REVIEW ARTICLE | DECEMBER 11 2024

Physics-informed neural networks for solving incompressible Navier–Stokes equations in wind engineering

Kang Cai (蔡康) ; Jiayao Wang (王佳瑶)

Physics of Fluids 36, 121303 (2024)
https://doi.org/10.1063/5.0244094

Physics of Fluids REVIEW pubs.aip.org/aip/pof

Physics-informed neural networks for solving incompressible Navier–Stokes equations in wind engineering

Cite as: Phys. Fluids **36**, 121303 (2024); doi: 10.1063/5.0244094
Submitted: 16 October 2024 · Accepted: 20 November 2024 ·
Published Online: 11 December 2024

View Online Export Citation CrossMark

Kang Cai (蔡康)[1,2,a] and Jiayao Wang (王佳瑶)[2,b]

AFFILIATIONS

[1]Institute of Structural Engineering, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, People's Republic of China
[2]Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong Special Administrative Region, People's Republic of China

[a]Electronic mail: kangc_hk@163.com
[b]**Author to whom correspondence should be addressed:** jiayao.wang@connect.ust.hk

ABSTRACT

Despite the substantial advancements made over the past 50 years in solving flow problems using numerical discretization of the Navier–Stokes (NS) equations, seamlessly integrating noisy data into existing algorithms remains a challenge. In addition, mesh generation is intricate, and addressing high-dimensional problems governed by parameterized NS equations is difficult. The resolution of inverse flow problems is notably resource-intensive, often necessitating complex formulations and the development of new computational codes. To address these challenges, a physics-informed neural network (PINN) has been proposed to seamlessly integrate data and mathematical models. This innovative approach has emerged as a multi-task learning framework, where a neural network is tasked with fitting observational data while reducing the residuals of partial differential equations (PDEs). This study offers a comprehensive review of the literature on the application of PINNs in solving two-dimensional and three-dimensional NS equations in structural wind engineering. While PINN has demonstrated efficacy in many applications, significant potential remains for further advancements in solving NS equations in structural wind engineering. This work discusses important areas requiring improvement, such as addressing theoretical limitations, refining implementation processes, and improving data integration strategies. These improvements are essential for the continued success and evolution of PINN in computational fluid dynamics.

I. INTRODUCTION

In the past 50 years, significant advancements have been made in computational fluid dynamics (CFD) for numerically solving the incompressible and compressible Navier–Stokes (NS) equations in structural wind engineering. This progress is largely attributed to the adoption of many numerical methodologies such as finite elements, spectral, and meshless methods (Brooks and Hughes, 1982; Karniadakis and Sherwin, 2005; Katz, 2009). However, seamlessly integrating (multi-fidelity) data into existing algorithms for practical applications continues to pose significant challenges. Additionally, mesh generation for industrial-complexity problems is time-consuming. Furthermore, solving inverse problems, such as determining unknown boundary conditions or conductivities, frequently incurs high costs due to the necessity for unique formulations and the development of new computational codes. Finally, computer programs like OpenFOAM (Jasak *et al.*, 2007) consist of over 100,000 lines of code, making maintenance and updates challenging across generations.

To address these challenges, deep neural networks (DNN) can be considered for solving the NS equations in structural wind engineering, which have gained increasing popularity and demonstrated success in fields such as computer vision and natural language processing. Furthermore, DNN is being progressively employed to address traditional applied mathematics issues like partial differential equations (PDEs) through application of machine learning (ML) and artificial intelligence techniques. Some PDEs present significant challenges for conventional numerical methods due to substantial nonlinearities and convection dominance. Deep learning with its capability as a universal approximation and its high expressiveness has recently emerged as a new paradigm in scientific computing. Recent research has demonstrated the potential of deep learning in constructing meta-models for rapid predictions of dynamic systems, and neural networks have proven effective in capturing the underlying nonlinear input–output relationships in complex systems. However, addressing such high-dimensional complex systems remains the subject to the curse of dimensionality, as initially described by Bellman (1966) in the context of optimal control problems. ML-based algorithms offer promising prospects for solving PDEs (Blechschmidt and Ernst, 2021). Pioneer work utilized relatively simple neural network models such as the multi-layer perceptron with few hidden layers to solve differential equations (Lagaris *et al.*, 1998). Contemporary studies utilize sophisticated neural network techniques, incorporating optimization frameworks and auto-differentiation capabilities, as evidenced by the unified DNN approach proposed by Berg and Nyström (2018) for estimating PDE solutions. Blechschmidt and Ernst (2021) anticipate that ML-based approaches for solving PDEs will remain a significant area of study in the coming years, particularly as deep learning continues to advance in methodological, theoretical, and algorithmic developments.

While deep learning provides powerful modeling capabilities in data-rich fields like vision, language, and speech, the development of interpretable and generalizable models remains challenging, particularly for domains with limited data such as complex physical systems (Vinuesa and Sirmacek, 2021; Rudin, 2019). Purely data-driven approaches based on deep learning require extensive datasets for training, which may not be available for many scientific problems. Moreover, these models often neglect physical constraints, resulting in models that might align well with observational data but fail to adhere to the fundamental physical laws. Therefore, integrating governing physical laws and domain knowledge into training process may lead to more accurate and robust models. Such domain knowledge can act as an informative prior, enhancing the model's understanding of the physical or mathematical frameworks of the system besides the observational data. Accordingly, the physics-informed neural network (PINN) was innovatively introduced by Raissi *et al.* (2017a, 2017b), which uses automatic differentiation to represent differential operators, hence eliminating the need for a mesh generation. This methodology allows for the direct incorporation of constraints such as the NS equations into the neural network's loss function, enabling the model to penalize deviations from desired values while appropriately weighting any available data. This approach has been applied to various flow problems, including incompressible flows (Jin *et al.*, 2021; Raissi *et al.*, 2020), compressible flows (Mao *et al.*, 2020), and biomedical flows (Yin *et al.*, 2021).

Notably, PINNs are not intended to replace existing CFD codes for numerically solving the incompressible and compressible NS equations. The current generation of PINNs does not match the precision or efficacy of high-order CFD codes in solving standard forward problems due to the high-dimensional non-convex nature of the loss function (Karniadakis and Sherwin, 2005). However, PINNs present a better performance compared to traditional CFD solver when analyzing problems with scattered partial spatiotemporal data. Therefore, comprehensively investigating the potential application of PINNs in structural wind engineering is particularly important. This paper aims to comprehensively review the PINN framework in detail and discuss its application to the incompressible NS equation in structural wind engineering.

II. DEVELOPMENT OF PHYSICS-INFORMED NEURAL NETWORKS

PINNs are a type of scientific ML method used for addressing problems related to PDEs. PINNs approximate solutions to PDEs through training neural networks to minimize a loss function that incorporates terms representing both the initial and boundary conditions along the boundary of the space–time domain, as well as the PDE residual at specified collocation points within the domain (referred to as collocation points). PINNs are deep learning networks that yield an estimated solution at a given input point within the integration domain after training. A key innovation of PINNs is the incorporation of the governing physics equations and neural networks, transforming PDE solutions from direct solving to an optimization problem where the loss function serves as a penalizing factor, thus confining the range of acceptable solutions.

The incorporation of prior knowledge into ML algorithms is not an entirely novel concept. The work by Dissanayake and Phan-Thien in 1994 is often cited as an early precursor to PINNs. Their study built upon the universal approximation achievements in the late 1980s (Hornik *et al.*, 1989), and in the early 90 s, various methodologies were proposed to utilize neural networks for approximating PDEs, such as the research on constrained neural networks by Lagaris *et al.* (1998), Lee and Kang (1990), and Psichogios and Ungar (1992). Specifically, Dissanayake and Phan-Thien (1994) employed a simple neural network to approximate a PDE solution, where the output of the network aimed to approximate the solution value at a specified input position. The network incorporated two hidden layers, each containing 3, 5, or 10 nodes. The loss was evaluated using a quasi-Newtonian approach, with the gradient calculations performed via finite difference methods. In a subsequent refinement, Lagaris *et al.* (1998) proposed the solution of a differential equation as a constant term and an adjustable term designed by unknown parameters, optimized through neural network training. However, this method was initially restricted to problems defined within regular borders, which was later expanded by Lagaris *et al.* (2000) to address problems with irregular borders. PINNs shift the focus from reliance solely on extensive data sets to incorporating the physics of the problem, thereby enhancing the solution's accuracy by integrating structured prior knowledge about the system's behavior. This concept was further explored by Owhadi (2015), who demonstrated the significant potential of leveraging structured prior knowledge in computational approaches. Building on this foundation, Raissi *et al.* (2017c) utilized Gaussian process regression to develop a representation of the linear operator that could accurately infer solutions and provide uncertainty estimates for a diverse range of physical problems. This work was subsequently expanded upon in Raissi and Karniadakis (2018a) and Raissi *et al.* (2018b). In 2017, PINNs were introduced as a novel category of data-driven solvers in a two-part article (Raissi *et al.*, 2017a, 2017b), with further consolidation in 2019 (Raissi *et al.*, 2019b). In this latter work, the authors demonstrated the application of PINNs in solving nonlinear PDEs such as the Schrödinger, Burgers, and Allen-Cahn equations. The development of these PINNs has enabled not only the estimation of solutions to forward problems by directly governing mathematical models, but also the resolution of inverse problems by learning model parameters from observable data. This pioneering work has significantly advanced the field of physics-informed ML for solving complex physical problems through an integrated scientific computing framework.

As computational capabilities expanded in the 2000s, the feasibility of deploying more sophisticated models featuring an increasing number of parameters and layers became more general (Özbay *et al.*, 2021). Various deep models utilizing the Multilayer Perceptron and Radial Basis Functions were introduced (Kumar and Yadav, 2011). The interest in using neural networks for solving PDEs surged in the late 2010s, fueled by advances in hardware capabilities, enhanced training practices, and the advent of open-source software like TensorFlow (Haghighat and Juanes, 2021), which included automatic differentiation features (Paszke *et al.*, 2017). Recent advancements by Kondor and Trivedi (2018) and Mallat (2016) were further evolved by Raissi *et al.* (2019), who not only extended previous approaches but also introduced novel methods, including a discrete time-stepping scheme that effectively leverages the predictive capacity of neural networks (Kollmannsberger *et al.*, 2021).

The significant impact of PINNs is underscored by the high citation rate of key publications such as Raissi *et al.* (2019b) and the observed exponential growth in citations over recent years. Nonetheless, PINN represent just one of several neural net frameworks developed for solving PDEs. Among these, the Deep Ritz method (E and Yu, 2018) defines the loss function as the energy of the solution to the problem. Other approaches utilize the Galerkin method or Petrov–Galerkin method, wherein the loss is determined by multiplying the residual with a test function. The deep Galerkin method is employed when dealing with volumetric residuals (Sirignano and Spiliopoulos, 2018). Further variations include hp-VPINN (Kharazmi *et al.*, 2021), which applies a Galerkin approach at collocation points, and conservative PINN (CPINN), which preserves certain conservation properties within the solution domain (Jagtap *et al.*, 2020b). Another approach is physics-constrained neural networks (PCNNs) (Liu and Wang, 2021; Sun *et al.*, 2020a; Zhu *et al.*, 2019). While PINNs include both the PDE and its initial/boundary conditions in the training loss function, PCNNs are "data-free" neural networks, which do not require data for enforcing initial/boundary conditions like PINNs, instead using a customized neural network architecture that embeds these constraints directly within the network's structure. This soft form technique is detailed in Raissi *et al.* (2019b), where the term "PINN" was introduced. Despite the diversity of approaches, this review will primarily focus on PINNs due to their broader documentation and foundational status within this family of neural network methodologies.

Various review papers have been published, including review papers by Karniadakis *et al.* (2021), explores the potentials, limitations, and diverse applications of PINNs for both forward and inverse problems. Specific studies, such as those by Cai *et al.* (2021), have examined PINNs for incompressible three-dimensional (3D) flows, while others have performed comparative analyses with other ML techniques (Blechschmidt and Ernst 2021). PINN has also been compared against other methods for solving PDEs, such as the one based on the Feynman–Kac theorem (Blechschmidt and Ernst, 2021). Furthermore, PINNs have been extended to solve integrodifferential equations (Pang *et al.*, 2019) and stochastic differential equations (Yang *et al.*, 2020; Zhang *et al.*, 2019). The advantages of PINNs are as follows: they are mesh-free, enable on-demand computation of solutions after training, and allow solutions to be made differentiable using analytical gradients. Moreover, PINNs provide a straightforward approach to jointly solving forward and inverse problems through a single optimization framework. PINNs have further been adapted to solve inverse problems, such as characterizing fluid flows from sensor data with minimal modification to the forward problem-solving code. In the context of inverse design, PDEs can be enforced as hard constraints (Lu *et al.*, 2021). In conclusion, PINNs not only provide a framework to address PDEs in complex geometries or high dimensions, but also solve inverse and constrained optimization problems that are challenging to numerically simulate.

III. THE APPLICATION OF PINN IN THE STRUCTURAL WIND ENGINEERING

A. PINN for solving Navier–stokes equation

In structural wind engineering, the widely employed equation is the NS equation, which utilizes tensor notation and the Einstein summation convention, given by

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (1)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \upsilon \frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i, \quad (2)$$

$$u_i(\boldsymbol{x}, 0) = V_i(\boldsymbol{x}), \quad p_i(\boldsymbol{x}, 0) = P_i(\boldsymbol{x}), \quad (3)$$

$$u_i|_\Sigma = U_i(\boldsymbol{x}, t), \quad p(\boldsymbol{x}_0) = P_0, \quad (4)$$

where x is the spatial coordinate; t is the time; ρ represents the fluid density; υ denotes the kinematic viscosity of the fluid; f_i is the body force per unit mass acting on the fluid; $p_i(\boldsymbol{x}, 0)$ denotes the initial pressure field of the fluid at time $t = 0$; $p(\boldsymbol{x}_0)$ indicates the pressure at the boundary; and $u_i(\boldsymbol{x}, t)$ is the solution of the PDE with the initial condition $u_i(\boldsymbol{x}, 0)$ and boundary condition $u_i|_\Sigma$.

To significantly reduce computational time compared to directly solving all turbulent fluctuations, the Reynolds-Averaged Navier–Stokes (RANS) equations were developed to model the average behavior of turbulence flows as follows:

From $\left\langle \frac{\partial u_i}{\partial x_i} \right\rangle = 0$, get

$$\frac{\partial \langle u_i \rangle}{\partial x_i} = 0, \quad (5)$$

$$\left\langle \frac{\partial u_i}{\partial t} \right\rangle + \left\langle u_j \frac{\partial u_i}{\partial x_j} \right\rangle = \left\langle -\frac{1}{\rho}\frac{\partial p}{\partial x_i} \right\rangle + \left\langle \upsilon \frac{\partial^2 u_i}{\partial x_j \partial x_j} \right\rangle + \langle f_i \rangle, \quad (6)$$

$$\left\langle u_j \frac{\partial u_i}{\partial x_j} \right\rangle = \left\langle \frac{\partial u_i u_j}{\partial x_j} - u_i \frac{\partial u_j}{\partial x_j} \right\rangle = \left\langle \frac{\partial u_i u_j}{\partial x_j} \right\rangle = \frac{\partial \langle u_i u_j \rangle}{\partial x_j}, \quad (7)$$

$$\langle u_i u_j \rangle = \langle u_i \rangle \langle u_j \rangle + \langle u_i' u_j' \rangle, \quad (8)$$

$$\left\langle u_j \frac{\partial u_i}{\partial x_j} \right\rangle = \langle u_j \rangle \frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_i' u_j' \rangle}{\partial x_j}, \quad (9)$$

$$\frac{\partial \langle u_i \rangle}{\partial t} + \langle u_j \rangle \frac{\partial \langle u_i \rangle}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \langle p \rangle}{\partial x_i} + \upsilon \frac{\partial^2 \langle u_i \rangle}{\partial x_j \partial x_j} - \frac{\partial \langle u_i' u_j' \rangle}{\partial x_j} + \langle f_i \rangle. \quad (10)$$

For the 2D incompressible NS equations, the location and speed of every fluid particle was $x_i = \{x, y\}$, $u_i = \{u, v\}$, while $x_i = \{x, y, z\}$ and $u_i = \{u, v, w\}$ was given for the 3D incompressible flow.

In the context of the vanilla PINNs (Raissi *et al.*, 2019b), a fully connected feed-forward neural network with multiple hidden layers is used to approximate the solution of the PDE $\hat{u}$ by taking the space and time coordinates $(\boldsymbol{x}, t)$ as inputs. Let the hidden variable of the *kth* hidden layer be denoted by Z^k, then the neural network can be expressed as

$$Z^0 = (x, t), \quad (11)$$

$$Z^k = \sigma(W^k Z^{k-1} + b^k), \quad 1 \le k \le K-1, \quad (12)$$

$$Z^k = W^k Z^{k-1} + b^k, \quad k = K, \quad (13)$$

where K is the total layers of this neural network and the output of the last layer is utilized to approximate the true solution, namely $\hat{u} = Z^L$. W^k and b^k denote the weight matrix and bias vector of the *kth* layer; $\sigma(\cdots)$ is a nonlinear activation function. All the trainable model parameters, i.e., weights and biases, are denoted by ϑ in this paper.

In PINNs, solving a PDE system is transformed into an optimization problem by iteratively updating ϑ with the objective of minimizing the loss function *Lo*,

$$Lo = w_1 L_{PDE} + w_2 L_{data} + w_3 L_{IC} + w_4 L_{BC}, \quad (14)$$

where $w_i (i = 1, 2, 3, 4)$ are the weighting coefficients for different loss terms. The first term L_{PDE} penalizes the residual of the governing equations. The other terms are imposed to satisfy the model predictions for the measurements L_{data}, the initial condition L_{IC}, and the boundary condition L_{BC}, respectively. Typically, the mean square error, computed using the $L2$-norm of the sampling points, is utilized to calculate the losses in Eq. (14). The sampling points are represented as a data set $\{\boldsymbol{x}_i, t_i\}_{i=1}^N$, where the number of points (N) for different loss terms can vary. Generally, the Adam optimizer (Kingma and Ba, 2014), an adaptive algorithm for gradient-based first-order optimization, was used to optimize the model parameters ϑ. The detailed process of PINN is given in Fig. 1.

To construct the PDE loss in Eq. (14), a key procedure involves computing partial derivatives using automatic differentiation (AD). AD leverages the chain rule to calculate derivatives of the outputs with respect to the network inputs directly in the computational graph. This approach allows for explicit expression of partial derivatives, avoiding the introduction of discretization and truncation errors seen in conventional numerical methods. However, parameterizing the PDE and its derivatives with neural networks in PINNs may lead to generalization errors and optimization errors based on the training data and the optimizer, respectively (Shin *et al.*, 2020). Currently, AD has been integrated into various deep learning frameworks (Abadi *et al.*, 2016; Paszke *et al.*, 2019), facilitating the development of PINNs.

B. Case study for incompressible flows using PINN

The aerodynamic analysis of supertall buildings and large-span bridges, with their varied external surfaces and surrounding complex wind fields, has always been a challenging area of research (Cai *et al.*, 2023; Wang *et al.*, 2025). Historically, simplifications were made by reducing 3 D flow field problems to two-dimensional cross sections. With the evolution of computational methods, there is an innovative application of PINN to obtain the flow field distribution of two-dimensional cross sections. This method offers a precise representation of complex flow field, and a better understanding of the structural response characteristics in wind fields. Raissi *et al.* (2017b) employed PINN to analyze the incompressible flow and dynamic vortex shedding around a circular cylinder at Reynolds number Re = 100. The focus of Raissi *et al.* (2019a) is to predict the lift and drag forces on a structure using some limited and scattered information on the velocity field. This challenging inverse problem is addressed using DNN extended to encode the incompressible NS equations coupled with the structure's dynamic motion equation. The proposed algorithm is applicable to non-Newtonian, compressible, or partially known flows, allowing inference of unknown parameters such as Reynolds and Péclet numbers, in addition to velocity and pressure fields. Their work was further extended in Raissi *et al.* (2019b), where PINN were applied to the prototype problem of incompressible flow past a circular cylinder, demonstrating the model's ability to capture the periodic steady-state behavior and asymmetrical vortex shedding patterns, known as the Kármán vortex street.

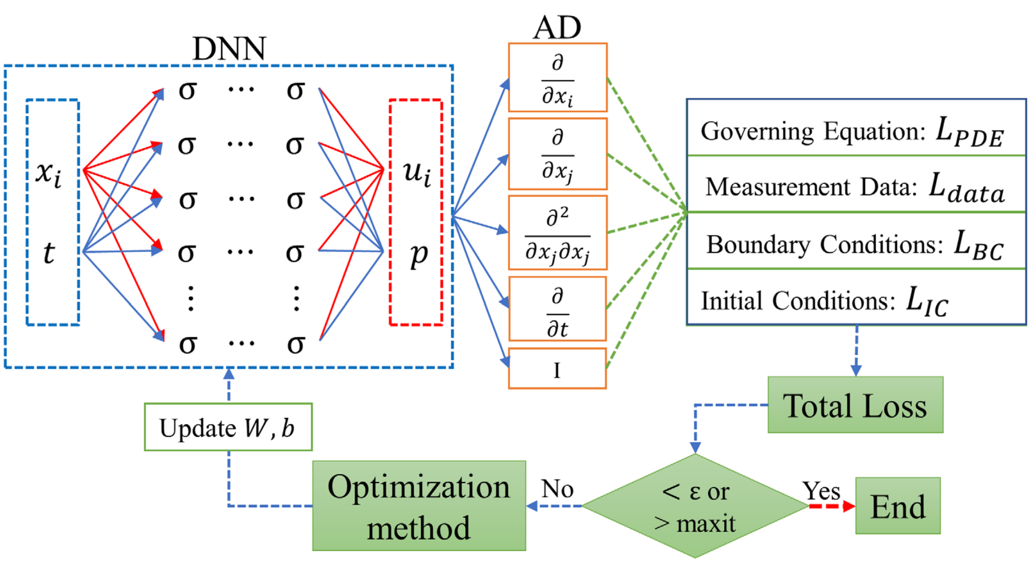

FIG. 1. Schematic of the PINN.

Zhang *et al.* (2020) introduced a frequency-compensated PINN framework capable of constructing a highly accurate surrogate for predicting the 2D incompressible fluid flow past a cylinder. This model employs a subnetwork that learns the Fourier components as functions of inlet velocity and cylinder size, enhancing the system's ability to generalize across different spatial and temporal domains, even for novel geometrical configurations. Furthermore, advancements were reported by Cheng and Zhang (2021), who proposed integrating PINN with Resnet blocks to solve dynamic vortex shedding and vortex-induced vibrations around a circular cylinder at high Reynolds numbers. In contrast to traditional data-driven neural networks, they embedded the RANS equations with an additional turbulent eddy viscosity into the loss function. Training and validation data were obtained using CFD techniques. The performance of PINN in solving vortex-induced vibration of cylinders and wake-induced vibration of a cylinder behind another cylinder was validated through three scenarios. In a related development, Eivazi *et al.* (2022) employed PINNs to solve the RANS equations for incompressible turbulent flows by taking only the data on the domain boundaries for the simulation of four turbulent flow cases, i.e., zero-pressure-gradient boundary layer, adverse-pressure-gradient boundary layer, and turbulent flows over a NACA4412 airfoil and the periodic hill. Analysis results show the remarkable accuracy with predictions of PINNs for laminar flows showing less than 1% error in cases involving strong pressure gradients. Most recently, Yan *et al.* (2023) used PINN to investigate the 2D incompressible fluid flow across a generic bridge deck section at varying Reynolds numbers. The network successfully extracted velocity and pressure fields from the concentration field and accurately predicted drag and lift coefficients, showcasing results comparable to those obtained via CFD.

Notably, simple two-dimensional sectional flow field studies cannot always accurately represent 3D flow field around supertall buildings or large-span bridges. This indicates the importance of directly solving for the 3D flow field to comprehensively understand the characteristics and evolution of such flow field. The complexity of neural networks based on PINN for solving the 3D NS equations presents both challenges and opportunities for advancement in this field. Research in this area is currently in its preliminary stage. A notable advance was made by Raissi *et al.* (2018c), who used PINN to infer hidden quantities of interest such as velocity and pressure fields merely from spatiotemporal visualizations of a passive scaler (e.g., dye or smoke). Their algorithm successfully predicted the pressure and velocity fields in both 2D and 3D flows past the circular cylinder for several benchmark problems relevant to real-world applications. Furthermore, exploration by Jin *et al.* (2021) led to the development of the velocity–pressure (VP) and the vorticity–velocity (VV) forms of the governing equations based on PINNs to solve the incompressible 3D NS equation. Their findings indicated that the VV form method yielded higher solution accuracy compared to the VP form when applied to flow around a cylinder at a Reynolds number of 100. Rui *et al.* (2023) reconstructed a 3D wind field around a building model in wind tunnel by proposing a novel PINN framework. This framework introduced a dynamic prioritization self-adaptive loss balance strategy, which adaptively reconciles the loss terms during PINN training to optimize multi-objective optimization, thereby facilitating convergence. Meanwhile, the zero-equation turbulence model and the wind velocity data near wall were embedded in PINN. The result demonstrated that novel PINN framework could be a powerful mean for wind field simulation and reconstruction in wind engineering applications. Currently, the application of PINN to real-world 3D flow field problems remains largely developmental. A primary challenge is often insufficient data available in real-world flow fields. To address this, a derivative method based on PINN has been proposed. This method enables the reconstruction of complex 3D flow fields using a limited number of data samples, thus broadening the potential applications of PINNs in practical.

C. The notable derivation of PINN: Knowledge-enhanced deep learning

As mentioned earlier, expertise in the domain can be utilized to enhance the performance of ML models in wind engineering applications. A good understanding of fundamental physics and other relevant types of domain knowledge specific to each subfield of wind engineering would facilitate a more efficient utilization of ML tools. The integration of such knowledge is crucial for improving the accuracy and reliability of predictions in structural wind engineering, especially when dealing with complex phenomena like typhoons, tornadoes, and thunderstorm (Wang *et al.*, 2022, 2023; Cai *et al.*, 2021a, 2021b, 2022). It is noted that the fundamental physics in terms of governing equations is a special type of domain knowledge, and recent studies have demonstrated that incorporating the underlying physics into training process can significantly reduce the need for labeled datasets, enhancing the regularization mechanism (e.g., Raissi *et al.*, 2017a; 2017b). Additionally, other equation-based domain knowledge such as empirical/semi-empirical formulas was also regarded as part of the loss function in deep learning to provide machine-readable prior knowledge for the effective regularization of neural networks in simulating tropical cyclone winds (Snaiki and Wu, 2019, 2022). In this reference, the simulation of tropical cyclone boundary-layer wind requires both the spatial coordinates and standard storm parameters (denote by $\varnothing = (\Delta p; r_m; c; \theta_0; \varphi; z_0)$) as inputs, shown in Fig. 2. Different from the traditional PINN, an additional term L_{ef} that denotes the loss in terms of the empirical/semi-empirical formulas was embedded into the PINN framework. The central pressure difference Δp, radius of maximum winds r_m, translational speed c, approach angle θ_0, and hurricane center location φ can be readily obtained from the National Hurricane Center, and the surface roughness z_0 can be obtained using the available database on Land Use/Land Cover. This knowledge-enhanced deep learning (KEDL) approach provided the 3D boundary-layer wind field with high computational efficiency and accuracy. Furthermore, Zhang *et al.* (2023) adopted an KEDL-based approach to reconstruct the tornado vortices from limited observed data, where the simplified tornado governing equations was embedded in a DNN as the prior physical information. This approach was validated by the reconstruction of different numerical tornado vortices generated by large eddy simulation and then applied to reconstruct a real tornado vortex.

This recent success of this novel KEDL approach underscores its potential to substantially advance ML applications in wind engineering. To fully realize the promise of KEDL, systematic research efforts are needed to efficiently identify knowledge representations (such as physics equations, empirical formulas, probabilistic relations, logic rules, and others) in structural wind engineering. Following this, it is crucial to integrate these elements into each component of the ML

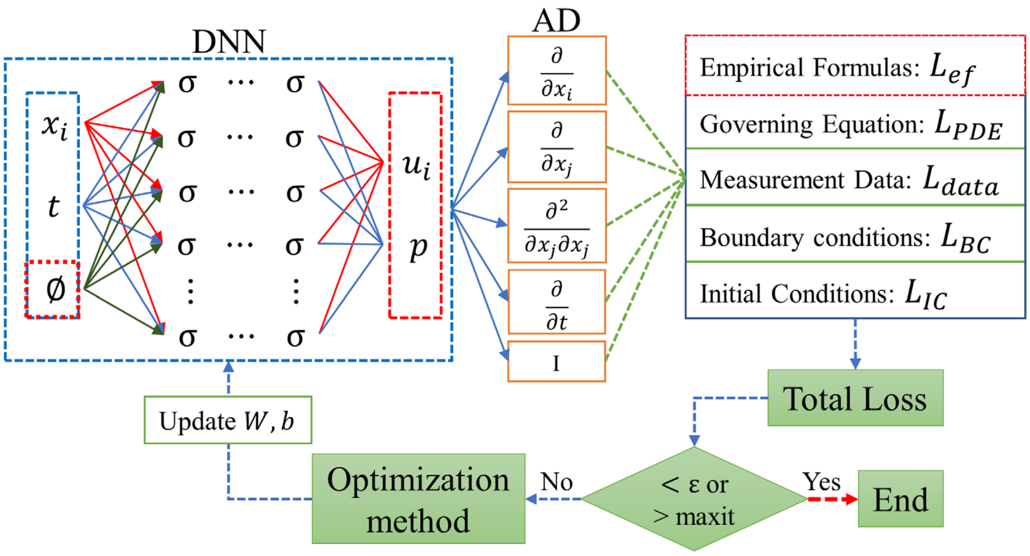

FIG. 2. Schematic of a knowledge-enhanced deep learning.

framework. While domain knowledge could be employed to enhance purely data-driven ML tools, it is expected that learning machines could be utilized for harnessing data to discover new knowledge in wind engineering (e.g., governing laws characterizing transport of turbulence quantities or optimization of wind-structure system). Ultimately, the synergy between domain knowledge and machine learning will pave the way for innovative solutions to complex challenges in wind engineering, enhancing the resilience and performance of structures in wind-prone environments.

IV. PINN FUTURE DIRECTIONS

The next step for advancing the theoretical or practical framework of PINN is not yet clearly defined. While recent literatures have improved PINN capabilities, several unresolved issues remain unaddressed. These include practical applications to real-world situations and theoretical concerns related to convergence and stability, and implementation challenges such as managing boundary conditions, designing neural network, and refining optimization strategies. Although PINNs and other deep learning methods that incorporate physics priors demonstrate promise in effectively solving high-dimensional PDEs, they often fall short of accurately approximating solutions to PDEs when compared to traditional numerical methods. This is particularly evident in cases involving complex physical phenomena that display multi-scale, chaotic, or turbulent characteristics.

A. A aspect of overcoming theoretical drawback in PINN

A PINN consists of three main components: a neural network for approximation, a module to refine this approximation, and a module for minimizing the losses. The neural network architecture determines its ability to approximate a function, and the error we make in approximating is called approximation error. The approach to iteratively improving the approximator is influenced by how the loss is defined and the quantity of data points considered. The quality of iterations in minimizing the losses depends on the optimization process, leading to the optimization error. These elements catalyze a multitude of research questions concerning the future of PINNs, particularly regarding their convergence to the correct solutions of a differential equations. To achieve convergence and stability, approximation errors must asymptotically approach zero, a process significantly influenced by the network's topology. The output of the current research on the topology of PINNs is extremely limited. For instance, Mo *et al.* (2022) altered the number of hidden layers and neurons per layer to calculate the relative error for various neural architectures. Similarly, Blechschmidt and Ernst (2021) demonstrated the number of successes for different network topologies after ten iterations, considering the number of layers, neurons, and activation function. Conversely, Mishra and Molinaro (2022) derived error estimates and identified potential methods for PINNs to approximate PDEs. It appears that the initial hidden layers may be responsible for encoding low-frequency components, while subsequent layers may handle higher-frequency components (Markidis, 2021). This concept could be an expansion of the Frequency principle (Xu, *et al.*, 2020). According to this principle, DNNs fit target functions from low to high frequencies during training, indicating a low-frequency bias in DNNs and explains why DNNs do not generalize well on randomized dataset.

The impact of initialization and loss function on generalization error in DNN learning is a critical area of research that need further exploration. While numerous theoretical studies have focused on loss estimation via quadrature rules applied to randomly and identically distributed points, the strategy employed by some PINNs involves selectively positioning collocation points within specific regions of the space–time domain, a method that merits additional scrutiny (Nabian *et al.*, 2021). Moreover, the application of dynamic loss weighting within PINNs presents a promising avenue for future research, as highlighted by Nandi *et al.* (2022).

Another interesting research topic is the apparent insensitivity of PINNs to the dimensionality of problems. Research indicates that PINNs can scale up effectively, independent of the size of the problems, without the computational cost exponentially increasing with problem dimensionality—a characteristic that remains somewhat enigmatic within neural network architectures (Mishra and Molinaro, 2021; De Ryck and Mishra, 2021). Recent work by Bauer and Kohler (2019) has demonstrated that least squares estimates based on feedforward neural networks (FFNNs) can circumvent the curse of dimensionality in nonparametric regression. Additionally, Zubov *et al.* (2021) have effectively utilized the PINN technique with quadrature methods to solve high-dimensional problems. In general, the failures of PINNs to accurately approximate solutions are often not attributable to a lack of expressivity within the neural network architecture, but due to the challenges associated with optimizing soft PDE constraints (Krishnapriyan *et al.*, 2021).

B. A aspect of improving implementation in PINN

When developing a PINN, it is important to thoroughly investigate additional configurations and consider various tweaks and best practices. This involves addressing each of the PINN's three modules, from neural network architecture options to activation function types. Recent studies have suggested that extending research beyond traditional FFNNs could significantly advance PINN capabilities. There is a growing interest in exploring the potential of non-FFNN architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) within the PINN framework (Wang *et al.*, 2022). Additionally, even within FFNNs, critical questions about optimal network size and configuration remain (Haghighat, *et al.*, 2021), with various alternative architectures from the broader deep learning field suggesting possible benefits for PINN development (Dargan *et al.*, 2020). For example, Fourier neural operator has been proposed for learning generalized functional spaces, potentially enhancing PINN performance across various applications (Wen *et al.*, 2022; Rafiq *et al.*, 2022). N-BEATS, a deep stack of fully connected layers connected with forward and backward residual links, is suitable for analyzing time series-based phenomena (Oreshkin *et al.*, 2020). The Transformer architecture is capable of managing long-range dependencies by representing global relationships in intricate physical problems (Kashinath *et al.*, 2021). Additionally, sinusoidal representation networks are well-suited for capturing complex natural signals and their derivatives and could be applied in PINN (Sitzmann *et al.*, 2020). Some initial findings can be found in Huang *et al.* (2021) and Wong *et al.* (2022). Another area of research aims to investigate whether enhancing the width or depth of the FFNN yields better results in improving PINN outcomes. This raises the question of whether there exists a minimum depth/width threshold necessary for networks to effectively capture and compute complex physical laws (Torabi Rad *et al.*, 2020). Future research in PINN will crucially focus on enhancing compatibility and understanding activation functions more deeply, which are essential for improving network performance in terms of convergence rates and solution accuracy (Rudin *et al.*, 2022). Jagtap *et al.* (2020a) demonstrated that the scalable activation function can be optimized to enhance network performance, in terms of convergence rate and solution accuracy. Furthermore, investigation can explore alternative or combined approaches to differentiate the differential equations. To accelerate PINN training, Chiu *et al.* (2022) introduced an innovative approach by defining the loss function using both numerical and automatic differentiation. The proposed coupled-automatic-numerical differentiation PINN (can-PINN) has proved to be more efficient and precise compared to traditional PINN setups, where automatic differentiation alone requires numerous collocation points to achieve high accuracy. While PINNs training points can be distributed spatially and temporally and making them highly versatile, the position of training locations affects the quality of the results. A notable limitation of PINNs is that the boundary conditions must be determined during the training phase, necessitating the creation of a new network if these conditions change (Wiecha *et al.*, 2021). In terms of loss, considering that neural networks tend to prioritize minimizing the largest loss component during training, it follows that all loss components should have similar magnitudes. If there is a significant disparity in the sizes of the loss components, the network may overlook the smaller components, leading to insufficient optimization in certain aspects of the model. Different loss components play distinct roles within the system, making it reasonable to assign them different weights to reflect their importance. In essence, this weight adjustment aims to control the influence of each loss component within a relatively similar range, thereby preventing any single loss from dominating the optimization process. However, the current literature lacks an objective method for determining these weight variables, which represents a significant research gap (Nandi *et al.*, 2021).

Optimization tasks within PINNs also require deeper investigation. Standard optimization algorithms like Adam and Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithms are predominantly employed (Wong *et al.*, 2021). The Adam algorithm establishes a workflow that can be analyzed using dynamical systems theory, creating a gradient descent dynamic. Moreover, to alleviate stiffness in gradient flow dynamics, investigating the limiting neural tangent kernel is necessary. Considering that significant efforts to solve optimization problems or improve these approaches in ML has been done, there remains considerable potential to enhance these techniques within the context of PINNs (Sun *et al.*, 2020b). The L-BFGS-B is the most commonly used BFGS in PINNs, and it is currently the most crucial PINN technology according to Markidis (2021). Furthermore, the full impact of learning rate on PINN training behavior has not been thoroughly explored. Gradient normalization is another important area of research (Nandi *et al.*, 2022). This method dynamically assigns weights to different constraints to eliminate the dominance of any component of the global loss function. To accelerate the solution process of the NS equations, Wei *et al.* (2019) proposed a unified framework based on deep reinforcement learning, which proved more effective in capturing shock waves compared to traditional finite element methods in solving the Burgers equation. This approach also demonstrated an increase in solution speed with the number of solution steps, exhibiting transfer learning characteristics. Research on training strategies for network parameters when transitioning from solving low Reynolds number to high Reynolds number has also been conducted by Wei *et al.* (2019), which improves computational efficiency of PINN. Additionally, Meng *et al.* (2020) proposed a temporally parallel approach for PINNs by dividing the time domain into subdomains. Subsequently, a coarse-scale network corresponds to the complete time domain and N fine-scale networks is, respectively, linked to N subdomains. By alternately training these two types of networks, this method can enhance the computational efficiency of PINNs when solving problems with longer time domains. In addition, the computational efficiency of PINNs can be greatly improved by designing parallel computing frameworks, such as the parallel computing program (i.e., SimNet developed by

NVIDIA (Hennigh *et al.*, 2021)). In a related development, Dong and Li (2021) combined extreme learning machines, domain decomposition, and local neural network, introducing more prior functions to effectively enhance the efficiency of network solution.

C. An aspect of integrating field measurement and simulated wind data

One of the primary challenges in applying PINN to solve real 3D typhoon flow field problems is the limited availability of training data. While KEDL methods can mitigate the data requirement, increasing the data volume can significantly improve model convergence and accuracy. Hence, it is crucial to procure an extensive range of wind field data, encompassing not only wind speed but also diverse parameters like temperature and air pressure fields. In terms of data collection, traditional meteorological wind measurement towers have been complemented by modern lidar wind profilers and simulated data, such as typhoon wind field data from the Weather Research and Forecasting Model (WRF). Inspired by the work of Mao *et al.* (2020), it is noted that while the data obtained from WRF can be distributed throughout the space, the measured data are mainly concentrated near the ground. This solution suggests that dividing domain into several subdomains and using different deep networks may improve the solution accuracy.

In practical applications, it is crucial to integrate these different data sources effectively, ensuring their accuracy and consistency. This integration requires rigorous quality control and calibration of observational data, as well as validation and verification of simulated data. Given the inherent differences and uncertainties among various data sources, systematic research into data fusion and integration methods is necessary. Such research aims to maximize the advantages of various data sources and minimize the limitations of each data type.

Simultaneously, attention must be given to the spatiotemporal distribution characteristics of data collection to ensure extensive and representative coverage. This entails scientifically planning and appropriately arranging the deployment of observation devices and the selection of simulated regions, as well as effectively integrating and utilizing data across different temporal and spatial scales. By fully leveraging diverse data collection methods and considering various factors, a more comprehensive understanding of the features of typhoon 3 D flow fields can be achieved, providing more reliable support for model training and prediction.

V. CONCLUSIONS

PINNs present a novel method for simulating realistic fluid flows, utilizing data from multimodal sources, whereas the boundary conditions or initial conditions are unknown. Although this is commonly encountered in practical scenarios, current CFD solvers are unable to address such ill-posed problems. Therefore, PINNs can be considered a supplementary tool to traditional numerical methods in CFD, offering new avenues for tackling idealized and real-world issues.

This review article provides an overview of the application of PINNs in solving the NS equations in structural wind engineering. It presents numerous case studies for solving two-dimensional and 3D flow fields, along with the KEDL method for reconstructing wind fields for typhoons, tornadoes, and other phenomena. Additionally, this study also discusses potential improvements in utilizing PINNs to solve the NS equations in wind engineering, considering the theoretical drawback, implementation process and data integration.

ACKNOWLEDGMENTS

This work was partially supported by the Research Grants Council of Hong Kong Theme-based Research Scheme (Grant No. T22-501/23-R). The authors would like to express sincere gratitude to Professor Yi-Qing Ni from The Hong Kong Polytechnic University for his support.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Kang Cai: Conceptualization (equal); Methodology (equal); Writing – original draft (equal). **Jiayao Wang:** Methodology (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

REFERENCES

Abadi, M., Barham, P., Chen, J. *et al.*, "Tensorflow: A system for large-scale machine learning," In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (USENIX, 2016), pp. 265–283.
Bauer, B. and Kohler, M., "On deep learning as a remedy for the curse of dimensionality in nonparametric regression," Ann. Statist. **47**(4), 2261–2285 (2019).
Bellman, R., "Dynamic programming," Science **153**, 34–37 (1966).
Berg, J. and Nyström, K., "A unified deep artificial neural network approach to partial differential equations in complex geometries," Neurocomputing **317**, 28–41 (2018).
Blechschmidt, J. and Ernst, O. G., "Three ways to solve partial differential equations with neural networks A review," GAMM-Mitteilungen **44**, e202100006 (2021).
Brooks, A. N. and Hughes, T. J., "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," Comput. Methods Appl. Mech. Eng. **32**, 199–259 (1982).
Cai, K., Li, X., and Zhi, L. H., "Extracting time-varying mean component of nonstationary winds utilizing vondrak filter and genetic algorithm: A wind engineering perspective," Int. J. Struct. Stab. Dyn. **21**(11), 2150155 (2021a).
Cai, K., Li, X., Zhi, L. H., and Han, X. L., "Extraction of optimal time-varying mean of non-stationary wind speeds based on empirical mode decomposition," Struct. Eng. Mech. **77**(3), 355–368 (2021b).
Cai, K., Huang, M. F., Xu, H. W., and Kareem, A., "Analysis of nonstationary typhoon winds based on optimal time-varying mean wind speed," J. Struct. Eng. **148**(12), 4022199 (2022).
Cai, K., Huang, M. F., Li, X., Xu, H. W., Li, B. B., and Yang, C., "Modal parameter identification of tall buildings based on variational mode decomposition and energy separation," Wind Struct. **37**(6), 445–460 (2023).
Cai, S., Mao, Z., Wang, Z *et al.*, "Physics-informed neural networks (PINNs) for fluid mechanics: A review," Acta Mech. Sin. **37**(12), 1727–1738 (2021).
Cheng, C. and Zhang, G. T., "Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems," Water **13**, 423 (2021).
Chiu, P. H., Wong, J. C., Ooi, C *et al.*, "CAN-PINN: A fast physics-informed neural network based on coupled-automatic–numerical differentiation method," Comput. Methods Appl. Mech. Eng. **395**, 114909 (2022).

Dargan, S., Kumar, M., Ayyagari, M. R. *et al.*, "A survey of deep learning and its applications: A new paradigm to machine learning," Arch. Comput. Methods Eng. **27**(4), 1071–1092 (2020).
De Ryck, T. and Mishra, S. "Error analysis for physics informed neural networks (PINNs) approximating Kolmogorov PDEs," arXiv:2106.14473 (2021).
Dissanayake, M. W. M. G. and Phan-Thien, N., "Neural-network-based approximations for solving partial differential equations," Commun. Numer. Methods Eng. **10**(3), 195–201 (1994).
Dong, S. C. and Li, Z. W., "Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations," Comput. Methods Appl. Mech. Eng. **387**, 114129 (2021).
Eivazi, H., Tahani, M., Schlatter, P., and Vinuesa, R., "Physics-informed neural networks for solving reynolds-averaged navier–stokes equations," Phys. Fluids **34**, 075117 (2022).
E, W. and Yu, B., "The deep ritz method: A deep learning-based numerical algorithm for solving variational problems," Commun. Math. Stat. **6**(1), 1–12 (2018).
Haghighat, E., Bekar, A. C., Madenci, E *et al.*, "A nonlocal physics-informed deep learning framework using the peridynamic differential operator," Comput. Methods Appl. Mech. Eng. **385**, 114012 (2021).
Haghighat, E. and Juanes, R., "SciANN: A Keras/Tensorflow wrapper for scientific computations and physics informed deep learning using artificial neural networks," Comput. Methods Appl. Mech. Eng. **373**, 113552 (2021).
Hennigh, O., Narasimhan, S., Nabian, M. A. *et al.*, "NVIDIA SimNet^{TM}: An AI-accelerated multi-physics simulation framework," *International Conference on Computational Science* (ACM, 2021), pp. 447–461.
Hornik, K., Stinchcombe, M., and White, H., "Multilayer feedforward networks are universal approximators," Neural Networks **2**(5), 359–366 (1989).
Huang, X., Liu, H., Shi, B *et al.*, "Solving partial differential equations with point source based on physics-informed neural networks," arXiv:2111.01394 (2021).
Jasak, H., Jemcov, A., Tukovic, Z *et al.*, "OpenFOAM: A C++ library for complex physics simulations," Int. Workshop Coupled Methods Numer. Dyn. **1000**, 1–20 (2007).
Jagtap, A. D., Kawaguchi, K., and Karniadakis, G. E., "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," J. Comput. Phys. **404**, 109136 (2020a).
Jagtap, A. D., Kharazmi, E., and Karniadakis, G. E., "Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems," Comput. Methods Appl. Mech. Eng. **365**, 113028 (2020b).
Jin, X., Cai, S., Li, H *et al.*, "NSFnets (Navier-Stokes flow nets): physics-informed neural networks for the incompressible Navier-Stokes equations," J. Comput. Phys. **426**, 109951 (2021).
Karniadakis, G. E., Kevrekidis, I. G., Lu, L *et al.*, "Physics-informed machine learning," Nat. Rev. Phys. **3**(6), 422–440 (2021).
Karniadakis, G. E. and Sherwin, S., *Spectral/hp Element Methods for Computational Fluid Dynamics*, 2nd edn (Oxford University Press, Oxford, 2005).
Kashinath, K., Mustafa, M., Albert, A *et al.*, "Physics-informed machine learning: Case studies for weather and climate modelling," Phil. Trans. R Soc. A **379**, 20200093 (2021).
Katz, A. J., *Meshless Methods for Computational Fluid Dynamics* (Stanford University Stanford, Stanford, 2009).
Kharazmi, E., Zhang, Z., and Karniadakis, G. E., "hp-VPINNs: Variational physics-informed neural networks with domain decomposition," Comput. Methods Appl. Mech. Eng. **374**, 113547 (2021).
Kingma, D. P. and Ba, J. "Adam: A method for stochastic optimization," arXiv:1412.6980 (2014).
Kollmannsberger, S., D'Angella, D., Jokeit, M *et al. Deep Learning in Computational Mechanics, Studies in Computational Intelligence* (Springer International Publishing, Cham, 2021).
Kondor, R. and Trivedi, S., "On the generalization of equivariance and convolution in neural networks to the action of compact groups," Proc. 35th Int. Conf. Mach. Learn. **80**, 2747–2755 (2018).
Krishnapriyan, A., Gholami, A., Zhe, S *et al.* "Characterizing possible failure modes in physics-informed neural networks," arXiv:2109.01050 (2021).
Kumar, M. and Yadav, N., "Multilayer perceptrons and radial basis function neural network methods for the solution of differential equations: A survey," Computers Mathematics with Appl. **62**(10), 3796–3811 (2011).
Lagaris, I., Likas, A., and Fotiadis, D., "Artificial neural networks for solving ordinary and partial differential equations," IEEE Trans. Neural Networks **9**(5), 987–1000 (1998).
Lagaris, I., Likas, A., and Papageorgiou, D., "Neural-network methods for boundary value problems with irregular boundaries," IEEE Trans. Neural Networks **11**(5), 1041–1049 (2000).
Lee, H. and Kang, I. S., "Neural algorithm for solving differential equations," J. Comput. Phys. **91**(1), 110–131 (1990).
Liu, D. and Wang, Y., "A Dual-Dimer method for training physics-constrained neural networks with minimax architecture," Neural Networks **136**, 112–125 (2021).
Lu, L., Pestourie, R., Yao, W. *et al.* "Physics-informed neural networks with hard constraints for inverse design," arXiv:2102.04626 (2021).
Mallat, S., "Understanding deep convolutional networks," Phil. Trans. R Soc. A **374**, 20150203 (2016).
Mao, Z., Jagtap, A. D., and Karniadakis, G. E., "Physics-informed neural networks for high-speed flows," Comput. Methods Appl. Mech. Eng. **360**, 112789 (2020).
Markidis, S., "The old and the new: Can physics-informed deep-learning replace traditional linear solvers?," Front. Big Data. **4**, 669097 (2021).
Meng, X. H., Li, Z., Zhang, D. K. *et al.*, "PPINN: Parareal physics-informed neural network for time-dependent PDEs," Comput. Methods Appl. Mech. Eng. **370**, 113250 (2020).
Mishra, S. and Molinaro, R., "Physics informed neural networks for simulating radiative transfer," J. Quant. Spectrosc. Radiat. Transfer **270**, 107705 (2021).
Mishra, S. and Molinaro, R., "Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs," IMA J. Numer. Anal. **42**, 981–1022 (2022).
Mo, Y., Ling, L., and Zeng, D., "Data-driven vector soliton solutions of coupled nonlinear Schrödinger equation using a deep learning algorithm," Phys. Lett. A **421**, 127739 (2022).
Nabian, M. A., Gladstone, R. J., and Meidani, H., "Efficient training of physics-informed neural networks via importance sampling," Comput-Aided. Civ. Infrastruct. Eng. **36**(8), 962–977 (2021).
Nandi, T., Hennigh, O., Nabian, M *et al.*, "Progress towards solving high reynolds number reacting flows in SimNet," *NVIDIA GPU Technology Conference, USA* (NVIDIA, 2021).
Nandi, T., Hennigh, O., Nabian, M *et al.*, "Developing Digital Twins for Energy Applications Using Modulus," *NVIDIA GPU Technology Conference, USA* (NVIDIA, 2022).
Oreshkin, B. N., Carpov, D., Chapados, N *et al.*, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," arXiv:1905.10437 (2020).
Owhadi, H., "Bayesian numerical homogenization," Multiscale Model. Simul. **13**(3), 812–828 (2015).
Özbay, A. G., Hamzehloo, A., Laizet, S *et al.*, "Poisson CNN: Convolutional neural networks for the solution of the Poisson equation on a Cartesian mesh," DCE. **2**, e6 (2021).
Pang, G., Lu, L., and Karniadakis, G. E., "fPINNs: Fractional physics-informed neural networks," SIAM J. Sci. Comput. **41**(4), A2603–A2626 (2019).
Paszke, A., Gross, S., Chintala, S *et al.*, "Automatic differentiation in PyTorch," *31st Conference on Neural Information Processing Systems, USA* (Curran Associates, 2017).
Paszke, A., Gross, S., Massa, F *et al.*, "Pytorch: An imperative style, high-performance deep learning library," arXiv preprint arXiv:1912.01703 (2019).
Psichogios, D. C. and Ungar, L. H., "A hybrid neural network-first principles approach to process modeling," AlChE. J. **38**(10), 1499–1511 (1992).
Rafiq, M., Rafiq, G., and Choi, G. S., "DSFA-PINN: Deep Spectral feature aggregation physics informed neural network," IEEE Access **10**, 22247–22259 (2022).
Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations," arXiv:1711.10561 (2017a).
Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics informed deep learning (Part II): Data-driven discovery of nonlinear partial differential equations," arXiv:1711.10566 (2017b).

Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Machine learning of linear differential equations using Gaussian processes," J. Comput. Phys. **348**, 683–693 (2017c).
Raissi, M. and Karniadakis, G. E., "Hidden physics models: Machine learning of nonlinear partial differential equations," J. Comput. Phys. **357**, 125–141 (2018a).
Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Numerical Gaussian processes for time-dependent and nonlinear partial differential equations," SIAM J. Sci. Comput. **40**, A172–A198 (2018b).
Raissi, M., Yazdani, A., and Karniadakis, G. E., "Hidden fluid mechanics: A Navier-Stokes informed deep learning framework for assimilating flow visualization data," arXiv:1808.04327 (2018c).
Raissi, M., Wang, Z., Triantafyllou, M. S. *et al.*, "Deep learning of vortex-induced vibrations," J. Fluid Mech. **861**, 119–137 (2019a).
Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Fluid Mech. **378**, 686–707 (2019b).
Raissi, M., Yazdani, A., and Karniadakis, G. E., "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," Science **367**, 1026–1030 (2020).
Rui, E. Z., Chen, Z. W., Ni, Y. Q., Yuan, L., and Zeng, G. Z., "Reconstruction of 3D flow field around a building model in wind tunnel: A novel physics informed neural network framework adopting dynamic prioritization self-adaptive loss balance strategy," Eng. Appl. Comput. Fluid Mech. **17**(1), 2238849 (2023).
Rudin, C., "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," Nat. Mach. Intell. **1**, 206–215 (2019).
Rudin, C., Chen, C., Chen, Z *et al.*, "Interpretable machine learning: Fundamental principles and 10 grand challenges," Stat. Survey **16**, 1–85 (2022).
Sirignano, J. and Spiliopoulos, K., "DGM: A deep learning algorithm for solving partial differential equations," J. Comput. Phys. **375**, 1339–1364 (2018).
Snaiki, R. and Wu, T., "Knowledge-enhanced deep learning for simulation of tropical cyclone boundary-layer winds," J. Wind Eng. Ind Aerodyn. **194**, 103983 (2019).
Snaiki, R. and Wu, T., "Knowledge-enhanced deep learning for simulation of extratropical cyclone wind risk," Atmosphere **13**(5), 757 (2022).;
Shin, Y., Darbon, J., and Karniadakis, G. E., "On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs," Comm. Comput. Phys. **28**, 2042–2074 (2020).
Sitzmann, V., Martel, J. N. P., Bergman, A. W. *et al.*, "Implicit neural representations with periodic activation functions," arXiv:2006.09661 (2020).
Sun, L., Gao, H., Pan, S *et al.*, "Surrogate modeling for fluid flows based on physics constrained deep learning without simulation data," Comput. Methods Appl. Mech. Eng. **361**, 112732 (2020a).
Sun, S., Cao, Z., Zhu, H *et al.*, "A Survey of Optimization Methods from a Machine Learning Perspective," IEEE Trans. Cybern. **50**(8), 3668–3681 (2020b).
Torabi Rad, M., Viardin, A., Schmitz, G. J. *et al.*, "Theory-training deep neural networks for an alloy solidification benchmark problem," Comput. Mater. Sci. **180**, 109687 (2020).
Vinuesa, R. and Sirmacek, B. "Interpretable deep-learning models to help achieve the sustainable development goals," arXiv:2108.10744 (2021).
Wang, J., Tse, T. K., Li, S., and Fung, J. C., "Prediction of the typhoon wind field in Hong Kong: Integrating the effects of climate change using the Shared Socioeconomic Pathways," Clim. Dyn. **59**(7), 2311–2329 (2022).
Wang, J., Tse, T. K., Li, S., and Fung, J. C., "A model of the sea–land transition of the mean wind profile in the tropical cyclone boundary layer considering climate changes," Int. J. Disaster Risk Sci. **14**(3), 413–427 (2023).
Wang, J., Cao, S., Tim, K. T., Li, S., Fung, J. C., and Li, Y., "A novel life-cycle analysis framework to assess the performances of tall buildings considering the climate change," Eng. Struct. **323**, 119258 (2025).
Wang, S., Yu, X., and Perdikaris, P., "When and why PINNs fail to train: A neural tangent kernel perspective," J. Comput. Phys. **449**, 110768 (2022).
Wei, S. Y., Jin, X. W., and Li, H., "General solutions for nonlinear differential equations: A rule-based self-learning approach using deep reinforcement learning," Comput. Mech. **64**(5), 1361–1374 (2019).
Wen, G., Li, Z., Azizzadenesheli, K *et al.*, "U-FNO–An enhanced Fourier neural operator-based deep learning model for multiphase flow," Adv. Water Resour. **163**, 104180 (2022).
Wiecha, P. R., Arbouet, A., Arbouet, A *et al.*, "Deep learning in nano-photonics: Inverse design and beyond," Photonics Res. **9**(5), B182–B200 (2021).
Wong, J. C., Gupta, A., and Ong, Y. S., "Can transfer neuroevolution tractably solve your differential equations?," IEEE Comput. Intell. Mag. **16**(2), 14–30 (2021).
Wong, J. C., Ooi, C., Gupta, A *et al.*, "Learning in sinusoidal spaces with physics-informed neural networks," arXiv:2109.09338 (2022).
Xu, Z. J., Zhang, Y. Y., Luo, T., Xiao, Y. Y., and Ma, Z., "Frequency principle: Fourier analysis sheds light on deep neural networks," Comm. Comput. Phys. **28**(5), 1746–1767 (2020).
Yan, H., Wang, Y., Yan, Y., and Cui, J., "Physics-informed neural network for flow prediction based on flow visualization in bridge engineering," Atmosphere **14**, 759 (2023).
Yang, L., Zhang, D., and Karniadakis, G. E., "Physics-informed generative adversarial networks for stochastic differential equations," SIAM J. Sci. Comput. **42**(1), A292–A317 (2020).
Yin, M., Zheng, X., Humphrey, J. D. *et al.*, "Non-invasive inference of thrombus material properties with physics-informed neural networks," Comput. Methods Appl. Mech. Eng. **375**, 113603 (2021).
Zhang, T. T., Dey, B., Kakkar, P., Dasgupta, A., and Chakraborty, A., "Frequency-compensated PINNs for fluid-dynamic design problems," arXiv:2011.01456 (2020).
Zhang, H., Wang, H., Xu, Z., Liu, Z., and Khoo, B. C., "A physics-informed neural network-based approach to reconstruct the tornado vortices from limited observed data," J. Wind Eng. Ind. Aerodyn. **241**, 105534 (2023).
Zhang, D., Lu, L., Guo, L *et al.*, "Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems," J. Comput. Phys. **397**, 108850 (2019).
Zhu, Y., Zabaras, N., Koutsourelakis, P. S. *et al.*, "Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data," J. Comput. Phys. **394**, 56–81 (2019).
Zubov, K., McCarthy, Z., Ma, Y *et al.*, "Neural PDE: Automating physics-informed neural networks (PINNs) with error approximations," arXiv:2107.09443 (2021).