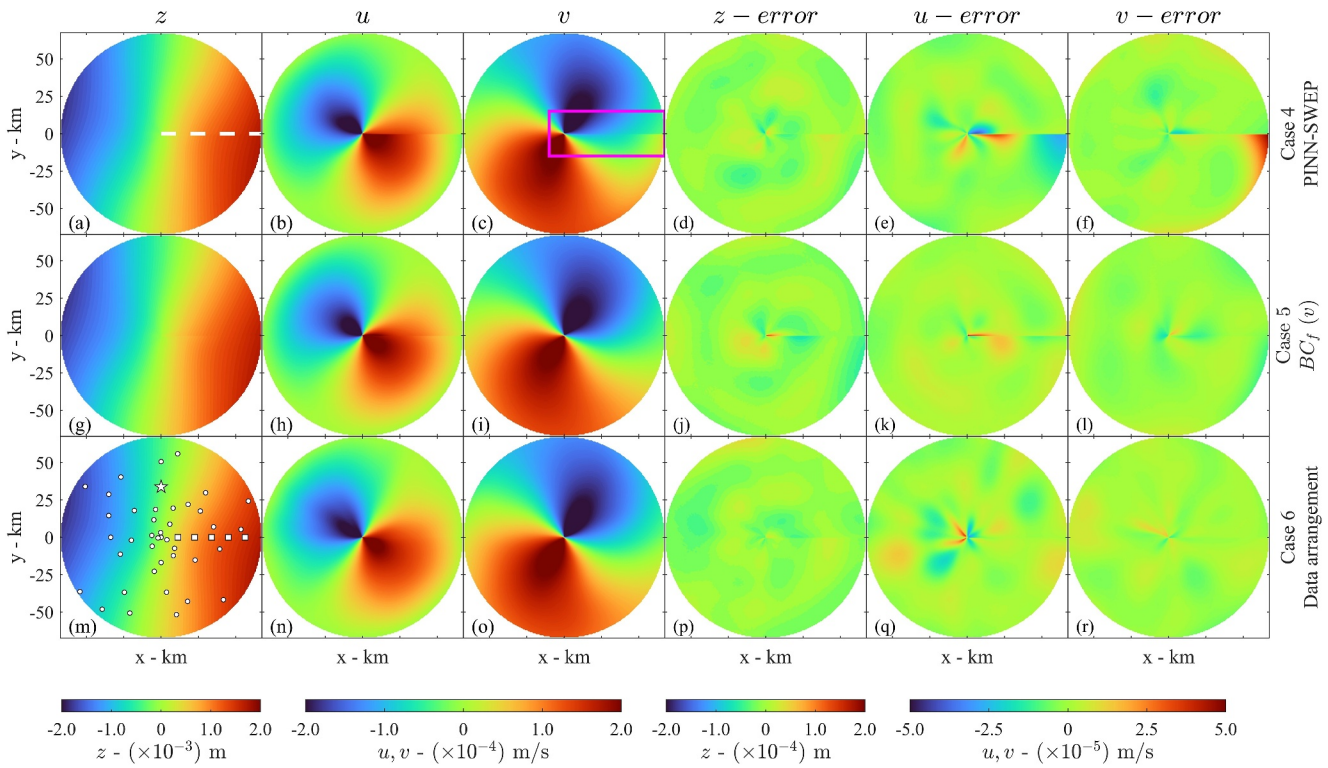

Figure 4. The wind-induced water fluctuation z (first column), velocity u in r-axis (second column), velocity v in θ-axis (third column), and errors (from fourth to sixth columns) of (z, u, v) compared to Case 1 at t = 195 min derived from 3 methods, including the basic PINN-SWEP method (Case 4, first row), PINN-SWEP with a fictious BC for variable v (Case 5, second row), and PINN-SWEP using specifically arranged observational sites (Case 6, third row, 5 sites on $\theta = 0$). The white dashed line in panel (a) is the boundary of θ. The magenta box in panel (c) shows the discontinuity when $\theta = 0$ (2π). White round and squared dots in panel (m) are the positions of the assumed observational sites, and the pentagram is selected to perform a time-series validation.

3.2. PINN-SWEP, Boundary Problem, and Solutions

The performance of the model greatly improves by imposing PDEs (PINN-SWEP cases, Figure 4). The results of the basic PINN-SWEP model and the optimized models bear far greater resemblance to the analytical solution (Case 1, Figures 3a–3c) than the cases without PDEs (Case 3, Figures 3j–3o), and are also much better than the interpolated data (Case 2, Figures 3d–3i). This is also proved by the RMSEs calculated based on the difference between each case and the benchmark Case 1 (Table 3).

A drawback of NN models used in polar coordinate systems is the discontinuity (magenta box in Figure 4c) at $\theta = 0 = 2\pi$ (white dashed line in Figure 4a), which is even more apparent in Case 3 (Figures 3j–3o). The pronounced discontinuity at $\theta = 0$ (2π) in all ML-based cases results from the overlap of two ends of θ dimension in the polar coordinate system. Small errors during the extrapolation toward these two ends can lead to serious discontinuity problem. This discontinuity does not exist for the AS-interpolation case because both the interpolation and figure plotting are conducted in the rectangular (non-polar) coordinate system. Among the three output variables, this problem is most serious for v at t = 195 min, as pattern conversion can be found within the magenta box in Figures 4c and 4f. Taking variable v as an example, the boundary problem can be resolved by defining a loss term L_{BC_f} along the $\theta = 0$ (2π) line (Case 5) using Q_3:MSE ($NN_{(0,r,t)}$,$NN_{(2\pi,r,t)}$) for variable v. This extra loss term L_{BC_f} significantly reduces the coordinate-induced discontinuity problem (Figures 4g–4l). Even though the RMSE in Table 3 is based on information about the whole basin rather than focusing on the θ boundary, the RMSE of v has been significantly improved from 5.3×10^{-6} m/s to 2.9×10^{-6} m/s due to the additional BC_f. On the other hand, although the BC_f only involves variable v, the RMSE of u has also been improved from 5.7×10^{-6} m/s to 3.7×10^{-6} m/s. Moreover, an extra test case which includes the BC_f for variable u is presented (Appendix C)

Table 3
The RMSEs of Case 2–6 Compared to the Benchmark Case 1 at t = 195 Minutes

	Case 2	Case 3	Case 4	Case 5	Case 6
z($\times 10^{-5}$ m)	3.7	17.3	1.3	1.4	1.3
u($\times 10^{-6}$ m/s)	16.1	33.2	5.7	3.7	4.9
v($\times 10^{-6}$ m/s)	14.7	31.2	5.3	2.9	2.8

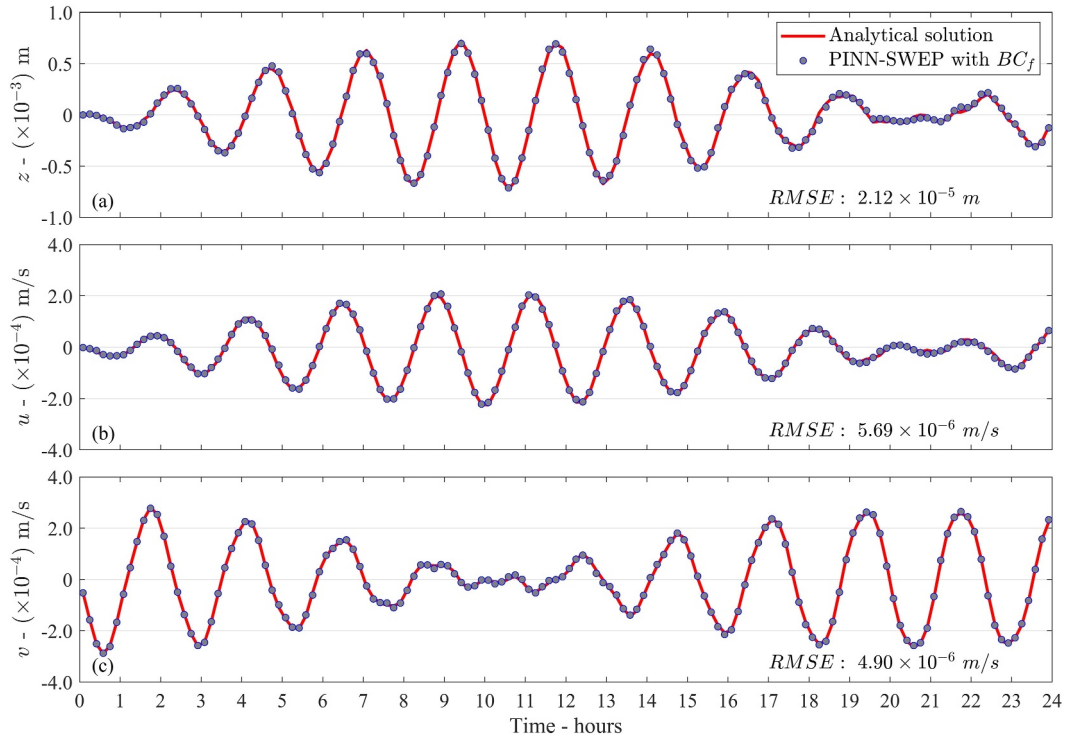

Figure 5. A time-series validation for Case 5 at $\theta = \pi/2$, $r = R/2$ (pentagram in Figure 4m); panels (a–c) are results of z, u, and v, respectively.

in Supporting Information S1, providing similar results, that is, these two cases both improve the overall model performance on the (u, v) simulation at $\theta = 0\ (2\pi)$ while the performance on z remains similar. This is related to the inner relationship between u and v, as they are both components of flow velocities, more information on one component can simultaneously improve the simulation of the other, indicating the effectiveness of imposing a simple fictitious BC. As explained in Section 2.2, more complicated BC_f that includes all or combinations of (z, u, v) can be applied as needed, but a new balance of weighting coefficients should be achieved as the model complexity is enhanced.

The discontinuity can be further reduced by imposing a stronger constraint on the boundaries of θ by defining observational sites along the $\theta = 0\ (2\pi)$ line. The previous fictitious BC method only proposed a constraint that the output values of $\theta = 0$ should be equal to those of $\theta = 2\pi$, but without specific ground truth data. However, the data arrangement measure steps up to specify the exact values of the output on the θ boundary. Based on Case 4, which uses 45 randomly distributed observational sites, we can spare 5 of these sites specifically on $\theta = 0$, resulting in 40 remaining random sites (Case 6, Figures 4m–4r). A slight discontinuity in Case 5 is further eliminated by only 5 specified observational sites, indicated by the comparison of errors (Figures 4j–4l and 4p–4r). This strategy has two advantages compared to the fictitious boundary method. Firstly, it does not impose any new loss term, therefore does not enhance model complexity, benefiting the training. Secondly, since more pseudo-observational data are distributed at the θ boundary and are incorporated into the model as hard constraints which are stronger than self-constraints, a better simulation of this high-deviation region can be achieved, potentially improving the overall performance of the model. The second advantage is reflected by the RMSEs of the validation data set, as the RMSEs of (z, u, v) are 3.76×10^{-5} m, 1.26×10^{-5} m/s and 8.8×10^{-6} m/s for Case 5 (with fictitious BC, averaged results of three runs), and are 2.86×10^{-5} m, 1.05×10^{-5} m/s and 8.97×10^{-6} m/s for Case 6 (data arrangement, averaged results of three runs). Even though the fictitious BC method in Case 5 mainly targets at improving model accuracy of v, these two cases have close RMSEs of v, while the data arrangement strategy obviously improves model performance on solving z and u. Thus, the data arrangement strategy helps to better address the boundary problem at the expense of enhancing the request (on sampling site selection rather than data size) for observations. Although this strategy has higher demand for observations, it can be adopted in applications where the observational positions are flexible, and the fluid dynamics near $\theta = 0$ are important.

3.3. Time-Series Validation

The PINN-SWEP model (Case 5) has been additionally validated in the time domain using the same position as Chen et al. (2007) ($\theta = \pi/2$; $r = R/2$, corresponding to the pentagram in Figure 4m). This site is not included in the 45 training points. The PINN-SWEP model outputs the result every 10 min, starting at minute 5 to avoid overlap with the training data set (starting at minute 0). The wind-induced fluid motions in the basin are relatively tiny compared to the geometric scale of the domain which is therefore difficult to accurately resolve. Nevertheless, the RMSEs are 2 orders of magnitude smaller than the small variation ranges of z, u, v (Figure 5). The PINN-SWEP method shows excellent performance for time-series reproduction, proving its capability to resolve fluid dynamics based on limited data.

3.4. Sensitivity Test on the Training Data Volume

The results presented in Figures 4 and 5 are based on 45 assumed observational sites. To analyze how PINN-SWEP is sensitive to the observational data volume, we conducted a sensitivity test in which the number of observational sites increased from 20 to 60 (with an interval of 5). For each scenario of the test, three runs of PINN-SWEP simulations are conducted and the results are averaged. The remaining RMSE of the validation data set computed after 600,000 training epochs is used to quantify the model's sensitivity to the observational data volume (Figures 6a–6c). For all cases, the positions of the observational sites follow the Sobol sequence to ensure repeatability.

Clear inflection points of all three output variables (z, u, v) appear when applying 25 assumed observational points (Figures 6a–6c). The remaining RMSE increases noticeably as the data volume is downsized to less than 25. However, after 25, the benefit of using more data points reduces, and reduces to near-zero after 50. When 10, 25, and 40 assumed observational sites are applied, counterpart time-series validations similar to Figure 5 were also conducted (Figures 6d–6f). To better visualize the deviations of each simulation, results in the period $t = 16$–24 hr are presented. The time-series validation for using 25 pseudo-observational sites shows that the model can well simulate the hydrodynamics at $\theta = \pi/2$, $r = R/2$, as the red lines with pentagrams in Figure 6 keep great resemblance to the analytical solution. Meanwhile, an increase or decrease of 15 pseudo-observational sites based on 25 sites causes significant difference in model performance. The improvement of accuracy from using 25 to 40 sites is not significant while the accuracy reduction from 25 to 10 sites is apparent, especially for the velocity in radius direction (u, Figure 6e).

The sensitivity test indicates that the recommended minimum data volume is from at least 25 observational sites. In this study, we assume the temporal frequency is 10 min. In real observations, for water elevation and velocity measurements, the frequency can surpass the 10-min assumption easily. In that case, more abundant data in dimension t may further improve the model performance.

3.5. Training Process and Model Convergence

The convergences of the loss function and the RMSE of Case 5 (PINN-SWEP with BC_f) are shown in Figure 7, and similar training processes (Appendix D) in Supporting Information S1 of other cases (Appendix B) in Supporting Information S1. The loss function gradually decreased and reached smaller values at the end of every 20,000 epochs. As introduced before, except for the pseudo-observational data, other collocation points were resampled every 20,000 epochs. The resampling measure was conducted by randomly sampling new points and did not increase the dependence on data volume. It also explains the fluctuations that occurred every 20,000 epochs in Figure 7. However, the fluctuations caused by the resampling measure became weaker as the training went on, indicating overall convergence during the whole training process. In other words, the resampling measure helps to avoid overfitting and increase the generalization of the PINN-SWEP model. After the training process, loss of data (L_{OBS}) remained the largest among all loss terms (title of Figure 7a), also suggesting the model was not overfitted with the limited data.

An example of typical overfitting can be found in the training process of Case 3 (no physics case). As the loss function continued to decrease in the whole 600,000 epochs, while the RMSE increased on the contrary after hitting a low point within the first 20,000 epochs (Figure S2 in Appendix D in Supporting Information S1). The significance of physical information is once again emphasized because the limited information from 45 pseudo-observational sites can be easily overfitted, according to Case 3.

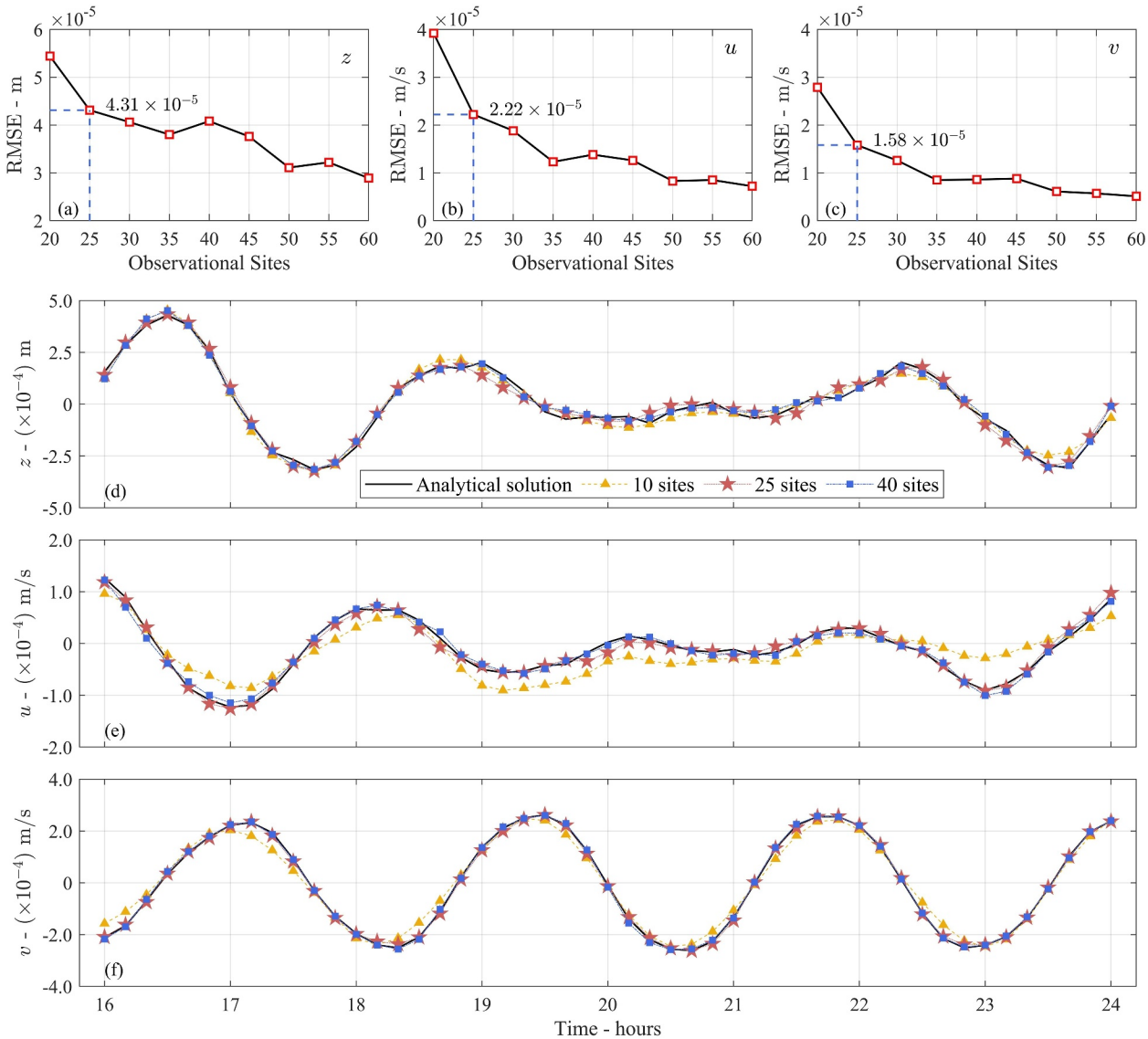

Figure 6. The remaining RMSEs of z (a), u (b), and v (c) of the validation set after 600,000 training epochs using different numbers of assumed observational sites. Time-series validations at $\theta = \pi/2$, $r = R/2$ (the pentagram in Figure 4m) using 10 (d), 25 (e), and 40 (f) pseudo-observational sites.

3.6. Model Robustness

In this study, we apply AS as pseudo-observational data. However, both systematic and random errors typically exist in real measurements. Therefore, we conducted a test for different levels of noise to evaluate the robustness of our model. In this test, the noise was added to the pseudo-observational data by setting different maximum noise percentages. Specifically, the noise-contaminated data was generated as:

$$\text{Data} = \text{Data}_{AS} + \underbrace{\text{MaxNoise} \times |\text{Data}_{AS}|}_{\text{noise amplitude}} \times p \tag{13}$$

where MaxNoise is a percentage that specifies the noise amplitude together with the absolute value of the pseudo-observational data ($|\text{Data}_{AS}|$), and p is a random probability in (0,1), that creates the randomness of noise. Case 5, which involves 45 pseudo-observational sites, is used as a benchmark case, and the noise is added based on Case 5. The MaxNoise is specified at 2%, 5%, 10%, 15%, 20%, 25%, and 40%.

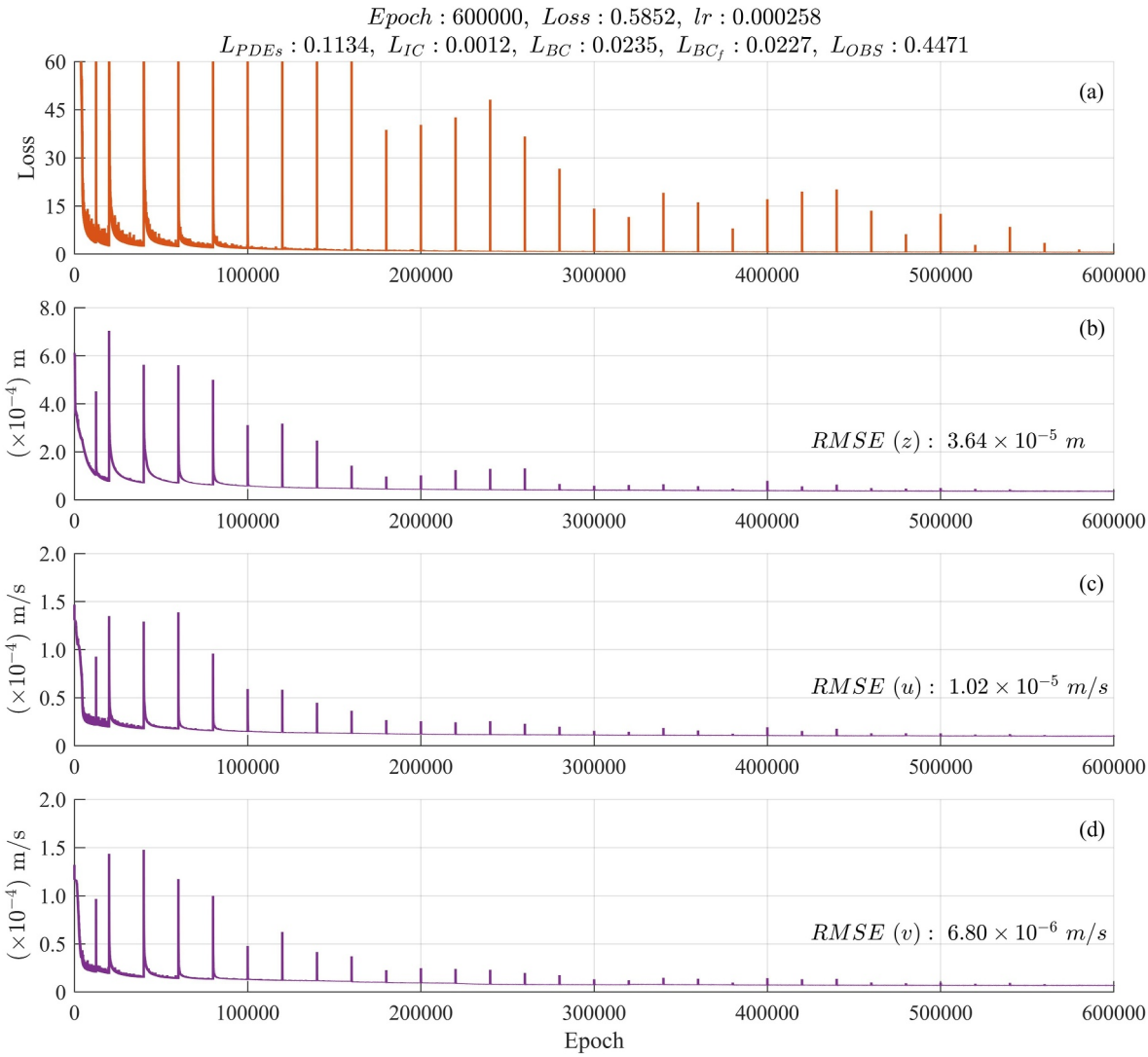

Figure 7. Training process of Case 5 (PINN-SWEP with BC_f); (a) the total loss, the text above panel (a) is the model status after 600,000 epochs; panels (b–d) show the variation of remaining root mean squared errors (after 600,000 epochs) of z, u, and v of the validation data set, respectively.

A 2%-maximum noise does not affect the model performance (Figures 8a–8c). On the contrary, such small noise can act as a form of regularization and enhance model generalization (Shen, 2018), resulting in a decrease of RMSEs when including this 2%-maximum noise. Model performance is significantly weakened by noise from 10% onward. For a 20%-maximum noise, the RMSEs of the validation set double the 0- and 2%-maximum noise cases. When the maximum noise percentage reaches 40%, the RMSE of water elevation z is an order of magnitude larger compared to other cases (Figure 8a).

In addition, we conducted a time-series validation, in consistent with the preceding validation position in Figures 5 and 6. This time-series validation visually demonstrates the deviations induced by different levels of noise. The 2%- and 10%-maximum noise leads to comparable results, which are acceptable in terms of both time-series resemblance to the analytical solution (Figures 8d–8f), and the overall RMSEs (Figures 8a–8c), under 5.0×10^{-5} m, 2.0×10^{-5} m/s, and 2.0×10^{-5} m/s, as mentioned in Section 2.2). However, a 40%-maximum noise leads to obvious deviations in all variables (z, u, v). As a result, when the noise is less than 10% of the data magnitude, the model is able to simulate the hydrodynamics in the system with sufficient accuracy. In real observations, a 10% measurement error is a relatively low (poor) standard. For instance, the typical bias of Acoustic Doppler Current Profiler (ADCP) is about 1.5%–3.5% (Huang, 2018), indicating that the bias tolerance of our model (10% of the data magnitude) holds great potential for practical applications based on in-situ observations.

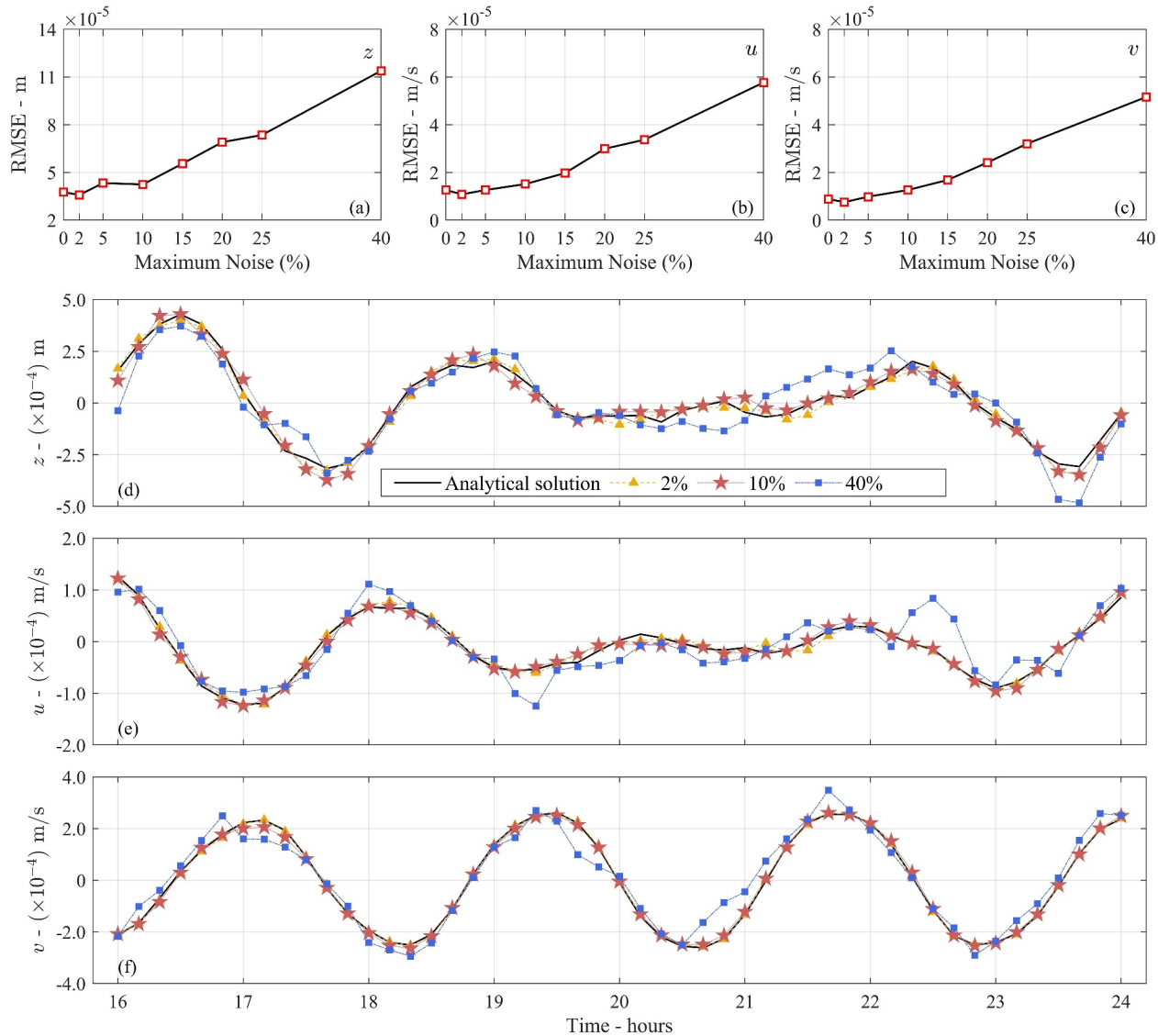

Figure 8. The remaining RMSEs of z (a), u (b), and v (c) of the validation set after 600,000 training epochs considering different levels of noise. Time-series validations at $\theta = \pi/2$, $r = R/2$ (the pentagram in Figure 4m) using 2% (d), 10% (e), and 40% (f) of the data magnitude as noise maximum.

3.7. General Discussion and Future Work

Although PINN models can simulate dynamic processes, they have substantial distinctions from numerical models (which are most widely used to model dynamic processes). PINN models are not constrained by the "output time step" present in numerical models. They can output all information within the spatial and temporal domains extremely fast (on the order of seconds or even less) without requiring the model to be re-run. This highlights a significant advantage of PINNs—their flexible usage after training. The success of the PINN method in solving SWEs strongly suggests that this method has the potential to solve other problems that can be described by the same or similar PDEs. Especially for PDEs in polar coordinate systems, our method provides implications for handling boundary discontinuity issues.

However, it should be noted that currently PINNs still face a wide variety of challenges that limit their application in real-world complex problems. One of the primary challenges is the computational efficiency during training. To achieve accurate results, a PINN training process usually requires thousands or hundreds of thousands of epochs, depending on the complexity of PDEs. For each epoch, the computational expense is largely determined by the architecture (size) of the model and the number of sample points involved in the calculation (Nabian

et al., 2021). On the other hand, an adequate number of sample points is important for ensuring model accuracy. As a result, sampling techniques that balance the coverage of sample points and the computational efficiency are important. In this study, the resampling method benefits the model accuracy but also increases computational cost. Other strategies such as the clustered training points sampling method can help achieve a better trade-off although it needs priori knowledge of the solution (Mao et al., 2020). In our future work, more efficient sampling methods, such as residual-based adaptive sampling (Hanna et al., 2022; Wu et al., 2023) can be applied or combined with the current resampling strategy to improve the training efficiency.

In addition, time-dependent problems still remain significant challenges for the majority of PINN models (Dong et al., 2023; Zhang et al., 2022). Typically, MLP-based PINNs do not distinguish between the temporal and spatial dimensions, resulting in the lack of information on temporal relationships (Wu et al., 2022). However, time-dependent processes are crucial in water research. In this study, the PINN-SWEP model successfully solved the time-dependent SWEs but the simulation period was limited. The model requires improvement in terms of capturing temporal relationships before realizing its potential for solving real-world complex issues. Previous efforts such as temporal domain splitting (Meng et al., 2020) and latent context generating (Jiang et al., 2020; Wu et al., 2022) provide information of great value on our future work related to time-space issues.

The architecture and hyperparameters of the model can be further optimized to enhance model generalization and training efficiency. For parameters in the SWEs such as the wind speed in our case, their changes do not significantly affect the model complexity and the loss function. Therefore, the current model architecture can be easily applied to these scenarios. While boundary and initial conditions can be much more complicated and can pose challenges for training the model. Dazzi (2024) provides an example for taking topography into consideration using an augmented system of SWEs, which enlightens future development of the PINN-SWEP model. Other techniques, such as dynamic weights and adaptive sampling, will be explored in the future to accelerate the training process. Despite the limitations identified here and, in the literature, PINNs successfully alleviate the dependence on data volume and have attracted attention from various scientific fields. In this study, we focus more on investigating their capability in hydrodynamics simulation but cannot ignore their advantages and suitability in inverse and parametrized problems. As the model develops, we will seek to apply the model to realistic applications and additional scenarios; for instance, resolving fluid dynamics in a rotating tank could be an initial attempt as real observations can be conveniently obtained.

4. Conclusions

We developed a PINN model, PINN-SWEP, which incorporates governing equations and sparse data to solve SWEs in a polar coordinate system. Classic wind-induced fluid motions in a circular basin were revisited with PINN-SWEP. The horizontal motions and water surface fluctuations were well resolved by PINN-SWEP, suggesting great potential for PINN-SWEP as an alternative method for analytical or numerical solutions. The single-site time-series validation further demonstrates the accuracy of our method to infer fluid dynamics, including subtle wind-induced motions.

The polar coordinate system may cause a discontinuity of NN model output at the overlapping boundaries of the θ-axis ($\theta = 0$ or 2π). A v-oriented fictitious boundary condition was imposed to equalize the model output for points on θ-axis boundaries. This measure helps to alleviate the discontinuity and pattern shift of v. A follow-up method to reduce discontinuity is also proposed, but at the cost of a higher data requirement. In that case, it is suggested to allocate these data (or observation) resources based on concrete demands if these resources are flexible.

This physics-informed methodology can significantly reduce the amount of data required for a data-only NN. The sensitivity analysis suggests that for such a large area, the fluid dynamics of the entire basin can be simulated based on data collected from only 25–45 observational sites using PINN-SWEP. The noise tolerance test further proves the robustness of the model, indicating the potential for being applied based on real observations. The ML-based model introduced here as well as the technique to deal with boundary-related discontinuity could also be applied to other geophysical systems, especially those with dynamical processes described in a polar coordinate system.

Data Availability Statement

All datasets (the pseudo-observational data) used in this study, as well as the code and a training-process video of the PINN-SWEP model are publicly available at: https://figshare.com/articles/dataset/2024_PINN-SWEP/25303330.

Acknowledgments
This research is supported by the National Key R&D Program of China (Grant 2022YFE0117500), the National Natural Science Foundation of China (Grants 42406161, 42476160, and 41776104), the Key Research Program of Shanghai Municipal Oceanic Bureau (Key Technology Research and Application of Marine Intelligent Grid Forecasting for Shanghai, Grant HHK-2023-06), and the KNAW Project (Grant PSA-SA-E-02).

References

Arcas, D., & Wei, Y. (2011). Evaluation of velocity-related approximations in the nonlinear shallow water equations for the Kuril Islands, 2006 tsunami event at Honolulu, Hawaii. *Geophysical Research Letters*, *38*(12), 1–6. https://doi.org/10.1029/2011GL047083

Bertels, D., & Willems, P. (2023). Physics-informed machine learning method for modelling transport of a conservative pollutant in surface water systems. *Journal of Hydrology*, *619*, 129354. https://doi.org/10.1016/j.jhydrol.2023.129354

Bihlo, A., & Popovych, R. O. (2022). Physics-informed neural networks for the shallow-water equations on the sphere. *Journal of Computational Physics*, *456*, 111024. https://doi.org/10.1016/j.jcp.2022.111024

Birchfield, G. E. (1969). Response of a circular model great lake to a suddenly imposed wind stress. *Journal of Geophysical Research*, *74*(23), 5547–5554. https://doi.org/10.1029/JC074i023p05547

Bratley, P., & Fox, B. L. (1988). Algorithm 659 implementing Sobol's quasirandom sequence generator. *ACM Transactions on Mathematical Software*, *14*(1), 88–100. https://doi.org/10.1145/42288.214372

Chen, C., Huang, H., Beardsley, R. C., Liu, H., Xu, Q., & Cowles, G. (2007). A finite volume numerical approach for coastal ocean circulation studies: Comparisons with finite difference models. *Journal of Geophysical Research*, *112*(C3), C03018. https://doi.org/10.1029/2006JC003485

Chen, Q., Wang, N., & Chen, Z. (2023). Simultaneous mapping of nearshore bathymetry and waves based on physics-informed deep learning. *Coastal Engineering*, *183*, 104337. https://doi.org/10.1016/j.coastaleng.2023.104337

Chen, T., & Chen, H. (1993). Approximations of continuous functionals by neural networks with application to dynamic systems. *IEEE Transactions on Neural Networks*, *4*(6), 910–918. https://doi.org/10.1109/72.286886

Chen, Y., de Ridder, S. A. L., Rost, S., Guo, Z., Wu, X., & Chen, Y. (2022). Eikonal tomography with physics-informed neural networks: Rayleigh wave phase velocity in the Northeastern Margin of the Tibetan Plateau. *Geophysical Research Letters*, *49*(21), 1–9. https://doi.org/10.1029/2022GL099053

Csanady, G. T. (1967). Large-scale motion in the great lakes. *Journal of Geophysical Research*, *72*(16), 4151–4162. https://doi.org/10.1029/JZ072i016p04151

Csanady, G. T. (1968). Motions in a model great lake due to a suddenly imposed wind. *Journal of Geophysical Research*, *73*(20), 6435–6447. https://doi.org/10.1029/JB073i020p06435

Csanady, G. T. (1968). Wind-driven summer circulation in the great lakes. *Journal of Geophysical Research*, *73*(8), 2579–2589. https://doi.org/10.1029/JB073i008p02579

Dazzi, S. (2024). Physics-informed neural networks for the augmented system of shallow water equations with topography. *Water Resources Research*, *60*(10). https://doi.org/10.1029/2023WR036589

Dong, W., Fritts, D. C., Liu, A. Z., Lund, T. S., Liu, H. L., & Snively, J. (2023). Accelerating atmospheric gravity wave simulations using machine learning: Kelvin-Helmholtz instability and mountain wave sources driving gravity wave breaking and secondary gravity wave generation. *Geophysical Research Letters*, *50*(15), 1–10. https://doi.org/10.1029/2023GL104668

Ellenson, A., Pei, Y., Wilson, G., Özkan-Haller, H. T., & Fern, X. (2020). An application of a machine learning algorithm to determine and describe error patterns within wave model output. *Coastal Engineering*, *157*, 103595. https://doi.org/10.1016/j.coastaleng.2019.103595

Feng, D., Tan, Z., & He, Q. Z. (2023). Physics-informed neural networks of the Saint-Venant equations for downscaling a large-scale river model. *Water Resources Research*, *59*(2). https://doi.org/10.1029/2022WR033168

Ge, J., Zhou, Z., Yang, W., Ding, P., Chen, C., Wang, Z. B., & Gu, J. (2018). Formation of concentrated benthic suspension in a time-dependent salt wedge estuary. *Journal of Geophysical Research: Oceans*, *123*(11), 8581–8607. https://doi.org/10.1029/2018JC013876

Gracia, S., Olivito, J., Resano, J., Martin-del-Brio, B., de Alfonso, M., & Álvarez, E. (2021). Improving accuracy on wave height estimation through machine learning techniques. *Ocean Engineering*, *236*, 108699. https://doi.org/10.1016/j.oceaneng.2021.108699

Guo, Q., Zhao, Y., Lu, C., & Luo, J. (2023). High-dimensional inverse modeling of Hydraulic Tomography by Physics Informed Neural Network (HT-PINN). *Journal of Hydrology*, *616*, 128828. https://doi.org/10.1016/j.jhydrol.2022.128828

Hanna, J. M., Aguado, J. V., Comas-Cardona, S., Askri, R., & Borzacchiello, D. (2022). Residual-based adaptivity for two-phase flow simulation in porous media using physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, *396*, 115100. https://doi.org/10.1016/j.cma.2022.115100

He, Q. Z., & Tartakovsky, A. M. (2021). Physics-informed neural network method for forward and backward advection-dispersion equations. *Water Resources Research*, *57*(7), 1–20. https://doi.org/10.1029/2020WR029479

Huang, H. (2018). Estimating bias limit of moving-boat ADCP streamflow measurements. *Journal of Hydraulic Engineering*, *144*(6). https://doi.org/10.1061/(ASCE)HY.1943-7900.0001465

Huang, S., Xia, J., Wang, Y., Wang, W., Zeng, S., She, D., & Wang, G. (2022). Coupling machine learning into hydrodynamic models to improve river modeling with complex boundary conditions. *Water Resources Research*, *58*(10), 1–15. https://doi.org/10.1029/2022WR032183

Iwasaki, Y., & Lai, C.-Y. (2023). One-dimensional ice shelf hardness inversion: Clustering behavior and collocation resampling in physics-informed neural networks. *Journal of Computational Physics*, *492*, 112435. https://doi.org/10.1016/j.jcp.2023.112435

Jiang, C. M., Marcus, P., Esmaeilzadeh, S., Azizzadenesheli, K., Kashinath, K., Mustafa, M., et al. (2020). MESHFREEFLOWNET: A physics-constrained deep continuous space-time super-resolution framework. In *SC20: International conference for high performance computing, networking, storage and analysis* (pp. 1–15). IEEE. https://doi.org/10.1109/SC41405.2020.00013

Jin, X., Cai, S., Li, H., & Karniadakis, G. E. (2021). NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, *426*, 109951. https://doi.org/10.1016/j.jcp.2020.109951

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*, 1–15. https://doi.org/10.48550/arXiv.1412.6980

Läuter, M., Handorf, D., & Dethloff, K. (2005). Unsteady analytical solutions of the spherical shallow water equations. *Journal of Computational Physics*, *210*(2), 535–553. https://doi.org/10.1016/j.jcp.2005.04.022

Li, C., Han, Z., Li, Y., Li, M., Wang, W., Dou, J., et al. (2023). Physical information-fused deep learning model ensembled with a subregion-specific sampling method for predicting flood dynamics. *Journal of Hydrology*, *620*, 129465. https://doi.org/10.1016/j.jhydrol.2023.129465

Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, *3*, 218–229. https://doi.org/10.1038/s42256-021-00302-5

Mao, Z., Jagtap, A. D., & Karniadakis, G. E. (2020). Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, *360*, 112789. https://doi.org/10.1016/j.cma.2019.112789

Meng, X., Li, Z., Zhang, D., & Karniadakis, G. E. (2020). PPINN: Parareal Physics-Informed Neural Network for time-dependent PDEs. *Computer Methods in Applied Mechanics and Engineering*, *370*, 113250. https://doi.org/10.1016/j.cma.2020.113250

Mhaskar, H. N., & Hahm, N. (1997). Neural networks for functional approximation and system identification. *Neural Computation*, *9*(1), 143–159. https://doi.org/10.1162/neco.1997.9.1.143

Mitsuyasu, H., & Honda, T. (1982). Wind-induced growth of water waves. *Journal of Fluid Mechanics*, *123*, 425–442. https://doi.org/10.1017/S0022112082003139

Mulia, I. E., Ueda, N., Miyoshi, T., Gusman, A. R., & Satake, K. (2022). Machine learning-based tsunami inundation prediction derived from offshore observations. *Nature Communications*, *13*(1), 5489. https://doi.org/10.1038/s41467-022-33253-5

Nabian, M. A., Gladstone, R. J., & Meidani, H. (2021). Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*, *36*(8), 962–977. https://doi.org/10.1111/mice.12685

Phillips, O. M. (1957). On the generation of waves by turbulent wind. *Journal of Fluid Mechanics*, *2*(05), 417. https://doi.org/10.1017/S0022112057000233

Raissi, M., Perdikaris, P., & 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*, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Raissi, M., Yazdani, A., & Karniadakis, G. E. (2020). Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, *367*(6481), 1026–1030. https://doi.org/10.1126/science.aaw4741

Rossi, F., & Conan-Guez, B. (2005). Functional multi-layer perceptron: A non-linear tool for functional data analysis. *Neural Networks*, *18*(1), 45–60. https://doi.org/10.1016/j.neunet.2004.07.001

Seyoum, S. D., Vojinovic, Z., Price, R. K., & Weesakul, S. (2012). Coupled 1D and noninertia 2D flood inundation model for simulation of urban flooding. *Journal of Hydraulic Engineering*, *138*(1), 23–34. https://doi.org/10.1061/(asce)hy.1943-7900.0000485

Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resources Research*, *54*(11), 8558–8593. https://doi.org/10.1029/2018WR022643

Song, Y., Shen, C., & Liu, X. (2023). A surrogate model for shallow water equations solvers with deep learning. *Journal of Hydraulic Engineering*, *149*(11), 1–15. https://doi.org/10.1061/JHEND8.HYENG-13190

Tarbiyati, H., & Nemati Saray, B. (2023). Weight initialization algorithm for physics-informed neural networks using finite differences. *Engineering Computers*, *40*(3), 1603–1619. https://doi.org/10.1007/s00366-023-01883-y

Verjans, V., & Robel, A. (2024). Accelerating subglacial hydrology for ice sheet models with deep learning methods. *Geophysical Research Letters*, *51*(2). https://doi.org/10.1029/2023GL105281

Wang, N., Chen, Q., & Chen, Z. (2022). Reconstruction of nearshore wave fields based on physics-informed neural networks. *Coastal Engineering*, *176*, 104167. https://doi.org/10.1016/j.coastaleng.2022.104167

Wu, B., Hennigh, O., Kautz, J., Choudhry, S., & Byeon, W. (2022). Physics informed RNN-DCT networks for time-dependent partial differential equations. In *International conference on computational science*, 2022, 372–379. https://doi.org/10.1007/978-3-031-08754-7_45

Wu, C., Zhu, M., Tan, Q., Kartha, Y., & Lu, L. (2023). A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, *403*, 1–23. https://doi.org/10.1016/j.cma.2022.115671

Zdyrski, T., & Feddersen, F. (2020). Wind-induced changes to surface gravity wave shape in deep to intermediate water. *Journal of Fluid Mechanics*, *903*, 1–42. https://doi.org/10.1017/jfm.2020.628

Zhang, X., Zhu, Y., Wang, J., Ju, L., Qian, Y., Ye, M., & Yang, J. (2022). GW-PINN: A deep learning algorithm for solving groundwater flow equations. *Advances in Water Resources*, *165*, 104243. https://doi.org/10.1016/j.advwatres.2022.104243

Zhou, Z., Ge, J., Wang, Z. B., Maren, D. S., Ma, J., & Ding, P. (2019). Study of lateral flow in a stratified tidal channel-shoal system: The importance of intratidal salinity variation. *Journal of Geophysical Research: Oceans*, *124*(9), 6702–6719. https://doi.org/10.1029/2019JC015307