Composites Part B 299 (2025) 112409

Contents lists available at ScienceDirect

Composites Part B

journal homepage: www.elsevier.com/locate/composites

Rapid estimation of residual stress in composite laminates using a deep operator network

Seung-Woo Lee [a], Teubes Christiaan Smit [b], Kyusoon Jung [a], Robert Grant Reid [b], Do-Nyun Kim [a,c,d,*]

[a] Department of Mechanical Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu Seoul 08826, Republic of Korea
[b] School of Mechanical, Industrial and Aeronautical Engineering, University of the Witwatersrand, Private Bag 3 Wits 2050, South Africa
[c] Institute of Advanced Machines and Design, Seoul National University, 1 Gwanak-ro, Gwanak-gu Seoul 08826, Republic of Korea
[d] Institute of Engineering Research, Seoul National University, 1 Gwanak-ro, Gwanak-gu Seoul 08826, Republic of Korea

ARTICLE INFO

Keywords:
Residual stress
Incremental hole drilling
Integral method
Composite laminate
Deep operator network

ABSTRACT

A deep operator network (DeepONet) is designed and developed for rapid estimation of residual stress in composite laminates, which traditionally requires intensive finite element method (FEM) calculations to calibrate the incremental hole-drilling (IHD) method used in measuring residual stresses. The proposed DeepONet model incorporates graph convolution, trigonometric series expansion, and Monte Carlo dropout to effectively learn the relationship between residual stress distribution and the corresponding deformation observed in the IHD procedure. This learning is based on FEM data from various symmetric composite laminate configurations, which are composed of eight layers of fiber-reinforced plates with possible ply orientations at −45°, 0°, 45°, and 90°. Trained on 30 configurations, the proposed model exhibits strong generalization capabilities over an additional 40 unseen configurations, achieving a forward strain prediction error of 1.59% and an inverse stress calculation error of 3.92%. These errors are within the range of experimental noise and corresponding stress uncertainty levels commonly encountered in real experiments. The performance of the model suggests the potential for establishing a comprehensive database for the IHD method as applied to composite materials, filling a significant gap in resources when compared to those available for metallic materials.

1. Introduction

Residual stress within materials is a critical factor in addressing a wide range of engineering problems such as material design [1], micromechanics [2,3], additive manufacturing [4,5], fracture mechanics [6,7], and biomechanics [8]. The incremental hole-drilling method (IHD), a prevalent semi-destructive method, is employed for measuring residual stresses by gradually drilling a hole in the stressed material and recording the strain response. The strain data obtained are then used to inversely determine the residual stresses released during the material's removal [9]. The IHD method is widely applicable in various materials, including metals [10], composites [11,12], and fiber-metal laminates [13]. It proves particularly beneficial for materials with non-crystalline microstructures, where non-destructive testing methods are constrained [14]. Nevertheless, the inverse calculation in IHD necessitates numerous calibration coefficients to correlate strain measurements with released stress, which are derived from repetitive finite element method (FEM) calculations [15]. The demand for rapid, efficient, and modern methods for measuring residual stress is increasingly pressing within the industrial sector, a challenge that traditional FEM-based approaches find difficult to address. This difficulty arises primarily from the high computational costs associated with FEM, particularly for composite materials, which necessitate unique calibration coefficients for each material combination [16].

The emergence of deep neural networks and data-driven modeling has opened promising pathways for enhancing efficiency in the scientific and engineering realms. This development is especially significant for its ability to streamline the complex and resource-demanding processes inherent in traditional physics-based models. The adaptability and broad applicability of deep neural networks have been demonstrated in their successful application to a range of challenging mechanics problems, including rapid topology optimization [17], multiscale analysis [18,19], as well as multiphysics analyses that integrate thermo-mechanical [20], electro-mechanical [21], and magneto-mechanical [22] couplings. Furthermore, neural network models have

* Corresponding author.Department of Mechanical Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu Seoul 08826, Republic of Korea.
E-mail address: dnkim@snu.ac.kr (D.-N. Kim).

https://doi.org/10.1016/j.compositesb.2025.112409
Received 19 August 2024; Received in revised form 2 December 2024; Accepted 13 March 2025
Available online 22 March 2025
1359-8368/© 2025 Elsevier Ltd. All rights reserved, including those for text and data mining, AI training, and similar technologies.

shown exceptional efficacy in integrating experimental data into numerical methods for material identification [23], highlighting their robustness in managing realistic measurement noise [24].

The deep operator network (DeepONet), developed by Lu et al. is characterized by a unique dual deep neural network architecture [25]. It includes a branch network designed to encode the discrete representation of input function and a trunk network that encodes the domain of the output functions. This innovative structure facilitates the approximation of the operator that connects an input function represented by the branch network to an output function in the domain represented by the trunk network. DeepONet is unique in its approach, shifting from traditional function approximation to operator approximation. This shift enables the efficient learning of both linear and nonlinear solution operators for parametric partial differential equations (PDEs). The trained operators serve as effective surrogates for otherwise computationally intensive physics-based models. DeepONet has found applications across a spectrum of mechanics problems, including heat conduction [26], boundary layer instability [27], phase field modeling [28], phase field modeling with governing free-energy flow [29], fracture mechanics [30], structures under sequential loading conditions [31], multiphysics problems in additive manufacturing [32], and multiscale problems employing domain decomposition [33]. Furthermore, significant efforts have been directed towards integrating advanced model architectures and training methods to elevate DeepONet's accuracy and reliability, fully capitalizing on its inherent flexibility as suggested in the original work on DeepONet [25]. These enhancements include the implementation of residual U-Net with information fusion [34], Fourier transform layers [35], training methodologies enriched with additional observations or governing equations [36], and uncertainty quantification [37,38]. Recent advancements in DeepONet have targeted large-scale and complex applications with strong generalization capabilities. Notable examples include zero-shot prediction aiming for robust digital twin systems [39], energy-dissipative frameworks for modeling classes of PDEs governed by gradient flows [40], and rapid full-field predictions on varying 3D geometries through advanced geometric encoding [41].

There has been an increasing interest in leveraging neural network models for enhancing residual stress measurement methods [16]. For instance, Mathew et al. deployed a neural network to predict welding-induced residual stress distributions across various geometries and welding parameters, targeting applications in nuclear plants. This model, validated against experimental data obtained through neutron diffraction and contour methods, demonstrated good agreement with experimental measurements [42,43]. In IHD, Held and Gibmeier proposed a novel neural network-based approach to determine the correction function for strain measurements in thick films. This model is capable of deriving the correction function from the parameters of thick film coatings, a task that traditionally necessitated FEM for each coating system [44]. Chupakhin et al. proposed a neural network-based method to accommodate the plasticity effect within IHD, effectively broadening the technique's stress level applicability up to yield strength and across large stress gradients in the thickness direction [45]. Despite these advancements, most previous efforts have been concentrated on predicting case-specific targets using fully connected layers, while little attention has been paid to the strategic design of model architectures that incorporate domain knowledge of the residual stress measurement process.

In this research, we present a DeepONet architecture specifically designed for surrogate modeling of the IHD process in composite laminates, representing a notable advancement in the application of neural network-based models for critical components of residual stress measurement. The DeepONet model is conceptually aligned with the general integral equation governing destructive residual stress measurement methods [46], closely associating with the kernel function and integration within the integral equation. The model, trained on FEM data, is adept at predicting the deformation caused by the release of residual stress. It serves dual purposes: as a forward surrogate model and for the swift generation of calibration matrices for inverse calculations, without the need for additional FEM. Remarkably, with training on merely 30 composite laminate configurations, the model demonstrates exceptional generalization ability to 40 additional configurations of varying stacking orders in both forward and inverse problems, achieving prediction errors within the experimental noise level and offering a significant acceleration over the traditional FEM-based approach. Additionally, the DeepONet architecture is enriched with features like graph convolution, trigonometric series expansion, and Monte Carlo dropout. This enhancement improves prediction accuracy and mitigates challenges related to the sparse dataset and the inherently ill-posed nature of the IHD inverse problem.

The contributions of this work can be summarized as follows:

1. To our knowledge, this study is the first to introduce a neural network model specifically designed for surrogate modeling and calibration matrix construction of IHD for composite laminates, marking a pioneering contribution in the field.
2. Drawing inspiration from the underlying general integral equation, our proposed DeepONet architecture facilitates a seamless extension to broad applications with the same background. These might include other destructive residual stress measurement methods such as Sachs' method, the layer removal method, and crack compliance, all of which are fundamentally based on the same mathematical structure of the Volterra equation of the first kind [46]. This highlights the potential versatility of our approach.
3. By incorporating advanced model architectures and model uncertainty quantification, the model exhibits solid generalization capabilities. This advancement paves the way for the establishment of a comprehensive database for the IHD method applied to composite materials.

The rest of this paper is structured into three main sections. Section 2 outlines the methodology, beginning with an overview of the IHD method and the integral method. This is followed by a detailed description of the FEM model, data generation process, and architecture of the proposed model. Section 3 presents the results of the model's training and prediction capabilities, including the hyperparameter tuning process, forward strain prediction, and inverse stress calculation. Additionally, an ablation study within this section evaluates the impact of individual components of the proposed DeepONet architecture on its overall performance. Section 4 concludes the paper by discussing the study's limitations and suggesting directions for future research.

2. Methods

2.1. Incremental hole-drilling and integral method

IHD is a semi-destructive residual stress measurement method where the removal of material is achieved by incrementally drilling a hole into the specimen. Removing a small portion of material from the specimen during each incremental depth leads to the release of residual stress, prompting the specimen to shift towards a new equilibrium state different from that before material removal. The magnitude of the released residual stress can be inversely determined by measuring the resulting deformations as the depth of the hole increases. Deformations are most commonly measured using a strain gauge rosette. Let h, $\sigma(H)$, and $\epsilon(h)$ denote the current depth of the hole, the residual stress released at depth H and the strain measured at hole depth h, respectively. The general relation between the released stress and the measured strain can be expressed by the Volterra equation of the first kind:

$$\epsilon(h) = \int_0^h K(H,h)\sigma(H)\,dH, \tag{1}$$

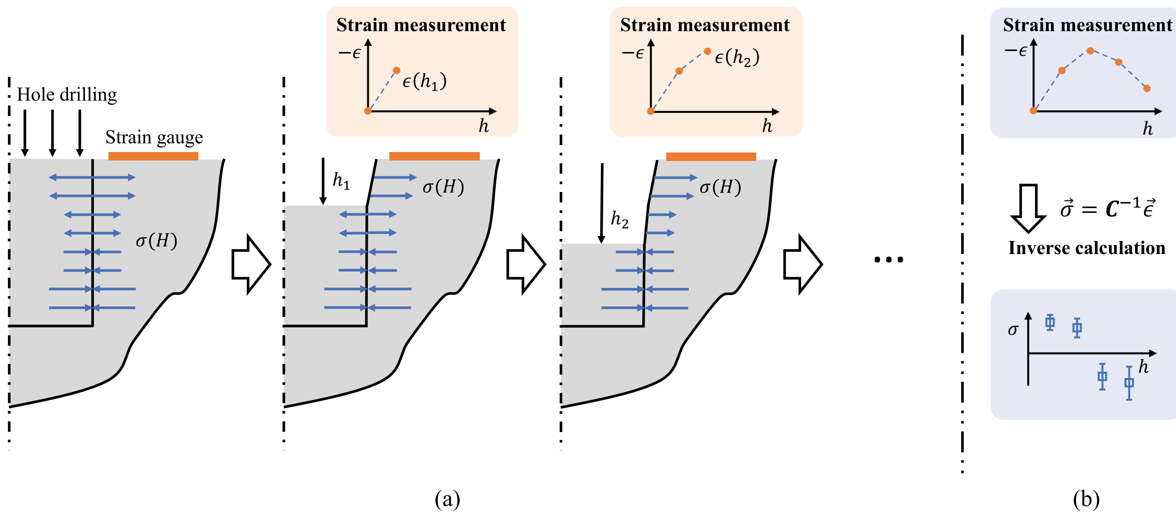


Fig. 1. Overview of incremental hole-drilling with the integral method. (a) Incremental hole-drilling and strain measurement, (b) inverse calculation using measured strain and calibration matrix.

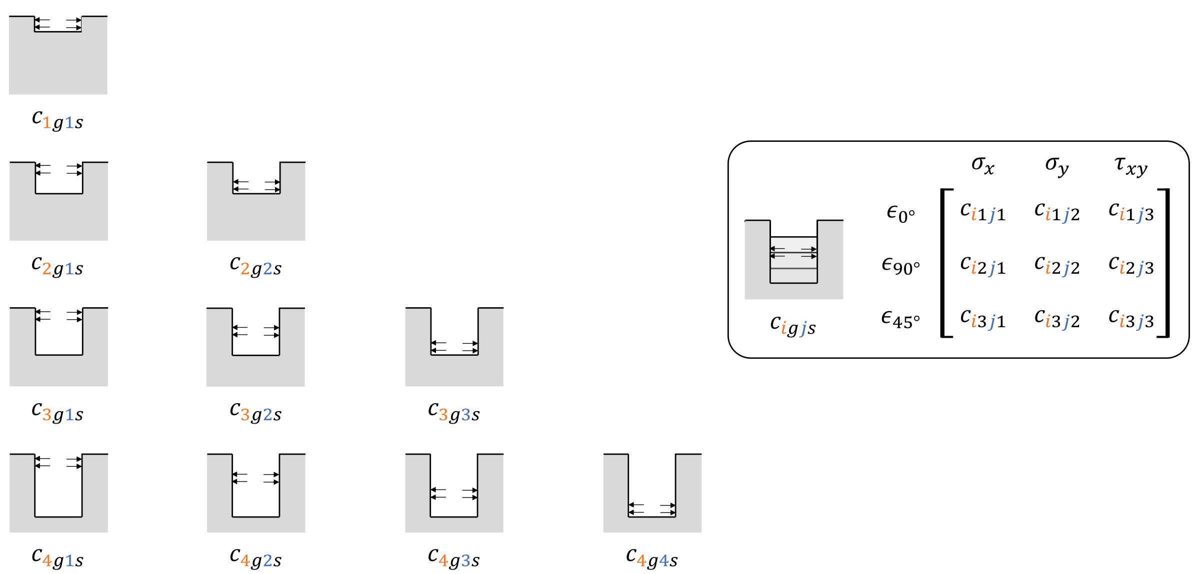


Fig. 2. Example of calibration matrix construction for 4 depth increment levels.

where $K(H,h)$ is the kernel function that describes the sensitivity of the measured strain $\epsilon(h)$ to the released stress $\sigma(H)$ [46]. In a real experiment, there are three components of residual stress, and strain is measured at three strain gauge locations. Assuming constant stress in each depth increment level $0 < h_1 < \cdots < h_j < \cdots < h_i$, the integral equation can be simplified as

$$\epsilon_{ig} = \sum_{s=1}^{3}\sum_{j=1}^{i} c_{igjs}\sigma_{js}, \tag{2}$$

where ϵ_{ig} denotes strain measured at gauge location g after depth increment i, and σ_{js} denotes stress component s released at the region of depth increment j. Specifically, we consider three strain gauges aligned with the x-axis ($\epsilon_{i1} = \epsilon_{i,0^\circ}$), the y-axis ($\epsilon_{i2} = \epsilon_{i,90^\circ}$), and at 45° orientation ($\epsilon_{i3} = \epsilon_{i,45^\circ}$). Also considered are the three stress components; the normal stresses $\sigma_{j1} = \sigma_{j,x}$ and $\sigma_{j2} = \sigma_{j,y}$, and the shear stress $\sigma_{j3} = \tau_{j,xy}$. Calibration coefficient c_{igjs} corresponds to the kernel function $K(h_j,h_i)$ in Eq. (1) for strain gauge g and stress component s. Eq. (2) can be written in matrix and vector notation.

$$\vec{\epsilon} = \boldsymbol{C}\vec{\sigma} \tag{3}$$

This integral equation and the procedure of incremental hole-drilling are depicted in Fig. 1. It is important to note that only the released stress (the residual stress present in the material at depths shallower than the current depth of the hole) influences the current strain measurement. Consequently, the calibration matrix $\boldsymbol{C}$ has a block lower triangular form [12]. Each element of the calibration matrix can be derived by applying a unit stress to a finite element model, as shown in Fig. 2. For instance, applying unit stress $\sigma_{2,x} = 1$ at the depth increment level 3 in the finite element model allows us to determine calibration coefficients based on strain values: $c_{3121} = \epsilon_{3,0^\circ}$, $c_{3221} = \epsilon_{3,90^\circ}$, and $c_{3321} = \epsilon_{3,45^\circ}$.

In this process, accurately modeling the strain gauge within the finite element model is crucial to address the non-uniform strain field experienced by a finite-sized strain gauge, as reflected in the strain ϵ_{ig} derived from the finite element solution. This study adopts the strain gauge modeling approach detailed in [11]. Within the finite element model, the strain gauge is depicted as a rectangular area, with nodal displacement interpolated at each edge of this rectangle. Considering the strain gauge to be aligned with the x-axis, the rectangular region

$$\begin{Bmatrix} \epsilon_{1,0^\circ} \\ \epsilon_{1,90^\circ} \\ \epsilon_{1,45^\circ} \\ \hline \epsilon_{2,0^\circ} \\ \epsilon_{2,90^\circ} \\ \epsilon_{2,45^\circ} \\ \hline \epsilon_{3,0^\circ} \\ \epsilon_{3,90^\circ} \\ \epsilon_{3,45^\circ} \\ \hline \vdots \\ \epsilon_{ig} \end{Bmatrix} = \left[\begin{array}{ccc|ccc|ccc|c} c_{1111} & c_{1112} & c_{1113} & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ c_{1211} & c_{1212} & c_{1213} & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ c_{1311} & c_{1312} & c_{1313} & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \hline c_{2111} & c_{2112} & c_{2113} & c_{2121} & c_{2122} & c_{2123} & 0 & 0 & 0 & \cdots \\ c_{2211} & c_{2212} & c_{2213} & c_{2221} & c_{2222} & c_{2223} & 0 & 0 & 0 & \cdots \\ c_{2311} & c_{2312} & c_{2313} & c_{2321} & c_{2322} & c_{2323} & 0 & 0 & 0 & \cdots \\ \hline c_{3111} & c_{3112} & c_{3113} & c_{3121} & c_{3122} & c_{3123} & c_{3131} & c_{3132} & c_{3133} & \cdots \\ c_{3211} & c_{3212} & c_{3213} & c_{3221} & c_{3222} & c_{3223} & c_{3231} & c_{3232} & c_{3233} & \cdots \\ c_{3311} & c_{3312} & c_{3313} & c_{3321} & c_{3322} & c_{3323} & c_{3331} & c_{3332} & c_{3333} & \cdots \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ c_{ig11} & c_{ig12} & c_{ig13} & c_{ig21} & c_{ig22} & c_{ig23} & c_{ig31} & c_{ig32} & c_{ig33} & c_{igjs} \end{array}\right] \begin{Bmatrix} \sigma_{1,x} \\ \sigma_{1,y} \\ \tau_{1,xy} \\ \hline \sigma_{2,x} \\ \sigma_{2,y} \\ \tau_{2,xy} \\ \hline \sigma_{3,x} \\ \sigma_{3,y} \\ \tau_{3,xy} \\ \hline \vdots \\ \sigma_{is} \end{Bmatrix} \quad (4)$$

Box I.

is specified as $G_{0^\circ} = \{(x,y)|1.8 \le x \le 3.3, -0.385 \le y \le 0.385\}$. Finite element nodes located within $1.7 \le x \le 3.6$ and $-0.4 \le y \le 0.4$ serve as data points and are denoted by $\vec{x}_{d,n}$, with $n = 1,2,\dots,N_d$. Query points are represented by equally spaced points along the edges of the rectangular region, denoted as $\vec{x}_{q,m}$, where $m = 1,2,\dots,N_q$. The Green's function matrix, which calculates pairwise distances between data points and between query points and data points, is computed accordingly.

$$\boldsymbol{G}_{dd} = \begin{bmatrix} g(\vec{x}_{d,1},\vec{x}_{d,1}) & g(\vec{x}_{d,1},\vec{x}_{d,2}) & \cdots & g(\vec{x}_{d,1},\vec{x}_{d,N_d}) \\ g(\vec{x}_{d,2},\vec{x}_{d,1}) & g(\vec{x}_{d,2},\vec{x}_{d,2}) & \cdots & g(\vec{x}_{d,2},\vec{x}_{d,N_d}) \\ \vdots & \vdots & \ddots & \vdots \\ g(\vec{x}_{d,N_d},\vec{x}_{d,1}) & g(\vec{x}_{d,N_d},\vec{x}_{d,2}) & \cdots & g(\vec{x}_{d,N_d},\vec{x}_{d,N_d}) \end{bmatrix}, \quad (5)$$

$$\boldsymbol{G}_{qd} = \begin{bmatrix} g(\vec{x}_{q,1},\vec{x}_{d,1}) & g(\vec{x}_{q,1},\vec{x}_{d,2}) & \cdots & g(\vec{x}_{q,1},\vec{x}_{d,N_d}) \\ g(\vec{x}_{q,2},\vec{x}_{d,1}) & g(\vec{x}_{q,2},\vec{x}_{d,2}) & \cdots & g(\vec{x}_{q,2},\vec{x}_{d,N_d}) \\ \vdots & \vdots & \ddots & \vdots \\ g(\vec{x}_{q,N_q},\vec{x}_{d,1}) & g(\vec{x}_{q,N_q},\vec{x}_{d,2}) & \cdots & g(\vec{x}_{q,N_q},\vec{x}_{d,N_d}) \end{bmatrix}, \quad (6)$$

where $g(\vec{x}_m,\vec{x}_n) = \|\vec{x}_m - \vec{x}_n\|_2^2(\ln\|\vec{x}_m - \vec{x}_n\|_2 - 1)$ represents Green's function for biharmonic spline in 2D. This interpolation process is executed once using the finite element data and is then frequently applied to the displacement output obtained from neural networks, a procedure that will be elaborated upon in subsequent sections. Consequently, the weight matrix $\boldsymbol{W}_{qd}$ can be precomputed, facilitating interpolation through a straightforward multiplication of this weight matrix with the displacement data at the data points. This approach not only ensures efficient interpolation but also maintains the full differentiability of the interpolation process. As a result, the procedure can be seamlessly integrated into the backpropagation algorithm during the training of the neural network, enhancing the efficiency of the neural network's learning process.

$$\boldsymbol{W}_{qd} = \boldsymbol{G}_{qd}\boldsymbol{G}_{dd}^{-1}, \quad \vec{u}_q = \boldsymbol{W}_{qd}\vec{u}_d = \boldsymbol{G}_{qd}\boldsymbol{G}_{dd}^{-1}\vec{u}_d, \quad (7)$$

Using interpolated displacements, longitudinal strain $\bar{\epsilon}$, transverse strain $\hat{\epsilon}$, and transverse sensitivity corrected strain ϵ are calculated as follows [11].

$$\bar{\epsilon} = \frac{u_{outer} - u_{inner}}{L}, \quad \hat{\epsilon} = \frac{v_{upper} - v_{lower}}{W}, \quad \epsilon = \frac{\bar{\epsilon} + \hat{\epsilon}K_{tp}}{1 - v_0 K_{tp}}, \quad (8)$$

where u_{outer} and u_{inner} represent averaged x displacement interpolated at outer edge $\{(x,y)|x = 3.3, -0.385 \le y \le 0.385\}$ and inner edge $\{(x,y)|x = 1.8, -0.385 \le y \le 0.385\}$, v_{upper} and v_{inner} represent averaged y displacement interpolated at upper edge $\{(x,y)|1.8 \le x \le 3.3, y = 0.385\}$ and lower edge $\{(x,y)|1.8 \le x \le 3.3, y = -0.385\}$, $L = 1.5$, and $W = 0.77$. K_{tp} denotes the transverse sensitivity coefficient of the strain gauge, and v_0 is the Poisson's ratio for the material used in measuring the gauge factor. Strain gauges in other directions, G_{90° and G_{45° can be regarded as rotation of G_{0°. Given nodal displacement $\hat{\boldsymbol{u}}$, we denote this strain calculation procedure compactly as

$$\vec{\epsilon} = [\epsilon_{0^\circ}, \epsilon_{90^\circ}, \epsilon_{45^\circ}] = \mathcal{G}(\hat{\boldsymbol{u}}) \quad (9)$$

All strain components and calibration coefficients in Eq. (4) in Box I are computed by $\mathcal{G}$.

After the calibration matrix is constructed, residual stress can be inversely calculated from strain measurements as follows.

$$\vec{\sigma} = \boldsymbol{C}^{-1}\vec{\epsilon} \quad (10)$$

It is important to recognize that this inversion process is also typically ill-conditioned, and the condition number of $\boldsymbol{C}$ tends to increase with the addition of depth increment levels or as the hole depth itself increases. This phenomenon occurs because residual stress is released at the location of the hole, whereas strain is measured on the specimen's surface. Consequently, according to Saint Venant's principle, the strain response rapidly decreases as the hole depth increases. Thus, a slight error or noise in the right-hand side of Eq. (10) can lead to significant variations in the calculated residual stress. This variation, or stress uncertainty, is briefly illustrated by the error bars in Fig. 1(b). Proper quantification of all sources of stress uncertainty is crucial. In real experiments, factors such as material properties, noise in strain measurements, and finite element calculations are considered sources of stress uncertainty. This uncertainty is quantified using Monte Carlo simulations, assuming an error distribution for each uncertainty source [12]. In this study, we also account for neural network model uncertainty, quantified through Monte Carlo dropout, as a source of stress uncertainty [47,48]. This aspect will be further explained in the section dedicated to model architecture.

2.2. Finite element model and data generation

In this section, we detail the finite element model and dataset construction. The overall finite element model geometry, boundary conditions, loading conditions, and strain gauge placement are illustrated in Fig. 3. To generate a single finite element model, inputs such as material properties, geometry, and loading conditions are required. These inputs are identified by four indices (l,i,j,s), each corresponding to laminate configuration $\vec{\theta}_l$, hole depth h_i, and load case $\vec{\sigma}_{js}$. The base geometry for the finite element model is a circular plate with a radius of 18 mm and a thickness of 1.6 mm, defined within the domain $\mathcal{B}_0 = \{(x,y,z)|x^2 + y^2 \le 18^2, 0 \le z \le 1.6\}$. To prevent rigid body motion, displacement boundary conditions are applied as $u(0,0,0) = v(0,0,0) = w(0,0,0) = 0$, $v(-18,0,0) = w(-18,0,0) = 0$, and $w(0,18,0) = 0$.

The material property for each layer in the composite laminate, defined by ply orientations $\vec{\theta}_l = [\theta_{l1},\theta_{l2},\theta_{l3},\theta_{l4}]$ $(l = 1,2,\dots,70)$, determines the constitutive matrix of orthotropic material assigned to each of the eight layers of the fiber-reinforced plate. The constitutive

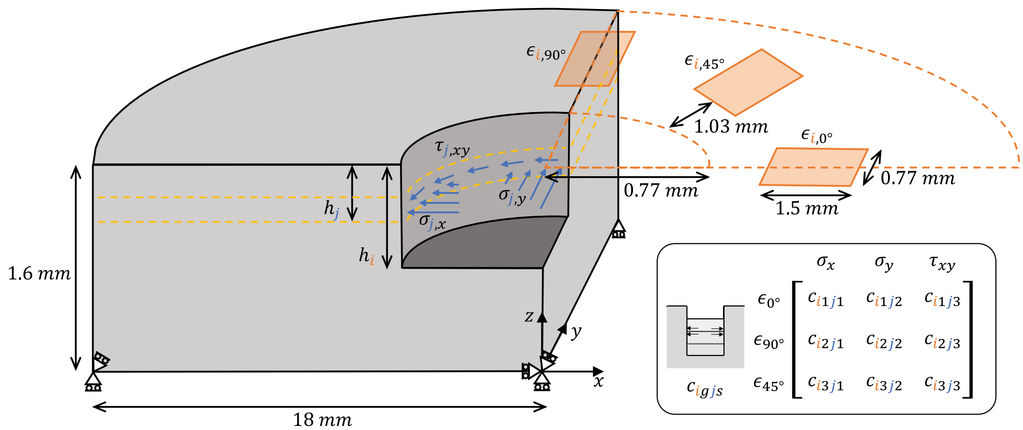


Fig. 3. Overview of the finite element model. Geometry, boundary conditions, loading conditions, and strain gauge placement of the finite element model with a quarter part depicted for visualization purpose.

matrix $\boldsymbol{E}_i$ for layer i is assigned within $1.6-0.2i \leq z \leq 1.6-0.2(i-1)$ for $i = 1, 2, \ldots, 8$, and material properties are symmetrically assigned about the $z = 0.8$ plane as

$$\boldsymbol{E}_i = \boldsymbol{E}_{9-i} = \boldsymbol{T}_{\theta_i} \boldsymbol{E} \boldsymbol{T}_{\theta_i}^T, \tag{11}$$

where $\boldsymbol{E}$ is the constitutive matrix for orthotropic material, whose properties are listed in Table 1, and $\boldsymbol{T}_{\theta_i}$ is the 6×6 stress transformation matrix for rotation θ_i in the xy plane.

The hole depth increments are represented by $h_i = 0.1i$ ($i = 1, 2, \ldots, D$) with $D = 8$ different depth increment levels, corresponding to the incremental removal of material from domain $\mathcal{B}_0$ to simulate drilling a hole up to 0.8 mm in depth. Each depth increment modifies the domain to $\mathcal{B}_i = \mathcal{B}_0 \setminus \{(x, y, z) | x^2 + y^2 \leq a^2, 1.6 - h_i \leq z \leq 1.6\}$, where $a = 0.77$ indicates the radius of the drilled hole.

The load cases $\vec{\sigma}_{js} \in \mathbb{R}^{3D}$ ($j = 1, 2, \ldots, D$, s=1, 2, 3) correspond to unit stress components s applied at depth level j, as shown in Fig. 3. Each load case $\vec{\sigma}_{js}$ is a vector matching the shape of $\vec{\sigma}$ in Eqs. (3) and (4), with a single component set to one and all others equal to zero. The index j denotes the depth at which the load is applied, while s specifies the stress component: $s = 1, 2, 3$ corresponds to $\sigma_{j,x}$, $\sigma_{j,y}$, and $\tau_{j,xy}$, respectively. For example, $\vec{\sigma}_{2,3}$ represents the vector defined in Eq. (4) with $\tau_{2,xy} = 1$ and all other components set to zero. The corresponding finite element model applies a unit stress $\tau_{2,xy} = 1$ at depth h_2. The unit stress σ_{js} is imposed as traction forces over the surface $\mathcal{S}_j = (x, y, z) \mid x^2 + y^2 = r^2, 1.6 - h_j \leq z \leq 1.6 - h_{j-1}$ for each stress component s, using the following stress transformation:

$$\sigma_{j,rr} = \sigma_{j,x} \cos^2\theta + \sigma_{j,y} \sin^2\theta + \tau_{j,xy} \sin 2\theta, \tag{12}$$

$$\tau_{j,r\theta} = -\frac{\sigma_{j,x} - \sigma_{j,y}}{2} \sin 2\theta + \tau_{j,xy} \cos 2\theta, \tag{13}$$

where $\sigma_{j,rr}$ and $\tau_{j,r\theta}$ denote the normal and shear traction forces applied at the surface $\mathcal{S}_j$ at angle θ from the x-axis in the xy plane.

Simulations were performed using the commercial finite element software Abaqus, employing 20-node quadratic hexahedral elements with reduced integration [49]. Validation of the finite element model was conducted using reference data from [12], which were developed using Nastran, validated through mesh independence studies, and applied in experimental investigations of cross-ply laminates. Strain responses at the strain gauge locations shown in Fig. 3 were compared with the reference data for a laminate configuration of $[0°, 0°, 90°, 90°]$. Validation was carried out for two drilled hole depths: half the maximum hole depth and the maximum hole depth. To ensure numerical accuracy, a mesh convergence study was performed, as presented in Fig. 4.

Table 1
Material properties of orthotropic glass fiber reinforced plastic lamina [12].

Longitudinal modulus, E_1	38154 MPa
Transverse modulus, E_2	10747 MPa
Shear modulus, G_{12}	3712 MPa
Poisson's ratio, ν_{12}	0.3113
Poisson's ratio, ν_{23}	0.4124

This study demonstrated a consistent reduction in relative strain error as the number of elements in the z-direction increased from 4 to 8, 16, and 32. The corresponding degrees of freedom at the maximum hole depth were 262,131 for the 16 elements case, and 2,005,731 for the 32 elements case. On our computational setup using 4 parallel CPU cores from an Intel i7-11700 and an NVIDIA RTX 2070 GPU, the time required to construct the calibration matrix in Eq. (4) was 12.28 min for 16 elements and 294.15 min for 32 elements in the z-direction. Based on the mesh convergence results (Fig. 4) and considering the balance between numerical accuracy and computational efficiency for data generation, the finite element model with 16 elements in the z-direction was selected for the following data generation.

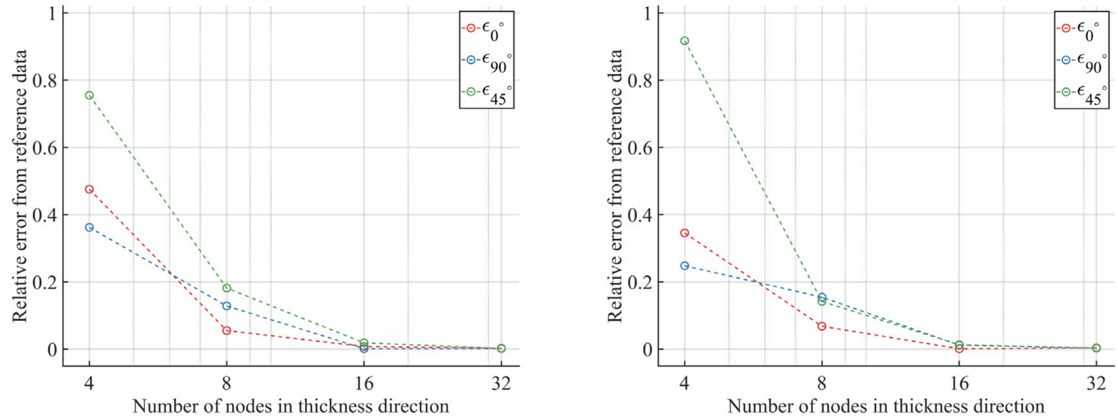
A single datapoint is derived from the finite element model (l, i, j, s), comprising input data and label data as summarized in Fig. 5a. The input data includes the graph G_{li}, based on a finite element model for a specific hole depth h_i and laminate configuration $\vec{\theta}_l$, and the load case $\vec{\sigma}_{js}$. The graph G_{li} consists of vertices V, representing the finite element nodes within the defined region of interest ($0.77^2 \leq x^2 + y^2 \leq 4^2, z = 1.6$), and edges E formed using a k-nearest neighbor approach where each vertex is connected to its eight closest neighbors based on spatial coordinate of the finite element node (Fig. 5b). Each vertex is associated with embedding vector $\vec{x}$, derived from the spatial coordinates of the finite element node and its relative angle to the ply orientation. The components of $\vec{x} \in \mathbb{R}^{d_{in}}$ with input dimension $d_{in} = 26$, denoted as $\vec{x}^{(\cdot)}$, are defined as follows:

$$\text{Normalized radius: } \vec{x}^{(1)} = \left[\frac{1}{1 + r/a}\right], \tag{14}$$

$$\text{Trigonometric series basis: } \vec{x}^{(2:9)} = [\cos k\theta, \sin k\theta] \quad (k = 1, \ldots, 4), \tag{15}$$

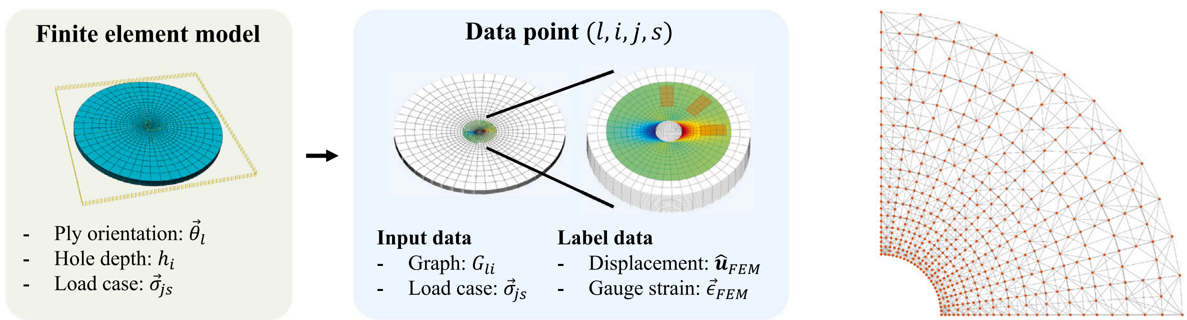
$$\text{Relative angle from ply orientation: } \vec{x}^{(10:25)} = \left[\cos^2 k(\theta - \theta_{li})\right] \quad (k = 1, \ldots, 4 \;\; i = 1, \ldots, 4), \tag{16}$$

$$\text{Normalized hole depth: } \vec{x}^{(26)} = \left[h/h_{max}\right], \tag{17}$$



(a) (b)

Fig. 4. Mesh convergence study. Relative strain response error at drilled hole depths of (a) half the maximum hole depth, and (b) the maximum hole depth, validated against reference data from [12].



(a) (b)

Fig. 5. Workflow of data generation. (a) data generation by finite element model and post-processing, (b) the first quadrant of the k-nearest neighbor graph constructed based on the finite element node coordinates.

where $r = \sqrt{x^2 + y^2}$, θ represents the angle from the x-axis, $a = 0.77$ is the radius of the drilled hole, and $h_{max} = 0.8$ indicates the maximum depth of the hole.

Label data is composed of nodal displacement $\hat{\boldsymbol{u}}_{FEM} \in \mathbb{R}^{N_n \times 3}$, sampled for N_n finite element nodes within the region of interest, and strain $\vec{\epsilon}_{FEM} = [\epsilon_{0°}, \epsilon_{90°}, \epsilon_{45°}] = \mathcal{G}(\hat{\boldsymbol{u}})$ calculated based on Eq. (9). Both displacement and strain are normalized according to the method proposed in [50], with the characteristic length $l_c = 0.77$ chosen as the radius of the hole, and displacement normalization factor $u_c = 10^{-6}$ selected based on the approximate ratio between the stress level and Young's modulus.

2.3. DeepONet-based model architecture and training

In this research, we introduce a model architecture that primarily utilizes the deep operator network (DeepONet) framework. Unlike traditional neural networks that approximate a function directly from the input to the label dataset, DeepONet is designed to learn an operator mapping from an input function space to an output function space [25]. The DeepONet architecture comprises two distinct neural networks: the branch net and the trunk net. The branch net processes the discrete representation of the input function, which, in our context, is the residual stress distribution along the hole depth, denoted as $\vec{\sigma}$. On the other hand, the trunk net manages the domain information of the output function, represented by the graph G derived from the finite element mesh in our study. Given that our training data involves the displacement field resulting from linear finite element model outputs, adopting a linear branch network and a graph neural network-based trunk network is thought to be a natural approach.

Specifically, the branch network is implemented as a single fully connected layer $\boldsymbol{W}$, without a nonlinear activation function. This maps the stress distribution $\vec{\sigma} \in \mathbb{R}^{3D}$ to branch network output $\vec{b} \in \mathbb{R}^{9D}$ as

$$\vec{b}(\vec{\sigma}) = \vec{\sigma} \cdot \boldsymbol{W}. \tag{18}$$

The trunk network employs a graph convolution network (GCN) [51], with the GCN block as proposed in [52]. Given the input feature of vertex i, $\vec{x}_i \in \mathbb{R}^{d_{in}}$ within a graph with edges E, the graph convolution operation maps $\vec{x}_i$ to an output vertice feature $\vec{x}'_i \in \mathbb{R}^{d_{out}}$ by aggregating vertice features of neighboring vertices $\vec{x}_j \in \mathbb{R}^{d_{in}}$ and itself. This operation is parameterized by a learnable weight matrix $\boldsymbol{\Theta} \in \mathbb{R}^{d_{in} \times d_{out}}$ and is defined as

$$\vec{x}'_i = \left(\sum_{j \in \Gamma(i) \cup \{i\}} \frac{e_{j,i}}{\sqrt{\hat{d}_j \hat{d}_i}} \vec{x}_j \right) \cdot \boldsymbol{\Theta}, \tag{19}$$

where $\Gamma(i)$ denotes the set of neighbors of vertex i, $\hat{d}_i = 1 + \sum_{j \in \Gamma(i)} e_{j,i}$ is the degree of vertex i, and $e_{j,i} = 1$ if there exists an edge in edges E connecting node j to node i and $e_{j,i} = 0$ otherwise.

The trunk network, constructed by stacking multiple GCN blocks, maps the input vertex feature $\vec{x} \in \mathbb{R}^{d_{in}}$ in a graph G to a sequence of hidden vertex features $\vec{x}_h(\vec{x}; G) \in \mathbb{R}^{d_h}$, before projecting it to an output vector of the trunk network in $\mathbb{R}^{9D(2K+1)}$. The output vector is consequently divided into $2K + 1$ vectors of equal length $\vec{t}_i \in \mathbb{R}^{9D}$

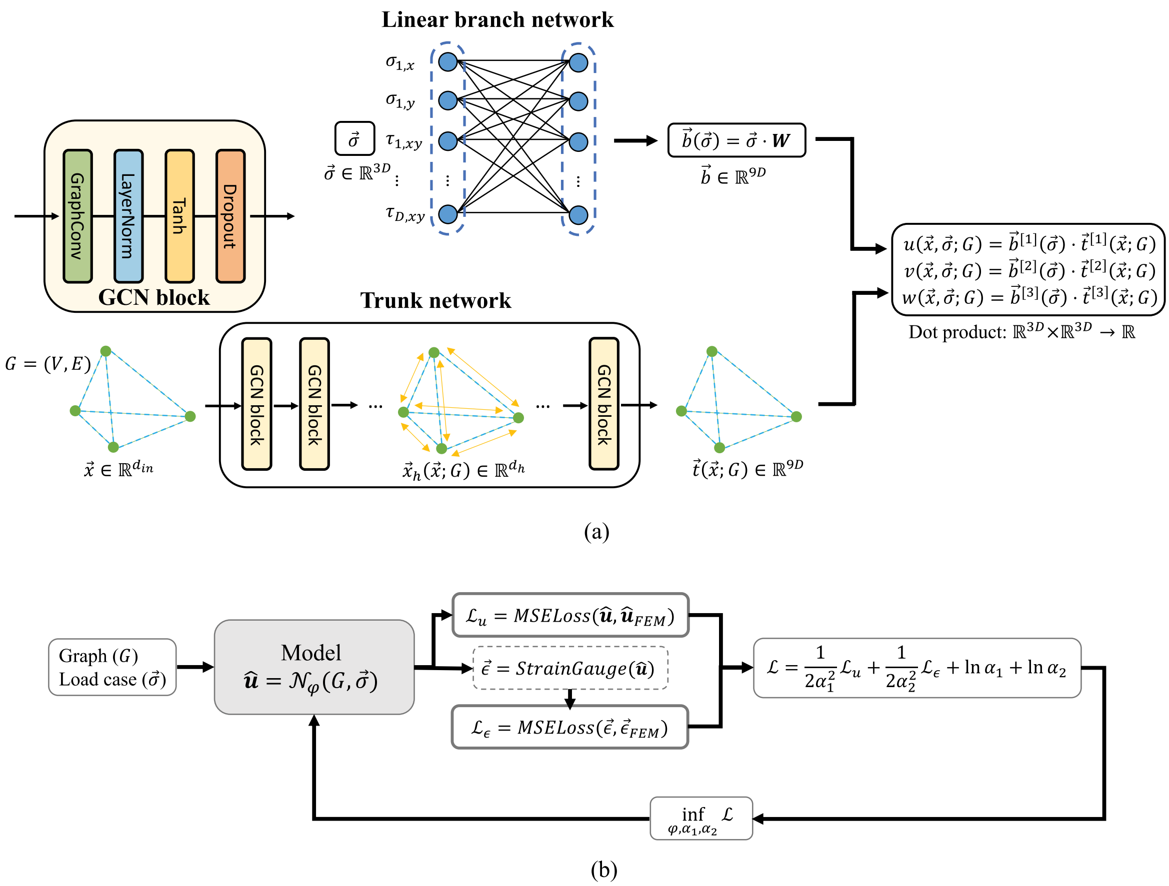


Fig. 6. Schematic of the model architecture and training procedure. (a) the proposed DeepONet architecture with linear branch network and trunk network based on GCN blocks, (b) training step of model prediction, loss calculation, and backpropagation.

to compute the final prediction of trunk network using trigonometric series expansion of order K as follows

$$\vec{t}(\vec{x}; G) = \vec{t}_0 + \sum_{k=1}^{K} \vec{t}_{2k-1} \cos k\theta + \vec{t}_{2k} \sin k\theta. \tag{20}$$

The incorporation of trigonometric series expansion is inspired by the Fourier finite element method [53], with the trigonometric series order $K = 4$ selected based on the typical angular displacement profiles reported in [54], yet without presupposing a specific functional form since the coefficients of the trigonometric series $\vec{t}_i$ remain dependent on θ. The efficacy of this method will be evaluated in an ablation study presented in the results section.

Monte Carlo dropout is applied within the trunk network to account for model uncertainty when predicting inputs beyond the training data [47]. The mean and standard deviation of 10 model predictions under different dropout configurations are taken as the model's prediction and uncertainty, respectively. Displacements at each vertex (u, v, w) constitute the model's final predictions and are calculated by the dot product of the trunk network's output $\vec{t}(\vec{x}; G)$ with the output of the branch network $\vec{b}(\vec{\sigma})$. Specifically, the trunk network's output is divided into three equal length vectors $\vec{t}^{[1]}, \vec{t}^{[2]}, \vec{t}^{[3]} \in \mathbb{R}^{3D}$, and the branch network's output is similarly divided into $\vec{b}^{[1]}, \vec{b}^{[2]}, \vec{b}^{[3]} \in \mathbb{R}^{3D}$.

$$u(\vec{x}, \vec{\sigma}; G) = \vec{b}^{[1]}(\vec{\sigma}) \cdot \vec{t}^{[1]}(\vec{x}; G), \quad v(\vec{x}, \vec{\sigma}; G) = \vec{b}^{[2]}(\vec{\sigma}) \cdot \vec{t}^{[2]}(\vec{x}; G),$$
$$w(\vec{x}, \vec{\sigma}; G) = \vec{b}^{[3]}(\vec{\sigma}) \cdot \vec{t}^{[3]}(\vec{x}; G) \tag{21}$$

The prediction on every N_n vertices of graph G will be denoted as $\hat{\boldsymbol{u}} = \mathcal{N}_\varphi(G, \vec{\sigma}) \in \mathbb{R}^{N_n \times 3}$, where $\mathcal{N}_\varphi$ represents the model with trainable parameters $\vec{\varphi}$. To compute N_n nodal displacement predictions at vertices from the single graph G, branch network output $\vec{b}(\vec{\sigma})$ is simply repeated N_n times for computing dot product in Eq. (21).

This DeepONet with linear branch network has direct implications for integral relation in Eqs. (1) and (2), where dot product of branch network weight and trunk network prediction $\boldsymbol{W} \cdot \vec{t}(\vec{x}; G)$ corresponds to kernel function $K(H, h)$ or discrete representation of kernel function c_{igjs} and integral operation. The final displacement prediction, $\vec{\sigma} \cdot \boldsymbol{W} \cdot \vec{t}(\vec{x}; G)$, highlights the linear dependency on the stress distribution $\vec{\sigma}$, while the highly nonlinear dependencies on the laminate configuration and hole depth, embedded in the graph G, are effectively captured by the GCN blocks within the trunk network. The effectiveness of this design can be further supported by a similar approach in [55], where a linear branch network demonstrated strong generalization for components with linear dependencies. The overall architecture of the proposed DeepONet is illustrated in Fig. 6a.

The displacement loss for prediction of batched B graphs each containing N_n vertices, $\mathcal{L}_u$, is defined as follows

$$\mathcal{L}_u = \frac{1}{3N_n B} \|\hat{\boldsymbol{u}} - \hat{\boldsymbol{u}}_{FEM}\|_2^2, \tag{22}$$

where $\hat{\boldsymbol{u}}$ represents the batched predictions of displacement for every vertex, and $\|\cdot\|_2$ denotes the L^2 norm.

The strain loss, $\mathcal{L}_\epsilon$, is defined as

$$\mathcal{L}_\epsilon = \frac{1}{3B} \|\vec{\epsilon} - \vec{\epsilon}_{FEM}\|_2^2, \tag{23}$$

where $\vec{\epsilon}$ and $\vec{\epsilon}_{FEM}$ denote the strains for batched displacement predictions $\hat{\boldsymbol{u}}$ and the label data $\hat{\boldsymbol{u}}_{FEM}$, calculated in accordance with Eq (9).

To integrate $\mathcal{L}_u$ and $\mathcal{L}_\epsilon$, we employ an adaptive weight balancing method proposed in [56]

$$\mathcal{L} = \frac{1}{2\alpha_1^2}\mathcal{L}_u + \frac{1}{2\alpha_2^2}\mathcal{L}_\epsilon + \ln\alpha_1 + \ln\alpha_2, \tag{24}$$

Table 2
k-fold cross-validation results.

	Common	Activation	Batch size	Avg.	Std.
model 1	- 6 GCN blocks - 512 hidden dimension - 0.01 weight decay	ReLU	16	0.0616	0.0137
model 2			32	0.0764	0.0208
model 3		Tanh	8	**0.0464**	0.0107
model 4			16	0.0478	**0.0062**

where α_1 and α_2 are initially set to 1 and are updated along with the model parameters $\vec{\varphi}$ to address the following optimization problem using standard backpropagation as depicted in Fig. 6(b).

$$\inf_{\vec{\varphi},\alpha_1,\alpha_2} \mathcal{L} \tag{25}$$

We utilize the AdamW optimization algorithm to apply weight decay effectively, benefiting from the stable optimization afforded by the adaptive learning rate of Adam combined with the learning rate scheduler $lr = 0.001 \times 0.99^{epoch}$ [57,58]. The utility of adaptive weight balancing in Eq. (24) for engineering applications of neural networks, as reported in [50], will be further explored in an ablation study specific to our application.

3. Results

This section details the training and prediction outcomes of our model. The dataset encompasses inputs and labels for 70 laminate configurations, with eight depth increments, i, up to the midplane and $3i$ load cases per depth increment i, resulting in a total of $70 \times (3 + 6 + \cdots + 24) = 7560$ data points. The dataset is divided into 30 training configurations (3240 data points) and 40 testing configurations (4320 data points), ensuring a balanced distribution of ply orientations across each layer for the training configurations. For the 40 testing laminate configurations, $\vec{\theta}_l$, pairs of virtual residual stress distribution $\vec{\sigma}_l$ and strain measurement $\vec{\epsilon}_l$ are generated. Initially, this section discusses the selection of hyperparameters and the outcomes of training with the 30 configurations designated for training. Subsequently, it illustrates the forward prediction and inverse calculation results for the virtual residual stress distribution of the 40 test configurations. Lastly, an ablation study is presented to assess the contribution of each component to the overall performance of the model.

3.1. Hyperparameter selection and training

Training a neural network model typically involves a nested loop process, where model parameters are optimized in the inner loop, and hyperparameters are optimized in the outer loop. While the inner loop benefits from well-established, theoretically grounded gradient-based optimization algorithms, optimizing hyperparameters in the outer loop is often viewed as more challenging. This challenge arises from the need for a robust estimate of model performance, which is computationally demanding [59]. To address this, we implemented k-fold cross-validation for four model candidates, chosen through a grid search of hyperparameters.

In the grid search, hyperparameters were divided into two categories. The first category focused on model size, including the number of GCN blocks (6,8), hidden dimensions of GCN outputs (d_h = 128, 256, 512), and weight decay ($\lambda = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}$). The second category addressed model architecture and optimization, encompassing the activation function (ReLU, Tanh), dropout probability (p = 0.005, 0.01, 0.05, 0.1), and batch size (B = 4, 8, 16, 32). Based on the training loss $\mathcal{L}_\epsilon$ over 30 training configurations after 1000 epochs, two candidates each using the ReLU and Tanh activation functions were selected as summarized in Table 2 considering a superiority of tanh activation in PDE-related tasks reported empirically and mathematically [60,61].

For the k-fold cross-validation, the 30 training configurations were randomly divided into 10 folds. Training results are subsequently obtained for 10 different dataset compositions using one fold for validation and the other nine folds for training. The stress L^2 relative error was calculated for the validation fold at every epoch of training as follows

$$e(\vec{\sigma}) = \frac{\|\vec{\sigma} - \vec{\sigma}_l\|_2}{\|\vec{\sigma}_l\|_2} \tag{26}$$

Here, $\vec{\sigma}$ denotes the stress using the calibration matrix composed by model predictions and virtual strain measurement $\vec{\epsilon}_l$. The average of the minimum validation errors across the 10 folds was used to estimate the model's generalization performance, with the standard deviation providing a measure of this estimate's reliability [62]. The training procedure of k-fold cross-validation is shown in Fig. 7a with a shaded area representing twice the standard deviation of 10 folds in each epoch. The results of the k-fold cross-validation are summarized in Fig. 7b and Table 2. Based on both the average and standard deviation of errors across the 10 folds, Model 4 was selected. Forward and inverse predictions in the subsequent section were performed using Model 4, trained with the full set of 30 training configurations.

3.2. Forward prediction and inverse calculation

In this section, we showcase the visualization of forward and inverse prediction results. Fig. 8 presents a histogram of the strain and stress relative L^2 error, as defined in Eq. (26), for 39 test laminate configurations. We exclude configurations where all ply orientations are aligned in a single direction, resulting in no residual stress. Three configurations representing the best, median, and worst cases of strain and stress relative L^2 errors are selected to illustrate the results of forward and inverse predictions.

For forward predictions, the model processes virtual residual stress distributions, $\vec{\sigma}_l$, corresponding to particular laminate configurations, $\vec{\theta}_l$. It predicts the displacement field and strain measurements related to the depth increments of the drilled hole. The displacement field and strain for a given depth increment level i are calculated as follows

$$\hat{\boldsymbol{u}}_{li} = \mathcal{N}_\varphi(G_{li}, \vec{\sigma}_{li}), \quad \vec{\epsilon}_{li} = \mathcal{G}(\hat{\boldsymbol{u}}_{li}) \tag{27}$$

where each element of $\vec{\sigma}_{li}$ equals $\vec{\sigma}_l$ up to the $3i^{th}$ component, with the remaining components set to zero since only the residual stress present in the material at depths shallower than the current depth of the hole influences the current strain measurement [12]. Fig. 9 displays the outcomes for three configurations chosen to span the range of strain prediction errors, averaging the strain and displacement predictions over 10 Monte Carlo trials as detailed in the methodology, with uncertainty represented by a shaded area indicating twice the standard deviation of these trials.

For inverse predictions, the model constructs the calibration matrix in Eq. (4) for specific laminate configurations. This involves multiple forward predictions for unit stress input at each depth region for each depth increment level. Calibration coefficients $c^{(l)}_{igjs}$ for a laminate configuration $\vec{\theta}_l$ are derived as follows

$$[c^{(l)}_{i1js}, c^{(l)}_{i2js}, c^{(l)}_{i3js}] = \mathcal{G}(\mathcal{N}_{\vec{\varphi}}(\vec{\sigma}_{js}, G_{li})), \tag{28}$$

for $i = 1, 2, \ldots, 8$, $j = 1, 2, \ldots, i$, and $s = 1, 2, 3$. Assuming strain measurements are provided, the residual stress distribution is calculated using Eq. (10). Fig. 10 illustrates the prediction outcomes, representing averages across 10 Monte Carlo trials, with error bars reflecting the uncertainty, quantified as twice the standard deviation of these trials. The histogram on the right of Fig. 10 displays the full statistics from the 10 Monte Carlo trials. The results reveal that the majority of dominant, non-zero components of residual stress are encompassed within the error bars. Additionally, an observable trend is that the magnitude of the error bars increases with the depth of the hole. This phenomenon aligns with the expected behavior according to Saint Venant's principle,

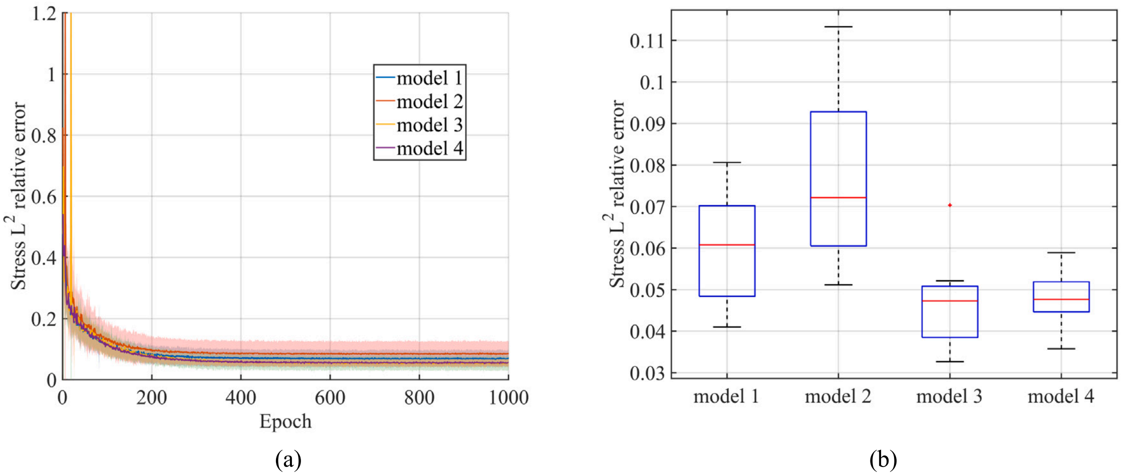


Fig. 7. k-fold cross-validation results for selected models. (a) Mean validation error curve with the shaded area representing $2 \times std$ for 10 folds, (b) box plot for minimum validation error distribution for 10 folds.

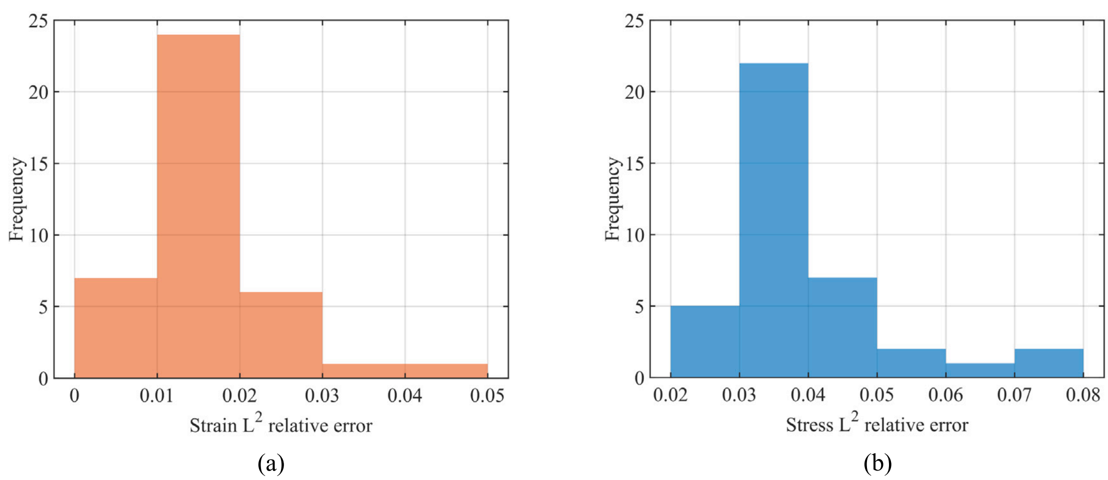


Fig. 8. Error histogram. (a) Strain L^2 relative error, (b) stress L^2 relative error.

illustrating a consistent feature of the IHD method related to the propagation of error with increasing depth.

Quantified model uncertainty provides further insight into prediction errors, as illustrated in Fig. 11. We analyzed 1680 stress predictions (70 laminate configurations × 8 hole depths × 3 components) by sorting them based on quantified model uncertainty, dividing them into eight equal groups, and computing the mean uncertainty and error for each group. Using Monte Carlo dropout, which approximates a Gaussian process, we found that higher uncertainty generally corresponds to inputs farther from the training data [47,48], where prediction errors tend to increase. The trends observed in Fig. 11 confirm a clear correlation between higher uncertainty and larger prediction errors, demonstrating how uncertainty quantification can effectively explain model behavior.

Due to the ill-conditioned nature of the equation referenced as Eq. (10), the selections for forward and inverse predictions differ. This condition also leads to a higher relative error in stress prediction compared to strain prediction, a consequence of the equation's ill-conditioned inversion. In practical settings of incremental hole-drilling, where a complete procedure for stress uncertainty quantification is applied, strain measurement noise is typically assumed to range between $1\% \sim 2\%$, such as the instrumentation error of 1.6% reported in [12]. Furthermore, the calibration coefficient's compact representation for metallic materials, which is the basis of the current American Society for Testing and Materials (ASTM) E837 standard, achieves an average accuracy within $1\% \sim 2\%$, with occasional deviations reaching about 3% [63]. An IHD experiment on a laminate configuration of $[0^\circ, 0^\circ, 90^\circ, 90^\circ]$ reported a stress uncertainty of $5.2\% \sim 8.5\%$ arising from strain noise, and when combined with other sources of uncertainty, the total quantified uncertainty amounted to up to 13.5% [12]. This level of experimental uncertainty can even increase for more complex laminate configurations, as observed in [11]. Considering these aspects, the average strain and stress errors in our predictions can be regarded as practically acceptable, taking into account the major sources of uncertainty.

3.3. Ablation study

This section details the results of an ablation study conducted to evaluate the contribution of each component within the proposed

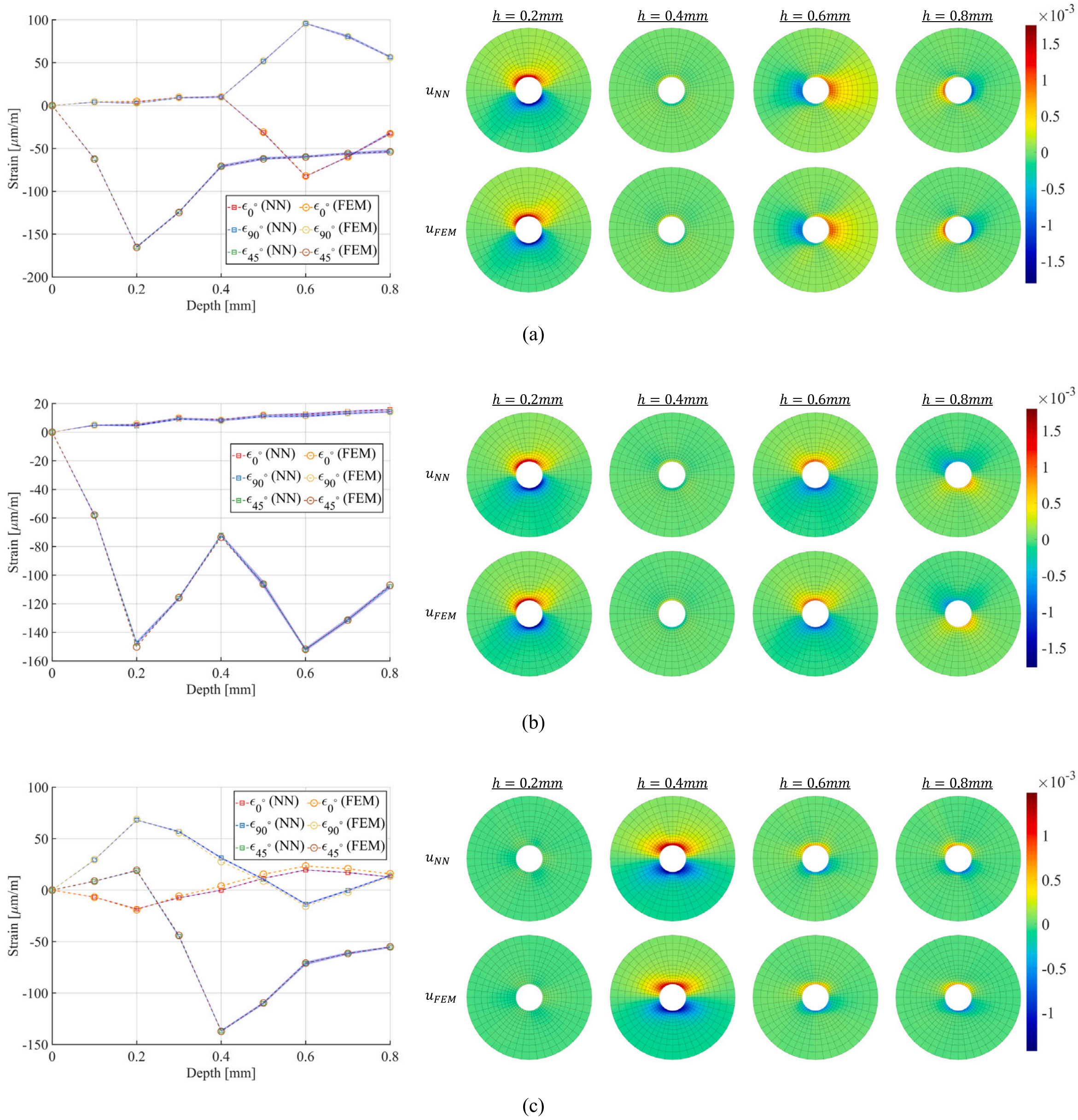


Fig. 9. Forward strain and displacement prediction results. (a) best case $\vec{\theta}_l = [45°, -45°, 0°, 90°]$, (b) median case $\vec{\theta}_l = [45°, -45°, 45°, -45°]$, (c) worst case $\vec{\theta}_l = [0°, 45°, -45°, 0°]$.

model. The study examined five different configurations by systematically removing specific components of the proposed model architecture or loss term. All other setups for training, including dataset generation, geometry, material properties, and loading conditions, remained identical to those described in Section 2. When fully connected layers are used in place of graph convolution, the operation in Eq. (19) simplifies to

$$\vec{x}_i' = \vec{x}_i \cdot \Theta + \vec{b}, \tag{29}$$

where $\Theta \in \mathbb{R}^{d_{in} \times d_{out}}$ and $\vec{b} \in \mathbb{R}^{d_{out}}$ are learnable parameters.

Below, we summarize the configurations analyzed in the study:

- Without GCN (w/o GCN): The GraphConv layer depicted in Fig. 6, whose operation is described in Eq. (19), is replaced by fully connected layers maintaining identical input, output, and hidden dimensions.
- Without trigonometric series expansion (w/o trig): The trigonometric series expansion in Eq. (20) is omitted, with the trunk network solely predicting $\vec{t} = \vec{t}_0 \in \mathbb{R}^{9D}$.
- Coordinate-based fully connected layer DeepONet (standard DeepONet): Both branch and trunk networks use fully connected layers with hidden dimensions matching the GCN blocks of our proposed model. The branch network processes load cases ($\vec{\sigma}_{js}$) and laminate configurations ($\vec{\theta}_l$), while the trunk network handles spatial coordinates and hole depth (x, y, h_i).
- Without displacement loss (w/o displ loss): The displacement loss term in Eq. (22), which aids in strain prediction, is removed, leading to $\mathcal{L} = \mathcal{L}_\epsilon$.
- Without strain loss (w/o strain loss): The strain loss term in Eq. (23) derived from displacement interpolation is omitted, leading to $\mathcal{L} = \mathcal{L}_u$.

Fig. 12 illustrates the results of the ablation study through a box plot for 39 test configurations. Overall, the complete model demonstrates superior performance over the variations, in terms of both average and worst-case errors. The significance of incorporating a GCN is highlighted in the 'w/o GCN' case, where predictions at a vertex are made without information about neighboring vertices. This limitation

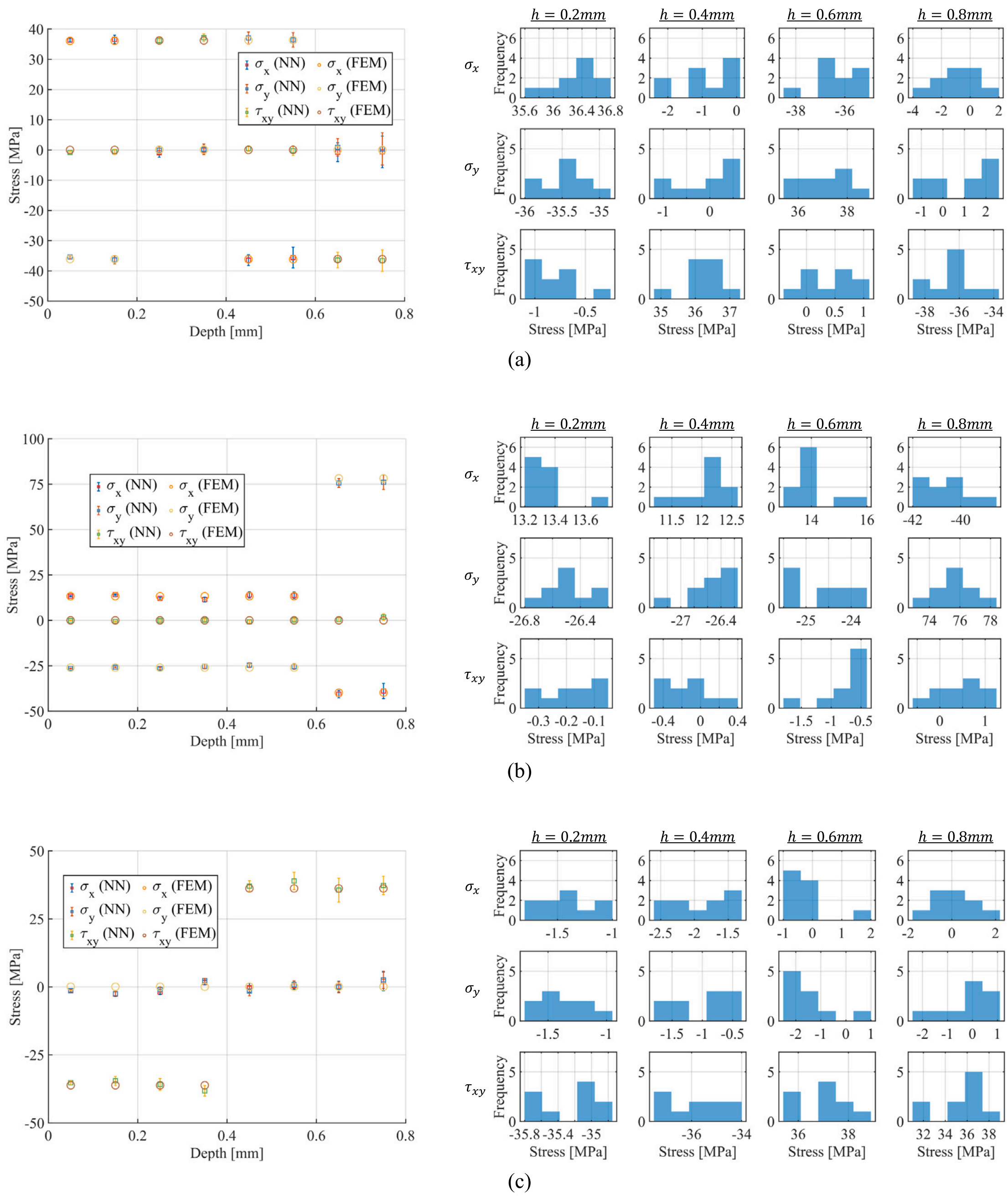


Fig. 10. Inverse stress calculation results and histogram for 10 Monte Carlo trials. (a) best case $\vec{\theta}_l = [0°, 45°, 90°, -45°]$, (b) median case $\vec{\theta}_l = [0°, 0°, 0°, 90°]$, (c) worst case $\vec{\theta}_l = [-45°, -45°, 45°, 45°]$.

restricts the contextual understanding of material properties, which are influenced by the relative angle from the ply orientation of each vertex. The importance of trigonometric series expansion is evident in the 'w/o trig' scenario, underscoring the known effectiveness of the Fourier finite element method, especially in handling non-axisymmetric loads on axisymmetric geometries. The 'standard DeepONet' employs fully connected layers in both branch and trunk networks, where the branch network processes load cases and ply orientation as PDE parameters, and the trunk network handles spatial coordinates and hole depth. In contrast, our model separates the linear contribution of load cases in the branch network and nonlinear ply orientation in the trunk network using a graph-based representation with relative ply orientations embedded as features of a vertex. This architecture achieves improved accuracy, showcasing the effectiveness of integrating a linear branch network with a GCN-based trunk network.

Furthermore, the ablation of either the displacement or strain loss terms results in reduced accuracy, indicating that while the full displacement field at the region of interest is not the direct prediction target, it plays a role in enhancing the model's generalization performance. Given the dataset's composition of discrete ply orientation combinations (−45°, 0°, 45°, and 90°), the full displacement field data furnish insights into the continuously varying relative ply orientations.

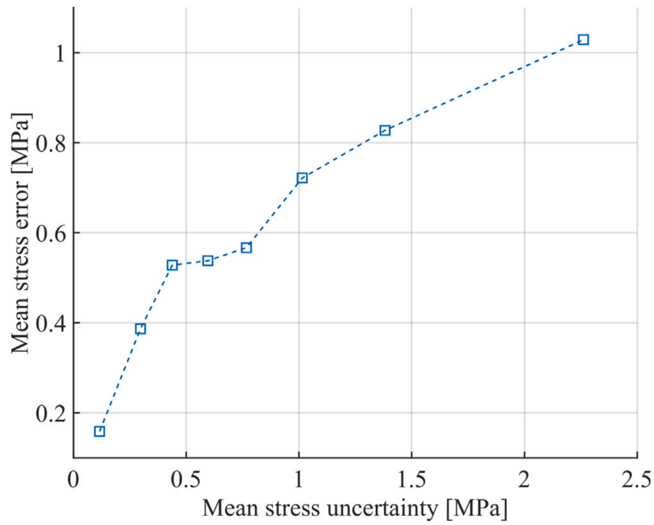


Fig. 11. Correlation between quantified uncertainty and prediction error.

This is facilitated by feature embedding, which incorporates the relative angle between the vertex and each ply orientation, thereby supporting the model's overall effectiveness.

4. Discussion

In this study, we introduced a deep operator network (DeepONet) architecture combining a linear branch network and a graph convolution trunk network to model the relationship between residual stress distribution and the resulting deformation. The model swiftly generates the calibration matrix for the inverse solution process, significantly reducing time compared to traditional methods that compute each coefficient individually using finite element simulations. Specifically, while training requires 12.23 min per laminate configuration on an NVIDIA RTX A6000 GPU and 12.28 min for FEM data generation as described in Section 2.2, the trained model predicts the calibration matrix for unseen laminate configurations in just 2.89 s, achieving two orders of magnitude speedup. We used a finite element model with 16 elements in the z-direction for efficiency in this study, but a 32-element model, as used in [12], requires 294.15 min for each laminate configuration using the traditional method, demonstrating an even greater potential for time reduction with higher-resolution models.

For metallic materials, IHD practitioners have access to the ASTM E837 standards with polynomial functions derived from FEM compliances to calculate calibration matrices, and IHD machines equipped with software that interprets stress results from strain measurements using standard processes. In contrast, such resources are not readily available for composite laminates, and it is unlikely they will be developed through conventional methods due to the vast and complex experimental parameter space introduced by the myriad possible combinations of ply orientations in composite materials. This gap highlights the potential of our study to facilitate establishing a large-scale calibration database for composite laminates using neural network methodologies, extending the standard of compact calibration data for metallic materials using a polynomial function [63].

To further illustrate the large-scale application potential highlighted earlier, we conducted additional experiments by varying the training dataset size and model complexity. As shown in Fig. 13, the results consistently showed a reduction in test error as both the data size and the number of model parameters increased, validating the scalability and robustness of the proposed DeepONet architecture based on linear branch network and graph convolution trunk network. The proposed architecture, grounded in the Volterra equation of the first kind which is a fundamental framework for relaxation-based residual stress measurement methods [46], can be extended through transfer learning, enabling pre-trained models to adapt efficiently to new material systems with additional data [50]. Leveraging uncertainty quantification results described in Section 3.2, the model can employ active learning to iteratively expand its training dataset, prioritizing the most informative samples to enhance performance in underrepresented configurations [64,65]. Furthermore, integrating advanced geometric encoding techniques can enable the model to generalize effectively across diverse shapes and systems [41,66].

Looking ahead, our future work will focus on expanding the model's capabilities and applying it to real experimental scenarios. Enhancing the framework to incorporate nonlinearities such as material nonlinearities [44], local plasticity [45], and contact phenomena including tool-part interactions [67] will significantly increase the model's applicability to real-world drilling processes and other manufacturing scenarios. Additionally, we aim to validate the framework with real experimental data, leveraging its computational efficiency and scalability to establish a robust calibration database for composite laminates. These efforts will pave the way for developing a neural network-based large-scale database for incremental hole drilling (IHD) in composite laminates.

CRediT authorship contribution statement

Seung-Woo Lee: Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Conceptualization. **Teubes Christiaan Smit:** Writing – review & editing, Visualization, Validation, Methodology, Investigation, Conceptualization. **Kyusoon Jung:** Writing – review & editing, Validation, Software, Methodology, Investigation, Conceptualization. **Robert Grant Reid:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Conceptualization. **Do-Nyun Kim:** Writing – review & editing, Visualization, Supervision, Methodology, Investigation, Conceptualization.

Code availability

All codes of this study are available from the corresponding author upon reasonable request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00346176). Financial support was also provided by the Youlchon Foundation (Nongshim Corporation and affiliated companies) in Korea.

Data availability

All data of this study are available from the corresponding author upon reasonable request.

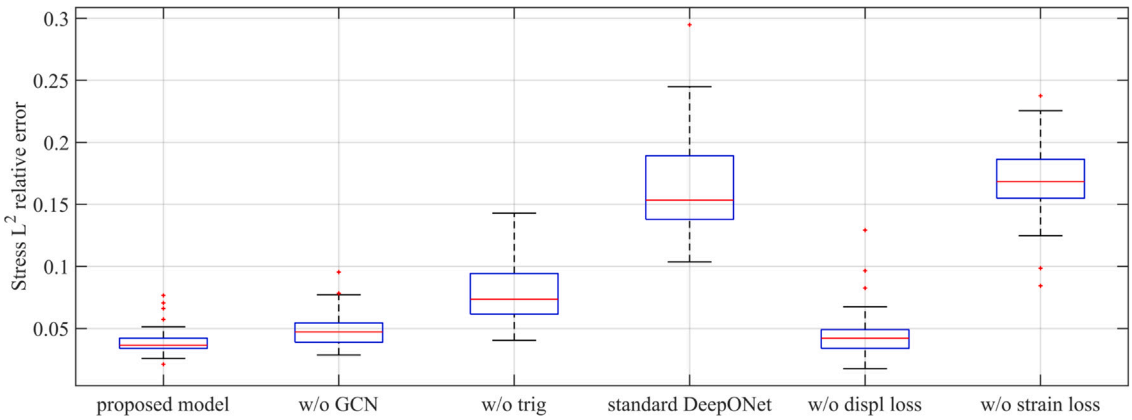


Fig. 12. Results of ablation study by stress L^2 relative error for test laminate configurations. .

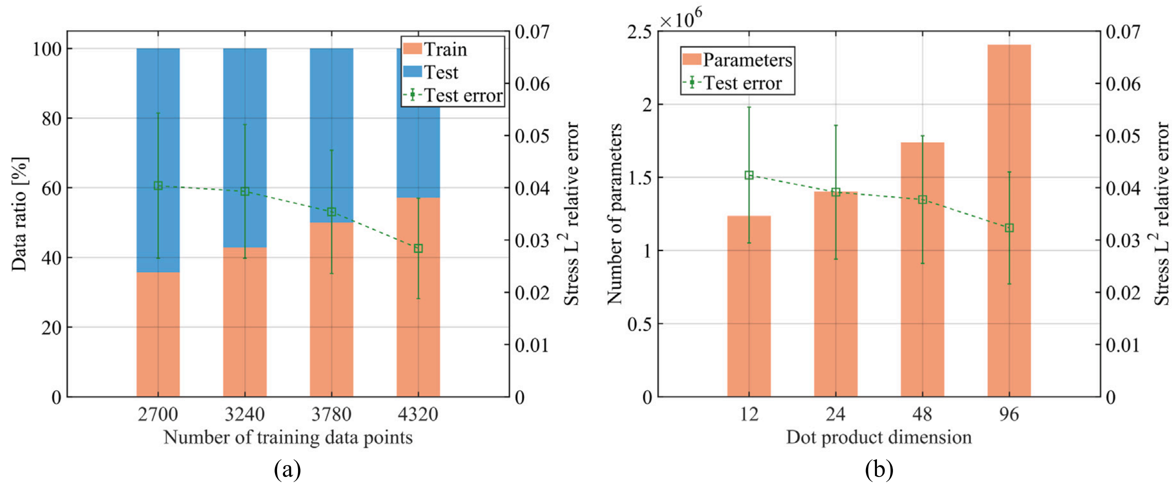


Fig. 13. Effect of training data size and model parameter size on test error. test error (mean and standard deviation across test laminate configurations) as a function of (a) training data points, and (b) model parameter size, adjusted by the dot product dimension in Eq. (21).

References

[1] Zhang Y, Wang WH, Greer AL. Making metallic glasses plastic by control of residual stress. Nat Mater 2006;5(11):857–60. http://dx.doi.org/10.1038/nmat1758.

[2] Schajer GS, Winiarski B, Withers PJ. Hole-drilling residual stress measurement with artifact correction using full-field DIC. Exp Mech 2013;53(2):255–65. http://dx.doi.org/10.1007/s11340-012-9626-0.

[3] Salvati E, Romano-Brandt L, Mughal M, Sebastiani M, Korsunsky A. Generalised residual stress depth profiling at the nanoscale using focused ion beam milling. J Mech Phys Solids 2019;125:488–501. http://dx.doi.org/10.1016/j.jmps.2019.01.007.

[4] Chen W, Voisin T, Zhang Y, Forien J-B, Spadaccini CM, McDowell DL, et al. Microscale residual stresses in additively manufactured stainless steel. Nat Commun 2019;10(1):4338. http://dx.doi.org/10.1038/s41467-019-12265-8.

[5] Hu D, Grilli N, Wang L, Yang M, Yan W. Microscale residual stresses in additively manufactured stainless steel: Computational simulation. J Mech Phys Solids 2022;161:104822. http://dx.doi.org/10.1016/j.jmps.2022.104822.

[6] Wang Y-D, Tian H, Stoica AD, Wang X-L, Liaw PK, Richardson JW. The development of grain-orientation-dependent residual stressess in a cyclically deformed alloy. Nat Mater 2003;2(2):101–6. http://dx.doi.org/10.1038/nmat812.

[7] Yang C, Mess F, Skenes K, Melkote S, Danyluk S. On the residual stress and fracture strength of crystalline silicon wafers. Appl Phys Lett 2013;102(2):021909. http://dx.doi.org/10.1063/1.4776706.

[8] Wang Y, Du Y, Xu F. Strain stiffening retards growth instability in residually stressed biological tissues. J Mech Phys Solids 2023;178:105360. http://dx.doi.org/10.1016/j.jmps.2023.105360.

[9] Schajer GS. Measurement of non-uniform residual stresses using the hole-drilling method. Part I—Stress calculation procedures. J Eng Mater Technol 1988;110(4):338–43. http://dx.doi.org/10.1115/1.3226059.

[10] Schajer G. Measurement of non-uniform residual stresses using the hole-drilling method. Part II—Practical application of the integral method. J Eng Mater Technol Trans ASME 1988.

[11] Smit TC, Reid RG. Residual stress measurement in composite laminates using incremental hole-drilling with power series. Exp Mech 2018;58(8):1221–35. http://dx.doi.org/10.1007/s11340-018-0403-6.

[12] Smit TC, Reid RG. Tikhonov regularization with incremental hole-drilling and the integral method in cross-ply composite laminates. Exp Mech 2020;60(8):1135–48. http://dx.doi.org/10.1007/s11340-020-00629-x.

[13] Smit T, Nobre J, Reid R, Wu T, Niendorf T, Marais D, et al. Assessment and validation of incremental hole-drilling calculation methods for residual stress determination in fiber-metal laminates. Exp Mech 2022;62(8):1289–304.

[14] Schajer GS. Hole-drilling residual stress measurements at 75: Origins, advances, opportunities. Exp Mech 2010;50(2):245–53. http://dx.doi.org/10.1007/s11340-009-9285-y.

[15] Schajer G. Application of finite element calculations to residual stress measurements. J Eng Mater Technol 1981.

[16] Tabatabaeian A, Ghasemi AR, Shokrieh MM, Marzbanrad B, Baraheni M, Fotouhi M. Residual stress in engineering materials: A review. Adv Eng Mater 2022;24(3):2100786. http://dx.doi.org/10.1002/adem.202100786.

[17] Lim J, Jung K, Jung Y, Kim D-N. Accelerating topology optimization using deep learning-based image super-resolution. Eng Appl Artif Intell 2024;133:108370. http://dx.doi.org/10.1016/j.engappai.2024.108370.

[18] Minh Nguyen-Thanh V, Trong Khiem Nguyen L, Rabczuk T, Zhuang X. A surrogate model for computational homogenization of elastostatics at finite strain using high-dimensional model representation-based neural network. Internat J Numer Methods Engrg 2020;121(21):4811–42.

[19] White DA, Arrighi WJ, Kudo J, Watts SE. Multiscale topology optimization using neural network surrogate models. Comput Methods Appl Mech Engrg 2019;346:1118–35. http://dx.doi.org/10.1016/j.cma.2018.09.007.

[20] Harandi A, Moeineddin A, Kaliske M, Reese S, Rezaei S. Mixed formulation of physics-informed neural networks for thermo-mechanically coupled systems and heterogeneous domains. Internat J Numer Methods Engrg 2024;125(4):e7388.
[21] Lee S-W, Truong-Quoc C, Ro Y, Kim D-N. Adversarial deep energy method for solving saddle point problems involving dielectric elastomers. Comput Methods Appl Mech Engrg 2024;421:116825. http://dx.doi.org/10.1016/j.cma.2024.116825.
[22] Kalina KA, Gebhart P, Brummund J, Linden L, Sun W, Kästner M. Neural network-based multiscale modeling of finite strain magneto-elasticity with relaxed convexity criteria. Comput Methods Appl Mech Engrg 2024;421:116739. http://dx.doi.org/10.1016/j.cma.2023.116739.
[23] Linka K, Hillgärtner M, Abdolazizi KP, Aydin RC, Itskov M, Cyron CJ. Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning. J Comput Phys 2021;429:110010. http://dx.doi.org/10.1016/j.jcp.2020.110010.
[24] Jeong I, Cho M, Chung H, Kim D-N. Data-driven nonparametric identification of material behavior based on physics-informed neural network with full-field data. Comput Methods Appl Mech Engrg 2024;418:116569. http://dx.doi.org/10.1016/j.cma.2023.116569.
[25] Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nat Mach Intell 2021;3(3):218–29. http://dx.doi.org/10.1038/s42256-021-00302-5.
[26] Koric S, Abueidda DW. Data-driven and physics-informed deep learning operators for solution of heat conduction equation with parametric heat source. Int J Heat Mass Transfer 2023;203:123809. http://dx.doi.org/10.1016/j.ijheatmasstransfer.2022.123809.
[27] Clark Di Leoni P, Lu L, Meneveau C, Karniadakis GE, Zaki TA. Neural operator prediction of linear instability waves in high-speed boundary layers. J Comput Phys 2023;474:111793. http://dx.doi.org/10.1016/j.jcp.2022.111793.
[28] Oommen V, Shukla K, Goswami S, Dingreville R, Karniadakis GE. Learning two-phase microstructure evolution using neural operators and autoencoder architectures. Npj Comput Mater 2022;8(1):190. http://dx.doi.org/10.1038/s41524-022-00876-7.
[29] Li W, Bazant MZ, Zhu J. Phase-field DeepONet: Physics-informed deep operator neural network for fast simulations of pattern formation governed by gradient flows of free-energy functionals. Comput Methods Appl Mech Engrg 2023;416:116299. http://dx.doi.org/10.1016/j.cma.2023.116299.
[30] Goswami S, Yin M, Yu Y, Karniadakis GE. A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. Comput Methods Appl Mech Engrg 2022;391:114587. http://dx.doi.org/10.1016/j.cma.2022.114587.
[31] He J, Kushwaha S, Park J, Koric S, Abueidda D, Jasiuk I. Sequential Deep Operator Networks (s-DeepONet) for predicting full-field solutions under time-dependent loads. Eng Appl Artif Intell 2024;127:107258. http://dx.doi.org/10.1016/j.engappai.2023.107258.
[32] Kushwaha S, Park J, Koric S, He J, Jasiuk I, Abueidda D. Advanced deep operator networks to predict multiphysics solution fields in materials processing and additive manufacturing. Addit Manuf 2024;88:104266. http://dx.doi.org/10.1016/j.addma.2024.104266.
[33] Yin M, Zhang E, Yu Y, Karniadakis GE. Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems. Comput Methods Appl Mech Engrg 2022;402:115027. http://dx.doi.org/10.1016/j.cma.2022.115027.
[34] He J, Koric S, Kushwaha S, Park J, Abueidda D, Jasiuk I. Novel DeepONet architecture to predict stresses in elastoplastic structures with variable complex geometries and loads. Comput Methods Appl Mech Engrg 2023;415:116277. http://dx.doi.org/10.1016/j.cma.2023.116277.
[35] Zhu M, Feng S, Lin Y, Lu L. Fourier-DeepONet: Fourier-enhanced deep operator networks for full waveform inversion with improved accuracy, generalizability, and robustness. Comput Methods Appl Mech Engrg 2023;416:116300. http://dx.doi.org/10.1016/j.cma.2023.116300.
[36] Zhu M, Zhang H, Jiao A, Karniadakis GE, Lu L. Reliable extrapolation of deep neural operators informed by physics or sparse observations. Comput Methods Appl Mech Engrg 2023;412:116064. http://dx.doi.org/10.1016/j.cma.2023.116064.
[37] Moya C, Zhang S, Lin G, Yue M. DeepONet-grid-UQ: A trustworthy deep operator framework for predicting the power grid's post-fault trajectories. Neurocomputing 2023;535:166–82. http://dx.doi.org/10.1016/j.neucom.2023.03.015.
[38] Psaros AF, Meng X, Zou Z, Guo L, Karniadakis GE. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. J Comput Phys 2023;477:111902. http://dx.doi.org/10.1016/j.jcp.2022.111902.
[39] Kobayashi K, Daniell J, Alam SB. Improved generalization with deep neural operators for engineering systems: Path towards digital twin. Eng Appl Artif Intell 2024;131:107844. http://dx.doi.org/10.1016/j.engappai.2024.107844.
[40] Zhang J, Zhang S, Shen J, Lin G. Energy-dissipative evolutionary deep operator neural networks. J Comput Phys 2024;498:112638. http://dx.doi.org/10.1016/j.jcp.2023.112638.
[41] He J, Koric S, Abueidda D, Najafi A, Jasiuk I. Geom-DeepONet: A point-cloud-based deep operator network for field predictions on 3D parameterized geometries. Comput Methods Appl Mech Engrg 2024;429:117130. http://dx.doi.org/10.1016/j.cma.2024.117130.
[42] Mathew J, Moat R, Paddea S, Fitzpatrick M, Bouchard P. Prediction of residual stresses in girth welded pipes using an artificial neural network approach. Int J Press Vessels Pip 2017;150:89–95. http://dx.doi.org/10.1016/j.ijpvp.2017.01.002.
[43] Mathew J, Moat RJ, Paddea S, Francis JA, Fitzpatrick ME, Bouchard PJ. Through-thickness residual stress profiles in austenitic stainless steel welds: A combined experimental and prediction study. Met Mater Trans A 2017;48(12):6178–91. http://dx.doi.org/10.1007/s11661-017-4359-4.
[44] Held E, Gibmeier J. Application of the incremental hole-drilling method on thick film systems—An approximate evaluation approach. Exp Mech 2015;55(3):499–507. http://dx.doi.org/10.1007/s11340-014-9962-3.
[45] Chupakhin S, Kashaev N, Klusemann B, Huber N. Artificial neural network for correction of effects of plasticity in equibiaxial residual stress profiles measured by hole drilling. J Strain Anal Eng Des 2017;52(3):137–51. http://dx.doi.org/10.1177/0309324717696400.
[46] Schajer GS, Prime MB. Use of inverse solutions for residual stress measurements. J Eng Mater Technol 2006;128(3):375–82. http://dx.doi.org/10.1115/1.2204952.
[47] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: Balcan MF, Weinberger KQ, editors. Proceedings of the 33rd international conference on machine learning. Proceedings of machine learning research, vol. 48, New York, New York, USA: PMLR; 2016, p. 1050–9.
[48] Kendall A, Gal Y. What uncertainties do we need in bayesian deep learning for computer vision? Adv Neural Inf Process Syst 2017;30.
[49] Smith M. ABAQUS/standard user's manual, version 6.9. United States: Dassault Systèmes Simulia Corp; 2009.
[50] Xu C, Cao BT, Yuan Y, Meschke G. Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios. Comput Methods Appl Mech Engrg 2023;405:115852. http://dx.doi.org/10.1016/j.cma.2022.115852.
[51] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. 2017, arXiv:1609.02907 [cs, stat].
[52] Shi Y, Huang Z, Feng S, Zhong H, Wang W, Sun Y. Masked label prediction: Unified message passing model for semi-supervised classification. 2021, arXiv:2009.03509 [cs, stat].
[53] Wilson EL. Structural analysis of axisymmetric solids. AIAA J 1965;3(12):2269–74. http://dx.doi.org/10.2514/3.3356.
[54] Schajer GS, Yang L. Residual-stress measurement in orthotropic materials using the hole-drilling method. Exp Mech 1994;34(4):324–33. http://dx.doi.org/10.1007/BF02325147.
[55] Patel D, Ray D, Abdelmalik MR, Hughes TJ, Oberai AA. Variationally mimetic operator networks. Comput Methods Appl Mech Engrg 2024;419:116536. http://dx.doi.org/10.1016/j.cma.2023.116536.
[56] Cipolla R, Gal Y, Kendall A. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: 2018 IEEE/CVF conference on computer vision and pattern recognition. Salt Lake City, UT, USA: IEEE; 2018, p. 7482–91. http://dx.doi.org/10.1109/CVPR.2018.00781.
[57] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2017, arXiv:1412.6980 [cs].
[58] Loshchilov I, Hutter F. Decoupled weight decay regularization. 2017, arXiv preprint arXiv:1711.05101.
[59] Cawley GC, Talbot NL. On over-fitting in model selection and subsequent selection bias in performance evaluation. J Mach Learn Res 2010;11:2079–107.
[60] Dung DV, Song ND, Palar PS, Zuhal LR. On the choice of activation functions in physics-informed neural network for solving incompressible fluid flows. In: AIAA SCITECH 2023 forum. 2023, p. 1803.
[61] Leng K, Thiyagalingam J. On the compatibility between neural networks and partial differential equations for physics-informed learning. 2023, arXiv:2212.00270 [physics].
[62] Jiang G, Wang W. Error estimation based on variance analysis of k -fold cross-validation. Pattern Recognit 2017;69:94–106. http://dx.doi.org/10.1016/j.patcog.2017.03.025.
[63] Schajer GS. Compact calibration data for hole-drilling residual stress measurements in finite-thickness specimens. Exp Mech 2020;60(5):665–78.
[64] Gao W, Wang C. Active learning based sampling for high-dimensional nonlinear partial differential equations. J Comput Phys 2023;475:111848. http://dx.doi.org/10.1016/j.jcp.2022.111848.
[65] Lye KO, Mishra S, Ray D, Chandrashekar P. Iterative surrogate model optimization (ISMO): An active learning algorithm for PDE constrained optimization with deep neural networks. Comput Methods Appl Mech Engrg 2021;374:113575. http://dx.doi.org/10.1016/j.cma.2020.113575.
[66] Sitzmann V, Martel J, Bergman A, Lindell D, Wetzstein G. Implicit neural representations with periodic activation functions. Adv Neural Inf Process Syst 2020;33:7462–73.
[67] Barnes J, Byerly G. The formation of residual stresses in laminated thermoplastic composites. Compos Sci Technol 1994;51(4):479–94. http://dx.doi.org/10.1016/0266-3538(94)90081-7.