

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/366339613>

# Physics-Informed Neural Networks for Material Model Calibration from Full-Field Displacement Data

Preprint · December 2022

DOI: 10.48550/arXiv.2212.07723

CITATIONS

0

READS

392

2 authors:



[David Anton](#)

Technische Universität Braunschweig

2 PUBLICATIONS 1 CITATION

[SEE PROFILE](#)



[Henning Wessels](#)

Technische Universität Braunschweig

27 PUBLICATIONS 296 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Convolutional Neural Networks (CNN) for engineering applications [View project](#)



Physics Informed Neural Networks (PINN) in Continuum Micromechanics [View project](#)

# Physics-Informed Neural Networks for Material Model Calibration from Full-Field Displacement Data

David Anton<sup>a,1,\*</sup>, Henning Wessels<sup>a</sup>

<sup>a</sup>*Institute for Computational Modeling in Civil Engineering, Technische Universität Braunschweig, Pockelsstr. 3, Braunschweig, 38106, Germany*

---

## Abstract

The identification of material parameters occurring in constitutive models has a wide range of applications in practice. One of these applications is the monitoring and assessment of the actual condition of infrastructure buildings, as the material parameters directly reflect the resistance of the structures to external impacts. Physics-informed neural networks (PINNs) have recently emerged as a suitable method for solving inverse problems. The advantages of this method are a straightforward inclusion of observation data. Unlike grid-based methods, such as the finite element method updating (FEMU) approach, no computational grid and no interpolation of the data is required. In the current work, we aim to further develop PINNs towards the calibration of the linear-elastic constitutive model from full-field displacement and global force data in a realistic regime. We show that normalization and conditioning of the optimization problem play a crucial role in this process. Therefore, among others, we identify the material parameters for initial estimates and balance the individual terms in the loss function. In order to reduce the dependence of the identified material parameters on local errors in the displacement approximation, we base the identification not on the stress boundary conditions but instead on the global balance of internal and external work. In addition, we found that we get a better posed inverse problem if we reformulate it in terms of bulk and shear modulus instead of Young's modulus and Poisson's ratio. We demonstrate that the enhanced PINNs are capable of identifying material parameters from both experimental one-dimensional data and synthetic full-field displacement data in a realistic regime. Since displacement data measured by, e.g., a digital image correlation (DIC) system is noisy, we additionally investigate the robustness of the method to different levels of noise.

**Keywords:** Model Calibration, Inverse Problems, Physics-Informed Neural Networks, Realistic Data Regime, Structural Health Monitoring

---



---

\*Corresponding author

Email address: [d.anton@tu-braunschweig.de](mailto:d.anton@tu-braunschweig.de) (David Anton )

<sup>1</sup><https://orcid.org/0000-0002-0888-0220>

## 1. Introduction

The identification of material parameters occurring in constitutive models is a major research subject in the field of solid mechanics and has a wide range of applications in practice. Probably the most obvious application is the characterization of unknown materials from experimental data. Another application that motivates this paper is continuous structural health monitoring (SHM). Building structures and materials age during service life due to chemical and physical processes. This in turn leads to a deterioration in both reliability and quality of the structure. At the same time, external impacts to the structure, e.g., traffic loads, often increase more than assumed at design. Therefore, continuous SHM is the prerequisite of a reliable prediction of the remaining service life of an infrastructure building [1, 2]. For an assessment of the actual building condition, it is crucial to consider the actual parameters of the building material. These material parameters indicate damage or material degradation, since they directly reflect the resistance of the structure to external impacts. Since we focus on steel structures in this paper, we assume linear-elastic, isotropic material behavior. The material condition of steel is thus directly reflected by Young’s modulus and Poisson’s ratio. Once the material parameters have been identified, they can be fed into a forward simulation, e.g., a finite element (FE) simulation, to calculate the actual resistance of the infrastructure building. Based on the simulation results, maintenance intervals and reinforcement measures can be derived.

The material parameters of interest can be identified from displacement data, e.g., measured by DIC [3], by solving an inverse problem. The underlying equation of the inverse problem is the balance of linear momentum. Traditionally, this inverse problem is solved by numerical methods, such as the finite element method updating (FEMU) approach or the virtual fields method (VFM) [4]. Recently, it has also been shown that PINNs [5] are particularly suitable for solving inverse problems. PINNs are a framework for solving forward and inverse problems involving nonlinear partial differential equations (PDEs) from the field of physics-informed machine learning [6]. Although the idea behind this method goes back to the 1990s [7, 8], it became applicable only recently due to developments in automatic differentiation [9], software frameworks, such as TensorFlow [10] and JAX [11], and more powerful hardware. In addition to the data, PINNs also exploit the physical laws behind the data. This can improve generalization capability and reduce the amount of required training data. Recently, PINNs have been applied to inverse problems from versatile fields in engineering and science, including the simulation of unsaturated groundwater flow [12], super sonic flows [13], tunneling [14], biomechanics [15], nano-optics [16] and epidemiology [17], to name only a few. The main advantages of PINNs over traditional numerical methods are a straightforward inclusion of observation data and that PINNs can work directly on the strong form of the PDE. Thus, no computational grid is required when using PINNs.

In the context of solid mechanics, *Shukla et al.* [18] applied PINNs to quantify the microstructural properties of polycrystalline nickel using ultrasound data. *Rojas et al.* [19] used PINNs in combination with some classical estimation

methods to identify parameters of a damage model. For the analyses of internal structures and defects, *Zhang et al.* [20] presented a general framework for identifying unknown geometric and material parameters. *Haghighat et al.* [21] proposed a multi-network model for the identification of material parameters from displacement and stress data. They applied their approach to linear elasticity and further extended it to von Mises plasticity. *Hamel et al.* [22] developed a framework for the calibration of constitutive models from full-field displacements and global force-displacement data and demonstrated it by means of hyperelastic material models. Contrary to the conventional PINN approach, they imposed the physical constraints by using the weak form of the PDE. This method, however, does not realize the full potential of PINNs as it relies on a computational grid for integrating the weak form of the PDE. Furthermore, in [23], *Zhang et al.* considered the identification of heterogeneous, incompressible, hyper-elastic material from full-field displacement data using two independent artificial neural networks (ANNs). One ANN was used for the approximation of the displacement field and another ANN for approximating the spatially dependent material parameter.

In summary, the previous contributions have demonstrated that PINNs can in principle be successfully applied to inverse problems in solid mechanics and particular for material parameter identification from displacement data. However, a severe restriction is that the assumptions made in the aforementioned contributions do not match the conditions of real-world inverse problems. This mainly concerns the magnitude and quality of the measured displacements as well as the magnitude of the material parameters to be identified. In the literature, normalized domains and material parameters have been predominantly considered so far. Apart from the simplifying assumptions mentioned above, in real-world applications, we cannot assume that we know the stress state within the domain of interest.

To the best of the authors knowledge, in this paper, the linear-elastic constitutive model is the first time calibrated from full-field displacement data in a realistic regime using PINNs. The enhanced PINN we propose in this contribution works with the strong form of the PDE and does not rely on a computational grid and thus realizes the full potential of PINNs. In contrast, when applying the standard PINN as proposed in [5] without further extensions to the same displacement data, we observed that it fails in solving the inverse problem [24]. Some of the possible failure modes when training PINNs have already been pointed out in the literature. Training of PINNs is a multi-objective optimization problem where a minimum in various loss terms with respect to data and physics is searched. Hence, a trade-off between all loss terms has to be found. The convergence issues caused by the multi-objective optimization problem are investigated in [25, 26]. In order to improve the convergence, loss term weighting [25] and the adaption of the neural tangent kernel [26] were proposed. However, as reported in [27], we found that adaptive loss term weighting does not really improve the training dynamics for problems with many different loss terms, such as those encountered in solid mechanics. The neural tangent kernel is not of much interest in practice, since it involves considerably high computational effort.

Other proposed approaches to improve convergence and stability of PINN training include adaptive activation functions [28, 29], adaptive sampling strategies [30] and self-adaptive PINNs using a soft attention mechanism [31].

In this contribution, we further develop PINN towards the calibration of linear-elastic constitutive models from full-field displacement and global force data in a realistic regime. We demonstrate that the standard PINN fails in identifying material parameters from displacement data in a realistic regime due to ill-posedness of the optimization problem. Based on our observations, we therefore introduce the following extensions:

1. We normalize both the inputs and outputs of the PINN and scale the parameters to be optimized by providing initial estimates for the material parameters. We also condition the loss function by giving more weight to the data loss term. Note that the normalization and scaling are introduced in a way that does not affect the physics.
2. We found that when using stress boundary conditions to account for the force information, the error in the identified material parameters is strongly dependent on the local error of the displacement approximation or the data quality in the boundary region. To account for this issue, we base the material parameter identification on global information and consider the balance law of internal and external work to account for the force situation.
3. We figured out that formulating the constitutive model in terms of bulk and shear modulus leads to more robust results than trying to identify Young’s modulus and Poisson’s ratio directly. Thus, we reformulate the optimization problem with respect to the material parameters to be identified.

We demonstrate that the enhanced PINNs are able to identify the material parameters from both one-dimensional analytical as well as experimental and two-dimensional synthetic displacement data in a realistic regime. In addition, we conduct sensitivity analyses to show that the relative errors of the identified parameters are independent of the initial estimates in a range reasonable for real-world applications. As we approximate the integrals when calculating the internal and external work, we also perform a convergence study with respect to the number of collocation points used. Finally, we investigate the robustness of the enhanced PINNs to different levels of artificial noise imposed to the synthetic displacement data of a plate with a hole.

The remainder of this paper is structured as follows. In Section 2 the governing equations of solid mechanics are reviewed and a brief introduction to ANNs and particularly PINNs is provided. The further developments of the PINNs towards realistic data are presented in Section 3. In Section 4, the proposed PINN is first validated using one-dimensional analytical and experimental displacement data. Subsequently, in Section 5, the enhanced PINN is applied to clean as well as artificially noisy, synthetic displacement data of a plate with a hole. Finally, we conclude our investigations and point out possible directions of further research in Section 6.

## 2. Preliminaries

In this section, we provide an overview of the governing equations and introduce the general notation of ANNs and PINNs which are essential for the following sections.

### 2.1. Governing equations

In computational mechanics, modelling usually starts with the balance of linear momentum. Assuming a static and stationary setting, the balance of linear momentum at the material point  $\mathbf{x}$  in the domain  $\Omega$  can be derived as

$$\operatorname{div} \boldsymbol{\sigma}(\mathbf{x}) + \rho(\mathbf{x}) \mathbf{b}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} \in \Omega, \quad (1)$$

where  $\boldsymbol{\sigma}(\mathbf{x})$  denotes the Cauchy stress tensor,  $\rho(\mathbf{x})$  the density and  $\mathbf{b}(\mathbf{x})$  contains accelerations related to external body forces, such as gravity. In order to close the above equation, the Cauchy stress tensor  $\boldsymbol{\sigma}(\mathbf{x})$  is expressed as a function of strain by a constitutive model via the kinematics.

For an isotropic, linear-elastic material, the constitutive model states

$$\boldsymbol{\sigma}(\mathbf{x}) = \frac{E}{1+\nu} \left( \boldsymbol{\varepsilon}(\mathbf{x}) + \frac{\nu}{1-2\nu} (\operatorname{tr} \boldsymbol{\varepsilon}(\mathbf{x})) \mathbf{I} \right). \quad (2)$$

The strain tensor  $\boldsymbol{\varepsilon}(\mathbf{x})$  can be expressed by the kinematic law in terms of the primary variables, which in solid mechanics are the displacements  $\mathbf{u}(\mathbf{x})$ , as

$$\boldsymbol{\varepsilon}(\mathbf{x}) = \frac{1}{2} \left( \operatorname{grad} \mathbf{u}(\mathbf{x}) + \operatorname{grad}^T \mathbf{u}(\mathbf{x}) \right). \quad (3)$$

Constitutive modeling comes along with the introduction of material parameters  $\boldsymbol{\kappa}$ . According to Eq. (2), the material is characterized by the Young's modulus  $E$  and Poisson's ratio  $\nu$ , so that  $\boldsymbol{\kappa} = [E, \nu]$ . Under the same assumptions, in one dimension, the constitutive model in Eq. (2) simplifies to

$$\sigma(x) = E \varepsilon(x). \quad (4)$$

with  $\kappa = E$ . In a two-dimensional setting, equivalently to Eq. (2), the constitutive model can also be formulated in terms of bulk modulus  $K$  and shear modulus  $G$  with  $\boldsymbol{\kappa} = [K, G]$ , such that

$$\boldsymbol{\sigma}(\mathbf{x}) = 2G\boldsymbol{\varepsilon}_D(\mathbf{x}) + 2K\boldsymbol{\varepsilon}_V(\mathbf{x}), \quad (5)$$

where  $\boldsymbol{\varepsilon}_V(\mathbf{x}) = \operatorname{tr} \boldsymbol{\varepsilon}(\mathbf{x}) / 2 \mathbf{I}$  is the volumetric part of the strain tensor and  $\boldsymbol{\varepsilon}_D(\mathbf{x}) = \boldsymbol{\varepsilon}(\mathbf{x}) - \boldsymbol{\varepsilon}_V(\mathbf{x})$  the deviatoric part. In plane strain state, the relation between these two sets of material parameters is given by

$$\hat{K} = \frac{E}{2(1+\nu)(1-2\nu)}, \quad G = \frac{E}{2(1+\nu)}. \quad (6)$$

Note that  $\hat{K}$  is the modified bulk modulus for a two-dimensional manifold embedded in three-dimensional space in plane strain state.

Substituting the stress in Eq. (1) by the constitutive model in Eq. (2) or Eq. (5), we obtain a parameterized system of PDEs

$$\begin{aligned}\operatorname{div} \boldsymbol{\sigma}(\mathbf{x}, \boldsymbol{\kappa}) + \rho(\mathbf{x}) \mathbf{b}(\mathbf{x}) &= \mathbf{0}, \quad \mathbf{x} \in \Omega, \\ \boldsymbol{\sigma}(\mathbf{x}_N) \cdot \mathbf{n}(\mathbf{x}_N) &= \bar{\mathbf{t}}(\mathbf{x}_N) \quad \mathbf{x}_N \in \Gamma_N \subset \partial\Omega, \\ \mathbf{u}(\mathbf{x}_D) &= \bar{\mathbf{u}}(\mathbf{x}_D) \quad \mathbf{x}_D \in \Gamma_D \subset \partial\Omega,\end{aligned}\tag{7}$$

where  $\Omega$  is the domain,  $\partial\Omega = \Gamma_N \cup \Gamma_D$  its boundary,  $\mathbf{n}(\mathbf{x})$  the normal vector and  $\bar{\mathbf{t}}(\mathbf{x})$  and  $\bar{\mathbf{u}}(\mathbf{x})$  are the boundary functions for the Neumann and Dirichlet boundary condition, respectively. Finally, Eq. (7) demonstrates that the relation between the displacements and the material parameters can be established by the balance of linear momentum from Eq. (1).

In addition to the balance of linear momentum, the balance of internal and external work applies. The internal work corresponds to the strain energy and is defined as

$$W_{int} = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{x}) : \boldsymbol{\sigma}(\mathbf{x}) d\Omega,\tag{8}$$

and the external work is composed of a surface and a volume part and can be calculated from

$$W_{ext} = \frac{1}{2} \int_{\partial\Omega} \bar{\mathbf{t}}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) d\partial\Omega + \frac{1}{2} \int_{\Omega} \mathbf{b}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) d\Omega.\tag{9}$$

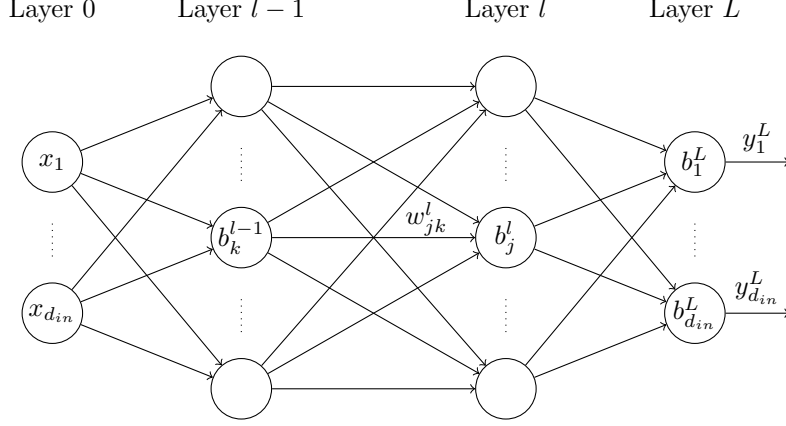
Assuming that the loads are static and no energy is dissipated in the form of heat, the balance of internal and external work thus states

$$\begin{aligned}W_{int} &= W_{ext}, \\ \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{x}) : \boldsymbol{\sigma}(\mathbf{x}) d\Omega &= \frac{1}{2} \int_{\partial\Omega} \bar{\mathbf{t}}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) d\partial\Omega + \frac{1}{2} \int_{\Omega} \mathbf{b}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) d\Omega.\end{aligned}\tag{10}$$

## 2.2. Artificial Neural Networks

Artificial neural networks (ANNs) are global, smooth function approximators defining a mapping  $\mathbb{R}^{d_{in}} \rightarrow \mathbb{R}^{d_{out}}$  from an input tensor to an output tensor. The computational units of an ANN are called neurons and are typically arranged in an input, an output and any number of hidden layers. In the following, we consider fully connected feed-forward neural networks (FFNNs) with  $L+1$  layers. These FFNNs each have in total  $L-1$  hidden layers, where layer 0 is the input layer and layer  $L$  the output layer. In a fully connected FFNN, the neurons of each two successive layers are connected. The weight of the connection between neuron  $k$  in layer  $l-1$  and neuron  $j$  in layer  $l$  is denoted by  $w_{jk}^l$ . All weights between layer  $l-1$  and  $l$  can then be combined in the weight matrix  $\mathbf{W}^l$ , whose entries are  $w_{jk}^l$ . In addition, the neurons in the hidden layers and the output layer each have a bias, where  $b_j^l$  denotes the bias of neuron  $j$  in layer  $l$ . Similarly, the biases of all neurons in layer  $l$  can be combined in vector  $\mathbf{b}^l$  with entries  $b_j^l$ . Thus, the weights and biases of all neurons are the trainable parameters of the ANN  $\boldsymbol{\theta} = \{\mathbf{W}^l, \mathbf{b}^l\}_{1 \leq l \leq L}$ . The schematic structure of an ANN is illustrated in

Fig. 1. In the forward pass, the output of the neurons in the hidden layers and the output layer are computed from the sum of their weighted inputs and their bias as an argument of an activation function.



**Fig. 1:** Schematic representation of a fully connected feed-forward neural network (FFNN) according to [32].

The mapping from an input to an output vector by a FFNN can be formulated recursively according to [32]. The weighted input  $z_j^l$  of neuron  $j$  in the hidden layer or output layer  $l$  with an upstream layer consisting of  $N_{l-1}$  neurons is defined as

$$z_j^l = \sum_{k=1}^{N_{l-1}} w_{jk}^l y_k^{l-1} + b_j^l, \quad \forall l \in [1, L]. \quad (11)$$

In Eq. (11),  $y_k^{l-1}$  is the output of neuron  $k$  in the upstream layer  $l-1$  given by

$$y_k^{l-1} = \phi_k^{l-1}(z_k^{l-1}). \quad (12)$$

Here,  $\phi_k^{l-1}$  is the activation function of neuron  $k$  in layer  $l-1$ . Throughout this paper, for all neurons in a layer the same activation function is used, denoted by  $\phi^l$  for layer  $l$ . Inserting Eq. (12) in Eq. (11) we obtain in symbolic notation

$$\mathbf{z}^l = \mathbf{W}^l \boldsymbol{\phi}^{l-1}(\mathbf{z}^{l-1}) + \mathbf{b}^l, \quad \forall l \in [1, L], \quad (13)$$

where  $\mathbf{z}^{l-1}$  contains the weighted inputs of all neurons in layer  $l-1$  and  $\boldsymbol{\phi}^{l-1}$  is applied elementwise. In the hidden layers, nonlinear functions, such as hyperbolic tangent, are usually used as activation functions. The activation of the output neurons is typically computed by the identity function. Since we do not use any activation function in the input layer, for the input layer  $l=0$  applies

$$\boldsymbol{\phi}^0(\mathbf{z}^0) = \mathbf{x}, \quad (14)$$



where  $\mathbf{x}$  is the input vector to the ANN. Given Eqs. (11)–(14), the output vector  $\mathbf{y}^L$  of the FFNN as a function of  $\mathbf{x}$  can be defined recursively as follows:

$$\begin{aligned}
\mathbf{y}^L &= \phi^L(\mathbf{z}^L) \\
\mathbf{z}^L &= \mathbf{W}^L \phi^{L-1}(\mathbf{z}^{L-1}) + \mathbf{b}^L \\
\mathbf{z}^{L-1} &= \mathbf{W}^{L-1} \phi^{L-2}(\mathbf{z}^{L-2}) + \mathbf{b}^{L-1} \\
&\vdots \\
\mathbf{z}^2 &= \mathbf{W}^2 \phi^1(\mathbf{z}^1) + \mathbf{b}^2 \\
\mathbf{z}^1 &= \mathbf{W}^1 \mathbf{x} + \mathbf{b}^1.
\end{aligned} \tag{15}$$

The definition in Eq. (15) demonstrates that ANNs are highly parameterized, nonlinear, composed functions.

The training of ANNs is an optimization problem. The objective of the optimization problem is the loss function  $\mathcal{L}(\boldsymbol{\theta}; \mathcal{T})$  which depends both on the trainable ANN parameters  $\boldsymbol{\theta}$  and the training data  $\mathcal{T}$ . For the optimization of the ANN parameters  $\boldsymbol{\theta}$ , usually gradient based optimization algorithms are used. The required gradient of the loss function with respect to the ANN parameters  $\boldsymbol{\theta}$  can be calculated using automatic differentiation [9].

It is well known that ANNs are universal function approximators [33, 34, 35]. Provided an ANN has a sufficient number of parameters, then according to the universal approximation theorem the ANN can theoretically approximate any continuous function and its derivatives to an arbitrarily small error with mild assumptions on the activation function only. It should be noted, however, that the question of optimal training of ANNs has not yet been solved. For a more in-depth introduction to deep learning, the reader is referred to standard text books, e.g., [36].

### 2.3. Physics-Informed Neural Networks

Physics-informed neural networks (PINNs) are a deep learning framework for solving forward and inverse problems involving PDEs recently emerged in the field of scientific machine learning [5]. The key characteristic of PINNs is the design of the loss function that enables leveraging physical knowledge during the training process. This succeeds by directly including the governing PDEs into the loss function as regularizing terms. The PINN acts as a global function approximator of the hidden solution of the PDE. Due to the regularizing terms in the loss function, the solution space is constrained and the ANN is enforced to satisfy the governing PDEs. For a more detailed overview of scientific machine learning, please refer to [6].

Although PINNs can be applied to more general problems [6], in the course of this paper we consider stationary elliptic, nonlinear PDEs. Following the notation from [37], these PDEs can be formulated as

$$\begin{aligned}
\mathcal{D}[\mathbf{u}(\mathbf{x}); \boldsymbol{\kappa}] &= \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \Omega, \\
\mathcal{B}_k[\mathbf{u}(\mathbf{x})] &= \mathbf{g}_k(\mathbf{x}), \quad \mathbf{x} \in \Gamma_k \subset \partial\Omega,
\end{aligned} \tag{16}$$

where  $\mathcal{D}$  is a differential operator acting on  $\mathbf{u}(\mathbf{x})$  and parameterized by  $\boldsymbol{\kappa}$ ,  $\mathbf{u}(\mathbf{x}) \in \mathbb{R}^{d_{out}}$  is the hidden solution of the PDE with  $\mathbf{x}$  representing spatial points in the domain  $\Omega \in \mathbb{R}^{d_{in}}$  and  $\mathbf{f}(\mathbf{x})$  is a forcing term. The system of PDEs is further defined by a set of suitable boundary condition operators  $\mathcal{B}_k$  acting on  $\mathbf{u}(\mathbf{x})$  and a set of boundary functions  $\mathbf{g}_k(\mathbf{x})$  which are imposed on the corresponding part  $\Gamma_k \subset \partial\Omega$  on the boundary with  $k = 1, \dots, n_{bc}$ . With Eq. (16), a general formulation of a variety of physical problems is provided.

When solving PDEs using PINNs, the PDEs are directly embedded into the loss function. Defining the loss function starts with approximating the hidden solution  $\mathbf{u}(\mathbf{x})$  by an ANN  $\mathcal{N}(\mathbf{x}; \boldsymbol{\theta})$ , such that

$$\mathbf{u}(\mathbf{x}) \approx \mathcal{N}(\mathbf{x}; \boldsymbol{\theta}). \quad (17)$$

Additionally, in the inverse setting, the PDE parameters  $\boldsymbol{\kappa}$  become trainable parameters and are optimized during the PINN training alongside the ANN parameters  $\boldsymbol{\theta}$  [5]. Thus, the loss function introduced in Section 2.2 depends on both  $\boldsymbol{\theta}$  and  $\boldsymbol{\kappa}$  as well as the training data  $\mathcal{T}$  and is defined as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\kappa}; \mathcal{T}) := \lambda_{pde} \mathcal{L}_{pde}(\boldsymbol{\theta}, \boldsymbol{\kappa}; \mathcal{T}_{pde}) + \mathcal{L}_{bc}(\boldsymbol{\theta}, \boldsymbol{\kappa}; \mathcal{T}_{bc}) + \lambda_o \mathcal{L}_o(\boldsymbol{\theta}; \mathcal{T}_o). \quad (18)$$

The loss terms  $\mathcal{L}_{pde}$ ,  $\mathcal{L}_{bc}$  and  $\mathcal{L}_o$  penalize the residual of the approximation  $\mathcal{N}(\mathbf{x}; \boldsymbol{\theta})$  with respect to the PDE, the boundary condition data and the observation data, respectively, and are defined as

$$\begin{aligned} \mathcal{L}_{pde}(\boldsymbol{\theta}, \boldsymbol{\kappa}; \mathcal{T}_{pde}) &= \frac{1}{N_{pde}} \sum_{i=1}^{N_{pde}} \|\mathcal{D}[\mathcal{N}(\mathbf{x}_i; \boldsymbol{\theta}); \boldsymbol{\kappa}] - \mathbf{f}(\mathbf{x}_i)\|_2^2, \\ \mathcal{L}_{bc}(\boldsymbol{\theta}, \boldsymbol{\kappa}; \mathcal{T}_{bc}) &= \lambda_{bc}^k \sum_k \frac{1}{N_{bc}^k} \sum_{j=1}^{N_{bc}^k} \|\mathcal{B}_k[\mathcal{N}(\mathbf{x}_j^k; \boldsymbol{\theta}); \boldsymbol{\kappa}] - \hat{\mathbf{u}}_j^k\|_2^2, \\ \mathcal{L}_o(\boldsymbol{\theta}; \mathcal{T}_o) &= \frac{1}{N_o} \sum_{l=1}^{N_o} \|\mathcal{N}(\mathbf{x}_l; \boldsymbol{\theta}) - \hat{\mathbf{u}}_l\|_2^2. \end{aligned} \quad (19)$$

Here, the training data  $\mathcal{T}$  is comprised of three sets,  $\mathcal{T}_{pde}$ ,  $\mathcal{T}_{bc}$  and  $\mathcal{T}_o$ . We refer to  $\mathcal{T}_{pde}$  as the  $N_{pde}$  collocation points  $\{\mathbf{x}_i\}_{i=1}^{N_{pde}}$  sampled from the domain  $\Omega$ . Furthermore,  $\mathcal{T}_{bc}$  is comprised of  $k$  subsets, each with  $N_{bc}^k$  training points  $\{\mathbf{x}_j^k, \hat{\mathbf{u}}_j^k\}_{j=1}^{N_{bc}^k}$  with  $\hat{\mathbf{u}}_j^k = \mathbf{g}_k(\mathbf{x}_j^k)$  sampled from the boundary  $\Gamma_k$  associated with the boundary condition.  $\mathcal{T}_o$  is set of  $N_o$  data points  $\{\mathbf{x}_l, \hat{\mathbf{u}}_l\}_{l=1}^{N_o}$  located in the domain  $\Omega$  for which observations of the hidden function  $\hat{\mathbf{u}}_l = \mathbf{u}(\mathbf{x}_l)$  are available. The different loss terms  $\mathcal{L}_{pde}$ ,  $\mathcal{L}_{bc}$  and  $\mathcal{L}_o$  can additionally be weighted by  $\lambda_{pde}$ ,  $\lambda_{bc}^k$  and  $\lambda_o$ , where  $\lambda_{bc}^k$  can in principle be different for each  $k$ . For calculating  $\mathcal{L}_{pde}(\boldsymbol{\theta}, \boldsymbol{\kappa}; \mathcal{T}_{pde})$  in Eq. (19), the partial derivatives of the ANN outputs with respect to the inputs are calculated using automatic differentiation [9].

For inverse problems, the optimization problem is to find the optimal PINN parameters  $\boldsymbol{\theta}^*$  and material parameters  $\boldsymbol{\kappa}^*$ :

$$[\boldsymbol{\theta}^*, \boldsymbol{\kappa}^*] = \arg \min_{\boldsymbol{\theta}, \boldsymbol{\kappa}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\kappa}; \mathcal{T}). \quad (20)$$

Solving the problem defined in Eq. (20) is called training. From Eq. (20), it can be seen that the PDE parameters  $\kappa$  are estimated while simultaneously the forward solution is approximated by the ANN. Thus, for inverse problems, PINNs follow the all-at-once approach. All-at-once approaches are outlined, e.g., in [38], [39] and in [40].

### 3. Further development of PINNs towards a realistic data regime

It is shown that standard PINNs, as presented in Section 2.3, fail in identifying material parameters from displacement data in a realistic regime without further modifications. Previous work, including [21] and [23], among others, have already proven PINNs to be generally capable of identifying material parameters from full-field displacement data. However, for this purpose, simplified assumptions were made which often do not match real-world conditions. These assumptions concern the domain size, the availability of stress data, the boundary conditions and the magnitude of both displacements and material parameters as well as inevitable noise in the data. In the following, we elaborate the necessary developments of the PINN approach, which essentially aim at conditioning the optimization problem and presenting it in an appropriate formulation with respect to the goal of material model calibration. Some of the extensions described below were first introduced in a similar form in [24]. In this paper, we present further improvements and demonstrate the applicability in the two-dimensional regime. We expect that an extension to the three-dimensional regime is straightforward, however it is difficult to measure three-dimensional displacement data.

#### 3.1. Normalization of inputs and outputs

For PINNs to be able to approximate the displacement field from given experimental displacement data, both the inputs and outputs of the PINN must be normalized. It is well known that the convergence of ANN training can be accelerated by normalizing its inputs, where the mean values of all input features should each be close to zero according to [41]. In addition, we found that in our use case, the output of the ANN also needs to be normalized. Therefore, we approximate the hidden solution in Eq. (7) by the normalized ANN  $\tilde{\mathcal{N}}$ , defined as

$$\tilde{\mathcal{N}}(\mathbf{x}) := \mathbf{T}^{out}(\mathcal{N}(\mathbf{T}^{in}(\mathbf{x}); \boldsymbol{\theta})). \quad (21)$$

Here, we first map the real inputs to the range  $[-1, 1]$  and forward them to the ANN. We then retransform the ANN outputs back to the real output data range and thus force the ANN to predict the outputs in the range  $[-1, 1]$ . The elementwise linear transformation  $\mathbf{T}^{in}$  and re-transformation  $\mathbf{T}^{out}$  are defined as

$$\begin{aligned} T_i^{in}(x_i) &= 2 \left( \frac{x_i - x_i^{min}}{x_i^{max} - x_i^{min}} \right) - 1, \quad \forall i \in [1, d_{in}], \\ T_j^{out}(\hat{u}_j) &= (\hat{u}_j^{max} - \hat{u}_j^{min}) \frac{\hat{u}_j + 1}{2} + \hat{u}_j^{min}, \quad \forall j \in [1, d_{out}], \end{aligned} \quad (22)$$

The transformation and re-transformation is based on  $x_i^{min}$ ,  $x_i^{max}$ ,  $\hat{u}_j^{min}$  and  $\hat{u}_j^{max}$  which are the minimum and maximum values of the PINN inputs and outputs for dimension  $i$ , respectively, and can be taken from the training data. As a consequence, the PINN learns the mapping between the normalized inputs and the normalized outputs instead of learning the mapping between the real inputs and the real outputs. Applying the chain and product rule, we obtain the first and second derivatives of the output of the normalized ANN with respect to the inputs in index notation:

$$\begin{aligned}\frac{d\bar{\mathcal{N}}_i}{dx_j} &= \frac{\partial T_i^{out}}{\partial \mathcal{N}_i} \frac{\partial \mathcal{N}_i}{\partial T_j^{in}} \frac{dT_j^{in}}{dx_j}, \\ \frac{d^2\bar{\mathcal{N}}_i}{dx_j dx_k} &= \frac{\partial T_i^{out}}{\partial \mathcal{N}_i} \frac{\partial^2 \mathcal{N}_i}{\partial T_j^{in} \partial T_k^{in}} \frac{dT_j^{in}}{dx_j} \frac{dT_k^{in}}{dx_k}.\end{aligned}\tag{23}$$

It is important to emphasize that at any time the real inputs are fed into the normalized ANN. Likewise, the normalized ANN always outputs the real outputs. Thus, the physics are not violated by the normalization when the outputs of the PINN are derived according to the inputs during training.

### 3.2. Conditioning of loss function

We also found that the loss function is ill-conditioned for small displacements, which prevents the PINN to accurately approximate the displacement field. In real-world problems, the measured displacements may be in the order of magnitude of  $10^{-3}$  mm or less. With displacements of this magnitude, even a relative error  $RE = 100\%$  of the approximation by the PINN results in a data loss in the order of magnitude of  $10^{-6}$  according to Eq. (19). The small losses result in small gradients of the loss with respect to the ANN parameters to be optimized. This in turn makes gradient based optimization difficult. Therefore, we first introduce the relative mean squared error. Applying this error metric in the data loss term in Eq. (19), we obtain

$$\mathcal{L}_o(\boldsymbol{\theta}; \mathcal{T}_o) = \frac{1}{N_o} \sum_{l=1}^{N_o} \|(\bar{\mathcal{N}}(\mathbf{x}_l; \boldsymbol{\theta}) - \hat{\mathbf{u}}_l) \oslash \mathbf{u}^{char}\|_2^2,\tag{24}$$

where  $\mathbf{u}^{char}$  is a vector of characteristic displacements and  $\oslash$  the Hadamard division operator. We propose to choose the characteristic displacements as the mean displacements from the training data set:

$$\mathbf{u}^{char} = \frac{1}{N_o} \sum_i^{N_o} \hat{\mathbf{u}}_i.\tag{25}$$

The displacement may have different orders of magnitudes in the different dimensions. As a result, the contributions of the different dimensions to the data loss would vary depending on their magnitude. The relative mean squared error takes this into account and adaptively equalizes the contributions to the same

level. Second, in order to increase the impact of the total data loss term  $\mathcal{L}_o$  and to balance the loss terms, it is necessary to increase the weight  $\lambda_o$  in Eq. (18). While the relative mean squared error is adaptive, the loss term weight currently has to be selected manually. Care must be taken to ensure that, on the one hand, the data loss term is given sufficient priority, but on the other hand, overfitting to the data is prevented. A proper conditioning of the data loss term in combination with normalization allows PINNs to approximate even displacement fields of small magnitude. Note, however, that an accurate approximation of the displacement field is a necessary, but not a sufficient, criterion for identifying material parameters from realistic displacement data.

### 3.3. Scaling of the parameters to be optimized

Due to different scales of the material and ANN parameters, the calibration with the standard PINN is not feasible for the optimizer. We examined the range of the optimized ANN parameters  $\theta$  for the standard PINN and found that for the analytical example in Section 4 these are in the range  $[-375.26, 260.72]$ , also see [24]. In contrast, the material parameters to be identified for construction steel are approximately  $\nu = 0.3$  and  $E = 210,000 \frac{\text{N}}{\text{mm}^2}$  or  $K = 201,923.08 \frac{\text{N}}{\text{mm}^2}$  and  $G = 80,769.23 \frac{\text{N}}{\text{mm}^2}$ . Since both the ANN and the material parameters are optimized simultaneously, this leads to a poorly scaled optimization problem [42]. We also assume that the optimizer encounters many local minima, which complicates the optimization. In order to obtain a similar scale for all parameters to be optimized, the real material parameters in the constitutive model will be replaced by

$$\kappa = (1.0 + \alpha_\kappa)\kappa_{est}. \quad (26)$$

Here  $\alpha_\kappa$  and  $\kappa_{est}$  are a correction factor and an initial estimate for the material parameter to be identified, respectively. From now on, the correction factor is identified instead of the material parameter. If we provide the exact Young's modulus, then the optimized parameters of the enhanced PINN  $\theta$  are now in the range  $[-5.3770, 4.2691]$  and the correction factor  $\alpha_\kappa$  very close to 1, as we will see in Section 4. Thus, the parameters to be optimized have a similar scale and the optimization problem is well posed. Unless otherwise specified, the correction factors are initialized with zeros throughout this paper.

### 3.4. Problem reformulation

The accuracy as well as the robustness of the identified material parameters strongly depend on how the optimization problem is formulated or with respect to which quantities the optimization problem is to be solved. The advantage of Young's modulus and Poisson's ratio over other types of elasticity parameters, such as bulk and shear modulus, are that they are often easier to interpret. However, in attempting to identify them directly, we observed that the results are less accurate and, more importantly, less robust than when the constitutive model is formulated with respect to bulk and shear modulus. We made this observation

for the case where we provide an estimate of  $\nu_{est} = 0.45$  for Poisson's ratio. For some random initializations of the PINN parameters, this estimate results in a failure of the calibration with respect to Young's modulus and Poisson's ratio. Therefore, we reformulated the optimization problem in terms of the material parameters to be identified. We first converted the estimated Young's modulus and Poisson's ratio into the corresponding estimates for bulk and shear modulus according to Eq. (6). With the scaling of the parameters proposed in Section 3.3, the modified material parameters in Eq. (7) then become  $\bar{\kappa} = [\alpha_K, \alpha_G]$ . These are optimized along with the PINN parameters  $\theta$ . Correspondingly, in one-dimensional problems, the modified material parameter is  $\bar{\kappa} = \alpha_E$ . Finally, the identified bulk and shear modulus are converted back to Young's modulus and Poisson's ratio. We assume that this formulation of the optimization problem is better posed because it introduces a constraint on the Poisson's ratio. As soon as the Poisson's ratio approaches the incompressibility limit of  $\nu = 0.5$ , the bulk modulus would approach infinity. Critical values  $\nu \geq 0.5$  can thus be prevented and the optimization becomes more robust as a result.

### 3.5. Introduce balance of internal and external work

In order to be able to identify material parameters, we need to account either for the global or at least some local force information. As soon as no volume force is present, at least some information on the force boundary conditions must be taken into account so that the material parameter is uniquely identifiable. Otherwise, the inverse problem would be ill-posed. There would exist an infinite number of combinations of material parameters and force situations, which lead to the same displacement field. In the standard PINN formulation, the accuracy of the material parameters depends strongly on the accuracy of the displacement field approximation in regions where the stress boundary conditions are applied. A large error in the approximation near the boundaries then inevitably leads to a large error in the identified material parameters. This can cause problems especially when dealing with erroneous or noisy data.

In order to remove this local dependency, instead of the stress boundary conditions, we include the balance of internal and external work in the loss function as supposed in [43]. According to Eq. (10), we enforce the balance of internal and external work by the loss term

$$\mathcal{L}_W(\theta, \bar{\kappa}; \mathcal{T}_W) = W_{int}(\theta, \bar{\kappa}; \mathcal{T}_{W_{int}}) - W_{ext}(\theta; \mathcal{T}_{W_{ext}}), \quad (27)$$

where the training data  $\mathcal{T}_W$  is comprised of  $\mathcal{T}_{W_{int}}$  and  $\mathcal{T}_{W_{ext}}$ . The internal energy can be approximated by

$$W_{int}(\theta, \bar{\kappa}; \mathcal{T}_{W_{int}}) \approx \frac{1}{2} \frac{V_\Omega}{N_{W_{int}}} \sum_{m=1}^{N_{W_{int}}} \varepsilon(\bar{\mathcal{N}}(\mathbf{x}_m; \theta)) : \sigma(\bar{\mathcal{N}}(\mathbf{x}_m; \theta); \kappa), \quad (28)$$

where  $\mathcal{T}_{W_{int}}$  is a set of  $N_{W_{int}}$  collocation points  $\{\mathbf{x}_m\}_{m=1}^{N_{W_{int}}}$  sampled from the domain  $\Omega$ . In one and two dimensions, the volume  $V_\Omega$  corresponds to the length and the area of the domain  $\Omega$ , respectively. The first derivative of the PINN

output with respect to its inputs, which is needed to calculate the strains  $\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}))$  from Eq. (3), can again be calculated using automatic differentiation. In the absence of body forces, the external energy can be approximated by

$$W_{ext}(\boldsymbol{\theta}; \mathcal{T}_{W_{ext}}) \approx \frac{1}{2} \frac{V_{\partial\Omega}}{N_{W_{ext}}} \sum_{n=1}^{N_{W_{ext}}} \bar{\mathcal{N}}(\mathbf{x}_n; \boldsymbol{\theta}) \cdot \mathbf{t}(\mathbf{x}_n) \quad (29)$$

Here,  $\mathcal{T}_{W_{ext}}$  is a set of  $N_{W_{ext}}$  collocation points  $\{\mathbf{x}_n\}_{n=1}^{N_{W_{ext}}}$  sampled from the boundary  $\partial\Omega$ . We assume that the traction  $\mathbf{t}(\mathbf{x})$  is known at least for the collocation points in  $\mathcal{T}_{W_{ext}}$  and that the collocation points are distributed as evenly as possible in the domain. Thus, we shift the dependence of the material parameter identification from the local accuracy of the displacement approximation in the boundary regions to the accuracy of the whole domain.

### 3.6. Summary of the method developments

Based on the general PINN formulation reviewed in Section 2.3 and the extensions introduced above, the PINN and the associated optimization problem for calibrating the linear-elastic material model are defined as follows: In two dimensions, the displacement field is approximated by two independent ANNs, such that

$$\mathbf{u}(\mathbf{x}) \approx \bar{\mathcal{N}}(\mathbf{x}; \bar{\boldsymbol{\theta}}) \approx \begin{bmatrix} \bar{\mathcal{N}}_x(\mathbf{x}; \bar{\boldsymbol{\theta}}_x) \\ \bar{\mathcal{N}}_y(\mathbf{x}; \bar{\boldsymbol{\theta}}_y) \end{bmatrix}, \quad (30)$$

with  $\bar{\boldsymbol{\theta}} = [\bar{\boldsymbol{\theta}}_x, \bar{\boldsymbol{\theta}}_y]$ . According to [21], this leads to a more accurate approximation then using one ANN with two outputs, since the cross-dependency between the outputs may not be accurately represented by a single network. This cross-dependency results from the kinematic law in Eq. (3) and the constitutive model. The loss function of the inverse problem is given by

$$\begin{aligned} \mathcal{L}(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\kappa}}; \mathcal{T}) &:= \mathcal{L}_{pde}(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\kappa}}; \mathcal{T}_{pde}) + \mathcal{L}_W(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\kappa}}; \mathcal{T}_W) \\ &\quad + \lambda_o \mathcal{L}_{o_x}(\boldsymbol{\theta}_x; \mathcal{T}_o) + \lambda_o \mathcal{L}_{o_y}(\boldsymbol{\theta}_y; \mathcal{T}_o), \end{aligned} \quad (31)$$

with the loss terms defined as

$$\begin{aligned} \mathcal{L}_{pde}(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\kappa}}; \mathcal{T}_{pde}) &= \frac{1}{N_{pde}} \sum_{i=1}^{N_{pde}} \left\| \operatorname{div} \boldsymbol{\sigma}(\bar{\mathcal{N}}(\mathbf{x}_i; \bar{\boldsymbol{\theta}}); \bar{\boldsymbol{\kappa}}) \right\|_2^2, \\ \mathcal{L}_W(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\kappa}}; \mathcal{T}_W) &= \frac{1}{2} \frac{V_\Omega}{N_{W_{int}}} \sum_{m=1}^{N_{W_{int}}} \boldsymbol{\varepsilon}(\bar{\mathcal{N}}(\mathbf{x}_m; \bar{\boldsymbol{\theta}})) : \boldsymbol{\sigma}(\bar{\mathcal{N}}(\mathbf{x}_m; \bar{\boldsymbol{\theta}}); \bar{\boldsymbol{\kappa}}) \\ &\quad - \frac{1}{2} \frac{V_{\partial\Omega}}{N_{W_{ext}}} \sum_{n=1}^{N_{W_{ext}}} \bar{\mathcal{N}}(\mathbf{x}_n; \bar{\boldsymbol{\theta}}) \cdot \mathbf{t}(\mathbf{x}_n), \\ \mathcal{L}_{o_x}(\boldsymbol{\theta}_x; \mathcal{T}_o) &= \frac{1}{N_o} \sum_{l=1}^{N_o} \left\| \frac{\mathcal{N}_x(\mathbf{x}_l; \boldsymbol{\theta}_x) - \hat{u}_x^l}{u_x^{char}} \right\|_2^2, \\ \mathcal{L}_{o_y}(\boldsymbol{\theta}_y; \mathcal{T}_o) &= \frac{1}{N_o} \sum_{l=1}^{N_o} \left\| \frac{\mathcal{N}_y(\mathbf{x}_l; \boldsymbol{\theta}_y) - \hat{u}_y^l}{u_y^{char}} \right\|_2^2, \end{aligned} \quad (32)$$

where  $[\hat{u}_x^l, \hat{u}_y^l]^T = \mathbf{u}(x_l)$  is known from the observation data.

In one dimension, the displacement is approximated by only one ANN, such that

$$u(x) \approx \bar{\mathcal{N}}(x; \boldsymbol{\theta}), \quad (33)$$

and the loss function simplifies to

$$\mathcal{L}(\boldsymbol{\theta}, \bar{\kappa}; \mathcal{T}) := \mathcal{L}_{pde}(\boldsymbol{\theta}, \bar{\kappa}; \mathcal{T}_{pde}) + \mathcal{L}_W(\boldsymbol{\theta}, \bar{\kappa}; \mathcal{T}_W) + \lambda_o \mathcal{L}_o(\boldsymbol{\theta}; \mathcal{T}_o) \quad (34)$$

Correspondingly, the loss terms are then defined as

$$\begin{aligned} \mathcal{L}_{pde}(\boldsymbol{\theta}, \bar{\kappa}; \mathcal{T}_{pde}) &= \frac{1}{N_{pde}} \sum_{i=1}^{N_{pde}} \left( E \frac{\partial}{\partial^2 x} \bar{\mathcal{N}}(x_i; \boldsymbol{\theta}) \right)^2, \\ \mathcal{L}_W(\boldsymbol{\theta}, \bar{\kappa}; \mathcal{T}_W) &= \frac{1}{2} \frac{V_\Omega}{N_{W_{int}}} \sum_{m=1}^{N_{W_{int}}} \varepsilon(\bar{\mathcal{N}}(x_m; \boldsymbol{\theta})) \cdot \sigma(\bar{\mathcal{N}}(x_m; \boldsymbol{\theta}); \bar{\kappa}) \\ &\quad - \frac{1}{2} V_{\partial\Omega} \bar{\mathcal{N}}(x_n; \boldsymbol{\theta}) \cdot t(x_n), \\ \mathcal{L}_o(\boldsymbol{\theta}; \mathcal{T}_o) &= \frac{1}{N_o} \sum_{l=1}^{N_o} \left( \frac{\mathcal{N}(\mathbf{x}_l; \boldsymbol{\theta}) - \hat{u}^l}{u^{char}} \right)^2. \end{aligned} \quad (35)$$

Here,  $x_n$  is the coordinate of the boundary where traction is applied.

Unless otherwise specified, the weight for the data loss terms  $\lambda_o$  and the characteristic displacements  $\mathbf{u}^{char}$  are set to  $\lambda_o = 10^5$  and according to Eq. (25). In addition, we use the same number and location of collocation points  $N_{col}$  to determine the PDE residual  $\lambda_{pde}$  and to determine the internal work  $W_{int}$ , such that  $N_{pde} = N_{W_{int}} = N_{col}$ .

#### 4. Parameter identification from one-dimensional displacement data

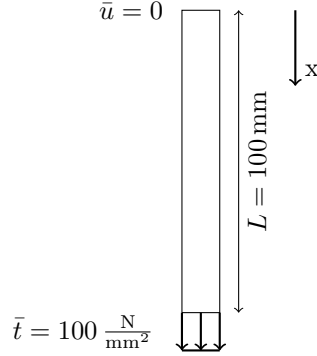
We first demonstrate the enhanced method for one-dimensional displacement data. Using analytical data, the method is validated for a realistic data regime and a sensitivity analysis is conducted with respect to the initial estimate of the material parameter to be identified. We then show that PINNs are, in principle, also able to identify the Young's modulus from one-dimensional experimental displacement data.

##### 4.1. Analytical displacement data

As a first test case, we consider analytical displacement data for a stretched rod. The stretched rod has a length of  $L = 100$  mm and its upper end is clamped. A traction of  $\bar{t} = 100 \frac{\text{N}}{\text{mm}^2}$  is applied at the free end, causing a deformation. External body forces, such as gravity, are neglected. We assume linear-elastic material with Young's modulus  $E_{true} = 210,000 \frac{\text{N}}{\text{mm}^2}$ . From the analytical solution, we calculate the displacements at  $N_o = 128$  equidistantly sampled points to train the PINN and at another  $N_{val} = 1,024$  points to validate



it. The collocation points for calculating the PDE residual as well as the internal work during training have the same number and location as the data points, such that  $N_{col} = N_o = 128$ . The geometry and the boundary conditions are schematically illustrated in Fig. 2.

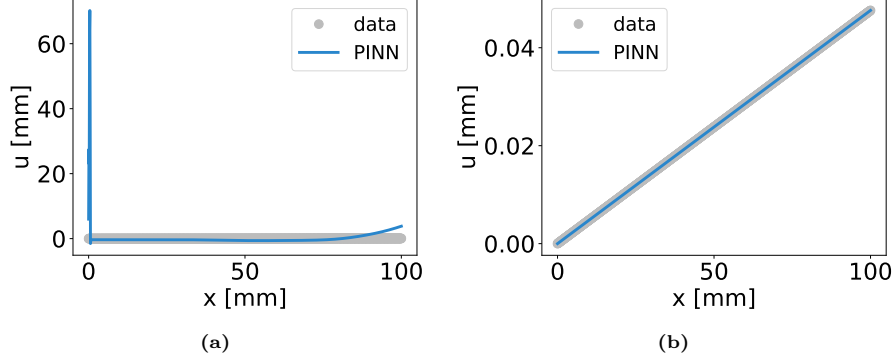


**Fig. 2:** Geometry and boundary conditions of a stretched rod under tension load. The stretched rod is clamped at the upper end and a traction is applied at the lower end. Body forces are neglected.

For the approximation of the displacement, we use a fully connected FFNN as defined in Section 2.2 and set the architecture and training hyperparameters as follows: The ANN has 2 hidden layers, each with 8 neurons. The hyperbolic tangent is used as activation function in the hidden layers. For optimization of the ANN parameters and the correction factor of Young’s modulus we use the BFGS [44, 45, 46, 47] optimizer. The weights are initialized using Glorot normal initialization [48]. The biases and the correction factor introduced in Section 3 are initialized with zeros. Optimization is performed in full batch mode. Since the chosen hyperparameters have proven to be suitable for this use case, no hyperparameter optimization is conducted.

In order to demonstrate the effectiveness of the enhanced method, we try to identify the Young’s modulus from the analytical displacement data with both the standard and the enhanced PINN. For the enhanced PINN, we compute the characteristic displacement  $u^{char} = 0.0238 \text{ mm}$  and set the data loss weight to  $\lambda_o = 10^5$ . The initial estimate is first set to the exact Young’s modulus  $E_{true} = 210,000 \frac{\text{N}}{\text{mm}^2}$  which we used for data generation. In Fig. 3, the displacement approximation for both the standard and the enhanced PINNs are shown. The comparison underlines that the standard PINN is not capable to accurately approximate the displacement of the stretched rod. As a consequence, the identified Young’s modulus is with  $E_{ident} = 320.47 \frac{\text{N}}{\text{mm}^2}$  far from the correct value. In contrast, the enhanced PINN succeeds in identifying a Young’s modulus of  $E_{ident} = 209,999.98 \frac{\text{N}}{\text{mm}^2}$  with a relative error (RE) of  $RE_E = -1.1054 \cdot 10^{-5} \%$ . In addition, the relative  $L^2$ -norm ( $rL^2$ -norm) of the displacement approximation is  $rL^2 = 6.1387 \cdot 10^{-8}$ .

The results presented above, and the method in general, are only useful as



**Fig. 3: Standard PINN vs. enhanced PINN:** Displacement of the stretched rod approximated by a) a standard PINN and b) the enhanced PINN. The comparison demonstrates that the standard PINN is not able to approximate the displacement of the stretched rod from the given data.

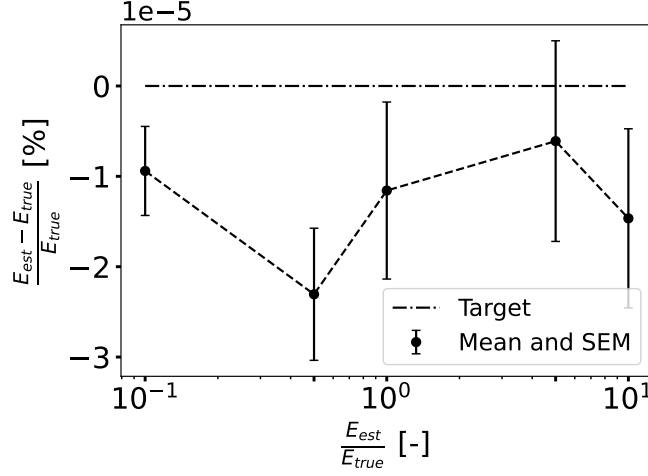
long as the results are satisfactory even if the exact material parameters are not known. Hence, we investigate the sensitivity of the relative error of the identified Young’s modulus with respect to the initial estimate. We perform a sensitivity analysis in which we solve the inverse problem for several initial estimates in the range  $E_{est} = [10\%, 1,000\%] \cdot E_{true}$  of the Young’s modulus used for data generation. The ANN parameters are initialized identically for each simulation of one analysis. Since we found that the optimization result depends on the initialization of the PINN parameters, we repeated the sensitivity analysis for a total of 10 different random initializations. From the 10 individual results for each initial estimate, we calculate the mean error and the standard error of the mean (SEM). The latter is defined as

$$SEM = \frac{\sigma}{\sqrt{n_s}}. \quad (36)$$

Here,  $\sigma$  is the standard deviation of the results and  $n_s$  is the number of samples, where in our case  $n_s = 10$ . The results in Fig. 4 suggest that the error of the identified material parameter is not sensitive to the initial estimate within a reasonable range.

#### 4.2. Experimental displacement data

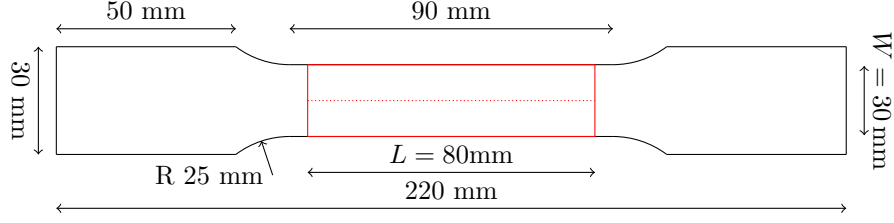
We test the enhanced PINN using one-dimensional experimental displacement data. For the material model calibration, we use the same PINN architecture and hyperparameters as in Section 4.1. To avoid potential overfitting of the PINN to the noisy displacement data, we reduce the data loss weight to  $\lambda_o = 10^3$ . We choose the initial estimate as  $E_{est} = 210,000 \frac{N}{mm^2}$ . The characteristic displacement is  $u^{char} = 0.0351$  mm. As a reference solution, we use the prediction of a FEMU simulation. For more information on this approach, we refer to [49, 50].



**Fig. 4: Sensitivity analysis:** Error sensitivity of the identified Young’s modulus  $E$  with respect to the initial estimate. We considered initial estimates in the range  $[10\%, 1,000\%]$  of the value used for data generation. The simulation for an estimated value was repeated a total of 10 times with different randomly initialized ANN parameters. The figure shows the mean values and standard errors of the means (SEM) of the relative errors calculated from the individual runs.

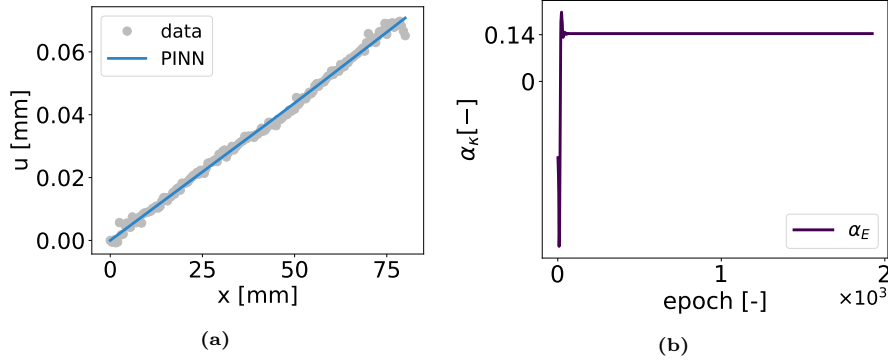
The experimental displacement data were measured in an uniaxial tensile test with the following setting: A specimen of TS275 steel was used, the geometry of which is shown in Fig. 5. The test was performed displacement-controlled using the tensile testing machine Z100 from ZwickRoell GmbH & Co. KG. While one end of the specimen was clamped, the testing machine pulled on the other end up to an averaged axial strain of  $\varepsilon^{mean} = 4.6860 \cdot 10^{-2}\%$ . Thus, the strain is still in the assumed linear-elastic range of the material under consideration. In addition, the displacements were measured by the DIC system Aramis 12M from Carl Zeiss GOM Metrology GmbH for an area of  $L = 80$  mm length and  $W = 19.25$  mm width in the center of the specimen. The displacements measured in two dimensions were then interpolated to a total of  $N_o = 161$  equidistant locations on the central axis of the specimen, as illustrated in Fig. 5. As in the analytical test case, the data points are also used as collocation points in the PDE loss term and for calculating the internal work when training the PINN, such that  $N_{col} = N_o = 161$ . Apart from the displacements, the force applied to initiate the displacements was measured by a load cell. The stress boundary condition can then be calculated from the force and the cross-sectional area of the specimen. The rigid body displacements are subtracted from the measured displacements.

The approximated displacement, which is shown in Fig. 6, is visually in good agreement with the measured displacements. Simultaneously, the PINN identified a Young’s modulus of  $E_{ident} = 240,097.30 \frac{N}{mm^2}$ , which differs by  $RE_E = 13.72\%$  from the value determined by the FEMU approach. We as-



**Fig. 5:** Geometry of the tensile specimen. The displacements were measured by a DIC system for the area which is outlined in red. For the material model calibration, the two-dimensional displacement data was interpolated onto a total of  $N_o = 161$  equidistant locations on the central axis of the specimen.

sume that the noisy data is the main reason for this deviation. In order to achieve the best possible comparability of the results of both methods, the FEMU approach is carried out with a stress boundary condition instead of a displacement boundary condition. Nevertheless, we assume that the methodological differences of the two approaches also influence the deviation of the determined material parameters.



**Fig. 6: One-dimensional experimental data:** a) Displacement of tension rod approximated by a PINN from experimental data and b) the evolution of the Young's modulus correction factor  $\alpha_E$  during training. The displacement approximation is visually in good agreement with the measured data.

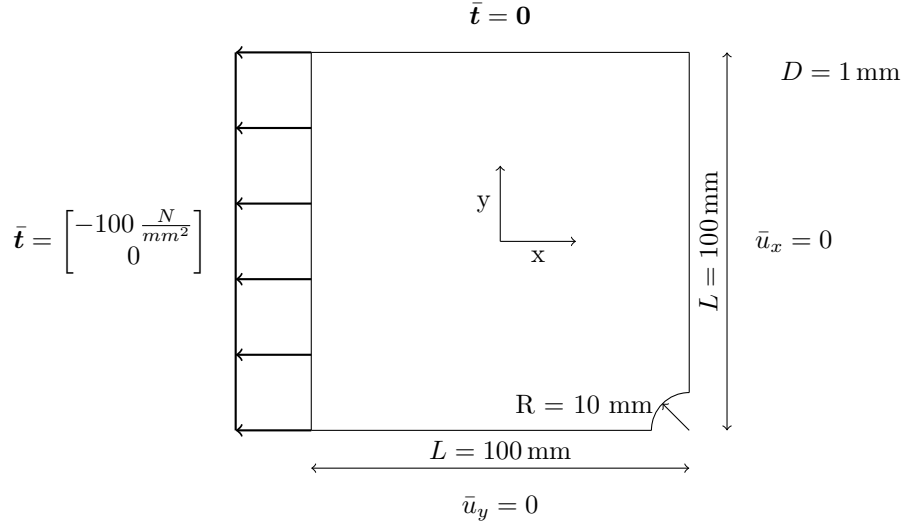
## 5. Parameter identification from full-field displacement data

In the next step, we apply the enhanced PINN formulation to synthetically generated displacement data of a plate with a hole, considering both clean and noisy data. As in the previous section, we first conduct an error sensitivity analysis for the identified Young's modulus and Poisson's ratio with respect to the initial estimate. Furthermore, we show how the relative errors of the identified material parameters evolve with an increasing number of collocation

points. Finally, we investigate the error sensitivity of the identified material parameters with respect to the noise level of the displacement data.

### 5.1. Clean synthetic displacement data

For the two-dimensional test case, we generate synthetic DIC data for a plate with a hole using the finite element method (FEM). Since the plate with a hole is two-fold symmetric, we consider only the quadrant at the top left of the plate and define symmetry boundary conditions on the bottom and right boundaries. The quadrant has an edge length of  $L = 100$  mm and the radius of the hole is  $R = 10$  mm. Furthermore, the thickness of the plate is set to  $T = 1$  mm and the plate is assumed to be in plain strain condition. We simulate uniaxial tension and load the left edge with  $\bar{\mathbf{t}} = [-100 \frac{\text{N}}{\text{mm}^2}, 0]^T$ . No force is applied to the upper boundary. External body forces are neglected. The geometry, dimensions and boundary conditions of the considered test case are illustrated in Fig. 7. We again assume isotropic, linear-elastic material. In two-dimensions, linear-elastic materials can be characterized by the Young's modulus and Poisson's ratio which are set to  $E_{true} = 210,000 \frac{\text{N}}{\text{mm}^2}$  and  $\nu_{true} = 0.3$ , respectively, emulating the behaviour of steel. For the FE simulation, the domain is meshed using linear triangular elements and the FE solution is evaluated and recorded at a total of 101,496 nodes. Discretization errors are neglected in the following due to the high resolution of the computational mesh. Under the given boundary conditions, the maximum calculated strain is  $\varepsilon_{max} = 0.20\%$  and thus in the assumed linear-elastic range of the considered material. The FEM code for data generation is implemented in Python and based on the FEniCS project [51].



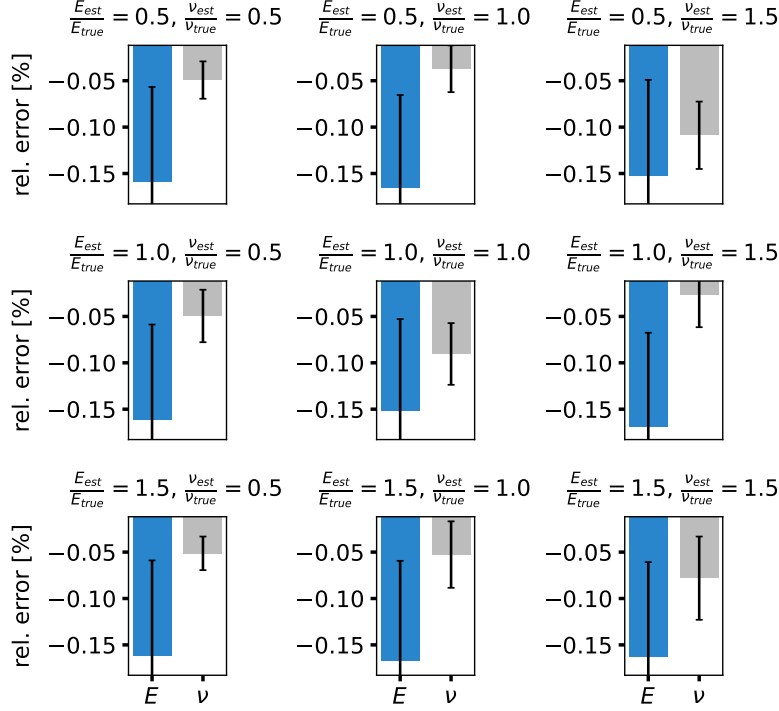
**Fig. 7:** Geometry and boundary conditions of the top left quadrant of a plate with a hole under uniaxial tension. The plate is assumed to be in plane strain condition. Body forces are neglected.

To solve the inverse problem, we only use a small subset of the synthetically generated displacement data in combination with stress data for the Neumann boundary conditions. The displacement data consists of  $N_o = 4,096$  data points within the domain of interest. We select the data points directly from the FE solution to avoid interpolation errors. In addition, the external work is approximated from  $N_{W_{ext}} = 32$  equidistantly distributed data points on the left and top Neumann boundary, respectively. While the traction causes stresses on the left boundary, the top edge is a free boundary. It should be noted that the displacement data points may be located on the right and bottom boundary, but not on the top and left boundary, as here the Neumann condition is defined. The architecture of the PINN and the training hyperparameters are defined as follows: We use two fully connected FFNNs to approximate the displacement fields  $u_x(\mathbf{x})$  and  $u_y(\mathbf{x})$ . Both ANN consist of 2 input neurons, 2 hidden layers with 16 neurons each, and 1 output neuron. The hyperbolic tangent is used as activation function in the hidden layers. The weights and biases of the ANN are initialized using Glorot normal initialization and zeros, respectively. As in the previous test cases, we initialize the correction factors for bulk and shear modulus with zeros. We solve the resulting optimization problem using the BFGS optimizer.

In order to investigate the error sensitivities of the identified Young's modulus and Poisson's ratio with respect to their initial estimates, we again perform a sensitivity analysis. For this purpose, we consider initial estimates for both Young's modulus and Poisson's ratio of 50%, 100% and 150% of the exact values used for data generation, respectively. Here we use the same number and locations of data and collocation points, so that  $N_o = N_{col} = 4,096$ . We then calibrate the linear-elastic material model from Eq. (5) for all of the resulting 9 combinations of initial estimates using a PINN. The ANN parameters are initialized identically for each of the 9 simulations of one analysis. To obtain more meaningful results, we again repeat the analysis a total of 10 times with different random initializations of the PINN parameters.

The results of the sensitivity analysis, summarized in Fig. 8, show that the maximum mean absolute relative error (MARE) of the identified material parameters averaged over all runs are  $MARE_E^{max} = 0.17\%$  and  $MARE_\nu^{max} = 0.11\%$  for Young's modulus and Poisson's ratio, respectively. If we consider all 90 simulations separately, in the worst case, the maximum absolute relative errors (AREs) are  $ARE_E^{max} = 0.84\%$  and  $ARE_\nu^{max} = 0.36\%$  for Young's modulus and Poisson's ratio, respectively. It is also noticeable that the mean relative error of Young's modulus is larger than the mean relative error of Poisson's ratio for all 9 combinations of initial estimates. This is also true for the standard errors of the means (SEM) of the relative errors. However, none of the nine combinations show significantly higher relative errors compared to the other combinations. Overall, the sensitivity analysis proves that the relative error is not sensitive to the required initial material parameter estimate within a reasonable range of estimates for real-world applications.

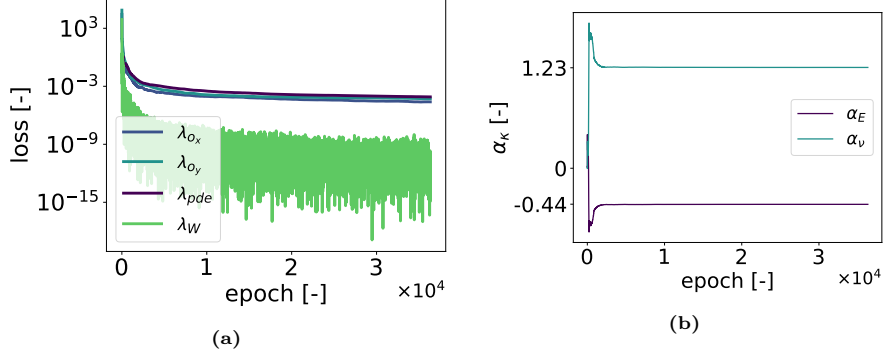
In Fig. 9, we show an example of how the loss terms and the correction factors for bulk and shear modulus for initial estimates of  $E_{est} = 105,000 \frac{\text{N}}{\text{mm}^2}$



**Fig. 8: Sensitivity analysis:** Error sensitivity of the identified Young’s modulus  $E$  and Poisson’s ratio  $\nu$  with respect to their initial estimates. We consider initial estimates for both Young’s modulus and Poisson’s ratio of 50%, 100% and 150% of the exact values used for data generation, respectively. The simulation for each of the 9 combinations of initial estimates is repeated a total of 10 times with different random initializations of ANN parameters. The figure shows the mean and standard errors of the means (SEM) of the relative errors calculated from the individual runs.

and  $\nu_{est} = 0.45$  evolve during training. The identified bulk and shear modulus for this example result in relative errors (REs) of  $RE_E = 0.14\%$  and  $RE_\nu = -0.03\%$  for Young’s modulus and Poisson’s ratio, respectively. To validate the accuracy of the displacement field, we use another  $N_{val} = 4,096$  data points randomly sampled from the FE solution and different from the training data. The displacement field is approximated with a relative  $L^2$ -norm (r $L^2$ -norm) of  $rL_x^2 = 2.9420 \cdot 10^{-5}$  in x- and  $rL_y^2 = 3.1787 \cdot 10^{-5}$  in y-direction.

Note that the approximation of the integrals for calculating the internal and external work in Eqs. (28) and (29) for heterogeneous strain and stress fields is erroneous for a finite number of collocation points. However, we assume that the accuracy of the approximation depends directly on the number of collocation points. To verify this assumption, we examine the convergence of the relative errors with respect to the number of collocation points within the range  $N_{col} = [512, 32,768]$ . The number of data points  $N_o = 4,096$  remains unchanged.



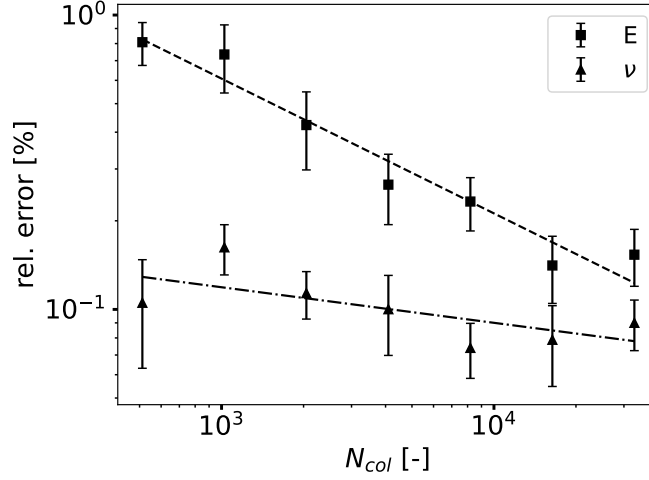
**Fig. 9: Clean two-dimensional data:** Exemplary evolution of a) the different loss terms and b) the correction factors  $\alpha_K$  and  $\alpha_G$ . The correction factors relate to the initial estimates  $E_{est} = 105,000 \frac{N}{mm^2}$  and  $\nu_{est} = 0.45$  for Young's modulus and Poisson's ratio, respectively.

We again repeat the convergence study a total of 10 times with different randomly initialized ANN parameters to obtain more meaningful and robust results. For this analysis, we use the same PINN architecture and hyperparameters as before and provide the material parameters we used for data generation as initial estimates. The results shown in Fig. 10 demonstrate that both the relative error of the identified Young's modulus and Poisson's ratio decrease with increasing number of collocation points.

### 5.2. Noisy synthetic displacement data

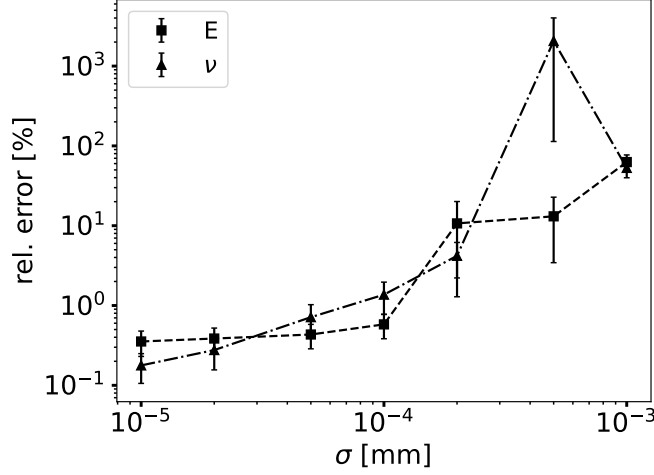
Finally, we investigate how sensitive the relative errors of the identified Young's modulus and Poisson's ratio are with respect to the level of noise. So far, we have not yet taken into account the fact that displacement data measured by a DIC system is inevitably affected by noise. In order to emulate real DIC data, we apply Gaussian noise  $\mathcal{N}(0, \sigma^2)$  with zero mean to the synthetic data used in Section 5.1. The noise in the DIC measurements and thus also the standard deviation depend on the pixel accuracy of the imaging device and is not proportional to the displacement magnitude. For this reason, we have determined the standard deviation relative to the order of magnitude of the mean displacement, which is  $10^{-2}$  mm. We applied the same absolute noise level to all nodal values of the FEM solution independently of the individual magnitude. For the sensitivity analysis, we apply varying levels of Gaussian noise  $\mathcal{N}(0, \sigma^2)$  to the clean data we used in the previous section in a range of [0.1%, 10%] relative to the order of magnitude of the mean displacement. This results in absolute standard deviations in the range of  $\sigma = [10^{-5}, 10^{-3}]$  mm. Just as with the sensitivity analysis in Section 5.1, we use  $N_o = N_{col} = 4,096$  training and collocation points. As initial estimates, we provide the same material parameters that we used to generate the clean displacement data. To avoid overfitting of the ANNs to the noisy data, we reduce the data loss weight in Eq. (31) to  $\lambda_o = 10^3$ . Apart from that, we use the same PINN architecture and hyperparameters as in Section 5.1.





**Fig. 10: Convergence study:** Relative errors of Young’s modulus  $E$  and Poisson’s ratio  $\nu$  with respect to the number of collocation points  $N_{col}$  used to determine the PDE residual and the internal work. The simulation for each tested number of collocation points is repeated a total of 10 times with different random initializations of ANN parameters. The figure shows the mean and standard errors of the means (SEM) of the relative errors calculated from the individual runs.

The results of the sensitivity analysis, shown in Fig. 11, indicate that PINNs can deal with moderate levels of noise when calibrating the linear-elastic material model. From the results, it can be seen that the relative errors of the identified material parameters slightly increase in their absolute values up to a standard deviation of  $\sigma = 10^{-4}$  mm, especially the relative error of the Young’s modulus. The mean absolute relative errors (MAREs) at this level of noise are  $MARE_E = 0.58\%$  and  $MARE_\nu = 1.37\%$  for Young’s modulus and Poisson’s ratio, respectively. For larger standard deviations, the relative errors increase significantly and are no longer in an acceptable range. According to [52], a representative value for the standard deviation of Gaussian random noise in DIC measurements is  $\sigma = 4 \cdot 10^{-4}$  mm. However, this requires optimal conditions of the experiment, which includes, among others, the optical setup as well as the imaging device. To account for the fact that these are not always met in practice, the authors test the PINN for a standard deviation of  $\sigma = 5 \cdot 10^{-4}$  mm. Under these assumptions, the mean absolute relative errors for a realistic standard deviation of the noise in DIC measurements are  $MARE_E = 13.10\%$  and  $MARE_\nu = 2,063.28\%$  for Young’s modulus and Poisson’s ratio, respectively. In future work, we aim to further enhance the method to increase the methods performance in the presence of noisy data as we discuss in the next section.



**Fig. 11: Sensitivity analysis:** Relative errors of the identified Young’s modulus  $E$  and Poisson’s ratio  $\nu$  with respect to the level of noise. We assume Gaussian noise with zero mean and test various standard deviations in the range  $\sigma = [10^{-5} \text{ mm}, 10^{-3} \text{ mm}]$ . The simulation for each tested standard deviation is repeated a total of 10 times with different random initializations of ANN parameters. The figure shows the mean and standard errors of the means (SEM) of the relative errors calculated from the individual runs.

## 6. Conclusion and outlook

PINNs have emerged as a suitable alternative approach to traditional numerical methods, such as FEM or VFM, to solve inverse problems in the field of solid mechanics. It has recently been shown that PINNs are capable of identifying the material parameters of constitutive models from full-field displacement data [20, 21, 22, 23]. However, the assumptions made in the literature often do not match real-world conditions and not all approaches realize the full potential of PINNs. This potential lies, among other things, in the fact that the inclusion of observation data is straightforward and no computational grid is required.

In this contribution, we further developed the standard PINN, as proposed in [5], towards full-field displacement data in a realistic regime. The realistic data regime refers to a realistic order of magnitude of the displacement data and material parameters as well as the fact that the data is inevitably affected by noise. In this process, we first normalized the input and output of the PINN and scaled the parameters to be optimized by providing initial estimates for the material parameters. A sensitivity analysis then demonstrated that the relative errors of the identified material parameters are not sensitive to the initial estimates in a reasonable range for real-world applications. Since we found that the cost function was ill-conditioned for realistic displacement data, we balanced the latter by weighting the data loss term. In order to reduce the dependence of the identified material parameters on local approximation errors in the boundary region, we based the identification not on the stress boundary

condition but instead on the global balance of internal and external work. In a convergence study, we were able to show that the error in approximating the integrals for calculating the internal and external energy converges with the number of collocation points. Furthermore, we found that the resulting optimization problem is better posed when it is formulated in terms of bulk and shear modulus. If Young’s modulus and Poisson’s ratio are to be identified, their estimates can be easily converted to bulk and shear modulus for optimization and vice versa for prediction. Finally, we investigated the sensitivity of the identified Young’s modulus and Poisson’s ratio with respect to various levels of noise.

Although the enhanced PINN shows satisfactory results for clean as well as moderately noisy data, future work is still needed for a real-world application of PINNs, e.g., as part of a monitoring system as motivated in the introduction. Future work includes, but is not limited to, increasing the robustness of the method to noise in the measurement data. It needs to be investigated whether this can be achieved by quantifying and accounting for the uncertainties in the identified material parameters. For this purpose, we will consider extended variants of PINNs [53, 54]. Another challenge in real-world applications is to take into account material inhomogeneities and to detect fracture at an early stage. Therefore, we intend to further develop the method towards heterogeneous materials and approximate the material field by another ANN, as proposed in [23]. The determined parameter field could then also be the basis for identifying fracture in the material.

Since the data loss term is currently weighted manually, we also plan to investigate methods for adaptively balancing the cost function. In addition, to accelerate the convergence of the identification process, future work should investigate and validate different approaches, such as adaptive activation functions [28]. In principle, the first four extensions presented in Section 3 are not restricted to linear problems. We would therefore apply the method presented here to more complex material models in combination with more sophisticated mechanical test cases. In order to be able to evaluate the quality of the identified material parameters in real-world applications, qualitative metrics, such as derived in [55] in the context of the FEMU approach, are required.

### **Declaration of competing interests**

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.

### **Acknowledgement**

The support of the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) is gratefully acknowledged in the following projects:

- DFG 255042459/GRK2075-2: *Modelling the constitutional evolution of building materials and structures with respect to aging.*
- DFG 501798687: *Monitoring data driven life cycle management with AR based on adaptive, AI-supported corrosion prediction for reinforced concrete structures under combined impacts.* Subproject of SPP 2388: *Hundred plus - Extending the Lifetime of Complex Engineering Structures through Intelligent Digitalization.*

In both projects, the solution of inverse problems is a key enabler to link measurement data and physical models. We further thank the Institute of Applied Mechanics at Technische Universität Clausthal for providing us the experimental data and the reference results of the FEMU approach. The first author wants also to thank Alexander Henkes, Jendrik-Alexander Tröger, Knut Andreas Meyer and Ralf Jänicke for the fruitful discussions.

### Data availability

The code will be published on zenodo.org upon acceptance of this manuscript.

### References

- [1] F.-K. Chang, Structural Health Monitoring: A Summary Report on the First Stanford Workshop on Structural Health Monitoring, Tech. rep., Stanford University (1998).
- [2] A. Entezami, Structural Health Monitoring by Time Series Analysis and Statistical Distance Measures, SpringerBriefs in Applied Sciences and Technology, Springer, 2021. doi:10.1007/978-3-030-66259-2.
- [3] M. A. Sutton, J.-J. Orteu, H. Schreier, Image Correlation for Shape, Motion and Deformation Measurements, 1st Edition, Springer, 2009. doi:10.1007/978-0-387-78747-3.
- [4] S. Avril, M. Bonnet, A.-S. Bretelle, M. Grédiac, F. Hild, P. Ienny, D. Lemosse, S. Pagano, E. Pagnacco, F. Pierron, Overview of identification methods of mechanical parameters based on full-field measurements, Experimental Mechanics 48 (4) (2008) 381–402. doi:10.1007/s11340-008-9148-y.
- [5] 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, Journal of Computational Physics 378 (2019) 686–707. doi:10.1016/j.jcp.2018.10.045.
- [6] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nature Reviews Physics 3 (6) (2021) 422–440. doi:10.1038/s42254-021-00314-5.

- [7] D. C. Psychogios, L. H. Ungar, A hybrid neural network-first principles approach to process modeling, *American Institute of Chemical Engineers Journal* 38 (10) (1992) 1499–1511. doi:10.1002/aic.690381003.
- [8] I. E. Lagaris, A. Likas, D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Transactions on Neural Networks* 9 (5) (1998) 987–1000. doi:10.1109/72.712178.
- [9] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: a survey, *Journal of Machine Learning Research* 18 (1) (2018) 5595–5637.
- [10] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattemberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous distributed systems, software available from <https://www.tensorflow.org> (2015).
- [11] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: Composable transformations of Python+NumPy programs, version 0.3.13. Software available from <http://github.com/google/jax> (2018).
- [12] I. Depina, S. Jain, S. M. Valsson, H. Gotovac, Application of physics-informed neural networks to inverse problems in unsaturated groundwater flow, *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards* 16 (1) (2021) 21–36. doi:10.1080/17499518.2021.1971251.
- [13] A. D. Jagtap, Z. Mao, N. Adams, G. E. Karniadakis, Physics-informed neural networks for inverse problems in supersonic flows, *Journal of Computational Physics* 466 (2022) 111402. doi:10.1016/j.jcp.2022.111402.
- [14] C. Xu, B. T. Cao, Y. Yuan, G. Meschke, Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios, *arXiv Preprint* (2022). doi:arXiv:2205.07731v2[cs.CE].
- [15] W. Li, K.-M. Lee, Physics informed neural network for parameter identification and boundary force estimation of compliant and biomechanical systems, *International Journal of Intelligent Robotics and Applications* 5 (3) (2021) 313–325. doi:10.1007/s41315-021-00196-x.
- [16] Y. Chen, L. Lu, G. E. Karniadakis, L. Dal Negro, Physics-informed neural networks for inverse problems in nano-optics and metamaterials, *Optics Express* 28 (8) (2020) 11618–11633. doi:10.1364/OE.384875.

- [17] E. Kharazmi, M. Cai, X. Zheng, Z. Zhang, G. Lin, G. E. Karniadakis, Identifiability and predictability of integer- and fractional-order epidemiological models using physics-informed neural networks, *Nature Computational Science* 1 (11) (2021) 744–753. doi:10.1038/s43588-021-00158-0.
- [18] K. Shukla, A. D. Jagtap, J. L. Blackshire, D. Sparkman, G. E. Karniadakis, A physics-informed neural network for quantifying the microstructural properties of polycrystalline nickel using ultrasound data: A promising approach for solving inverse problems, *IEEE Signal Processing Magazine* 39 (1) (2022) 68–77. doi:10.1109/MSP.2021.3118904.
- [19] C. J. G. Rojas, M. L. Bitterncourt, J. L. Boldrini, Parameter identification for a damage model using a physics-informed neural network, *arXiv Preprint* (2021). doi:arXiv:2107.08781v1[cond-mat.mtrl-sci].
- [20] E. Zhang, M. Dao, G. E. Karniadakis, S. Suresh, Analyses of internal structures and defects in materials using physics-informed neural networks, *Science Advances* 8 (7) (2022) eabk0644. doi:10.1126/sciadv.abk0644.
- [21] E. Haghighat, M. Raissi, A. Moure, H. Gomez, R. Juanes, A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, *Computer Methods in Applied Mechanics and Engineering* 379 (2021) 113741. doi:10.1016/j.cma.2021.113741.
- [22] C. M. Hamel, K. N. Long, S. L. B. Kramer, Calibrating constitutive models with full-field data via physics-informed neural networks, *Strain* (2022) e12431doi:10.1111/str.12431.
- [23] E. Zhang, M. Yin, G. E. Karniadakis, Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging, *arXiv Preprint* (2020). doi:arXiv:2009.04525v1[cs.LG].
- [24] D. Anton, H. Wessels, Identification of material parameters from full-field displacement data using physics-informed neural networks, in: M. Beer, E. Zio, K.-K. Phoon, B. M. Ayyub (Eds.), *Proceedings of the 8th International Symposium on Reliability Engineering and Risk Management*, Research Publishing, 2022, pp. 813–820. doi:10.3850/978-981-18-5184-1\_GS-07-026-cd.
- [25] S. Wang, Y. Teng, P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, *SIAM Journal on Scientific Computing* 43 (5) (2021) A3055–A3081. doi:10.1137/20M1318043.
- [26] S. Wang, X. Yu, P. Perdikaris, When and why PINNs fail to train: A neural tangent kernel perspective, *Journal of Computational Physics* 449 (2022) 110768. doi:10.1016/j.jcp.2021.110768.
- [27] A. Henkes, H. Wessels, R. Mahnken, Physics informed neural networks for continuum micromechanics, *Computer Methods in Applied Mechanics and Engineering* 393 (2022) 114790. doi:10.1016/j.cma.2022.114790.

- [28] A. D. Jagtap, K. Kawaguchi, G. E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, *Journal of Computational Physics* 404 (2020) 109136. doi:<https://doi.org/10.1016/j.jcp.2019.109136>.
- [29] A. D. Jagtap, Y. Shin, K. Kawaguchi, G. E. Karniadakis, Deep Kronecker neural networks: A general framework for neural networks with adaptive activation functions, *Neurocomputing* 468 (2022) 165–180. doi:[10.1016/j.neucom.2021.10.036](https://doi.org/10.1016/j.neucom.2021.10.036).
- [30] M. A. Nabian, R. J. Gladstone, H. Meidani, Efficient training of physics-informed neural networks via importance sampling, *Computer-Aided Civil and Infrastructure Engineering* 36 (8) (2021) 962–977. doi:[10.1111/mice.12685](https://doi.org/10.1111/mice.12685).
- [31] L. McClenny, U. Braga-Neto, Self-adaptive physics-informed neural networks using a soft attention mechanism, in: J. Lee, E. F. Darve, P. K. Kitanidis, M. W. Mahoney, A. Karpatsne, M. W. Farthing, T. Hesser (Eds.), *Proceedings of the AAAI 2021 Spring Symposium on Combining Artificial Intelligence and Machine Learning with Physical Sciences*, Stanford, CA, USA, March 22nd - to - 24th, 2021, Vol. 2964 of CEUR Workshop Proceedings, CEUR-WS.org, 2021, p. 5.
- [32] J. Berg, K. Nyström, A unified deep artificial neural network approach to partial differential equations in complex geometries, *Neurocomputing* 317 (2018) 28–41. doi:[10.1016/j.neucom.2018.06.056](https://doi.org/10.1016/j.neucom.2018.06.056).
- [33] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals, and Systems* 2 (4) (1989) 303–314. doi:[10.1007/BF02551274](https://doi.org/10.1007/BF02551274).
- [34] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Networks* 2 (5) (1989) 359–366. doi:[10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8).
- [35] X. Li, Simultaneous approximations of multivariate functions and their derivatives by neural networks with one hidden layer, *Neurocomputing* 12 (4) (1996) 327–343. doi:[10.1016/0925-2312\(95\)00070-4](https://doi.org/10.1016/0925-2312(95)00070-4).
- [36] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016, <http://www.deeplearningbook.org>.
- [37] B. Moseley, A. Markham, T. Nissen-Meyer, Finite basis physics-informed neural networks (FBPINNs): A scalable domain decomposition approach for solving differential equations, *arXiv Preprint* (2021). doi:[arXiv:2107.07871\[physics.comp-ph\]](https://doi.org/10.48550/arXiv.2107.07871).
- [38] E. Haber, U. M. Ascher, Preconditioned all-at-once methods for large, sparse parameter estimation problems, *Inverse Problems* 17 (6) (2001) 1847. doi:[10.1088/0266-5611/17/6/319](https://doi.org/10.1088/0266-5611/17/6/319).

- [39] A. Schlögl, B. Kaltenbacher, All-at-once formulation meets the Bayesian approach: A study of two prototypical linear inverse problems, in: B. Jadamba, A. A. Khan, S. Migórski, M. Sama (Eds.), *Deterministic and Stochastic Optimal Control and Inverse Problems*, 1st Edition, CRC Press, 2021, pp. 1–44. doi:10.1201/9781003050575.
- [40] P. A. Guth, C. Schillings, S. Weissmann, Ensemble kalman filter for neural network based one-shot inversion, in: R. Herzog, M. Heinkenschloss, D. Kalise, G. Stadler, E. Trélat (Eds.), *Optimization and Control for Partial Differential Equations: Uncertainty quantification, open and closed-loop control, and shape optimization*, De Gruyter, 2022, pp. 393–418. doi:10.1515/9783110695984-014.
- [41] Y. A. LeCun, L. Bottou, G. B. Orr, K.-R. Müller, Efficient BackProp, in: G. Montavon, G. B. Orr, K.-R. a. Müller (Eds.), *Neural Networks: Tricks of the Trade*, 2nd Edition, *Lecture Notes in Computer Science*, Springer, 2012, pp. 9–48. doi:10.1007/978-3-642-35289-8\_3.
- [42] J. Nocedal, S. J. Wright, *Numerical Optimization*, 2nd Edition, *Springer Series in Operations Research and Financial Engineering*, Springer, 2006. doi:10.1007/978-0-387-40065-5.
- [43] E. Samaniego, C. Anitescu, S. Goswami, V. M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, T. Rabczuk, An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, *Computer Methods in Applied Mechanics and Engineering* 362 (2020) 112790. doi:10.1016/j.cma.2019.112790.
- [44] C. G. Broyden, The convergence of a class of double-rank minimization algorithms 1. General considerations, *IMA Journal of Applied Mathematics* 6 (1) (1970) 76–90. doi:10.1093/imamat/6.1.76.
- [45] R. Fletcher, A new approach to variable metric algorithms, *The Computer Journal* 13 (3) (1970) 317–322. doi:10.1093/comjnl/13.3.317.
- [46] D. F. Shanno, Conditioning of Quasi-Newton methods for function minimization, *Mathematics of Computation* 24 (111) (1970) 647–656. doi:10.2307/2004840.
- [47] D. Goldfarb, A family of variable-metric methods derived by variational means, *Mathematics of Computation* 24 (109) (1970) 23–26. doi:10.2307/2004873.
- [48] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y. W. Teh, M. Titterton (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9 of *Proceedings of Machine Learning Research*, PMLR, 2010, pp. 249–256.



- [49] S. Hartmann, R. R. Gilbert, A. K. Marghzar, C. Leistner, P. K. Dileep, Material parameter identification of unidirectional fiber-reinforced composites, *Archive of Applied Mechanics* 91 (2) (2021) 687–712. doi:10.1007/s00419-021-01895-4.
- [50] S. Hartmann, R. R. Gilbert, Material parameter identification using finite elements with time-adaptive higher-order time integration and experimental full-field strain information, *Computational Mechanics* 68 (3) (2021) 633–650. doi:10.1007/s00466-021-01998-3.
- [51] M. S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, G. N. Wells, The FEniCS Project Version 1.5, *Archive of Numerical Software* 3 (100) (2015) 9–23. doi:10.11588/ans.2015.100.20553.
- [52] F. Pierron, S. Avril, V. T. Tran, Extension of the virtual fields method to elasto-plastic material identification with cyclic loads and kinematic hardening, *International Journal of Solids and Structures* 47 (22) (2010) 2993–3010. doi:10.1016/j.ijsolstr.2010.06.022.
- [53] D. Zhang, L. Lu, L. Guo, G. E. Karniadakis, Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems, *Journal of Computational Physics* 397 (2019) 108850. doi:10.1016/j.jcp.2019.07.048.
- [54] L. Yang, X. Meng, G. E. Karniadakis, B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, *Journal of Computational Physics* 425 (2021) 109913. doi:10.1016/j.jcp.2020.109913.
- [55] S. Hartmann, R. R. Gilbert, Identifiability of material parameters in solid mechanics, *Archive of Applied Mechanics* 88 (1) (2018) 3–26. doi:10.1007/s00419-017-1259-4.