Data-Centric Engineering (2024), 5: e6
doi:10.1017/dce.2024.4

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Physics-informed neural networks for structural health monitoring: a case study for Kirchhoff–Love plates

Anmar I. F. Al-Adly and Prakash Kripakaran

Department of Engineering, Faculty of Environment, Science and Economy, University of Exeter, Streatham Campus, Exeter. EX4 4QF, UK
Corresponding author: Anmar I. F. Al-Adly; Email: aa1224@exeter.ac.uk

Received: 04 July 2023; **Revised:** 19 December 2023; **Accepted:** 16 February 2024

Keywords: digital twins; neural networks; physics-informed machine learning; plate bending; structural health monitoring

Abstract

Physics-informed neural networks (PINNs), which are a recent development and incorporate physics-based knowledge into neural networks (NNs) in the form of constraints (e.g., displacement and force boundary conditions, and governing equations) or loss function, offer promise for generating digital twins of physical systems and processes. Although recent advances in PINNs have begun to address the challenges of structural health monitoring, significant issues remain unresolved, particularly in modeling the governing physics through partial differential equations (PDEs) under temporally variable loading. This paper investigates potential solutions to these challenges. Specifically, the paper will examine the performance of PINNs enforcing boundary conditions and utilizing sensor data from a limited number of locations within it, demonstrated through three case studies. Case Study 1 assumes a constant uniformly distributed load (UDL) and analyzes several setups of PINNs for four distinct simulated measurement cases obtained from a finite element model. In Case Study 2, the UDL is included as an input variable for the NNs. Results from these two case studies show that the modeling of the structure's boundary conditions enables the PINNs to approximate the behavior of the structure without requiring satisfaction of the PDEs across the whole domain of the plate. In Case Study (3), we explore the efficacy of PINNs in a setting resembling real-world conditions, wherein the simulated measurment data incorporate deviations from idealized boundary conditions and contain measurement noise. Results illustrate that PINNs can effectively capture the overall physics of the system while managing deviations from idealized assumptions and data noise.

Impact Statement

This paper investigates approaches for creating neural network models that satisfy the structure's physical constraints and measured deflection data. It also examines the ability of the models to generalize for uniformly distributed load (UDL) conditions beyond the training data if informed of the linear relationship between UDL and deflection. The overall goal is to evaluate the improvement in predictive accuracy, specifically in the behavior of structures, when the models are trained with limited physical knowledge and then applied to new, unseen data. The ability to predict structural behavior accurately will support maintenance, repair, and optimization decisions and consequently will be valuable for the areas of structural health monitoring, digital twins, and virtual sensing.

1. Introduction

Structural health monitoring (SHM) (Figueiredo and Brownjohn, 2022; Fremmelev et al., 2022) is increasingly employed on civil infrastructures such as bridges for preserving their integrity and safety, and more generally for their effective maintenance and management. A key unresolved challenge is the translation of SHM data to decision-making. This requires processing vast amounts of measurement data, such as that from the Queensferry Crossing Bridge in the United Kingdom with over 2,184 sensors (Cousins et al., 2022), to provide actionable information to bridge operators. Over the years, several studies have focused on this challenge with an emphasis particularly on damage detection (Middleton et al., 2016; Webb et al., 2015). Worden et al. (2015) offered a classification of the different stages of damage identification—from anomaly detection to damage prognosis, with most studies focusing primarily on the first two stages. Increasingly, however, the emphasis has turned to preventive maintenance and long-term structural management by leveraging recent developments in AI and machine learning to generate digital twins (Sacks et al., 2020) that not only detect anomalous events in real time from measured data but also offer detailed insight into structural behavior and performance, such as by tracking gradual deterioration of components and estimating remaining service life.

Modeling of digital twins in the context of structural identification and damage assessment can involve three kinds of approaches. First, model-based methods, which create an accurate physics-based model—typically a finite element model (FEM), of the structure (Sohn et al., 2002) by updating model parameters, based on actual observations. The updated FEM will subsequently serve as a benchmark against which new measurements will be compared for assessing the integrity and performance of the monitored structure. The drawbacks of model-based methods are that they are computationally intensive, have simplified assumptions and approximations, and are subject to uncertainties in modeling parameters (Farrar and Worden, 2012). Certain limitations may be addressed through uncertainty quantification analysis, as proposed by Sankararaman and Mahadevan (2013), or alternatively through probabilistic approaches like Bayesian model updating (Kamariotis et al., 2022).

The second class of approaches is data-driven methods. These typically rely on machine learning algorithms, primarily unsupervised learning algorithms, and detect anomalies by identifying changes to patterns in measurement data through comparisons with patterns observed during the structure's normal operation. In full-scale structures, distinguishing between feature changes due to damage and those caused by changing operational and environmental conditions such as temperature and wind remains a challenge (Figueiredo and Brownjohn, 2022). In addition, data covering a wide range of structural states are seldom available for training (Y. Ni et al., 2020), limiting the trained model's performance and ability to generalize. However, compared to model-based methods, these are computationally less expensive and are easier to apply for real-time monitoring.

The third class of methods—physics-informed machine learning (PIML) methods (Karniadakis et al., 2021)—are a relatively new development and are essentially machine learning algorithms that have been augmented with physics-based knowledge, provided in the form of constraints and loss function, or directly implemented in their architecture. PIML models, which can fully or partially satisfy the governing physics of a structure while being data-driven, have the potential for accurate predictions and improved generalization, defined as the model's ability to extrapolate predictions beyond the training set into unobserved regions.

In the field of SHM, PIML approaches are conducted through physics-informed neural networks (PINNs), which adjust their loss function to incorporate physics-based constraints, ensuring adherence to the governing physics equations of structural systems, which is the focus of this paper. Additionally, hybrid models that combine physics-based methods with machine learning techniques provide a noteworthy example. Specifically, the integration of Gaussian processes (GPs) with physics-based models. The mean function of the GP is used in this approach to integrate prior physical knowledge, while the observed data are incorporated using covariance functions, as discussed in the work of Cross et al. (2022).

Another hybrid approach involves using the output from physics-based models as input for machine learning algorithms. For instance, studies by Bud et al. (2019, 2023) demonstrated this method, where

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

monitoring data reflecting normal operational conditions of an undamaged structure, along with numerical data from FEM under extreme environmental conditions or damage scenarios, are utilized to train a machine learning algorithm. For further exploration, Xu et al. (2023) undertook a comprehensive review of PIML methods for reliability and system safety applications, and Karniadakis et al. (2021) discussed applications of PIML for forward and inverse problems as well as capabilities and their limitations.

The concept of PINNs is developed by Raissi et al. (2017a,b) in a two-part article, which is consolidated into a merged version published in 2019 (Raissi et al., 2019). PINNs were introduced as a new class of data-driven solvers and as a novel approach in scientific machine learning for dealing with problems involving partial differential equations (PDEs) (Cuomo et al., 2022) using automatic differentiation. Several studies have since applied PINNs for a range of scientific problems (Shukla et al., 2022), involving stochastic PDEs, integro-differential equations, fractional PDEs, non-linear differential equations (Uddin et al., 2023), and optimal control of PDEs (Mowlavi and Nabi, 2023). Within engineering, PINNs have been applied to problems in fluid dynamics (Mao et al., 2020; Raissi et al., 2020; Sliwinski and Rigas, 2023), heat transfer (Zobeiry and Humfeld, 2021), tensile membranes (Kabasi et al., 2023), and material behavior modeling (Zheng et al., 2022). However, there are only a few applications to SHM, the subject area of focus in this paper.

Previous studies on PINNs for SHM have primarily focused on training PINNs to satisfy the PDEs that govern the dynamic behavior of structural systems. Lai et al. (2021) and W. Liu et al. (2022) investigated physics-informed neural ordinary differential equations and physics-guided deep Markov models for structural identification and tested their prediction capabilities for a series of numerical and experimental examples. Lai et al. (2022) presented a framework that combines physics-based modeling and deep learning techniques to model civil and mechanical dynamical systems. They showed that the generated models have the ability to effectively reconstruct the structural response using data from only a limited number of sensors, although performance was observed to deteriorate when the dynamic regime deviated significantly from the training data. P. Ni et al. (2022) used a multi-end convolutional autoencoder to reconstruct the full seismic response of multi-degree of freedom (DOF) systems. They found that their methodology could reconstruct the seismic response of multi-DOF systems given a small training dataset, and PINNs performed better than conventional neural networks (NNs) with small datasets, particularly when the training data contained noise. Yuan et al. (2020) investigated the use of PINNs to solve forward and inverse PDE-based problems based on dynamic modeling of beam structures using synthetic sensor data analysis.

Further research is needed into the application of PINNs for quasi-static monitoring, which involves analyzing a structure's response at discrete time intervals. This type of monitoring is critical for assessing the long-term monitoring of structures (Kromanis and Kripakaran, 2014, 2016). In contrast to dynamic monitoring, quasi-static monitoring frequently uses sampling rates that may not sufficiently capture vibration properties, instead focusing on static responses such as strains and displacements. PINNs, by leveraging their physics-informed nature, may be able to compensate for insufficient sampling rates by directly integrating physical laws into the learning process, enhancing the model's ability to infer accurate responses even from limited data. This introduces other challenges such as temporally variable loading, which complicate modeling and verification of the PDE within PINNs. This study will investigate this aspect. It also leads to the other key novel contribution, which is to examine how model performance is influenced by the different physics-based constraints implemented within the loss function of a PINN. Specifically, we aim to examine whether a PINN that enforces the structure's boundary conditions and uses sensor data from a limited number of locations across the structure can accurately model the structural system despite not explicitly incorporating the governing PDE, which can be difficult to evaluate due to variable loading. The hypothesis is that deflections from a sparse set of sensors and constraints on satisfying force boundary conditions can allow PINNs to generalize for the whole structure. The generated PINNs will also be tested for their capability to predict deflections and internal forces at locations where there are no sensors. Lastly, we will also evaluate their potential for predicting deflections and forces for loads unseen in the training datasets.

The paper uses the numerical model of a square concrete slab, representative of a building structural element, to illustrate and test the developed ideas. It states the governing PDE and the boundary conditions of the slab, and describes how these can be enforced through the loss function for a PINN. The paper then investigates the effect of the various terms in the loss function on PINN model performance through three case studies. Case Study 1, which assumes constant unchanging load, examines five different setups of PINNs, each with different loss functions incorporating varying levels of physical knowledge. Each of the PINN setups is evaluated for four measurement cases, representing increasing amounts of measurement data derived from simulated FEM analytical solution. Case Study 2 includes the applied loading as an input variable for the NNs. Case Study 3 is aimed at investigating the robustness of the PINNs to (i) deviations from idealized boundary conditions, and (ii) varying levels of noise in data. The case studies are primarily conducted in the spatial domain. Extending them to include a temporal component would pose challenges, such as the need for more complex model architectures and potentially larger datasets to effectively capture temporal dynamics.

The outline of this paper is as follows. In Section 2, the plate problem and the methodology adopted for generating PINNs are introduced. Case Study 1 and the corresponding results are discussed in Section 3. Using the results as a basis, Case Study 2 is set up and this process and the results are presented in Section 4. Section 5 is dedicated to Case Study 3, which explores additional complexities. Finally, Section 6 summarizes the key conclusions of this study.

2. Problem setup and methodology

This section outlines the structure—a typical reinforced concrete floor slab, that is to be modeled using a PINN. It describes the underlying physics of the structure and the scenarios that are considered. It then describes the structure of the PINN, its parameters and the approaches adopted for imposing the physical constraints on the PINN.

2.1. Kirchhoff–Love plate

The Kirchhoff–Love plate theory (Kirchhoff, 1850; Love, 1888) is a mathematical model for calculating stresses and deflections in thin plates subjected to forces and moments. This model, which was first proposed by Love in 1888, is a two-dimensional extension of the Euler–Bernoulli beam theory. Kirchhoff's assumptions provided a basis for the development of this theory. The model is employed for the analysis and design of a wide range of structural components in civil engineering including concrete slabs.

2.1.1. Governing equations

Figure 1 presents a square plate with four simply supported edges labeled A, B, C, and D. The plate is subjected to a uniformly distributed load (UDL) q across its entire surface. The magnitude of the UDL directly influences the deflections in the plate, a relationship we will attempt to capture in modeling. The vertical deflection $w(x,y)$, that is, along the z-direction, in a Kirchhoff–Love plate must satisfy the PDE (Equation (1)) below (Timoshenko and Woninowsky-Kerieger, 1959):

$$\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} - \frac{q}{D} = 0. \tag{1}$$

D in Equation (1) represents the flexural rigidity of the plate and is determined from Young's modulus (E), the plate's thickness (h), and Poisson's ratio v as given below:

$$D = \frac{Eh^3}{12(1-v^2)}. \tag{2}$$

The bending moments represented by M_x, M_y, and M_{xy} and shear forces represented by Q_x and Q_y constitute the main internal forces within the plate structure from a structural engineer's perspective.

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

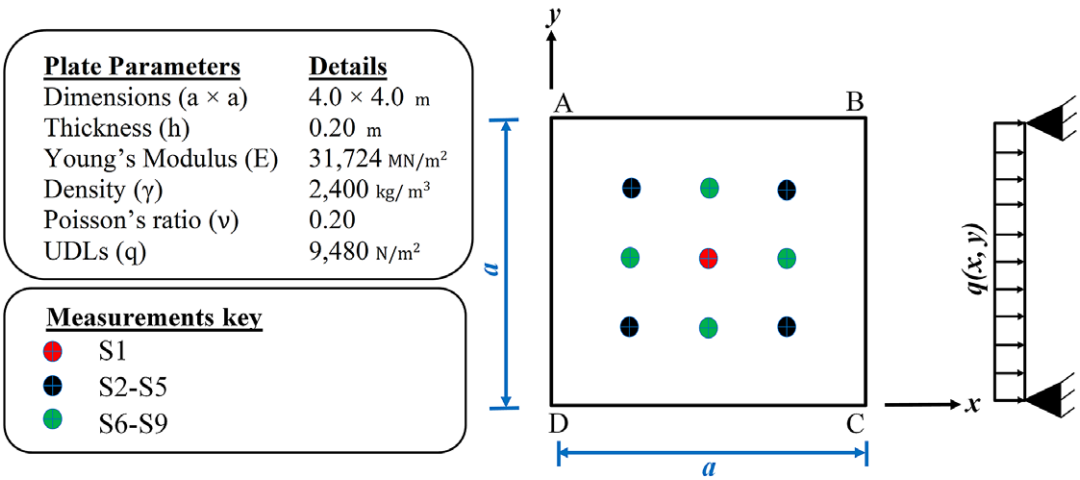


Figure 1. *A simply supported plate under uniformly distributed load and its assumed distribution of sensors.*

Computation of the bending moments is contingent upon an understanding of the deflection of the plate and knowledge of the flexural rigidity mentioned earlier. The following Equations (3)–(5) can be used to compute the moments from the deflections:

$$M_x = -D\left(\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right), \tag{3}$$

$$M_y = -D\left(\frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2}\right), \tag{4}$$

$$M_{xy} = -M_{yx} = -D(1-\nu)\frac{\partial^2 w}{\partial x \partial y}. \tag{5}$$

2.1.2. Boundary conditions

Two types of boundary conditions—displacement and force boundary conditions—are enforced on a structure during structural analysis to model how the structure interacts with its environment (G. R. Liu and Quek, 2014). Displacement boundary conditions define restrictions on the movement of specific points or regions of the structure. These are also referred to as essential boundary conditions as they are imposed directly on displacements, which are the main parameters in the governing equation, and ultimately result in equilibrium equations supporting the solution process when solving numerically.

Force boundary conditions define how internal forces and stresses at the boundaries relate to external forces (and moments) acting on the structure. These are also called natural boundary conditions since they are inherently derived from the displacements (G. R. Liu and Quek, 2014). Both force and displacement boundary conditions are essentially physics-based constraints that a solution to a given plate problem, that is, a $w(x,y)$ solution for Equation (1), must satisfy.

For the plate defined in Figure 1, the four edges are assumed as simple supports. This implies that there will be no displacement at its four boundaries; this constitutes its displacement boundary conditions. The simple supports also imply that the plate is allowed to rotate freely about the edges defined by $x=0$, $x=a$, $y=0$, and $y=a$; these constitute the force boundary conditions for the plate. In other words, these require that the bending moments M_x is zero along $y=0$ and $y=a$, and M_y is zero along $x=0$ and $x=a$.

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

Equations (6)–(8) express these boundary conditions mathematically:

$$w = 0 \text{ along } x = 0,\ x = a, y = 0, \text{ and } y = a, \tag{6}$$

$$M_x = 0 \text{ along } y = 0 \text{ and } y = a, \tag{7}$$

$$M_y = 0 \text{ along } x = 0 \text{ and } x = a. \tag{8}$$

2.2. Synthetic sensor data

Since this study is about generating PINNs that emulate measured structural behavior, the plate is assumed to have sensors. We assume that displacement sensors are present at a set of locations across the surface as depicted in Figure 1. We will use data predicted at these locations by an FEM representing the slab in Figure 1 as measured data for training the PINNs. While we have assumed displacement sensors, the PINNs concept outlined in the subsequent section would also be applicable if other types of measurements such as strains and rotations were to be available. Consequently, this study is not concerned with the actual sensor types or the process of measurement. Researchers interested in sensor types, their failures, and their optimization for SHM can refer to relevant recent literature (Kechavarzi et al., 2016; Middleton et al., 2016; Wang et al., 2022; Oncescu and Cicirello, 2023).

The FEM was set up in the Ansys (2023) software and the simulations performed using the PyAnsys (Kaszynski et al., 2021) toolkit that enables Ansys integration with Python. The model utilized Shell Element 181, which incorporates six DOFs and conforms to the Kirchhoff–Love plate theory, for modeling plate bending behavior. The numerical model with a mesh made up of 64 elements and 80 nodes provides a detailed illustration of the structural behavior. Typical material properties for concrete slabs were adopted and are also shown in Figure 1. The UDL of 9,480 N/m^2 is assumed, which is in the range of design loads in practice. Contour plots of the resulting deflections w, moments M_x, and moment M_y for the plate are presented in Figure 2. Table 1 presents the predicted data from the model at the assumed sensor locations (see Figure 1); these are treated as measurements at the respective locations for the purpose of this study.

2.3. Development of PINNs

An NN denoted as a function $NN(X)$ is shown in Equation (9). The function accepts an input variable X, which is subjected to a series of weight matrices W and bias vectors b to generate an output vector. The process of transformation involves the passage of input through multiple activation functions, which are represented by the σ. The incorporation of nonlinearities in this process endows the NN with the capability to discern and comprehend intricate data patterns. When the NN is used for prediction, the outcome from NN is regarded as the anticipated outcome for the system that the NN is emulating:

$$NN(X) = W_n \sigma_{n-1}(W_{n-1}\sigma_{n-2}(\ldots(W_2(W_1 X + b_1) + b_{n-1}) + b_n. \tag{9}$$

For the plate structure, the x and y coordinates, indicating the location at which the displacement is desired, constitutes the input X. The loading on the plate q may also be assumed as an input and the study shall

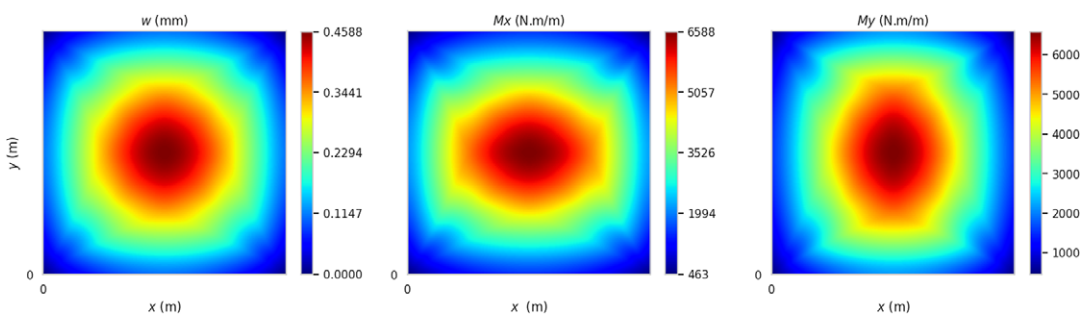


Figure 2. *Deflection and moments predicted by finite element model of plate described in Figure 1.*

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

Table 1. *Summary of measured data and locations (UDL = 9,480 N/m²)*

Sensor no.	Coordinates (x,y) (m)	Deflection (mm)	M_x (N.m/m)	M_y (N.m/m)
S1	(2,2)	0.4589	6,589	6,589
S2	(1,1)	0.2404	3,524	3,524
S3	(3,1)	0.2403	3,524	3,524
S4	(1,3)	0.2404	3,524	3,524
S5	(3,3)	0.2404	3,524	3,524
S6	(2,1)	0.3315	4,518	5,239
S7	(2,3)	0.3314	4,518	5,239
S8	(1,2)	0.3315	5,239	4,518
S9	(3,2)	0.3314	5,239	4,518

explore models that accept q as input later in Section 4. The output from the NN is the displacement w. Other derived parameters such as bending moments M_x, M_y, and M_{xy} may also be predicted using the NN as will be demonstrated later on.

The training of NNs typically requires the minimization of a loss function, such as the mean squared error (MSE) computed between predictions and expected results for a training dataset comprising preexisting information. In the case of PINNs, the loss function incorporates components beyond this MSE. Specifically, terms associated with the governing PDE for the system and its boundary conditions are included. In this manner, the PINN is not only able to offer accurate predictions for the training dataset but also aims to satisfy physical constraints for the system. This study models the loss function as an aggregate loss, represented as $\mathcal{L}$, that is computed as the summation of four different loss components —$\mathcal{L}_f$, $\mathcal{L}_w$, $\mathcal{L}_m$, and $\mathcal{L}_d$, as shown in Equation (10):

$$\mathcal{L} = \mathcal{L}_f + \mathcal{L}_w + \mathcal{L}_m + \mathcal{L}_d. \tag{10}$$

$\mathcal{L}_f$, $\mathcal{L}_w$, and $\mathcal{L}_m$ evaluate how well the NN satisfies the governing PDE, the displacement boundary conditions, and the force boundary conditions, respectively. $\mathcal{L}_d$ represents the performance of the NN for the training dataset. Each of these loss terms is written as the product of a weight α_i and the related MSE. Consequently, $\mathcal{L}$ can be formulated as follows, wherein α_f, α_w, α_m, and α_d represent the weights associated with the respective MSE values:

$$\mathcal{L} = \alpha_f MSE_f + \alpha_w MSE_w + \alpha_m MSE_m + \alpha_d MSE_d. \tag{11}$$

Each of the MSE terms in Equation (11) is described in detail below.

1. MSE_f represents the average residual for the governing PDE outlined in Equation (1). The mean of the residual $f(x,y)$ (see Equation (12)) evaluated at a set of "collocation points" —$\{(x_i,y_i): 0 \le i \le N_f\}$—within the domain of the plate is taken as MSE_f as shown in Equation (13). The partial derivatives will be evaluated using automatic differentiation:

$$f(x,y) = \frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} - \frac{q}{D}, \tag{12}$$

$$MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f} \|f(x_i,y_i)\|^2. \tag{13}$$

2. MSE_w addresses the plate's displacement boundary conditions (Equation (6)). A set of $4N_w$ points is generated by randomly choosing N_w points from each of the four edges of the plate boundaries,resulting in a total of $4N_w$ points given by $\{(x_i,y_i): 0 \le i \le 4N_w\}$. The first $2N_w$ points

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

are from edges AB and CD, and the last $2N_w$ points are from edges AD and BC. We predict the displacements at these points using the PINN and compute the average error MSE_w in the predictions, which should be zero to satisfy the boundary conditions, using Equation (14) below:

$$MSE_w = \frac{1}{N_w}\sum_{i=1}^{N_w} \|w(x_i, y_i)\|^2. \tag{14}$$

3. MSE_m is similar to MSE_w but addresses the force boundary condition (Equations (7) and (8)). The same N_w points used for MSE_w are also employed here. MSE_m is computed as given in Equation (15), wherein $M_x = 0$ is required for points along edges AD and BC, and $M_y = 0$ is required for the points along edges AB and CD. The moments, being derivatives of deflection w (Equations (3) and (4)), are evaluated using automatic differentiation:

$$MSE_m = \frac{1}{4N_w}\left[\sum_{i=1}^{2N_w} \|M_x(x_i, y_i)\|^2 + \sum_{i=2N_w+1}^{4N_w} \|M_y(x_i, y_i)\|^2\right]. \tag{15}$$

4. MSE_d relates to the reconstruction of the deflection field based on measurements from a sparse sensor array. The sensor locations for the plate are presented in Figure 1. If N_d sensors are employed on the plate at locations $\{(x_i, y_i) : 0 \le i \le N_d\}$, then the measurements $D(x_i, y_i)$ at these locations can be compared against the corresponding displacement predictions from the PINN to compute MSE_d as shown below:

$$MSE_d = \frac{1}{N_d}\sum_{i=1}^{N_d} \|D(x_i, y_i) - w(x_i, y_i)\|^2. \tag{16}$$

2.4. PINN setup and tuning parameters

Figure 3 illustrates the schematic overview of the PINNs framework. The diagram depicts a feed-forward NN that is fully connected and accepts coordinates (x, y) as inputs to predict a solution $w(x, y)$, the expected deflection at the specified location. The technique of automatic differentiation is utilized for the computation of the derivatives of w in relation to the inputs. The derivatives are subsequently evaluated at specific locations and employed to find MSE_f, MSE_w, MSE_m, and MSE_d as explained in Equations (13)–(16). The number of points for N_f and N_w is equal to 800 and 100, respectively.

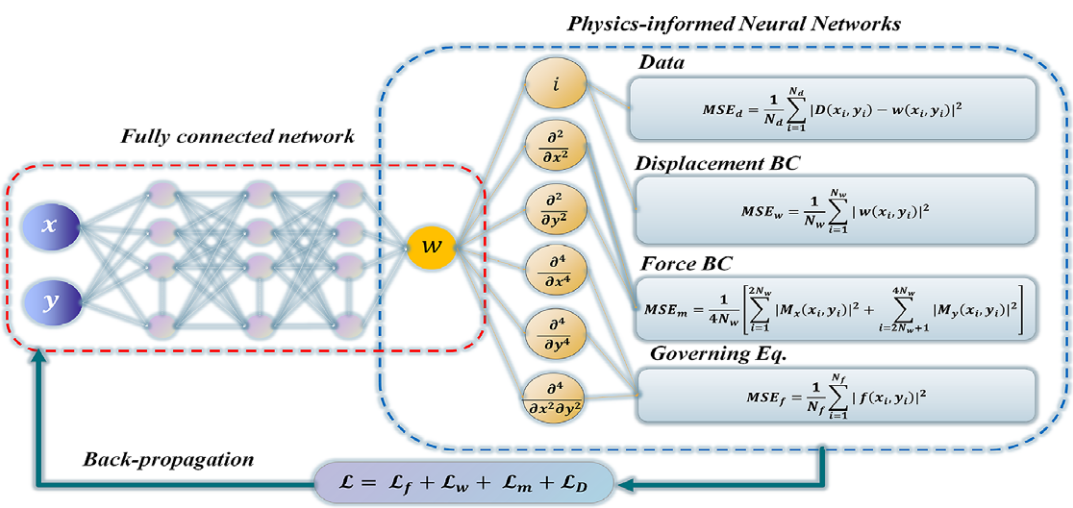


Figure 3. *Schematic of the physics-informed neural networks.*

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

Table 2. *Hyperparameters related to the neural network's architecture and training*

Hyperparameters	Details
Learning rate	0.001
Layers	4
Neurons per layer	64
Activation function	Tanh
Epoch	1,500
Optimizer	Adam
Early stopping	Yes
ReduceLROnPlateau	Yes

Our code is based on the work of Bischof and Kraus (2022), which examined the application of PINNs to a comparable plate with the aim of achieving weight balance across the loss function terms. We have adapted their open-source code for our case studies and this can be found at GitHub. The TensorFlow (Abadi et al., 2015) library is used to implement this computational process, and the model is trained on Google Colab, with GPU capabilities.

Table 2 provides a detailed summary of the hyperparameters that were selected for the PINN model. These hyperparameters were carefully optimized through a series of iterative trial and error processes to achieve the best possible performance of the model. The learning rate, an essential parameter that governs the magnitude of the step taken during each iteration during the minimization of the loss function, is set as 0.001. The architecture of the NN employed in this study consists of four hidden layers, each containing 64 neurons. This configuration is selected after examining various architectures, ranging from two to six hidden layers. The choice of four hidden layers attains a balance aimed at reducing computational time and avoiding over-fitting. Additionally, the Tanh activation function is employed in each layer. The Tanh function is chosen due to its non-linear characteristics, which are essential for PINNs. Unlike linear activation functions, such as ReLU, Tanh enhances the network's ability to learn and predict nonlinear patterns. This is a key requirement in the framework of PINNs, where handling derivative terms in the governing equations is crucial for accurate prediction and pattern acquisition. The model is subjected to a training process spanning 1,500 epochs, where each epoch involves a complete iteration over the entire training dataset. The optimizer utilized is Adam. Further techniques for enhancing the model's efficacy include (i) early stopping, which halts the training process as soon as the model's performance begins to deteriorate on a validation dataset, thus preventing overfitting, and (ii) ReduceLROnPlateau, which limits the learning rate when the model's learning advancement decelerates, enabling more precise calibration (Goodfellow et al., 2016).

2.5. Methods for functional loss balancing

MSE_f, MSE_w, MSE_m, and MSE_d are based on different measurement units, and their magnitudes may vary by multiple orders of magnitude. Consequently, the overall loss $\mathcal{L}$ may be biased toward one of these loss terms if these are simply added up. To avoid this, we employ weights—α_f, α_w, α_m, and α_d, as already stated in Equation (11).

The gradient descent algorithm is chosen to determine the optimal weights for the various terms in the loss function. The algorithm considers the relative importance of each term in the loss function and avoids overemphasizing any one term by setting the sum of the weights to 1. This method is effective at determining the optimal weights for each term of the loss function, allowing us to train the model more efficiently and accurately. Figure 4 shows the contribution of loss function components with respect to the total loss and median of the log MSE (L_2) loss during optimization. The loss decreases rapidly at the

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

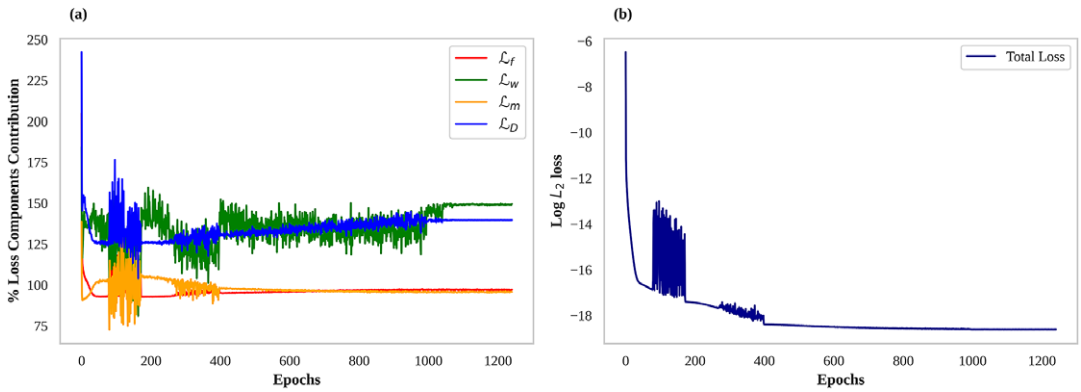


Figure 4. *(a) Percentage contribution of loss components. (b) Loss optimization observed with gradient descent.*

start of optimization as expected but slows after 400 epochs. After that, due to convergence of solution, the loss decreases very slowly.

2.6. Optimizing the number of collocation points

Collocation points are specialized samples in a computational domain that have been selected to embed physical laws into the network with the objective of proving computational simplicity for efficient evaluation, representatives to cover the computational domain, and feasibility, which ensures the points represent attainable system states or domains (Sholokhov et al., 2023). As consequently, selecting an appropriate set of collocation points not only improves model precision during PINNs training, ensuring that the NN learns to adhere to the governing physical principles of the system being modeled, but also reduces computational time. This investigation is divided into two distinct but interconnected sections: the examination of the influence on model performance of the number of collocation points within the model's internal domain and those located around its boundaries.

For internal domain, we initially focus on the number of collocation points ranging from 400 to 1,000. The number of boundary collocation points along the plate edges is fixed at 50 per edge. For each collocation point setting, PINNs are trained with 10 different seeds. These seeds are randomized for each setup, which means that the model is trained 10 times with 10 different, randomly chosen initial seeds for producing robust statistical metrics for each specific number of collocation points (400, 500, up to 1,000). Figure 5a,b shows the box plots of the root-mean-square error (RMSE) associated with deflection and bending moments (Mx), respectively. According to the analysis, the relationship between the number of internal domain collocation points and RMSE does not always follow a strictly linear or decreasing trend. In Figure 5a, while the RMSE for deflection generally decreases as the number of collocation points increases, there are notable exceptions to this pattern, particularly at points ranging from 600 to 900. These examples show that increasing the number of collocation points does not always result in improved model accuracy for deflection predictions. Figure 5b, on the other hand, shows a somewhat more consistent trend for the RMSE associated with Mx, though it, too, exhibits fluctuations, emphasizing the complexities of the relationship between collocation points and model accuracy. Remarkably, the variance in RMSE tends to narrow at higher collocation points (specifically, 800 and 900), implying that model performance at these levels is more stable and consistent.

The influence of boundary collocation points on model performance is subsequently investigated. The number of internal collocation points is now fixed at 800, as identified in the previous stage. Six different counts of boundary collocation points ranging from 50 to 100 per edge (i.e., 200 to 400 in total), while adhering to the analytical methodology established for the internal domain. Figure 6a,b shows the RMSE values for the chosen number of boundary points. Figure 6a shows that the RMSE for deflection increases

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

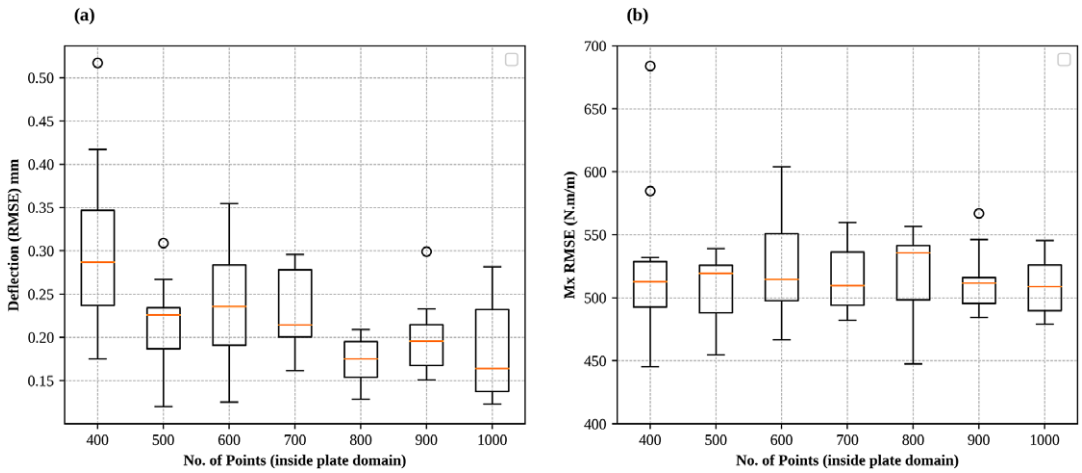


Figure 5. *(a) Deflection root-mean-square (RMSE) and (b) moment RMSE for different numbers of internal domain collocation points.*

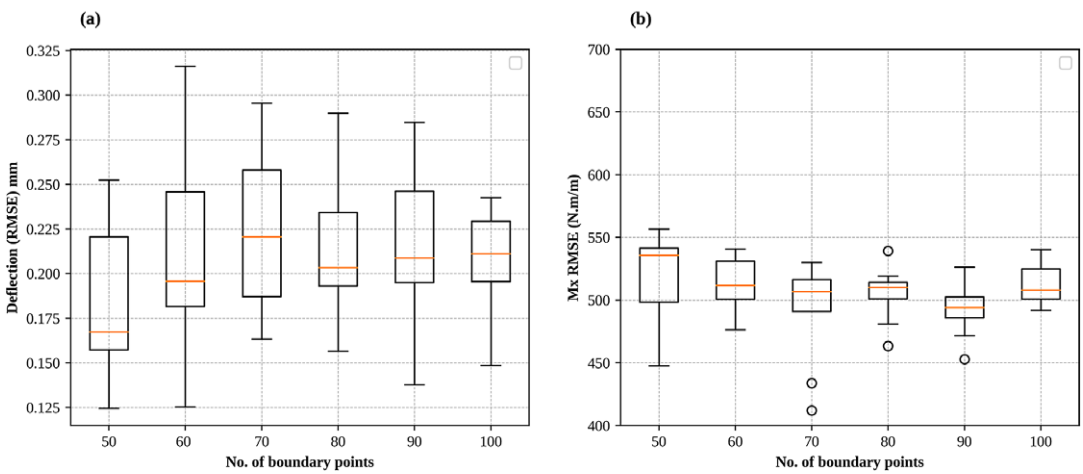


Figure 6. *(a) Deflection root-mean-square (RMSE) and (b) moment RMSE for different numbers of boundary collocation points.*

as the number of boundary collocation points increases, with the lowest RMSE observed at the fewest number of boundary points. Despite this trend, the RMSE variance tends to narrow in the range of 80 to 100 points per edge, indicating that model performance may stabilize at higher collocation point counts. Figure 6b, on the other hand, shows a different pattern for the RMSE associated with moment (Mx), where a decrease in RMSE values is observed as the number of collocation points increases. The best and most consistent RMSE values are found between 70 and 90 points per edge. Based on the analysis presented here, the study proceeds with 800 collocation points within the internal domain, which is supplemented with 400 points around the boundaries.

3. Case Study 1

In this section, the UDL q is assumed to be constant for the duration of monitoring. The study then evaluates the influence of the loss terms—$\mathcal{L}_f$, $\mathcal{L}_w$, $\mathcal{L}_m$, and $\mathcal{L}_d$ as given in Equation (10), on PINN

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

Table 3. *Summary of the loss function scenarios employed in the study*

Scenario	Loss function $\mathcal{L}$
SN1	$\mathcal{L}_d$
SN2	$\mathcal{L}_w + \mathcal{L}_d$
SN3	$\mathcal{L}_m + \mathcal{L}_d$
SN4	$\mathcal{L}_w + \mathcal{L}_m + \mathcal{L}_d$
SN5	$\mathcal{L}_f + \mathcal{L}_w + \mathcal{L}_m + \mathcal{L}_d$

performance. For this purpose, PINN models are generated for various combinations of loss terms—specifically the five scenarios, as listed in Table 3. In SN1, the loss function only includes $\mathcal{L}_d$, which represents the fit to the measured data. In SN2 and SN3 scenarios, displacement and force boundary conditions, respectively, are included alongside $\mathcal{L}_d$. Scenario SN4 refers to a PINN generated with a loss function combining loss terms for force and displacement boundary conditions—$\mathcal{L}_m$ and $\mathcal{L}_w$, with $\mathcal{L}_d$. Scenario SN5 represents the case when the loss $\mathcal{L}_f$—corresponding to the satisfaction of the governing PDE, is also included.

For each of the scenarios in Table 3, the study also evaluates the influence of the quantity of measurements on model performance. For this, PINN models are generated for varying number of assumed sensors (i.e., measurements)—from zero to nine, as shown by the four cases in Table 4. The case with no measurements is employed only with loss scenarios SN2–SN5 to evaluate whether enforcing constraints on governing PDE and/or boundary conditions, without having measurements, would lead to an accurate PINN.

3.1. Error metrics

The performance of the PINNs generated for the various scenarios is evaluated using multiple metrics. The RMSE given in Equation (17) captures the average magnitude of the errors between predicted and observed values, thereby providing a scale-dependent snapshot of the model's general accuracy. The coefficient of variation (CV) of the RMSE from Equation (18) provides a scale-independent perspective by normalizing the RMSE relative to the mean of the observed values. This enables a more careful comparison of model performance across various scenarios, which is particularly useful when dealing with structures or conditions that vary in size. Lastly, the normalized mean bias error (NMBE) in Equation (19) is used to evaluate the model's tendency for systematic bias, such as consistent overestimation or underestimation. These are computed by comparing PINN predictions against the results from the FEM (see Section 2), which is taken as the exact solution to the modeled plate. The comparison is performed at $n = 13$ locations. Nine of the locations are the same as the locations where sensors are assumed to be present; the coordinates for these are given in Table 2. The coordinates for the other four locations are: $(2.0,3.5), (3.5,2.0), (2.0,0.5), (0.5,2.0)$:

Table 4. *Summary of the measurement cases used in the study*

Cases	Sensor no.
Case 0	No measured data included
Case 1	S1
Case 2	S1–S5
Case 3	S1–S9

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Measurements_i - Model\ predictions_i)^2}, \tag{17}$$

$$CV = \frac{1}{MeasurementsMean}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Measurements_i - Model\ predictions_i)^2}, \tag{18}$$

$$NMBE = \frac{\sum_{i=1}^{n}(Model\ predictions_i - Measurements_i)}{\sum_{i=1}^{n}(Measurements_i)}. \tag{19}$$

3.2. Results and discussion

PINNs are trained for each of the scenarios described in Section 3, and the performance of the trained models are evaluated. Table 5 shows the results obtained for the different scenarios. The performance of the PINNs are compared in terms of various parameters such as training time and loss as well as RMSE, CV, and NMBE with respect to w, M_x, and M_y. Several observations can be made from the presented results.

The general trend is that the model performance improves as the loss function includes more terms to impose increasing number of physics-based constraints, that is, as one moves from SN1 to SN5. However, the training time tends to increase, with a notably steep increase for the loss function that also requires satisfaction of the governing PDE. For each of the loss function scenarios—namely SN1–SN5, performance is observed to improve with inclusion of more measurements. This is clearly as per expectations.

SN1, which represents a regular NN trained only using measurements, shows poor performance, which is in line with expectations. The training times for these models range roughly between 1 and 4 minutes. The loss values are all extremely small indicating that the model predicts accurately at the sensor locations used for model training. The RMSE for w varies significantly with the number of measurements used for training, with a value of 1.938 mm for Case 1 but reducing to 0.0219 mm for Case 3. The CV and NMBE for w also improve with increasing number of measurements. However, the RMSE, CV, and NMBE for M_x and M_y are excessively high, indicating a high level of variability and bias for these parameters.

SN2 is similar to SN1 but includes displacement boundary conditions, which essentially implies additional training data points across the four edges of the plate. When compared to SN1, models for SN2 offer more accurate predictions for displacements and moments. Training times are roughly 5 minutes with loss values that are marginally higher than for SN1. RMSE and CV for w are generally consistent and not varying as dramatically as in SN1. Predictions for M_x and M_y are better compared to SN1, but the errors, as indicated by RMSE, CV, and NMBE, are all still large; consequently, SN2 models are not usable for reliably predicting these parameters.

SN3 represents the scenario where the loss function is based on the errors in predicting w at the sensor locations as well as the errors in satisfying force boundary conditions. However, unlike SN2, displacement boundary conditions are not modeled. The training times for this model are similar to SN2, and the loss values are only slightly higher. However, when compared to models for SN1 and SN2, the predictions for w are less accurate as evident in the corresponding RMSE, CV, and NMBE metrics. In contrast, M_x and M_y are predicted more accurately with the corresponding error metrics being more consistent and balanced, indicating improved reliability and precision.

SN4 brings together the features in SN1–SN3. It incorporates displacement and force boundary conditions, which are important factors in physical structure behavior. However, the governing PDE is not included in the learning process of SN4; the expectation is that the monitoring data can overcome this limitation. The results demonstrate that the models become more accurate with increasing number of measurements. All metrics—RMSE, CV, and NMBE, for the three predicted parameters w, M_x, and M_y show improvement with increasing number of measurements, with results particularly for M_x and M_y significantly more accurate compared to predictions obtained using models from the earlier scenarios.

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

Table 5. *Error metric results for the various scenarios in Case Study 1*

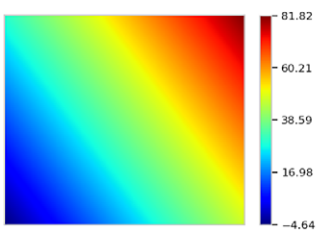
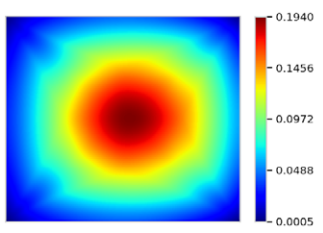
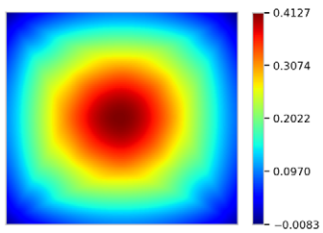
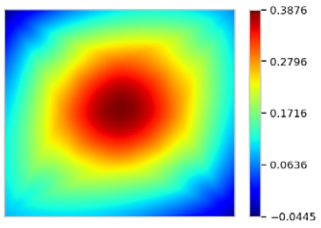
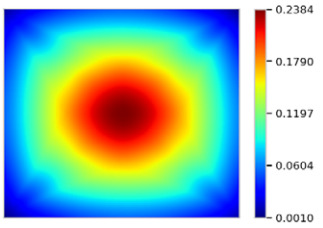
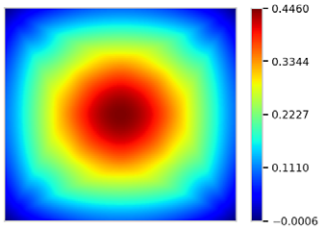
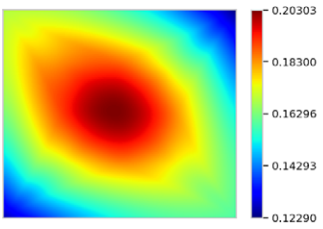
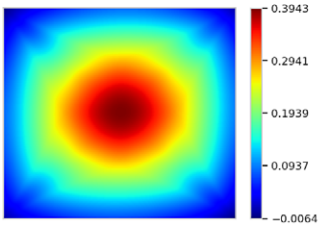
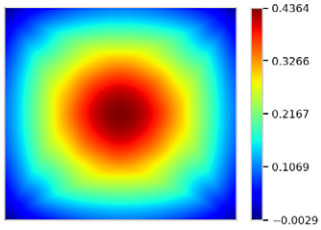
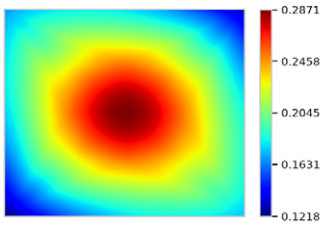
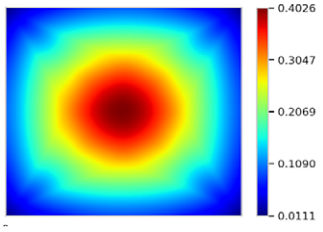
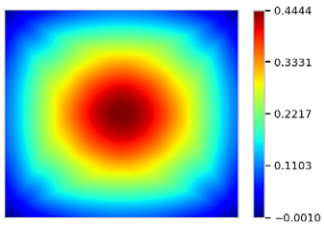
				Deflection (mm)			M_x (N.m/m)			M_y (N.m/m)		
SN	Cases	Training time	Loss	RMSE	CV	NMBE	RMSE	CV	NMBE	RMSE	CV	NMBE
SN1	Case 1	1 m 30s	1.5291e-19	1.938	917.41%	−158.96%	64,450	1,992.44%	−1,256%	60,203	1,861.11%	401%
	Case 2	2 m 50s	2.1539e-11	0.547	258.97%	−9.764%	38,590	1,192%	−1,046%	32,713	1,011%	987%
	Case 3	3 m 40s	5.2454e-12	0.0219	10.35%	−1.035%	28,044	867%	410%	17,616	544%	188%
SN2	Case 0	4 m 30s	5.1805e-09	0.16	76.16%	−19.51%	15,801	488%	75.55%	17,422	538%	222%
	Case 1	4 m 54 s	2.8829e-10	0.096	45.26%	38.22%	9,953	308%	−102%	11,214	347%	−120%
	Case 2	4 m 54 s	2.1066e-10	0.091	43.19%	35.52%	10,451	323%	−123%	5,067	157%	−47.28%
	Case 3	5 m 20s	3.0258e-10	0.092	43.71%	13.10%	10,673	330%	−106%	8,545	264%	−86.59%
SN3	Case 0	4 m 23 s	3.5790e-09	43.5	20,621%	−18,251%	2,242	69.33%	59.54%	2,250	69.58%	59.67%
	Case 1	4 m 39 s	4.4927e-09	0.056	26.66%	13.45%	695	21.51%	17.57%	1,019	31.50%	16.98%
	Case 2	4 m 41 s	4.8458e-10	0.139	66.13%	19.38%	3,310	102.32%	86.98%	3,284	101.53%	86.12%
	Case 3	5 m 19 s	7.5379e-09	0.10	50.93%	0.184%	2,577	79.68%	69.81%	2,575	79.63%	68.78%
SN4	Case 0	6 m 10s	7.0818e-10	0.154	72.84%	59.09%	2,148	66.40%	56.36%	2,094	64.75%	55.32%
	Case 1	6 m 54 s	4.2294e-09	0.128	60.63%	48.66%	1,738	53.74%	47.51%	1,756	54.28%	46.84%
	Case 2	6 m 54 s	1.6367e-08	0.046	21.73%	17.67%	513	15.86%	11.41%	730	22.58%	11.16%
	Case 3	7 m 20s	4.3398e-09	0.039	18.53%	13.20%	410	12.69%	5.00%	417	12.90%	6.00%
SN5	Case 0	1 hr 10 m	1.2270e-08	0.0288	15.73%	7.72%	520	16.29%	6.54%	500	15.73%	7.72%
	Case 1	2 hr 4 m	7.5788e-09	0.0089	4.25%	3.49%	531	16.42%	−1.01%	539	16.68%	−0.93%
	Case 2	1 hr 33 m	8.7162e-09	0.0189	6.55%	4.96%	500	15.48%	1.75%	493	15.26%	1.21%
	Case 3	2 hr 20 m	7.6921e-09	0.0097	4.61%	3.82%	528	16.33%	−0.33%	522	16.15%	−0.36%

Furthermore, the time required for training is only slightly more than the time needed for SN2 and SN3, while the overall loss is similar.

Scenario SN5 builds on SN4 by incorporating the loss term for satisfying the governing PDE. These models have the longest training time (up to 2 hours 20 minutes), but also produce the best results. The error metrics are noticeably more stable and consistent compared to the previous scenarios. For example, this has the lowest CV for deflection w. Models also offer good prediction accuracy for the moments M_x and M_y. Overall, SN5 models perform with the least amount of error and bias, and generally offer the most accurate predictions among the five models.

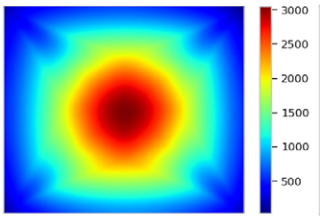
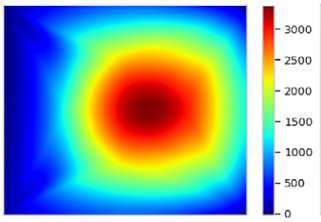
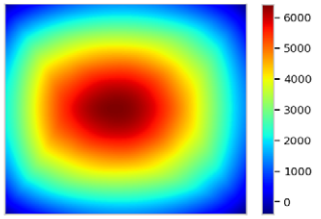
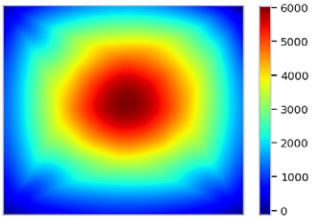
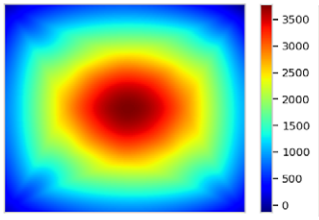
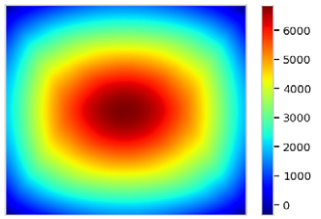
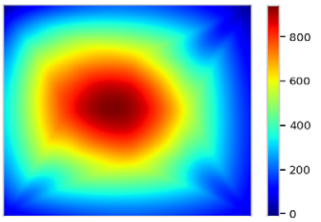
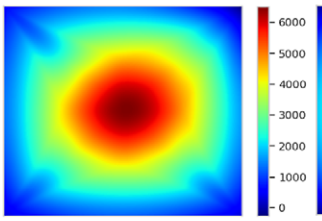
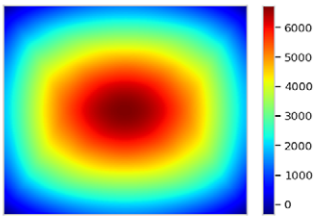
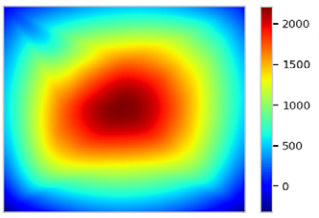
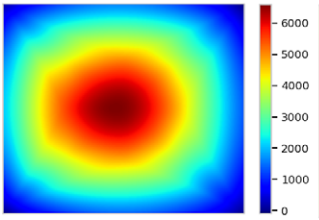
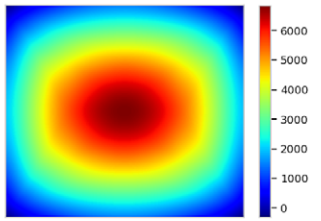
Models from SN3–SN5 offer significantly better performance than SN1 and SN2, particularly for Case 3, which has the most number of measurements. Consequently, we explore the generalization capability of these models in more detail. Tables 6 and 7 show the contour plots for w and M_x and M_y, respectively. The plots in these tables need to be compared with the contours from finite element analysis given in Figure 2.

Table 6. *Deflection predictions of physics-informed neural networks for Case Study 1*

	Scenarios		
w (mm)	SN3	SN4	SN5
Case 0			
Case 1			
Case 2			
Case 3			

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

Table 7. *Moment predictions (M_x) of physics-informed neural networks for Case Study 1*

	Scenarios		
M_x(N.m/m)	SN3	SN4	SN5
Case 0			
Case 1			
Case 2			
Case 3			

Notably, despite the omission of governing equations during the learning process, SN4 yields promising results. This is possibly due to the monitoring data, particularly with more sensors offering improved spatial coverage and hence being able to compensate for the lack of the loss term requiring satisfaction of the governing PDE. Consequently, SN4 may offer a simplified learning model that combines the advantages of both physical knowledge and data-driven learning, while being computationally modest. We will explore this further in the next section, where we focus primarily on how the learning model in SN4 can be adapted for scenarios where the loading is also allowed to vary.

4. Case Study 2

To predict for different UDL conditions, we modify the PINN to accept the load as an additional input. This shift toward considering UDL as an additional input closely reflects real-world scenarios where the

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

measured data are often obtained under varying load conditions. Implicitly, the premise is that the load is known, which is appropriate for practical monitoring scenarios.

To ensure the accuracy and robustness of the modified PINN's predictive capabilities, a loss term that imposes a constraint on how loading and deflection are related is incorporated into the existing loss function. In this study, a simple linear relationship between load and displacement is assumed; this implies that the second derivative of displacement with respect to load should ideally equal zero. This linear relationship is a fundamental assumption in numerous areas of structural analysis and engineering. Equation (20) shows the implementation of this loss. The loss is evaluated at a set of N_q points, wherein N_q is 278, 486, and 702, for Cases 1–3, respectively. These data represent deflection measurements for a total of 54 different load cases, spanning a range of 0 to 400 KN/m^2. It is noted that UDL values beyond 10 KN/m^2 are unrealistic but are used here solely to evaluate the generalization capability of the PINN models:

$$MSE_q = \frac{1}{N_q}\sum_{i=1}^{N_q} \| \frac{\partial^2 w}{\partial q^2}(x_i, y_i, q_i) \|^2. \tag{20}$$

The total modified loss function is shown in Equation (21):

$$\mathcal{L} = \alpha_w MSE_w + \alpha_m MSE_m + \alpha_d MSE_d + \alpha_q MSE_q. \tag{21}$$

The schematic in Figure 7 illustrates the NN's architecture modified to incorporate UDL as an additional input. The same PINN hyperparameters referenced in Table 2 are used with one notable exception. The number of epochs was increased to 50,000 to enhance prediction accuracy. However, it was observed that the early stopping mechanism typically activated between 20,000 and 30,000 epochs, thereby preventing potential over-fitting while ensuring model efficiency.

4.1. Impact of varying UDL on PINNs' predictions

Results for this PINN setup are given in Table 8, which sheds light on the performance of the trained PINNs for varying UDL conditions. To begin with, the predictive accuracy improves from Case 1 to Case 3, as evidenced by decreasing RMSE values for deflection and bending moments. For example, at a UDL of 150KN/m^2, the RMSE drops from 0.2741 mm in Case 1 to 0.1211 mm in Case 3. This decrease in RMSE can be attributed to the increased number of points N_q used for evaluating the loss function in each subsequent case, demonstrating that more comprehensive training data improve the model's effectiveness.

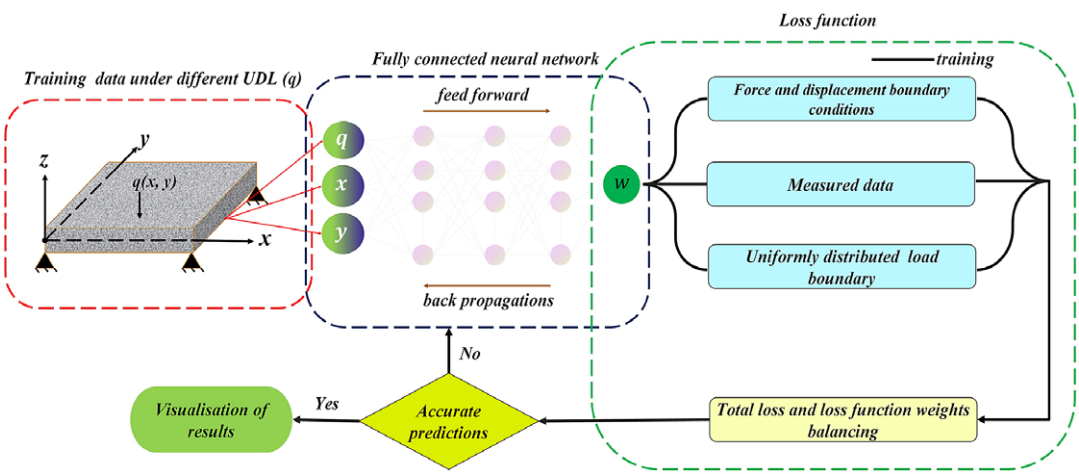


Figure 7. *Physics-informed neural networks setup for predicting response to varying uniformly distributed load.*

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

Table 8. *Error metric results for Case Study 2 with varying uniformly distributed load*

		w (mm)			M_x (N.m/m)			M_y (N.m/m)		
UDL (N/m^2)	Cases	RMSE	CV	NMBE	RMSE	CV	NMBE	RMSE	CV	NMBE
150,000	Case 1	0.2741	8.61%	−2.55%	15,165	27.68%	3.08%	12,031	21.96%	2.64%
	Case 2	0.1701	5.35%	−2.82%	7,131	13.02%	2.91%	9,801	17.89%	3.30%
	Case 3	0.1211	3.81%	−0.44%	8,478	15.47%	6.15%	8,302	15.15%	6.14%
450,000	Case 1	0.8132	8.84%	−5.45%	36,093	21.00%	8.68%	32,162	18.71%	8.68%
	Case 2	0.8251	8.97%	−6.58%	28,614	16.65%	7.79%	30,993	18.03%	7.89%
	Case 3	0.5382	5.85%	−3.89%	33,249	19.35%	11.05%	32,573	18.95%	10.96%
500,000	Case 1	0.5029	4.61%	1.27%	35,415	19.11%	5.95%	30,832	16.64%	5.96%
	Case 2	0.2437	2.23%	0.17%	27,290	14.73%	5.01%	30,112	16.25%	5.08%
	Case 3	0.4868	4.46%	2.80%	31,287	16.88%	8.41%	30,435	16.42%	8.34%
550,000	Case 1	0.6842	5.86%	−1.55%	55,686	25.41%	12.55%	52,431	23.92%	12.57%
	Case 2	0.6151	5.27%	−2.72%	49,598	22.63%	11.67%	52,058	23.75%	11.72%
	Case 3	0.4608	3.95%	−0.04%	55,186	25.18%	14.83%	53,991	24.63%	14.78%

Turning to the error metrics, RMSE and CV offer distinct but complementary perspectives on model performance. While RMSE provides an absolute measure of the prediction errors, CV normalizes these errors relative to the magnitude of the quantities being predicted. Consider the RMSE for deflection at a UDL of 150 KN/m^2 in Case 3, which is 0.1211 mm, and compare it with the deflection RMSE at a UDL of 550 KN/m^2 for the same case, which is 0.4608 mm. The RMSE increases as the load increases, but this is expected since the deflections themselves are larger under higher loads. The true strength of the model becomes apparent when considering the CV, which remains relatively stable (3.81%–3.95% for 150–550 KN/m^2) across these conditions. The CV helps to contextualize the RMSE by indicating that the percentage error in the predictions remains consistent, even as the load increases. Interestingly, the NMBE shows both positive and negative values (less than $\pm 7\%$), indicating that the model's predictions are neither consistently overestimated nor underestimated. This suggests that the additional loss term, MSE_q, aimed at ensuring a realistic relationship between load and deflection, is effective in mitigating systemic bias.

The performance of the model can also be understood through a careful analysis of the bending moment predictions,M_x and M_y. Taking Case 3 as an example, the RMSE for M_x is 8,478 N.m/m with a CV of 15.47% and the RMSE for M_y is 8,302 N.m/m with a CV of 15.15% when the UDL is 150 KN/m^2. In the same Case 3, the RMSE for M_x rises to 55,186 N.m/m and the RMSE for M_y to 53,991 N.m/m for a higher UDL of 550 KN/m^2, while the CV for M_x is 25.18% and for M_y it is 24.63%. The CV is thus an important indicator here, showing that the RMSE values are rising, but only in a proportional fashion to the rising bending moment magnitudes. This is further corroborated by the NMBE, which for M_x is 6.15% at 150 KN/m^2 and 14.83% at 550 KN/m^2. Similarly, for M_y, the NMBE is 6.14% at 150 KN/m^2 and 14.78% at 550 KN/m^2. These NMBE values suggest that the model maintains a relatively consistent bias across different loading scenarios.

Overall, this case study demonstrates the reliability of the PINN models for predicting floor slab behavior under varying loads. Remarkably, they also exhibit strong predictive capabilities for inputs beyond the range of training data, making it highly applicable for real-world SHM scenarios.

5. Case Study 3

This case study aims to explore the robustness of PINNs to uncertainties in modeling and measurement data. In structural engineering, connections are usually idealized as fully fixed or pinned for design

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

purposes. These idealized assumptions, while often conservative from a design perspective, may not be appropriate when interpreting measurements from real structures. Real-world connections tend to be semi-rigid, that is, with partially fixity with stiffness somewhere between a fully fixed connection and a pinned connection. The semi-rigid behavior could arise from imperfections in connections, for example, due to aging materials, but typically is a direct consequence of the way the connection is fabricated, as, for example, in the case of beam–column connections and truss element connections, which are commonly assumed to be perfectly pinned for modeling but research indicates that partial fixity is more realistic representation of their behavior (Kartal et al., 2010). Consequently, PINN models that are employed to interpret measurements from real-world conditions must be capable of handling deviations from idealized assumptions on boundary conditions. We evaluate this capability through simulated measurement data obtained from the FEM implementing semi-rigid connections along the plate boundaries.

SHM data are also subject to a degree of noise. Noise can be attributed to a range of environmental and operational issues, including but not limited to changes in temperature, live load variations, and inaccuracies in sensor measurements. To replicate this characteristic, we incorporate white Gaussian white noise into our simulated SHM data. The incorporation of this noise is intended to enhance the fidelity of the synthetic data, making it more reflective of the challenges engineers would face when analyzing SHM data collected in real-world scenarios. This case study thus considers two additional layers of complexity, namely semi-rigid connections and data affected by noise. Our aim is to evaluate the adaptability and performance of PINNs in two scenarios: with and without the presence of the PDEs that govern the system.

5.1. Synthetic semi-rigid connections data

The FEM of the plate used in Case Studies 1 and 2 is modified to emulate semi-rigid connections along its edges. The initial setup parameters for load distribution, geometric dimensions, and material properties are the same as defined in Section 2.2. However, specialized connection elements—Combin14 elements that serve as torsional springs (Ansys, 2023)—are employed to simulate semi-rigid behavior. These elements will offer restraint for rotation about the plate boundary. The stiffness (K) of the elements determines the degree of restraint with 0 representing a perfectly pinned connection and very large values offering fully fixed behavior. For the purposes of this case study, a stiffness value of 8,261,458 N · m was chosen. This value is derived from theoretical calculations assuming a reinforced concrete beam having dimensions 300 × 400 mm exists along the plate boundaries, as may be the case in a building floor. Table 9 provides a summary of the number and locations of these synthetic torsional springs.

Table 9. *Predictions from finite element model incorporating semi-rigid connection behavior along plate edges*

Sensor no.	Coordinates (x,y) (m)	Deflection (mm)	M_x (N.m/m)	M_y (N.m/m)
S1	(2,2)	0.3712	5,321	5,878
S2	(1,1)	0.1870	3,084	2,945
S3	(3,1)	0.1870	3,084	2,945
S4	(1,3)	0.1870	2,393	2,159
S5	(3,3)	0.1870	3,084	2,945
S6	(2,1)	0.2561	4,289	5,239
S7	(2,3)	0.2561	3,604	3,902
S8	(1,2)	0.2561	4,422	4,306
S9	(3,2)	0.2561	4,422	4,306

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

5.2. PINNs under varying signal-to-noise ratios

The purpose of this section is to evaluate the performance of PINNs under various noise levels evaluated in terms of signal-to-noise ratios (SNRs). White Gaussian noise is incorporated into the deflection data predicted by the FE model and presented in Table 9. The noisy data w_{noise} are formulated as in Equation (22)):

$$w_{\text{noise}} = w + \mathcal{N}\left(0, \sigma^2\right). \tag{22}$$

Here, w represents the actual displacement acquired from the FEM, and $\mathcal{N}(0,\sigma^2)$ signifies white Gaussian noise with zero mean and a standard deviation σ determined by the desired SNR. The SNR in decibels (dB) is expressed as in Equation (23)):

$$SNR(dB) = 10\log_{10}\left(\frac{\text{Signal Power}}{\text{Noise Power}}\right). \tag{23}$$

Four different SNRs are considered: 10, 20, 30, and 40 dB as presented in Figure 8. A lower SNR such as 10 dB implies a high level of noise, posing a significant challenge for any predictive model. Conversely, a higher SNR like 40 dB indicates a minimal effect of noise.

5.3. Results and discussion

PINNs are trained under two distinct scenarios: SN4, which excludes PDEs, and SN5, which incorporates them. The efficacy of these trained models is evaluated across a range of SNRs. Table 10 gives the

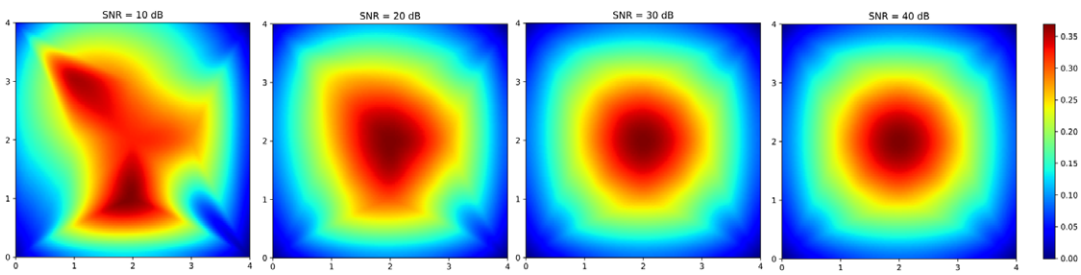


Figure 8. *Effect of signal-to-noise ratio on the quality of deflection w_{noise} data.*

Table 10. *Error metrics for the Case Study 3 scenarios*

		w (mm)			M_x (N.m/m)			M_y (N.m/m)		
SN	SNR	RMSE	CV	NMBE	RMSE	CV	NMBE	RMSE	CV	NMBE
SN4	10 dB	0.0443	26.87%	3.07%	1,150	43%	33%	1,182	45%	30.32%
	20 dB	0.0312	18.93%	6.33%	798	29.85%	23.93%	808	31%	20.24%
	30 dB	0.0147	8.91%	1.37%	876	32.78%	5.61%	639	24.52%	4.06%
	40 dB	0.0116	7.05%	0.80%	596	22.28%	4.28%	484	18.59%	1.67%
	—	0.0068	4.11%	0.56%	804	29.62%	11.97%	778	30.65%	−8.27%
SN5	10 dB	0.0245	14.86%	−4.72%	719	26.88%	−13.59%	723	27.73%	−16.45%
	20 dB	0.0178	10.78%	−3.74%	743	27.78%	−13.80%	729	27.96%	−16.47%
	30 dB	0.0132	8.03%	−3.87%	383	14.31%	4.50%	478	18.32%	0.37%
	40 dB	0.0128	7.80%	1.02%	409	15.29%	5.42%	407	15.60%	1.89%
	—	0.0092	5.56%	0.65%	308	11.55%	7.25%	380	14.6%	4.687%

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

outcomes for both scenarios, employing multiple error metrics, as already done in Case Study 1, for the predicted parameters w, M_x, and M_y. The data yield several key insights.

For the SN4 scenario, the results reveal that the model accuracy improves as the noise level decreases. All error metrics exhibit an improvement trend with diminishing noise for the three parameters w, M_x, and M_y. Specifically, the CV decreases from 26.87% at a 10 dB SNR to 4.11% with clean data. However, the moments M_x and M_y are more sensitive to noise compared to deflection. For instance, the RMSE nearly doubles when the SNR changes from 40 to 10 dB. The deflection NMBE generally remains within a $\pm 10\%$ range and is mostly positive, indicating underestimation compared to the measured data. This suggests that the PINNs in SN4 provide a robust generalization, even when faced with substantial noise. The NMBE for M_x and M_y follows the same trend as deflection, with values reaching up to 33%.

In the SN5 scenario, the models outperform those in SN4, even in noisier conditions. The CV for deflection ranges from just below 6% to 15% for clean data and a 10 dB SNR, respectively. The moments M_x and M_y are less affected by noise in SN5 compared to SN4. For instance, the RMSE values are 719 and 723 N.m/m for M_x and M_y at a 10 dB SNR, compared to 409 and 407 N.m/m at a 40 dB SNR. Remarkably, the NMBE for deflection remains below $\pm 5\%$ in all tested cases. It turns negative in noisier conditions, implying that the predicted values surpass the measured ones. This behavior suggests that as the data become noisier, the PINNs in the SN5 scenario adapt their predictions and rely more on the PDEs, thereby achieving more accurate and generalized outcomes. This trend is also observed for M_x and M_y, although the NMBE can extend up to -17%.

In summary, Case Study 3 helps in assessing the robustness of the PINNs to deviations from idealized boundary conditions and noisy data. The results show that including PDEs improves accuracy and reliability, especially under noisy conditions. The study also emphasizes that, while deflection is less sensitive to noise in general, moments (M_x) and (M_y) can be significantly affected.

6. Conclusions

Using three case studies, this research work investigates the influence of different loss function terms modeling physics-based constraints on the performance of the resulting PINN models. The first case study considers a constant load and investigates five distinct loss function configurations. Each loss function configuration is assessed across four measurement scenarios, which correspond to increasing numbers of measurement data obtained from a simulated FEM. The second case study uses the loads also as an input variable for the PINN, which is also informed by the structure's boundary conditions and synthetic sensor data. In the third case study, we assessed the adaptability and predictability of PINNs within a modeling context that incorporates semi-rigid connections and noisy data. The following points summarize the main study outcomes:

1. The PINNs that implement loss terms for physics-based constraints related to displacement and force boundary conditions and governing PDE as well as for matching measured data accurately predict deflection (CV = 4.25%) and bending moments (CV $\approx$ 16%).
2. The PINNs that implement force and displacement boundary conditions and satisfy monitoring data while not requiring satisfaction of the governing PDE show significant promise as they accurately predict deflections (CV $\approx$ 20%) and moments (CV $\approx$ 13%). These indicate strong potential for generalization and have the advantage of needing less computational time for training than with PINNs with loss terms for governing PDE.
3. The PINNs for scenarios with loss functions considering force and displacement boundary conditions or governing PDE are capable of offering virtual sensing capability, that is, they reliably predict internal force quantities which are difficult to measure in real structures.
4. The PINNs demonstrate remarkable predictive capabilities for inputs outside of the range of training data, which is essential for real-world monitoring scenarios.
5. The PINNs that comply with the PDEs are able to effectively deal with noisy data and deviations from idealized boundary conditions. In Case Study 3, for SN5 with loss function term for PDEs, the

https://doi.org/10.1017/dce.2024.4 Published online by Cambridge University Press

CV for deflection predictions is $\approx 14\%$ for an SNR of 10 dB, and $\approx 5.00\%$ for an SNR of 40 dB, whereas, in SN4 with the absence of PDEs, the CV begins at a higher value of approximately $\approx 26.00\%$ for 10 dB and decreases to approximately $\approx 4.00\%$ for 40 dB.

Overall, this research contributes to the field of SHM by investigating insights into the role of structural boundary conditions and measured data in training PINNs. This investigation enhances the understanding of model performance and highlights the importance of undertaking more detailed case studies in SHM applications.

Data availability statement. A demonstrative code that implements the proposed method is openly available through the public GitHub repository at https://github.com/AnmarAl-Adly/PINNs_FOR_SHM.git. The data used throughout this paper were synthetically generated. The authors have ensured to provide sufficient information to replicate the analytical and numerical models used to generate the synthetic data.

Author contribution. Conceptualization, methodology, and validation: both authors; Data visualization, formal analysis, and writing—original draft: A.I.F.A.-A. Both authors approved the final submitted draft.

Funding statement. The first author would like to acknowledge the financial support of The Higher Committee for Education Development in Iraq (HCED) scholarship reference D-14-2968.

Competing interest. The authors declare no competing interests exist.

Ethical standard. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

References

Abadi M, **Agarwal A**, **Barham P**, **Brevdo E**, **Chen Z**, **Citro C**, **Corrado GS**, **Davis A**, **Dean J**, **Devin M**, **Ghemawat S**, **Goodfellow I**, **Harp A**, **Irving G**, **Isard M**, **Jozefowicz R**, **Jia Y**, **Kaiser L**, **Kudlur M**, **Levenberg J**, **Mane D**, **Monga R**, **Moore S**, **Murray D**, **Olah C**, **Schuster M**, **Shlens J**, **Steiner B**, **Sutskever I**, **Talwar K**, **Tucker P**, **Vanhoucke V**, **Vasudevan V**, **Viegas F**, **Vinyals O**, **Warden P**, **Wattenberg M**, **Wicke M**, **Yu Y and Zheng X** (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Available at https://www.tensorflow.org/.

Ansys (2023). ANSYS Worckbench, Release Number 2023 R1, Help System, Theory Guide, ANSYS, Inc.

Bischof R and Kraus M (2022) Multi-objective loss balancing for physics-informed deep learning. https://doi.org/10.13140/RG.2.2.20057.24169

Bud MA, **Moldovan ID**, **Nedelcu M and Figueiredo E** (2023) Hybrid training of supervised machine learning algorithms for damage identification in bridges. In Rizzo P and Milazzo A (eds), *European Workshop on Structural Health Monitoring (EWSHM 2022)*. Lecture Notes in Civil Engineering, vol. 270. Cham: Springer, pp. 482–491. https://doi.org/10.1007/978-3-031-07322-9_49

Bud MA, **Nedelcu M**, **Radu L**, **Moldovan I and Figueiredo E** (2019) On the reliability of finite element models for the training of machine learning algorithms for damage detection in bridges. *Structural Health Monitoring 14*(4), S97–S97. https://doi.org/10.12783/shm2019/32204

Cousins DP, **McAra D and Hill C** (2022) Monitoring of the Queensferry Crossing, Scotland. In *Proceedings of the Institution of Civil Engineers: Bridge Engineering*. https://doi.org/10.1680/jbren.22.00018

Cross EJ, **Gibson SJ**, **Jones MR**, **Pitchforth DJ**, **Zhang S and Rogers TJ** (2022) Physics-informed machine learning for structural health monitoring. *Structural Integrity 21*, 347–367. https://doi.org/10.1007/978-3-030-81716-9_17

Cuomo S, **Di Cola VS**, **Giampaolo F**, **Rozza G**, **Raissi M and Piccialli F** (2022) Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing 92*(3), 1–62. https://doi.org/10.1007/s10915-022-01939-z

Farrar CR and Worden K (2012) *Structural Health Monitoring: A Machine Learning Perspective*. Hoboken, NJ: Wiley. https://doi.org/10.1002/9781118443118

Figueiredo E and Brownjohn J (2022) Three decades of statistical pattern recognition paradigm for SHM of bridges. *Structural Health Monitoring 21*(6), 3018–3054. https://doi.org/10.1177/14759217221075241

Fremmelev MA, **Ladpli P**, **Orlowitz E**, **Bernhammer LO**, **Mcgugan M and Branner K** (2022) Structural health monitoring of 52-meter wind turbine blade: Detection of damage propagation during fatigue testing. *Data-Centric Engineering 3*, e22. https://doi.org/10.1017/dce.2022.20

Goodfellow IJ, **Bengio Y and Courville A** (2016) *Deep Learning*. Cambridge, MA: MIT Press. https://www.deeplearningbook.org/

Kabasi S, **Marbaniang AL and Ghosh S** (2023) Physics-informed neural networks for the form-finding of tensile membranes by solving the Euler–Lagrange equation of minimal surfaces. *Thin-Walled Structures 182*, 110309. https://doi.org/10.1016/j.tws.2022.110309

Kamariotis A, **Chatzi E and Straub D** (2022) Value of information from vibration-based structural health monitoring extracted via Bayesian model updating. *Mechanical Systems and Signal Processing 166*, 108465. https://doi.org/10.1016/j.ymssp.2021.108465

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

Kartal ME, **Basaga H**, **Bayraktar A and Muvafık M** (2010) Effects of semi-rigid connection on structural responses. *Electronic Journal of Structural Engineering 10*, 22–35. https://doi.org/10.56748/ejse.10122

Kaszynski A, **Derrick J**, **Kaszynski A**, **Ans F**, **Jleonatti S**, **Correia D**, **Addy D and Guyver J** (2021) pyansys/pymapdl: v0.60.1 (v0.60.1). Zenodo. https://doi.org/10.5281/ZENODO.5726008

Kechavarzi C, **Soga K**, **de Battista N**, **Pelecanos L**, **Elshafie MZEB and Mair RJ** (2016) *Distributed Fibre Optic Strain Sensing for Monitoring Civil Infrastructure—A Practical Guide*. London: ICE Publishing. https://doi.org/10.1680/dfossmci.60555

Kirchhoff GR (1850) Über das Gleichgewicht und die Bewegung einer elastischen Scheibe [On the equilibrium and the motion of an elastic plate]. *Journal fur die Reine und Angewandte Mathematik 40*, 51–88.

Kromanis R and Kripakaran P (2014) Predicting thermal response of bridges using regression models derived from measurement histories. *Computers & Structures 136*, 64–77. https://doi.org/10.1016/j.compstruc.2014.01.026

Kromanis R and Kripakaran P (2016) SHM of bridges: Characterising thermal response and detecting anomaly events using a temperature-based measurement interpretation approach. *Journal of Civil Structural Health Monitoring 6*(2), 237–254. https://doi.org/10.1007/S13349-016-0161-Z

Lai Z, **Liu W**, **Jian X**, **Bacsa K**, **Sun L and Chatzi E** (2022) Neural modal ordinary differential equations: Integrating physics-based modeling with neural ordinary differential equations for modeling high-dimensional monitored structures. *Data-Centric Engineering 3*, e34. https://doi.org/10.1017/dce.2022.35

Lai Z, **Mylonas C**, **Nagarajaiah S and Chatzi E** (2021) Structural identification with physics-informed neural ordinary differential equations. *Journal of Sound and Vibration 508*, 116196. https://doi.org/10.1016/j.jsv.2021.116196

Liu W, **Lai Z**, **Bacsa K and Chatzi E** (2022) Physics-guided deep Markov models for learning nonlinear dynamical systems with uncertainty. *Mechanical Systems and Signal Processing 117*, 109276. https://doi.org/10.1016/j.ymssp.2022.109276

Liu GR and Quek SS (2014) *The Finite Element Method: A Practical Course*, 2nd Edn. Oxford and Waltham, MA: Butterworth-Heinemann. https://doi.org/10.1016/C2012-0-00779-X

Love AEH (1888) The small free vibrations and deformation of a thin elastic shell. *Philosophical Transactions of the Royal Society A 179*, 491–546.

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

Middleton CR, **Fidler PR and Vardanega PJ** (2016) *Bridge Monitoring: A Practical Guide*, 1st Edn. London: Thomas Telford (ICE Publishing.

Mowlavi S and Nabi S (2023) Optimal control of PDEs using physics-informed neural networks. *Journal of Computational Physics 473*, 111731. https://doi.org/10.1016/j.jcp.2022.111731

Ni P, **Sun L**, **Yang J and Li Y** (2022) Multi-end physics-informed deep learning for seismic response estimation. *Sensors 22*(10), 1–23. https://doi.org/10.3390/s22103697

Ni Y, **Wang Y and Zhang C** (2020) A Bayesian approach for condition assessment and damage alarm of bridge expansion joints using long-term structural health monitoring data. *Engineering Structures*, *212*, 110520. https://doi.org/10.1016/j.engstruct.2020.110520

Oncescu AM and Cicirello A (2023) A self-supervised classification algorithm for sensor fault identification for robust structural health monitoring. In Rizzo P and Milazzo A (eds), *European Workshop on Structural Health Monitoring (EWSHM 2022)*, Lecture Notes in Civil Engineering, Vol. *253*. Cham: Springer, pp. 564–574. https://doi.org/10.1007/978-3-031-07254-3_57

Raissi M, **Perdikaris P and Karniadakis GE** (2017a) Inferring solutions of differential equations using noisy multi-fidelity data. *Journal of Computational Physics 335*, 736–746. https://doi.org/10.1016/j.jcp.2017.01.060

Raissi M, **Perdikaris P and Karniadakis GE** (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics 378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Raissi M, **Yazdani A and Karniadakis GE** (2017b) Hidden fluid mechanics: A Navier–Stokes informed deep learning framework for assimilating flow visualization data. Preprint. http://arxiv.org/abs/1808.04327

Raissi M, **Yazdani A and Karniadakis GE** (2020) Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science 367*(6481), 1026–1030.

Sacks R, **Brilakis I**, **Pikas E**, **Xie HS and Girolami M** (2020) Construction with digital twin information systems. *Data-Centric Engineering 1*, e14. https://doi.org/10.1017/dce.2020.16

Sankararaman S and Mahadevan S (2013) Bayesian methodology for diagnosis uncertainty quantification and health monitoring. *Structural Control and Health Monitoring 20*(1), 88–106. https://doi.org/10.1002/stc.476

Sholokhov A, **Liu Y**, **Mansour H and Nabi S** (2023) Physics-informed neural ODE (PINODE): Embedding physics into models using collocation points. *Scientific Reports 13*(1), 10166. https://doi.org/10.1038/s41598-023-36799-6

Shukla K, **Xu M**, **Trask N and Karniadakis GE** (2022) Scalable algorithms for physics-informed neural and graph networks. *Data-Centric Engineering 3*, e24. https://doi.org/10.1017/dce.2022.24

Sliwinski L and Rigas G (2023) Mean flow reconstruction of unsteady flows using physics-informed neural networks. *Data-Centric Engineering 4*, e4. https://doi.org/10.1017/dce.2022.37

Sohn H, **Farrar CR**, **Hemez F and Czarnecki J** (2002) *A Review of Structural Health Monitoring Literature 1996–2001*. Technical report. Available at http://www.doe.gov/bridge.

Timoshenko S and Woninowsky-Kerieger S (1959) *Theory of Plates and Shells*, 2nd Edn. New York: McGraw-Hill.

Uddin Z, **Ganga S**, **Asthana R and Ibrahim W** (2023) Wavelets-based physics-informed neural networks to solve non-linear differential equations. *Scientific Reports 13*(1), 1–19. https://doi.org/10.1038/s41598-023-29806-3

Wang T, **Barthorpe RJ**, **Wagg DJ and Worden K** (2022) On improved fail-safe sensor distributions for a structural health monitoring system. *Data-Centric Engineering 3*, e27. https://doi.org/10.1017/dce.2022.27

Webb GT, **Vardanega PJ and Middleton CR** (2015) Categories of SHM deployments: Technologies and capabilities. *Journal of Bridge Engineering 20*(11), 04014118. https://doi.org/10.1061/(asce)be.1943-5592.0000735

Worden K, **Cross EJ**, **Dervilis N**, **Papatheou E and Antoniadou I** (2015) Structural health monitoring: From structures to systems-of-systems. *IFAC-PapersOnLine 48*(21), 1–17. https://doi.org/10.1016/j.ifacol.2015.09.497

Xu Y, **Kohtz S**, **Boakye J**, **Gardoni P and Wang P** (2023) Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges. *Reliability Engineering and System Safety 230*, 108900. https://doi.org/10.1016/j.ress.2022.108900

Yuan F-G, **Zargar SA**, **Chen Q and Wang S** (2020) Machine learning for structural health monitoring: Challenges and opportunities. In Zonta D, Sohn H and Huang H (eds), *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*. Bellingham, WA: SPIE, p. 2. https://doi.org/10.1117/12.2561610

Zheng B, **Li T**, **Qi H**, **Gao L**, **Liu X and Yuan L** (2022) Physics-informed machine learning model for computational fracture of quasi-brittle materials without labelled data. *International Journal of Mechanical Sciences 223*, 107282. https://doi.org/10.1016/j.ijmecsci.2022.107282

Zobeiry N and Humfeld KD (2021) A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Engineering Applications of Artificial Intelligence 101*, 104232. https://doi.org/10.1016/j.engappai.2021.104232

Cite this article: Al-Adly AIF and Kripakaran P (2024). Physics-informed neural networks for structural health monitoring: a case study for Kirchhoff–Love plates. *Data-Centric Engineering*, 5, e6. doi:10.1017/dce.2024.4