Computer Methods in Applied Mechanics and Engineering 436 (2025) 117650

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

Deep learning for inverse material characterization

Yousef Ghaffari Motlagh [a,b,d,*], Farshid Fathi [b], John C. Brigham [c], Peter K. Jimack [d]

[a] School of Built Environment, Engineering and Computing, Leeds Beckett University Headingley Campus, Leeds, LS6 3QR, UK
[b] School of Mechanical, Aerospace and Civil Engineering, University of Sheffield, Sheffield, S1 3JD, UK
[c] Department of Civil and Environmental Engineering, University of Pittsburgh, Pittsburgh, PA 15261, United States
[d] School of Computer Science, University of Leeds, Leeds, LS2 9JT, UK

ARTICLE INFO

Keywords:
Inverse problem
Direct inversion
PINNs
Material characterization

ABSTRACT

This paper presents an approach for computationally efficient inverse material characterization using Physics-Informed Neural Networks (PINNs) based on partial-field response measurements. PINNs reconstruct the full spatial distribution of the system's response from the measured portion of the response field and estimate the spatial distribution of unknown material properties. The primary computational expense in this approach is the one-time generation of potential responses for the PINNs, resulting in significant computational efficiency. Furthermore, this study utilizes PINNs to train a model based on the underlying physics described by differential equations, and to quantify *aleatoric* uncertainty arising from noisy data. We demonstrate several one-dimensional and two-dimensional examples where the elastic modulus distribution is characterized based on static partial-field displacement response measurements. The inversion procedure efficiently provides accurate estimates of material property distributions, showcasing the potential of PINNs in practical applications.

1. Introduction

Inverse problems are commonly encountered in a wide range of applications, such as medical imaging, geophysics, and nondestructive testing of materials. For instance, structures, from industrial to biological, could be evaluated to determine their current state of health based upon their material properties, whether mechanical, thermal, or electrical. Corresponding to this substantial interest in material property characterization, a wide variety of inverse problem solution strategies have been developed relating to a variety of applications, such as structural health monitoring and nondestructive evaluation [1,2] and biomechanical imaging [3–7], among other applications [8,9].

In many practical inverse problems, it is not possible to find analytical solutions due to their inherent complexity or ill-posedness. As a result, computational inverse characterization methods have become increasingly popular for these problems. Overall, computational inverse characterization approaches, which are typically based around some type of computational representation of the mechanics of the system of interest (e.g., finite element analysis), have been shown through several studies [3,5,9] to provide generalized frameworks for treating and distinguishing between various contributions to a system response, while providing physically meaningful solutions that can be applied to predict future behaviours. In the study by Brigham and et al. [3], a novel methodology was introduced for the inverse estimation of material properties in vibrating solids based on acoustic emission. This

* Corresponding author at: School of Built Environment, Engineering and Computing, Leeds Beckett University Headingley Campus, Leeds, LS6 3QR, UK.
E-mail addresses: J.Ghaffari-Motlagh@leedsbeckett.ac.uk (Y. Ghaffari Motlagh), F.fathi@sheffield.ac.uk (F. Fathi), brigham@pitt.edu (J.C. Brigham), P.K.Jimack@leeds.ac.uk (P.K. Jimack).

https://doi.org/10.1016/j.cma.2024.117650
Received 12 July 2024; Received in revised form 3 December 2024; Accepted 4 December 2024
Available online 1 January 2025

approach combines the classical random search algorithm with a surrogate model-accelerated random search (SMARS) algorithm, enhancing the efficiency of the search process. Additionally, Guzina and Bonnet [9] developed a boundary integral method to derive an explicit formula for the topological derivative. This formula is articulated through the elastodynamics fundamental solution, achieved by an asymptotic expansion of a misfit-type cost functional concerning the creation of an infinitesimal hole in an elastic medium, whether semi-infinite or infinite. However, the landscape of computational approaches to inverse problems is as diverse as the applications they aim to solve. Each method presents distinct advantages and limitations, and their effectiveness is largely contingent on the specific requirements of the application at hand. Moreover, these computational techniques often involve a significant trade-off between computational efficiency and the accuracy of the solution.

Although mathematical and practical data assimilation efforts have expanded significantly, the spatiotemporal heterogeneity of available data and the absence of universally reliable models highlight the necessity for a transformative approach to inverse problem solution estimation. Such an approach would seamlessly integrate multi-fidelity data into existing physical models [10]. Machine learning (ML) can be used to explore massive design spaces, identify multi-dimensional correlations and manage ill-posed problems [10]. In particular,in many application domains, deep learning approaches can provide tools for naturally extracting features from massive amounts of multi-fidelity observational data that are currently available and characterized by unprecedented spatial and temporal coverage [11]. Deep learning empowers overparametrised neural networks, equipped with multiple layers, to progressively extract higher-level features from raw input data. While these networks excel in supervised learning tasks demanding ample labelled training data, their effectiveness diminishes in scenarios where data is sparse or its collection is prohibitively expensive, as often encountered in engineering applications [12]. In such contexts, it becomes imperative to employ methods that exhibit reduced data dependency, prioritizing the training of deep learning models based on constraints, particularly those derived from physical laws, rather than relying solely on data.

Physics-Informed Neural Networks (PINNs) [10,13] have emerged as a promising solution. PINNs possess the unique ability to harness mathematical models that describe the inherent laws of physics, enabling them to discern patterns within high-dimensional data stemming from experimental observations. However, Krishnapriyan et al. [14] have highlighted potential challenges and reported limitations associated with PINNs. Their work emphasizes that PINNs currently face hurdles to rival traditional numerical methods in terms of both accuracy and computational cost. PINNs have been employed to infer properties of biological materials using synthetic data [15] and to calibrate material models from full-field displacement data [16]. Additionally, PINNs have been used to address identification challenges in nonhomogeneous materials, as demonstrated by Zhang et al. [17]. They successfully identified heterogeneous, incompressible, hyper-elastic materials using full-field displacement data through two distinct artificial neural networks (ANNs): one ANN approximated the displacement field, while the other estimated the spatially dependent material parameters.

In this study, we utilize PINNs to address inverse material characterization challenges efficiently using partial-field response measurements. We demonstrate that PINNs can reconstruct the full spatial distribution of a system's response from merely a segment of the response field, estimate unknown material properties, and train models based on the physics described by differential equations. Additionally, we demonstrate that PINNs are effective in quantifying *aleatoric* uncertainty arising from noisy data. The paper is organized as follows: Section 2 introduces the formulation of PINNs and their application to inverse problems. Section 3 discusses the PINNs framework for uncertainty quantification. In Section 4, we present our numerical studies, including results for both 1-D and 2-D cases under conditions of noise and uncertainty. The study concludes in Section 5 with a summary of our findings.

2. Physics informed neural networks for elastic problems

In this section, we present the PINNs formulation for the elastostatic forward and inverse problem. Let $\Omega \subset \mathbb{R}^d, d = 2, 3$, denote the domain occupied by the body, and let Γ_D and Γ_N be its Dirichlet and Neumann boundary respectively. Then in the current configuration

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f} = \boldsymbol{0} \quad \text{in} \quad \Omega, \tag{1}$$

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on} \quad \Gamma_D, \tag{2}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{h} \quad \text{on} \quad \Gamma_N, \tag{3}$$

where $\boldsymbol{\sigma}$ denotes the symmetric Cauchy stress tensor and $\boldsymbol{f}$ is the body force per unit volume, $\boldsymbol{u}$ is the displacement, $\boldsymbol{g}$ and $\boldsymbol{h}$ are referred to as prescribed boundary displacements and tractions, respectively.

We consider the strong form of Eqs. (1) and assume that $\mathcal{F}_{NN}(\boldsymbol{z}; \boldsymbol{W}, \boldsymbol{b})$ is a Neural Network (NN) approximation of the displacements, $\boldsymbol{u}$ in (1), (2), and (3). A NN is comprised of ℓ hidden layers with $\mathcal{N}_i$ neurons in each layer and activation functions $\mathcal{A}$ that reads

$$\mathcal{F}_{NN}(\boldsymbol{z}; \boldsymbol{W}, \boldsymbol{b}) = \mathcal{K} \circ \mathcal{T}^{(\ell)} \circ \mathcal{T}^{(\ell-1)} \circ \cdots \circ \mathcal{T}^{(1)}(\boldsymbol{z}), \tag{4}$$

where $\mathcal{K} : \mathbb{R}^{\mathcal{N}_\ell} \longrightarrow \mathbb{R}^d$ is the linear mapping in the output and d is the spatial dimension; $\mathcal{T}^i(*) = \mathcal{A}\left(\boldsymbol{W}^i \times * + \boldsymbol{b}^i\right)$ is the nonlinear mapping in each hidden layer $i = 1, 2, 3, \ldots, \ell$. Note that $\boldsymbol{W}^i$, $\boldsymbol{b}^i$ are the weights and biases. The strong-form residual r^k for the elastostatic equation, and the boundary residual r_b^k can be defined as

$$r^k(\hat{\boldsymbol{u}}) = \frac{\partial \sigma_{kj}}{\partial x_j} + f_k \qquad k = 1, \ldots, d \tag{5}$$

$$r_b^k(\hat{\boldsymbol{u}}) = \hat{u}_k - g_k \tag{6}$$

where $\hat{\boldsymbol{u}}$ is neural network approximation of the displacements. Now, in order to construct the variational forms of the problem, the weighted integral of the residuals can be defined by mapping them onto properly chosen space of test function $\mathcal{V}$; and then set them to zero. We choose the test functions v_q^k $(q = 1, \dots, N_{ru})$ for (5) and v_p^k $(p = 1, \dots, N_{bu})$ for (6) receptively and then impose

$$\mathcal{R}_q^k(\hat{\boldsymbol{u}}) = \int_\Omega r^k(\hat{\boldsymbol{u}}) v_q^k \mathrm{dV} = 0, \tag{7}$$

$$\mathcal{R}_{b,p}^k(\hat{\boldsymbol{u}}) = \int_{\partial\Omega} r_b^k(\hat{\boldsymbol{u}}) v_p^k \mathrm{dA} = 0. \tag{8}$$

The following minimization problem to determine the displacement throughout the domain can be formulated in place of solving the nonlinear systems resulting from the above equations:

$$\min_{\boldsymbol{W},\boldsymbol{b}} \mathcal{J}(\hat{\boldsymbol{u}}, \boldsymbol{v}) \tag{9}$$

$$\mathcal{J}(\hat{\boldsymbol{u}}, \boldsymbol{v}) = \sum_{k=1}^{d} \left(\sum_{j=1}^{N_{ru}} \left(\mathcal{R}_j^k(\hat{\boldsymbol{u}}) \right)^2 + \tau_b \sum_{s=1}^{N_{bu}} \left(\mathcal{R}_{b,s}^k(\hat{\boldsymbol{u}}) \right)^2 \right) \tag{10}$$

In (10), the first term represents the weighted integral of the elastostatic residual and the last term indicates the weighted integral of the displacement boundary condition. N_{ru} is the number of test functions corresponding to the elastostatic residual. N_{bu} is the number of collocation points corresponding the displacement essential boundary conditions. Furthermore, τ_b is a penalty parameter that represents the weight coefficient in the loss function and may be user-specified or tuned manually or automatically [18], e.g., based on the numerical experiment in each problem. The optimal bound of the penalty parameter, however, is still an open problem in the literature [19].

Now, we derive standard PINNs [13] by starting from the variational form, (9). To this end, we assume each of the test functions to be a Dirac Delta function, $v^k(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_r)$, so that $\boldsymbol{x}_r$ are the collocation points for the elastostatic equation. In essence, by means of this test function we can project the residuals onto a finite set of collocation points and enforce the equation to be satisfied at these points. The loss function for the PINNs reads

$$\mathcal{L}_{PINNs} = \sum_{k=1}^{d} \left[\frac{1}{N_r} \sum_{i=1}^{N_r} \left| r^k(\boldsymbol{x}_r^i) \right|^2 + \tau_b \frac{1}{N_b} \sum_{i=1}^{N_b} \left| r_b^k(\boldsymbol{x}_b^i) \right|^2 \right] \tag{11}$$

where $\{\boldsymbol{x}_r^i\}_{i=1}^{N_r}$, and $\{\boldsymbol{x}_b^i\}_{i=1}^{N_b}$ are collocation points in their domains. Note that N_r is the number of collocation points for the elastostatic equation. In addition, N_b is the number of collocation points corresponding the displacement essential boundary conditions. The loss function in (11) does not incorporate any observed data, excluding the Dirichlet boundary. To account for observed data, specifically displacement, the loss function is modified as shown in (12):

$$\mathcal{L}_{PINNs\ Data} = \mathcal{L}_{PINNs} + \sum_{k=1}^{d} \left[\tau_{obs} \frac{1}{N_{obs}} \sum_{i=1}^{N_{obs}} \left| \hat{u}^k(\boldsymbol{x}_{obs,i}) - \bar{u}_{obs,i}^k \right|^2 \right] \tag{12}$$

where $\{\bar{\boldsymbol{u}}_{obs,i}\}_{i=1}^{N_{obs}}$ represent the observed displacements, τ_{obs} denotes a penalty parameter and $\{\boldsymbol{x}_{obs,i}\}_{i=1}^{N_{obs}}$ represent the observed positions.

3. Physics-informed neural networks for uncertainty quantification

Herein, we employ PINNs to train models based on the physics described by differential equations while also quantifying *aleatoric* uncertainty stemming from noisy data. Our approach strikes a balance between solving the partial differential equation and performing statistical regression to accommodate both forward and inverse problems with noisy observations. We aim to implement normal probability distribution functions at training points, effectively balancing PINNs and data regression objectives. This is depicted in Fig. 1 , where the dashed green and purple lines represent instances of the proposed distribution functions, and the solid blue line indicates the mean function. Note that a likelihood formulation has been employed in TensorFlow Probability, as described by Dillon et al. [20]. This formulation is notably comprehensive and aligns with a modern vision of probability theory, which is adapted to the deep learning paradigm that emphasizes end-to-end differentiable computation. The approach is built upon two fundamental abstractions: Distributions and Bijectors. In our work, we propose a straightforward approach in which we uniquely derive our formula from the likelihood function:

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) = \prod_{i=1}^{n_{\text{tot}}} \frac{1}{\sqrt{2\pi\sigma_u^2(x_i, \boldsymbol{\theta})}} \exp\left(-\frac{\left(y_i - \bar{u}(x_i, \boldsymbol{\theta})\right)^2}{2\sigma_u^2(x_i, \boldsymbol{\theta})} \right), \tag{13}$$

where $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{b}\}$ represents the hyperparameters, comprising weights and biases. Additionally, y_i denotes our available scattered noisy measurements of displacement u. Furthermore, $\bar{u}$ and σ_u represent the mean function and the standard deviation, respectively.

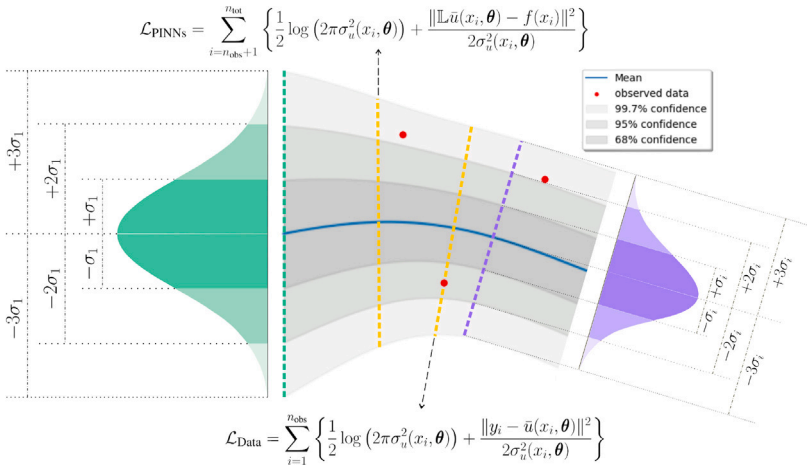

Fig. 1. Allocating normal distribution functions at training points in the presence of noisy data (red circles). The dashed green and purple lines are examples of distribution functions at the first and ith training points, and the solid blue line represents the mean function. The dashed yellow line with the red circle follows the loss term for data regression, while PINNs' loss function governs the other yellow line. The shades indicate different levels of uncertainties. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Consistent with customary minimization goals in Deep Learning, we also utilize the negative logarithm of marginal likelihood to give

$$-\log p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = \sum_{i=1}^{n_{\text{tot}}} \left\{ \frac{1}{2}\log\left(2\pi\sigma_u^2(x_i,\boldsymbol{\theta})\right) + \frac{\|y_i - \bar{u}(x_i,\boldsymbol{\theta})\|^2}{2\sigma_u^2(x_i,\boldsymbol{\theta})} \right\}. \tag{14}$$

Dividing the total points into observations and PINNs' training points, *i.e.* $n_{\text{tot}} = n_{\text{obs}} + n_{\text{PINNs}}$ where observation points are in the range $1, 2, \ldots, n_{\text{obs}}$

$$\begin{aligned} -\log p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = &\sum_{i=1}^{n_{\text{obs}}} \left\{ \frac{1}{2}\log\left(2\pi\sigma_u^2(x_i,\boldsymbol{\theta})\right) + \frac{\|y_i - \bar{u}(x_i,\boldsymbol{\theta})\|^2}{2\sigma_u^2(x_i,\boldsymbol{\theta})} \right\} + \\ &\sum_{i=n_{\text{obs}}+1}^{n_{\text{tot}}} \left\{ \frac{1}{2}\log\left(2\pi\sigma_u^2(x_i,\boldsymbol{\theta})\right) + \frac{\|y_i - \bar{u}(x_i,\boldsymbol{\theta})\|^2}{2\sigma_u^2(x_i,\boldsymbol{\theta})} \right\}. \end{aligned} \tag{15}$$

Analogous to the original concept of PINNs [13], we explicitly incorporate the differential equation of the general form $\mathbb{L}\boldsymbol{u} = f(\boldsymbol{x})$ into the loss function in order to avoid implicit data synthesis from the partial differential equation (PDE)

$$\begin{aligned} -\log p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = &\sum_{i=1}^{n_{\text{obs}}} \left\{ \frac{1}{2}\log\left(2\pi\sigma_u^2(x_i,\boldsymbol{\theta})\right) + \frac{\|y_i - \bar{u}(x_i,\boldsymbol{\theta})\|^2}{2\sigma_u^2(x_i,\boldsymbol{\theta})} \right\} + \\ &\sum_{i=n_{\text{obs}}+1}^{n_{\text{tot}}} \left\{ \frac{1}{2}\log\left(2\pi\sigma_u^2(x_i,\boldsymbol{\theta})\right) + \frac{\|\mathbb{L}\bar{u}(x_i,\boldsymbol{\theta}) - f(x_i)\|^2}{2\sigma_u^2(x_i,\boldsymbol{\theta})} \right\}, \end{aligned} \tag{16}$$

where $\mathbb{L}$ denotes the differential operator and $f(\boldsymbol{x})$ indicates the source function. The dashed yellow lines in Fig. 1 instantiate the objectives for data regression and PINNs. Finally, the minimization problem yields

$$\min_{\boldsymbol{\theta}} -\log p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}). \tag{17}$$

In practice, we adopt weights for the loss terms, *e.g.* τ_{PINNs} and τ_{Data}, which are shown in Fig. 2 for the proposed Neural Network architecture. The weights enable us to strike the balance between minimization goals. For instance, fitting observations may become the priority over PINNs where physics fail to aptly represent the behaviour of the problem. Thus, the weights of loss functions help weighing which term overrules the other.

An alternative to the proposed approach is Bayesian Physics-Informed Neural Networks (BPINNs) [21] which exploits PINNs' solution as the prior to enhance the posterior estimates through the Bayes' rule. The key difference between the current approach and BPINNs lies in directly targeting the mean and the standard deviation functions as the output of the Neural Networks, thereby bypassing costly integration schemes such as Markov Chain Monte Carlo (MCMC) utilized for Bayes' rule estimations. It is acknowledged that a more robust estimate is achieved in a fully Bayesian framework at the expense of costly sampling procedures [22].

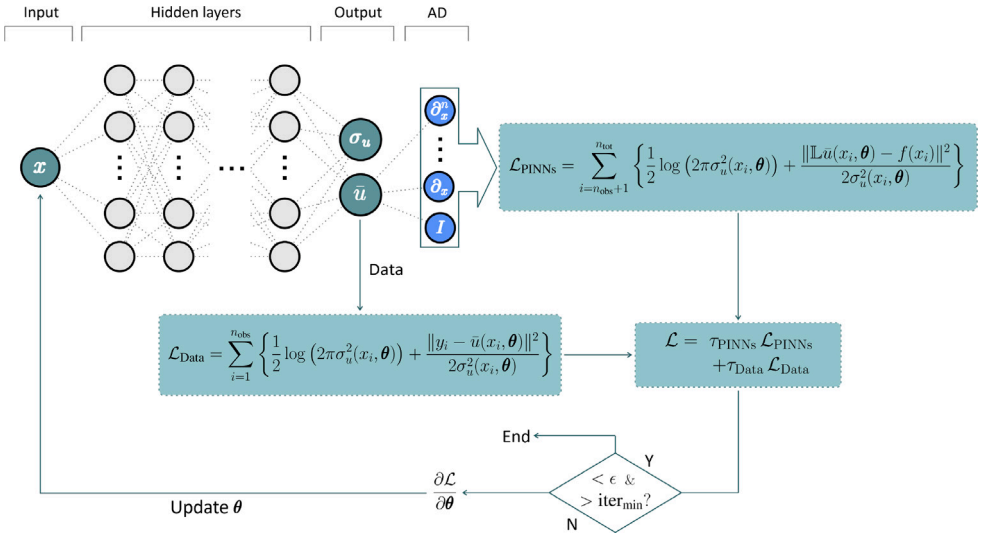

Fig. 2. Physics Informed Neural Networks augmented with purely statistical regression at observations for handling *aleatoric* uncertainty. The output of the Neural Networks, *i.e.* the mean function $\bar{\boldsymbol{u}}$ and the standard deviation $\boldsymbol{\sigma}_u$, are translated to PINNs and data regression loss functions to be minimized for the hyperparameters $\boldsymbol{\theta}$ containing weights and biases.

4. Numerical implementation

The proposed method is initially benchmarked against BPINNs, demonstrating that with sufficient data, our streamlined approach achieves comparable accuracy. Subsequently, various examples are methodically analysed to evaluate the potential benefits and capabilities of the proposed methods for characterizing the elastic modulus distribution in solids using partial-field measurements. In each scenario, partial-field measurements are assumed to be acquired through non-destructive testing within the linear/nonlinear range of solid behaviour. The investigation commences with pairs of numerical examples in 1D problems, encompassing variations in Young's modulus both with and without noise. Subsequently, the scope extended to 2D cases.

4.1. Comparison with BPINNs

We examine the following one-dimensional nonlinear PDE that is also considered in [21]:

$$\lambda \frac{\partial^2 u}{\partial x^2} + k \tanh(u) = f, \quad x \in [-0.7, 0.7] \tag{18}$$

where $\lambda = 0.01$ is a constant and k needs to be identified based on partial measurements of f and u. The solution for u is specified as $u = \sin^3(6x)$, enabling the derivation of the function f from Eq. (18) with $k = 0.7$. Following [21], we deploy various sensor configurations for f and u: 32 and 64 sensors are equidistantly placed across the domain $[-0.7, 0.7]$ to measure f, while two sensors for u are positioned at $x = -0.7$ and $x = 0.7$ to establish Dirichlet boundary conditions. Additionally, 6 and 12 interior sensors for u assist in identifying k. Measurements are subjected to two scales of Gaussian noise: the first scale includes $\epsilon_f \sim \mathcal{N}(0, 0.01^2)$, $\epsilon_b \sim \mathcal{N}(0, 0.01^2)$ and $\epsilon_u \sim \mathcal{N}(0, 0.01^2)$, while the second scale is characterized by $\epsilon_f \sim \mathcal{N}(0, 0.1^2)$, $\epsilon_b \sim \mathcal{N}(0, 0.01^2)$ and $\epsilon_u \sim \mathcal{N}(0, 0.1^2)$.

To solve equation (18), we define the following loss function, incorporating various statistical parameters specific to the problem domain.

$$\begin{aligned} -\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) = & \sum_{i=1}^{n_{\text{obs}_f}} \left\{ \frac{1}{2} \log\left(2\pi\sigma_f^2(x_i, \boldsymbol{\theta})\right) + \frac{\|f_i - \bar{f}(x_i, \boldsymbol{\theta})\|^2}{2\sigma_f^2(x_i, \boldsymbol{\theta})} \right\} + \\ & \sum_{i=1}^{n_{\text{obs}_{u_b}}} \left\{ \frac{1}{2} \log\left(2\pi\sigma_{u_b}^2(x_i, \boldsymbol{\theta})\right) + \frac{\|u_{bi} - \bar{u}_b(x_i, \boldsymbol{\theta})\|^2}{2\sigma_{u_b}^2(x_i, \boldsymbol{\theta})} \right\} + \\ & \sum_{i=1}^{n_{\text{obs}_u}} \left\{ \frac{1}{2} \log\left(2\pi\sigma_u^2(x_i, \boldsymbol{\theta})\right) + \frac{\|y_i - \bar{u}(x_i, \boldsymbol{\theta})\|^2}{2\sigma_u^2(x_i, \boldsymbol{\theta})} \right\} + \\ & \sum_{i=1}^{n_{\text{obs}_f}} \left\{ \frac{1}{2} \log\left(2\pi\sigma_{pde}^2(x_i, \boldsymbol{\theta})\right) + \frac{\|\mathbb{L}\bar{u}(x_i, \boldsymbol{\theta}) - f_i\|^2}{2\sigma_{pde}^2(x_i, \boldsymbol{\theta})} \right\}, \end{aligned} \tag{19}$$

The parameters σ_f and $\bar{f}$ represent the standard deviation and mean, respectively, for the function f. Similarly, σ_{u_b} and $\bar{u}_b$ denote the standard deviation and mean for the boundary conditions. Additionally, we introduce σ_{pde} to denote the standard deviation

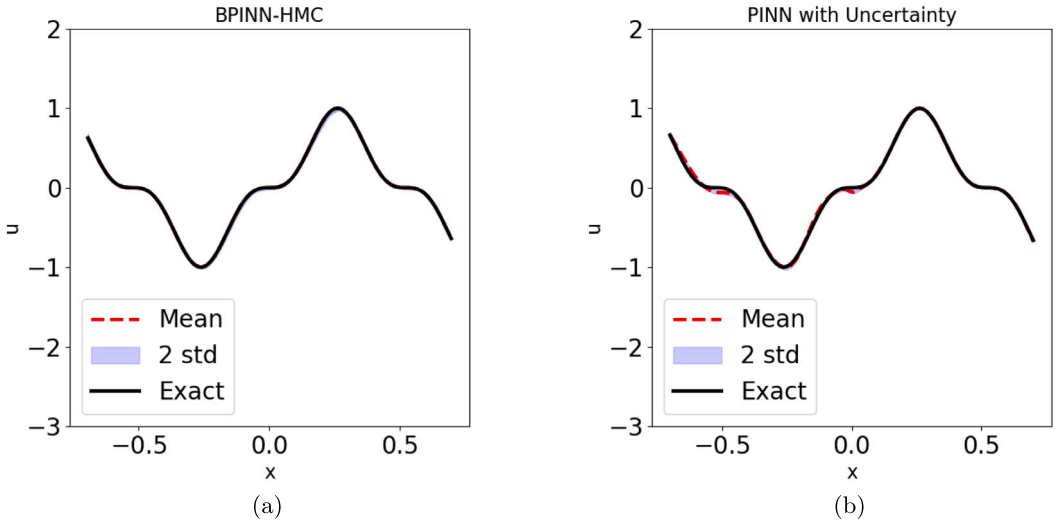

Fig. 3. Comparison of Displacements with Normal Distribution $\mathcal{N}(0, 0.01^2)$: (a) Displacements for BPINNs, as described in Ref. [21] (b) Displacements using the present method.

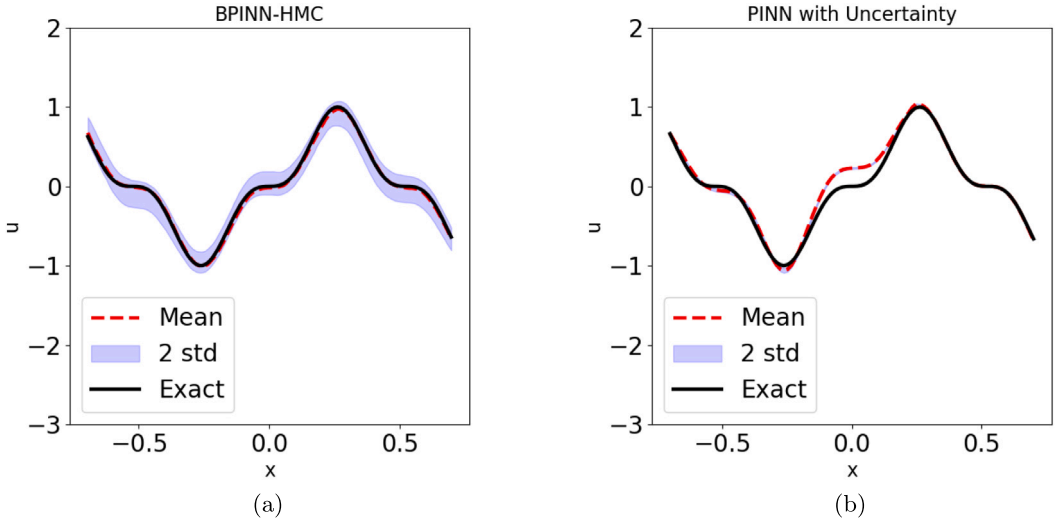

Fig. 4. Comparison of Displacements with Normal Distribution $\mathcal{N}(0, 0.1^2)$: (a) Displacements for BPINNs, as described in Ref. [21] (b) Displacements using the present method.

of the PDE values, distinguishing it from the standard deviations related to displacements. In (16), we assume that the standard deviations for displacement and the PDE need not be distinguished. However, to ensure comprehensive understanding and accuracy, this assumption warrants further examination through the defined loss function (19). For the implementation, we employ a neural network architecture consisting of two hidden layers. Each layer has a width of 100, and the network outputs five different parameters. Training is conducted using the Adam optimizer, set at a learning rate of 1×10^{-4}, and involves a total number of 100,000 training steps. This setup is designed to optimize the performance and accuracy of the model in predicting the solutions to the PDE under consideration.

Fig. 3 compares displacements modelled with a normal distribution $\mathcal{N}(0, 0.01^2)$ using a Bayesian physics-informed neural network with Monte Carlo estimations of the posterior [21], and our proposed approach which incorporates measurements from 64 sensors for f and 12 sensors for u. The results demonstrate that our method yields outcomes closely matching the accuracy of BPINNs. The predicted values of k for BPINN-HMC and our approach are 0.705 and 0.711, respectively. Fig. 4 addresses a similar comparison but under a higher noise scenario with a normal distribution $\mathcal{N}(0, 0.1^2)$. As expected, increasing the noise level reduces the accuracy of the results. This degradation is observed in both BPINNs and our method, although the impact is minimal in the former, as evidenced in Fig. 4. Nevertheless, our approach continues to produce commendable results with a high number of sensor measurements. The predicted values of k for BPINN-HMC and our approach are 0.665 and 0.628, respectively. Fig. 5 highlights that our method achieves

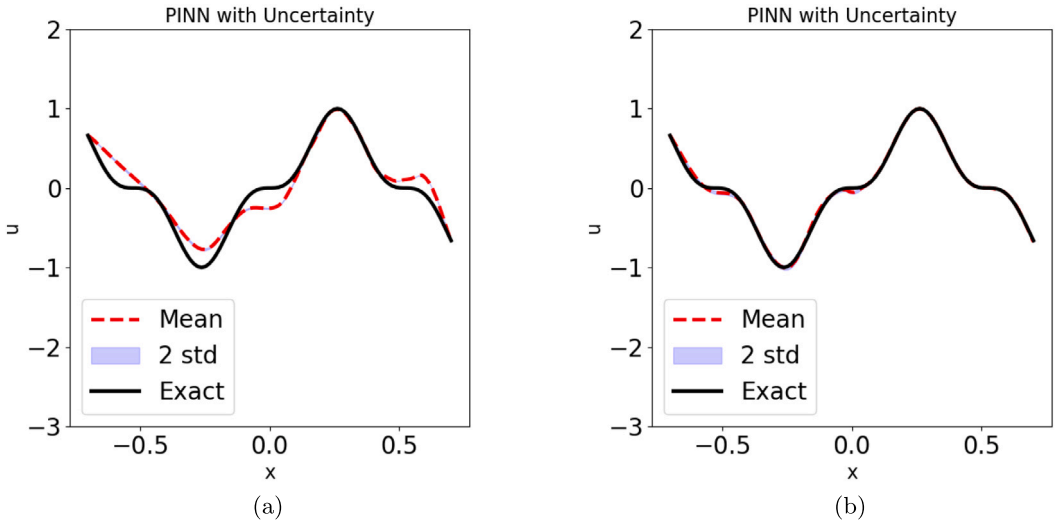

Fig. 5. Comparison of Displacements Modelled with a Normal Distribution $\mathcal{N}(0, 0.01^2)$: (a) Displacements monitored by 32 sensors for f and 6 sensors for u. (b) Displacements monitored by 64 sensors for f and 12 sensors for u.

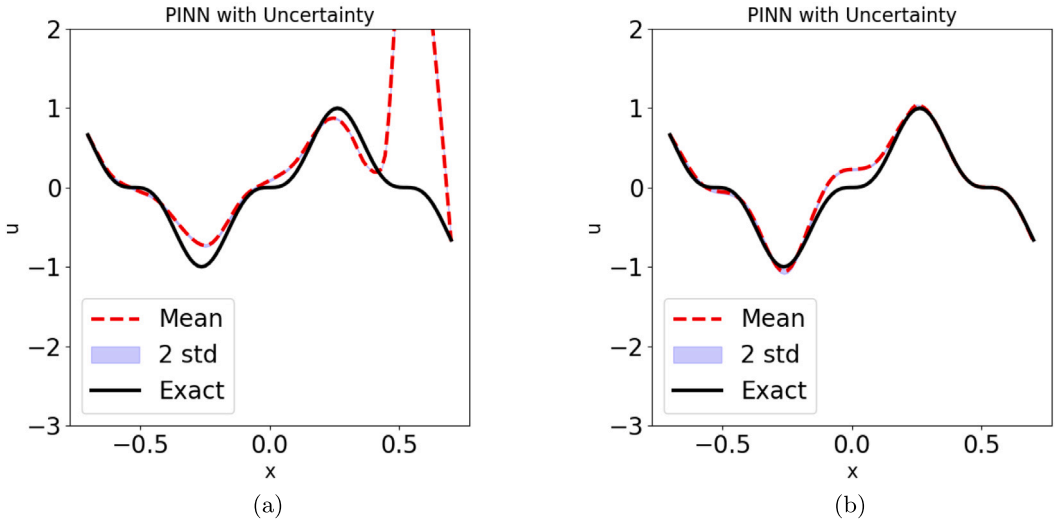

Fig. 6. Comparison of Displacements Modelled with a Normal Distribution $\mathcal{N}(0, 0.1^2)$: (a) Displacements monitored by 32 sensors for f and 6 sensors for u. (b) Displacements monitored by 64 sensors for f and 12 sensors for u.

greater accuracy by utilizing additional data. Specifically, the results with 12 and 64 sensor measurements for u and f, respectively, surpass those obtained with 6 and 32 sensor measurements. This improvement underscores the value of increasing the amount of informational input into our model. Fig. 6 demonstrates that increasing the noise level to $\mathcal{N}(0, 0.1^2)$ leads to a reduction in result accuracy. However, with 64 sensors for f and 12 sensors for u, there is still a relatively good agreement with the exact solution, as shown in Fig. 6-b. This contrasts with Fig. 6-a, which uses fewer sensors for both f and u and exhibits poorer agreement. This indicates that at higher noise levels, the present approach requires a greater number of measurements to maintain high fidelity in the results.

We consider a neural network with six outputs σ_f, $\bar{f}$, σ_{u_b}, $\bar{u}_b$, σ_u, and $\bar{u}_b$ instead of seven, which includes an additional σ_{pde}. This configuration is used to demonstrate that having the same standard deviation for the PDE and u does not significantly affect the results. Fig. 7 illustrates that this assumption has minimal impact on the outcomes. However, in more complex scenarios where displacements and the PDE residuals differ significantly in magnitude, the selection of distinct standard deviations becomes unavoidable.

In summary, the approach proposed in this work provides a simpler and more direct method for quantifying uncertainty when a sufficient quantity of sensor data with acceptable noise levels is available.

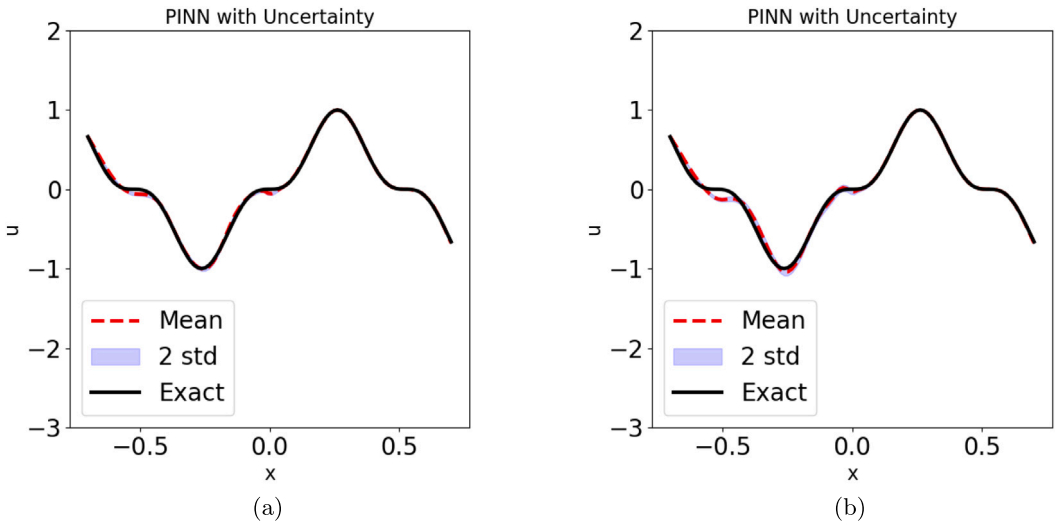

Fig. 7. Comparison of Displacements Modelled with a Normal Distribution $\mathcal{N}(0, 0.01^2)$: (a) Displacements modelled by a neural network with seven outputs, including the standard deviation for the PDE. (b) Displacements modelled by a neural network with six outputs.

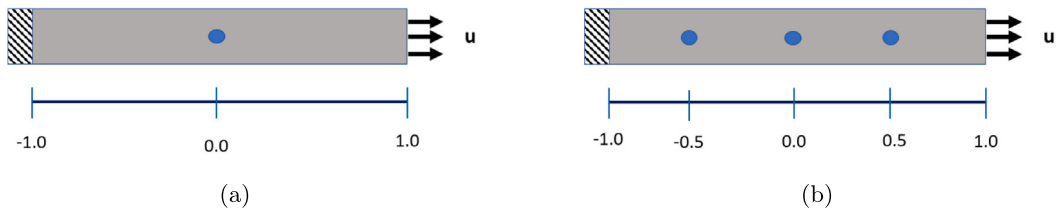

Fig. 8. One-dimensional model of bar with (a) a centre inclusion (b) three inclusions.

4.2. 1D problems

Here, the numerically simulated experiments are based upon characterization of elastic modulus distributions with circular inclusions (hard or soft), as shown schematically in Fig. 8-(a). The circular inclusion assumption is used for simplicity, but more importantly, it is based on several other related characterization works for a variety of applications [9,23,24]. A Gaussian radial basis function (RBF) representation is chosen to define the localized elastic modulus variations, which is based upon several other similar works [25–27]. The RBF representation of elastic modulus distribution is defined as:

$$E(x) = E_0\left[1 + \sum_{i=1}^{N_I} \alpha_i \exp\left(-\frac{(x - c_i)^2}{a_i^2}\right)\right] \tag{20}$$

where E_0 is the elastic modulus of the matrix material, α_i is the relative change in elastic modulus at the ith inclusion centre, c_i, is the location of the ith inclusion centre, a_i is the breadth of the ith inclusion, and N_I is the total number of inclusions. Furthermore, we also incorporate the following representation of the elastic modulus distribution to showcase the performance and versatility of the proposed PINNs approach:

$$E(x) = E_0\left[1 + \sum_{i=1}^{N_I} \alpha_i \left(\tanh\left(\frac{(x - c_i) + \beta_i}{a_i}\right) + \tanh\left(\frac{-(x - c_i) + \beta_i}{a_i}\right)\right)\right] \tag{21}$$

where β_i is the positive constant at the ith inclusion centre.

Initially, we address a forward problem involving a single inclusion, considering parameters such as $E_0 = 0.1$, $\alpha = 10.0$, $c = 0.0$ and $a = 0.1$. We consider a fully connected neural network with 8 hidden layers and 20 neurons in each hidden layer. We use a hyperbolic tangent as the activation function while a linear activation function has been implemented for the last hidden layer. The network architecture of the neural network and the number of collocation points are shown in Table 1. To optimize the network, the Adam optimizer [28] is utilized with a learning rate of 5×10^{-4}.

For the inverse problem we seek to discover the Young's modules based upon a set of 100 equally spaced displacements obtained from the forward problem. This involves utilizing the same neural network architecture, with the corresponding number of collocation points delineated in Table 1, and employing a hyperbolic tangent as the activation function.

Table 1
Neural network configuration settings for 1D elastostatic examples using PINNs and uncertainty-PINNs.

Method	Network architecture	No. collocation points
PINNs	[1,[20] × 8,1]	560
Uncertainty-PINNs	[1,[20] × 8,2]	560

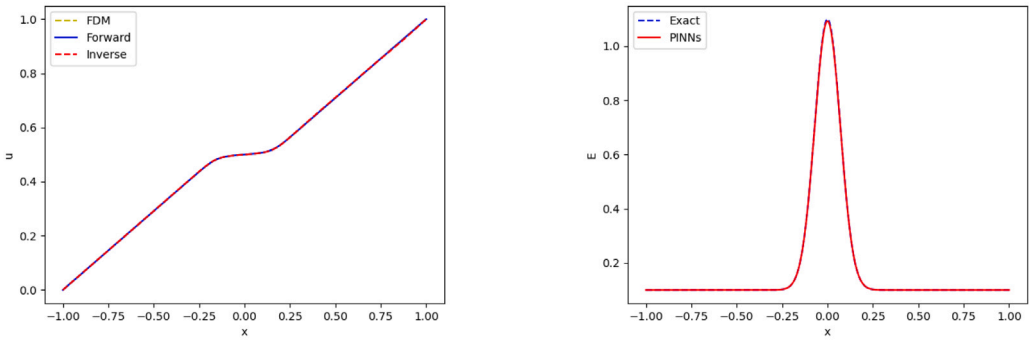

Fig. 9. (a) Comparison of displacements derived using the Finite Difference Method for forward and inverse 1D problems without noise. (b) Comparison of the Young's Modulus values determined from the inverse 1D problem without noise against the specified Young's Modulus distribution.

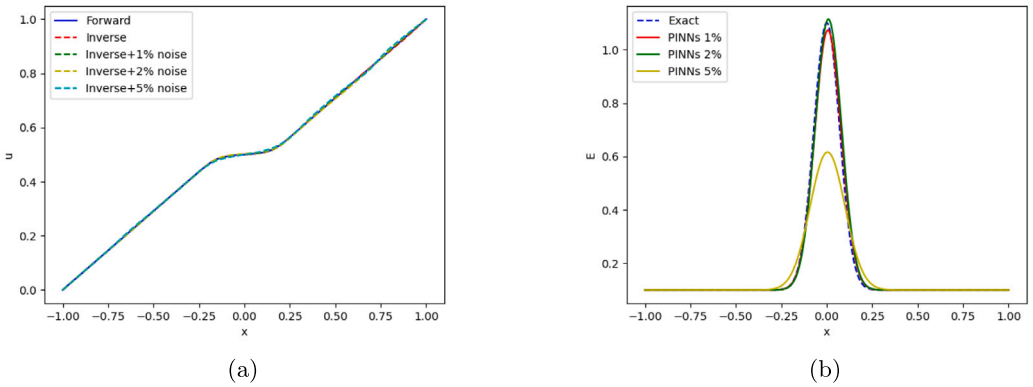

Fig. 10. Comparison of displacements (a) and Young's Modulus (b) for solutions obtained with Forward and Inverse 1D problem with noise.

In addressing the inverse problem, we employ a methodology similar to the successful Data-driven discovery of partial differential equations using PINNs, as demonstrated in diverse engineering and scientific contexts, such as the Navier–Stokes equations [13]. Our aim is to determine not only the parameters α, c, and a, but also the displacements across the entire domain. We utilize the loss function described in Eq. (12), incorporating the residual, r, from Eq. (11). The residual for the 1D case is defined as:

$$r = \frac{d\sigma}{dx} + f \tag{22}$$

where

$$\sigma = E\frac{du}{dx} \tag{23}$$

and E is defined by Eq. (20). It is important to note that the parameters α, c, and a become key components of the physics-informed neural network in defining r.

In Fig. 9-(a), the displacements for both the forward and inverse solutions are illustrated. As expected, PINNs exhibit the capability to reconstruct the entire structural response using only partial, known displacements. Moving to Fig. 9-(b), a comparison between the distributions of PINNs and the exact Young's modulus is presented, revealing a high degree of agreement between the two.

Now, introducing 1%, 2% and 5% noise in the 100 displacements, we proceed to solve the inverse problem using the same parameters as in the previous 1D scenario. In Fig. 10-(a), the displacements for the forward, inverse, and noisy inverse solutions are presented. Demonstrating remarkable capability, PINNs proves effective in reconstructing the entire structural response even

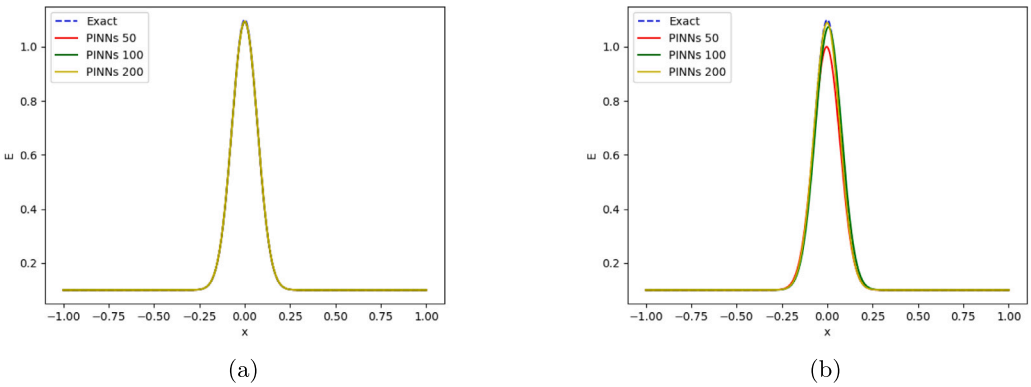

Fig. 11. Comparison of displacement counts for Young's Modulus: solutions derived without noise (a) versus with 1% noise (b).

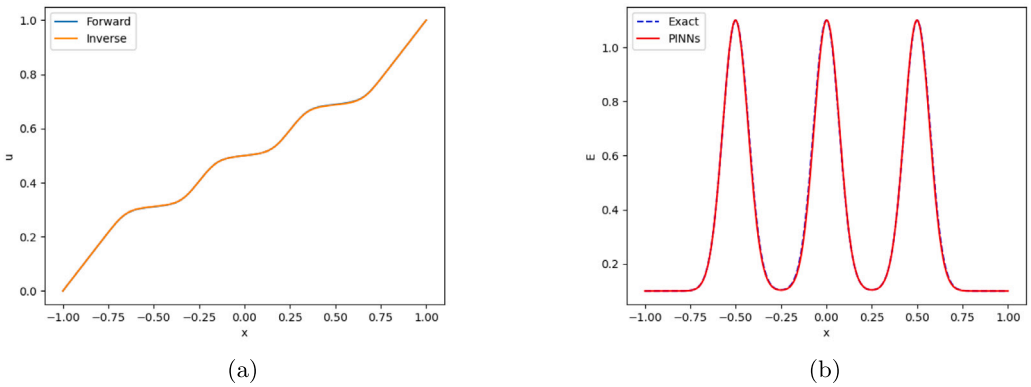

Fig. 12. Comparison of displacements (a) and Young's Modulus (b) for solutions obtained with Forward and Inverse 1D problem with three inclusion and with noise.

when provided with partial, noisy displacement data. Fig. 10-(b) further illustrates a comparison between the Young's modulus distributions obtained using PINNs with various noise percentages and the exact Young's modulus. The results demonstrate strong agreement at 1% and 2% noise levels, but a weaker correlation when the noise increases to 5%. In Figs. 11-(a) and 11-(b), we examine the impact of varying the number of displacements generated by the forward problem on the inverse problem solutions, both without and with noise. The data show that for the noise-free scenario, using 50, 100, or 200 displacements yields nearly identical estimates of Young's modulus. However, as illustrated in Fig. 11-(b), incorporating more displacements leads to more accurate determinations of Young's modulus when noise is present.

Next, we consider a bar with three inclusions as shown in Fig. 8-(b). In Fig. 12-(a), the displacements for both forward and inverse solutions in a 1D problem with three inclusions are presented. The parameters considered are $E_0 = 0.1$, $\alpha_1 = \alpha_2 = \alpha_3 = 10.0$, $c_1 = -0.5$, $c_2 = 0.0$, $c_3 = 0.5$, and $a_1 = a_2 = a_3 = 0.1$. The results show excellent agreement between the forward and inverse solutions. Furthermore, in Fig. 12-(b), a detailed comparison is presented, showcasing the distributions of PINNs alongside the exact Young's modulus. This comparison underscores a robust agreement between the predicted and actual values, in the absence of noise.

In Fig. 13-(a), the displacements are visualized for the forward, noise-free inverse, and 1 percent noisy inverse solutions in a 1D problem featuring Young's modulus derived from (21). Notably, Fig. 13-(b) provides a comprehensive comparison between the Young's modulus distributions, showcasing the exact values alongside those predicted by PINNs under both noise-free and noisy conditions.

Now, revisiting the previous 1D problem with a Gaussian radial basis function (RBF) representation of Young's modulus from (20), we employ PINNs to quantify the *aleatoric* uncertainty stemming from noisy data. This process involves utilizing a dataset of 100 displacements, evenly spaced and incorporating 1 percent noise, coupled with the application of the loss function as defined in (16). The output of this process yields the entire domain displacements, including mean values and standard deviations.

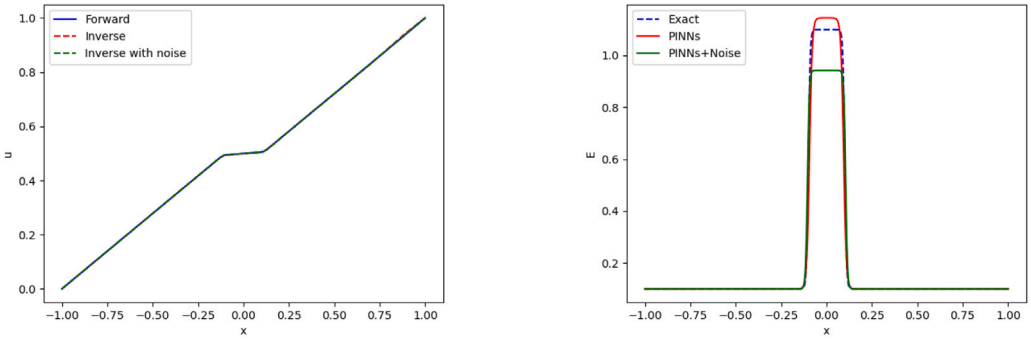

Fig. 13. (a) the displacements for the forward, noise-free inverse, and 1 percent noisy inverse solutions in a 1D problem featuring Young's modulus derived from (21) (b) comparison between the Young's modulus distributions.

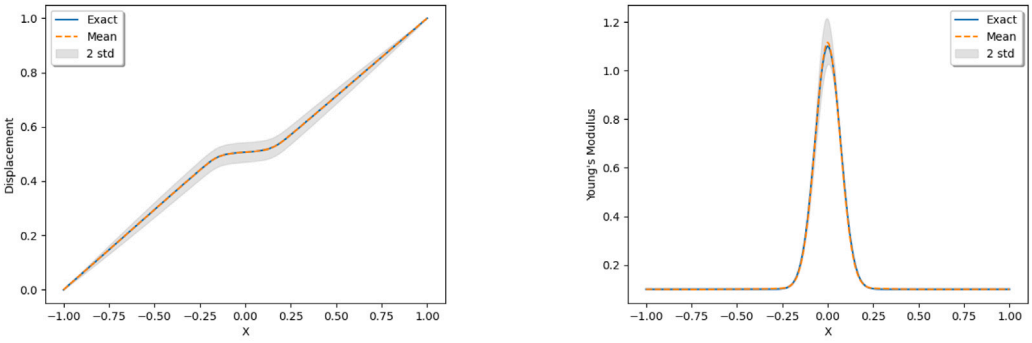

Fig. 14. Mean Value and Uncertainty Analysis with 1% Noise Level for 1D Elastostatic Models with Soft Matrix and Hard Inclusion (a) Displacements (b) Young's Moduli.

To estimate Young's modulus across various levels of uncertainty, we approach a direct inverse problem in the following manner:

Given: The mean and standard deviation of the displacements across the entire domain, which were calculated by addressing an inverse problem previously described in this subsection.

Objective: To determine the unknown distribution of material properties, including Young's modulus, E, and to quantify its associated uncertainty.

Methodology: Utilizing the calculated uncertainties in displacements, we solve equation (22) to infer Young's modulus, allowing us to translate displacement uncertainties into uncertainties in material properties. Our approach to solving this direct inverse problem builds on methodologies previously employed but differs in key aspects from those described by Wang et al. [29]. Unlike their method, which reconstructs the full domain response using finite element methods, our approach leverages PINNs.

Fig. 14-(a) illustrates the mean displacement values and their associated uncertainties. Fig. 14-(b) presents the mean values of Young's moduli along with their respective levels of uncertainty. Additionally, we explore a different distribution of Young's modulus characterized by parameters $E_0 = 69$, $\alpha = -0.8$, $a = 0.1$, and $c = 0.0$. This configuration, which simulates a hard matrix with soft inclusions, demonstrates the application potential for nondestructive evaluation in civil or aerospace structures. Fig. 15-(a) shows the mean displacement values and uncertainties under this setup. As in the previous case, we address the estimation of Young's modulus through a direct inverse problem as described by Eq. (1). Fig. 15-(b) then displays the mean values of Young's moduli, detailing the uncertainties associated with each.

4.3. 2D case studies

In this section, we explore two-dimensional examples, starting with the initial case study depicted in Fig. 16-(a), which involves a plate featuring a centrally located circular inclusion. The collocation/integration points are equally spaced, as illustrated in Fig. 16-(b), and amount to 11,827. A vertical upward displacement is imposed at the top edge. The material parameters are specified as

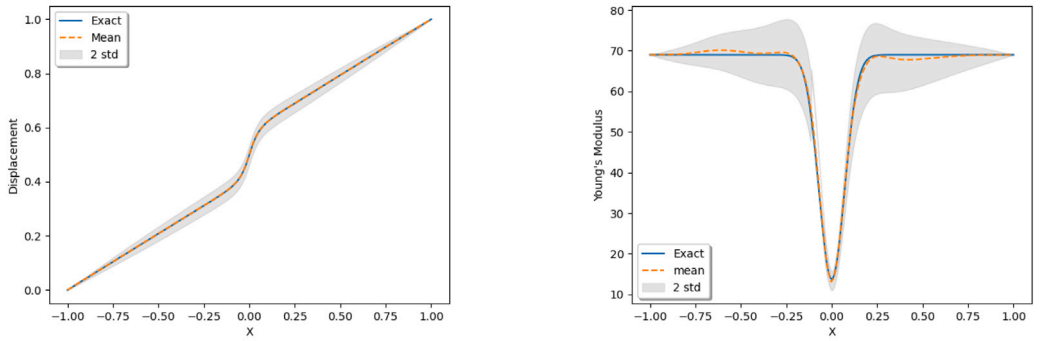

Fig. 15. Mean Value and Uncertainty Analysis with 1% Noise Level for 1D Elastostatic Models with Hard Matrix and Soft Inclusion (a) Displacements (b) Young's Moduli.

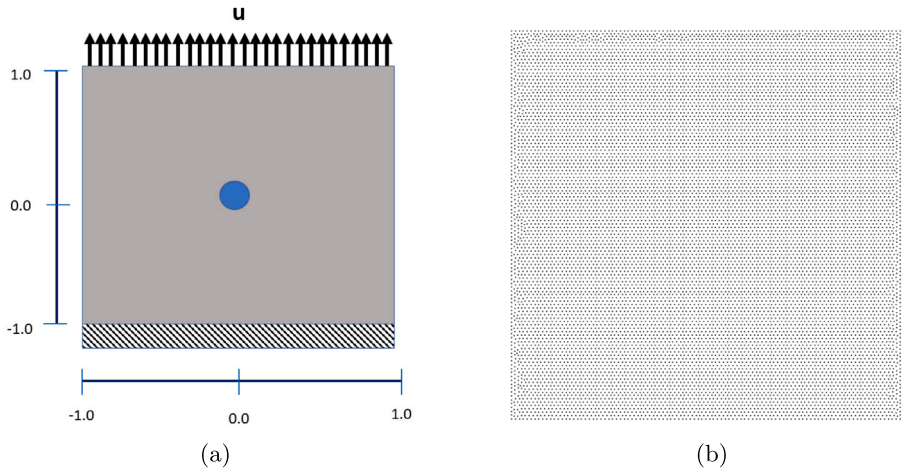

Fig. 16. (a) Two-dimensional model of a plate with a centre inclusion (b) collocation points.

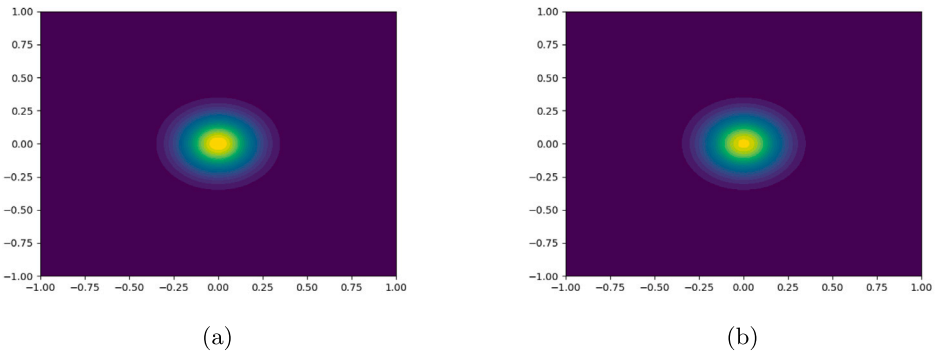

Fig. 17. Young's Modulus for (a) Exact (b) PINNs with noisy data.

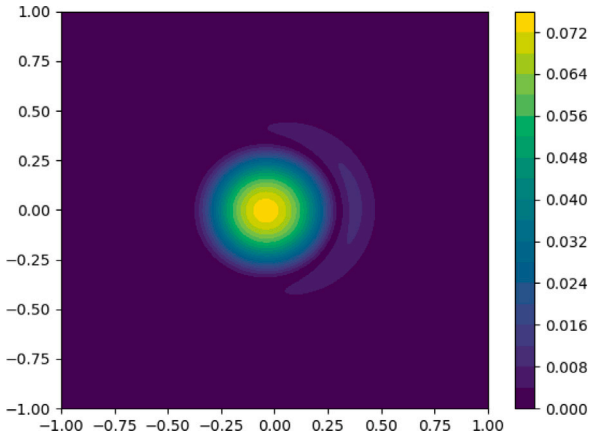

Fig. 18. Relative Error in Estimating Young's Modulus with a Central Inclusion: A Comparison Between Exact Values and PINNs Predictions.

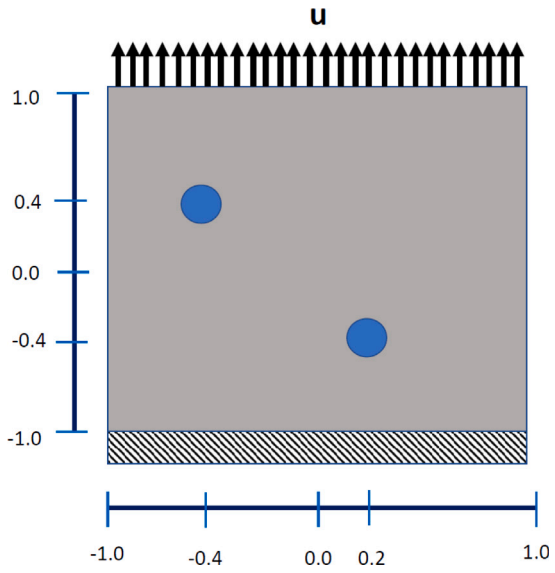

Fig. 19. Two-dimensional model of a plate with two inclusions.

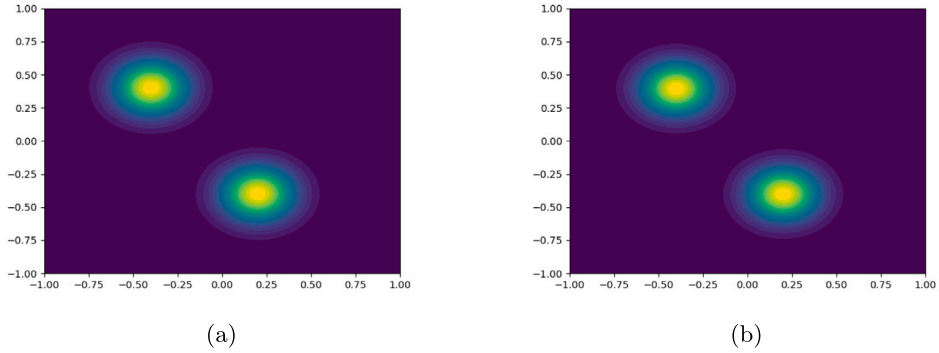

Fig. 20. Young's Modulus for (a) Exact (b) PINNs with noisy data.

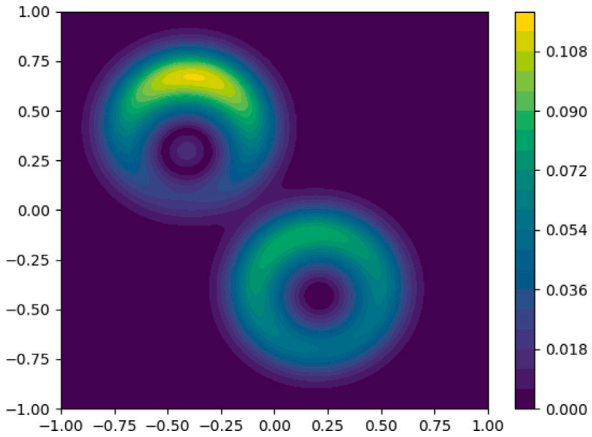

Fig. 21. Relative Error in Estimating Young's Modulus with Two Inclusions: A Comparison Between Exact Values and PINNs Predictions.

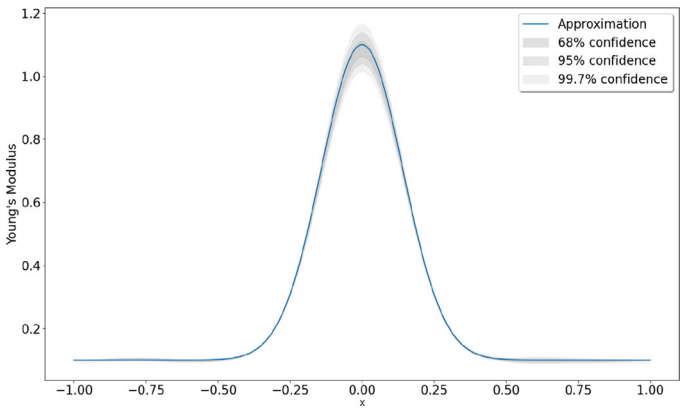

Fig. 22. Mean value and 68%, 95%, and 99% confidence levels of uncertainties for Young's moduli at $y = 0$.

Table 2
Neural network configuration settings for 2D elastostatic examples Using PINNs and uncertainty-PINNs.

Method	Network architecture	No. collocation points
PINNs	[2,[20] × 8,2]	11 827
Uncertainty-PINNs	[2,[20] × 8,4]	11 827

$E = 0.1$ GPa and $\nu = 0.3$. Additionally, the inclusion is defined based on Eq. (20), with parameters $E_0 = 0.1$, $\alpha = 10.0$, $\boldsymbol{c} = (0.0, 0.0)$, and $a = 0.2$.

For the neural network architecture, a fully connected structure with 8 hidden layers, each containing 20 neurons, is employed. The activation function is hyperbolic tangent for all hidden layers, and a linear activation function is applied to the last hidden layer. The network architecture and the number of collocation points are detailed in Table 2.

To optimize the network, the Adam optimizer [28] is utilized with a learning rate of 5×10^{-4}. Dirichlet boundary conditions are imposed strongly by altering the solutions, u and v, according to (24), where $\hat{u}$ and $\hat{v}$ are provided by the network.

$$u_1 = (x_1 + 1)\hat{u} \qquad u_2 = (x_2^2 - 1)\hat{v} + (x_2 + 1.0)\frac{\boldsymbol{u}}{2} \tag{24}$$

To tackle the two-dimensional (2D) inverse problem, we employ a strategy akin to the one used for the one-dimensional (1D) inverse problem, albeit with modifications to the residual calculation, r, as detailed in Eq. (11). In the context of the 2D scenario, the residuals are defined as follows:

$$r^1 = \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} + f_1 \tag{25}$$

$$r^2 = \frac{\partial \sigma_{21}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} + f_2 \tag{26}$$

Here, the stress components are given by:

$$\sigma_{11} = (\lambda + 2\mu)\frac{\partial u_1}{\partial x_1} + \lambda\frac{\partial u_2}{\partial x_2} \tag{27}$$

$$\sigma_{22} = (\lambda + 2\mu)\frac{\partial u_2}{\partial x_2} + \lambda\frac{\partial u_1}{\partial x_1} \tag{28}$$

$$\sigma_{12} = \sigma_{21} = \mu(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}) \tag{29}$$

where λ and μ are lamé parameters defined by:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \qquad \mu = \frac{E}{2(1+\nu)} \tag{30}$$

and E specified by Eq. (20). It is crucial to recognize that parameters such as α, c, and a are integral to the PINNs framework, influencing the definition of r. To solve the inverse problem, we introduced a 1 percent noise to 400 displacement measurements, evenly distributed with 20 measurements per direction. Fig. 17 presents a comparison between the estimated distributions of Young's modulus by PINNs and the exact values. The results demonstrate a strong correlation, highlighting the model's robustness despite the input data noise. Fig. 18 illustrates the relative error between the PINNs' predictions and the exact solutions for Young's modulus, providing a quantitative measure of the estimation accuracy.

Now, we turn our attention to another 2D example depicted in Fig. 19, featuring two inclusions. The material parameters remain consistent with the previous example. Furthermore, the inclusions are defined based on (20), with parameters $E_0 = 0.1$, $\alpha = 10.0$, $c_1 = (-0.4, 0.4)$, $c_2 = (0.2, -0.4)$, and $a = 0.2$. Introducing a 1 percent noise in 400 displacements in each direction, we proceed to solve the inverse problem using the same parameters as in the previous 2D scenario.

Fig. 20 presents a comparison between the distributions of Physics-Informed Neural Networks PINNs and the exact Young's modulus, demonstrating a robust agreement despite the presence of noise in the input data. Fig. 21 illustrates the relative error between the predictions made by PINNs and the exact solutions for Young's modulus. This comparison provides a quantitative evaluation of the prediction accuracy.

In the 2D scenario, we explore the uncertainty as depicted in Fig. 19. Similar to the 1D case, we approach this by solving a direct inverse problem to determine Young's modulus for each level of uncertainty. Fig. 22 presents the mean values of Young's modulus at $y = 0$ along with the corresponding uncertainty levels.

5. Conclusions

In this study, we employed PINNs to address inverse material characterization challenges by utilizing partial-field response measurements in a single attempt. We demonstrated that PINNs are capable of reconstructing the full spatial distribution of a system's response from only a portion of the measured response field, estimating unknown material properties, and training a model grounded in the underlying physics described by differential equations. Furthermore, PINNs are capable of handling aleatoric uncertainty, which stems from the presence of noisy data.

We compared our approach with Bayesian Physics-Informed Neural Networks and observed satisfactory results in solving a nonlinear inverse problem. Our 1D numerical experiments demonstrate that PINNs can accurately reconstruct the structural response and precisely determine mechanical properties, such as Young's modulus, both in the presence and absence of noise. We also conducted sensitivity analyses to evaluate the effects of various noise levels and different numbers of displacement samples on the accuracy of solving inverse problems for Young's modulus. This involved introducing noise to create uncertainty and estimating Young's modulus at various confidence levels of uncertainty. Our two-dimensional examples, which characterized the elastic modulus distribution based on static partial-field displacement response measurements, also showed strong agreement with exact solutions.

For future work, we aim to optimize the proposed approach in terms of computational cost and improve accuracy, for example, by testing different activation functions and optimizers. Additionally, we plan to reduce the dependency of the method on a priori information by diversifying the relative distributions for Young's modulus and Poisson's ratio.

CRediT authorship contribution statement

Yousef Ghaffari Motlagh: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Farshid Fathi:** Writing – review & editing, Methodology, Conceptualization. **John C. Brigham:** Writing – review & editing, Conceptualization. **Peter K. Jimack:** Writing – review & editing, Resources, Conceptualization.

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.

Acknowledgements

We are grateful to the N8 Centre of Computationally Intensive Research for supporting access the GPU-based Bede system for our computational work.

Data availability

No data was used for the research described in the article.

References

[1] S. Jung, S.-Y. Ok, J. Song, Robust structural damage identification based on multi-objective optimization, Internat. J. Numer. Methods Engrg. 81 (6) (2010) 786–804.
[2] M. Wang, J.C. Brigham, Assessment of multi-objective optimization for nondestructive evaluation of damage in structural components, J. Intell. Mater. Syst. Struct. 25 (9) (2014) 1082–1096.
[3] J. Brigham, W. Aquino, F. Mitri, J.F. Greenleaf, M. Fatemi, Inverse estimation of viscoelastic material properties for solids immersed in fluids using vibroacoustic techniques, J. Appl. Phys. 101 (2) (2007) 023509.
[4] J.C. Brigham, W. Aquino, Inverse viscoelastic material characterization using pod reduced-order modeling in acoustic–structure interaction, Comput. Methods Appl. Mech. Engrg. 198 (9–12) (2009) 893–903.
[5] P.E. Barbone, J.C. Bamber, Quantitative elasticity imaging: what can and cannot be inferred from strain images, Phys. Med. Biol. 47 (12) (2002) 2147.
[6] A.A. Oberai, N.H. Gokhale, S. Goenezen, P.E. Barbone, T.J. Hall, A.M. Sommer, J. Jiang, Linear and nonlinear elasticity imaging of soft tissue in vivo: demonstration of feasibility, Phys. Med. Biol. 54 (5) (2009) 1191.
[7] N.H. Gokhale, P.E. Barbone, A.A. Oberai, Solution of the nonlinear elasticity imaging inverse problem: the compressible case, Inverse Problems 24 (4) (2008) 045010.
[8] A. Sabelli, W. Aquino, A source sensitivity approach for source localization in steady-state linear systems, Inverse Problems 29 (9) (2013) 095005.
[9] B.B. Guzina, M. Bonnet, Topological derivative for the inverse scattering of elastic waves, Q. J. Mech. Appl. Math. 57 (2) (2004) 161–179.
[10] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422–440.
[11] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, et al., Deep learning and process understanding for data-driven Earth system science, Nature 566 (7743) (2019) 195–204.
[12] L. Alzubaidi, J. Bai, A. Al-Sabaawi, et al., A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications, J Big Data 10 (2023) 46.
[13] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[14] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, M.W. Mahoney, Characterizing possible failure modes in physics-informed neural networks, Adv. Neural Inf. Process. Syst. 34 (2021).
[15] M. Yin, X. Zheng, J.D. Humphrey, G.E. Karniadakis, Non-invasive inference of thrombus material properties with physics-informed neural networks, Comput. Methods Appl. Mech. Engrg. 375 (2021) 113603.
[16] D. Anton, H. Wessels, Physics-informed neural networks for material model calibration from full-field displacement data, 2022, arXiv preprint arXiv:2212.07723.
[17] E. Zhang, M. Yin, G.E. Karniadakis, Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging, 2020, arXiv preprint arXiv:2009.04525.
[18] E. Kharazmi, Z. Zhang, G.E. Karniadakis, Hp-VPINNs: Variational physics-informed neural networks with domain decomposition, Comput. Methods Appl. Mech. Engrg. 374 (2021) 113547.
[19] S. Wang, Y. Teng, P. Perdikaris, Understanding and mitigating gradient pathologies in physics-informed neural networks, 2020, arXiv preprint arXiv:2001.04536.
[20] J.V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, R.A. Saurous, Tensorflow distributions, 2017, arXiv preprint arXiv:1711.10604.
[21] L. Yang, X. Meng, G.E. Karniadakis, B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, J. Comput. Phys. 425 (2021) 109913.
[22] M. Raissi, P. Perdikaris, G.E. Karniadakis, Machine learning of linear differential equations using Gaussian processes, J. Comput. Phys. 348 (2017) 683–693.
[23] G.R. Feijoo, A new method in inverse scattering based on the topological derivative, Inverse Problems 20 (6) (2004) 1819.
[24] Y. Li, L. Udpa, S.S. Udpa, Three-dimensional defect reconstruction from eddy-current NDE signals using a genetic local search algorithm, IEEE Trans. Magn. 40 (2) (2004) 410–417.
[25] M. Contreras, S. Nagarajaiah, S. Narasimhan, Real time detection of stiffness change using a radial basis function augmented observer formulation, Smart Mater. Struct. 20 (3) (2011) 035013.
[26] M.A. Aguilo, W. Aquino, J.C. Brigham, M. Fatemi, An inverse problem approach for elasticity imaging through vibroacoustics, IEEE Trans. Med. Imaging 29 (4) (2010) 1012–1021.
[27] C.B. Compas, E.Y. Wong, X. Huang, S. Sampath, B.A. Lin, P. Pal, X. Papademetris, K. Thiele, D.P. Dione, M. Stacy, et al., Radial basis functions for combining shape and speckle tracking in 4D echocardiography, IEEE Trans. Med. Imaging 33 (6) (2014) 1275–1289.
[28] A. Gulli, S. Pal, Deep Learning with Keras, Packt Publishing Ltd, 2017.
[29] M. Wang, D. Dutta, K. Kim, J.C. Brigham, A computationally efficient approach for inverse material characterization combining Gappy POD with direct inversion, Comput. Methods Appl. Mech. Engrg. 286 (2015) 373–393.