ENGINEERING

Analyses of internal structures and defects in materials using physics-informed neural networks

Enrui Zhang[1], Ming Dao[2]*, George Em Karniadakis[1,3]*, Subra Suresh[4]*

Characterizing internal structures and defects in materials is a challenging task, often requiring solutions to inverse problems with unknown topology, geometry, material properties, and nonlinear deformation. Here, we present a general framework based on physics-informed neural networks for identifying unknown geometric and material parameters. By using a mesh-free method, we parameterize the geometry of the material using a differentiable and trainable method that can identify multiple structural features. We validate this approach for materials with internal voids/inclusions using constitutive models that encompass the spectrum of linear elasticity, hyperelasticity, and plasticity. We predict the size, shape, and location of the internal void/inclusion as well as the elastic modulus of the inclusion. Our general framework can be applied to other inverse problems in different applications that involve unknown material properties and highly deformable geometries, targeting material characterization, quality assurance, and structural design.

INTRODUCTION

Deep learning (*1*) approaches play an increasingly substantial role in a wide range of technologies that benefit computer vision (*2*), natural language processing (*3*), and other data-rich areas of societal interest. Despite the evolving sophistication of data analytics and neural networks (NNs), much of this work to date has not been predicated on a large volume of scientific data, through which predictive models can be constructed using experimentally validated mechanistic inferences and laws of physics. In most scientific applications, by contrast, physical conservation laws (such as those for momentum and energy) are framed by highly general, mathematical formulations [e.g., those invoking partial differential equations (PDEs) in areas such as solid mechanics, fluid mechanics, and material diffusion], along with experimental authentication by recourse to laboratory tests.

Emerging research reveals the profound untapped potential of physics-based, multidisciplinary, deep learning approaches with unprecedented opportunities for scientific and engineering advances in molecular analysis (*4*), design of materials with improved properties and performance (*5*, *6*) in structural and functional applications, and unique pathways for the characterization of properties of materials (*7*–*11*). To further realize this potential, broadly applicable methodologies in the area of NNs are needed to address a variety of issues that underpin deep learning analyses, governed by physical laws and guided by mathematical formulations. To this end, a physics-informed deep learning approach has recently been proposed (*12*) for the simulation of systems governed by physical laws that are represented by PDEs. While traditional methods based on deep learning implicitly encode these formulations by feeding training data governed by equations, this approach explicitly encodes known physical or scaling laws in the form of mathematical equations into the standard structure of NNs, formulating the so-called physics-informed NNs (PINNs) (*12*). Such an approach integrates any existing knowledge expressible in terms of PDEs during the learning process, thereby markedly improving predictability while reducing the amount of data required to achieve a desired level of accuracy. Studies have shown the applicability of PINNs in addressing a wide spectrum of forward and inverse problems spanning disciplines such as fluid mechanics (*13*–*15*), quantum mechanics (*12*), and solid mechanics (*16*–*22*). These applications have shown promise for enhancing predictability when the amount of data is limited or when the problem is ill posed, situations in which existing methods are not likely to yield accurate and reliable results. This approach has been further extended to offer unique pathways to address relevant mathematical formulations, such as stochastic PDEs (*23*) and fractional PDEs (*24*).

Here, we address geometry identification problems in the field of continuum solid mechanics. Geometry identification problems are a class of inverse problems of scientific, technological, and societal interest in fields as diverse as the following (*25*–*27*): safety and failure analysis of civil, mechanical, nuclear, and aeronautical structures; land, sea, and air transportations; reliability analysis in microelectronic devices; nondestructive testing of materials; and processing of engineered materials. In a geometry identification problem, the unknown geometric features and parameters are determined in a solid material/structure given measured material response under static or dynamic loading, thereby characterizing unknown structures including internal defects or boundaries such as voids, vacancies or holes (*28*–*32*), inclusions and reinforcements (*31*, *33*–*36*), and/or cracks (*30*–*32*, *37*). Traditionally, computational algorithms for geometry identification are established on the basis of the finite element method (FEM) (*38*) as the forward solver. Beyond the forward solver, considerable effort is required for the design and implementation of iterative algorithms for updating the estimated values of geometric parameters (*39*) (see section S1 for a brief review of the algorithms), through which the discrepancy (loss) between the observed data and the results of the forward solver is minimized. However, the embedded forward FEM solver as a mesh-based method inherently brings about complications in these algorithms. The estimated geometry is updated by repeatedly remeshing the domain through iterations (*33*). Alternatively, the

[1]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. [2]Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. [3]School of Engineering, Brown University, Providence, RI 02912, USA. [4]Nanyang Technological University, 639798 Singapore, Singapore.
*Corresponding author. Email: mingdao@mit.edu (M.D.); george_karniadakis@brown.edu (G.E.K.); ssuresh@ntu.edu.sg (S.S.)

unknown domain is embedded in a larger fixed domain while introducing an auxiliary field to track the presence of material (*28*, *36*, *40*). The problem becomes even more challenging when large deformations (i.e., geometric nonlinearity) and nonlinear mechanical properties (i.e., highly nonlinear constitutive behavior of the solid material) are involved. These issues are still not well resolved, and available methods are cumbersome and resource intensive for deriving automated solutions to such inverse problems involving unknown geometry.

Here, we present a unique, systematic approach based on PINNs for solving geometry identification problems in continuum solid mechanics. This method integrates known PDEs of importance in solid mechanics with NNs, composing a unified computational framework involving both the forward solver and the inverse algorithm. Notably, we propose a method for directly parameterizing the geometry of the solid in a differentiable and trainable manner. By using the workflow of NNs, our method can automatically update the geometry estimation through the deep learning process. To demonstrate the efficacy of our method, we study a two-dimensional prototypical problem on a matrix-void/inclusion system as a proof of concept (see Fig. 1). A square-shaped matrix material contains a void/inclusion with unknown geometry. To characterize the location, size, and shape of the void/inclusion, we apply loading P_0 on the matrix boundary and monitor the displacement response on the measurement points at the matrix boundary under such loading. We expect the PINN to inversely characterize the geometry of the void/inclusion according to the displacement data. To test the performance of our method with various parametric assessments, we build a set of detailed cases for this problem, including different shapes and topologies of the void and different constitutive models for describing the mechanical properties of the material. For the particular case of inclusion, the PINN is also required to estimate the unknown material parameter of the inclusion, through which we demonstrate the capability of our model in solving combined material and geometry identification problems. In addition to the major results shown in the main text, we report in the Supplementary Materials more systematic studies of additional cases and parametric analyses, highlighting the advantages and limitations of the method.

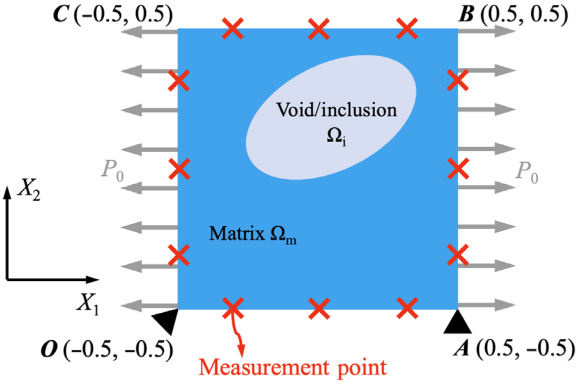


Fig. 1. General setup of the prototypical problem on geometry and material identification in this study. We consider a plane-strain problem in the X_1-X_2 plane about a square-shaped matrix specimen Ω_m with a void/inclusion Ω_i. Displacements are measured on the outer boundary of the matrix when loading P_0 is applied. The goal is to characterize the unknown geometry of the internal void/inclusion according to the measurement data. For the case of inclusion, material properties of the inclusion are also characterized.

RESULTS

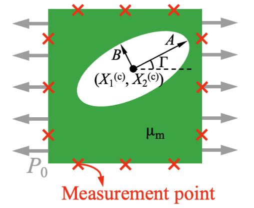
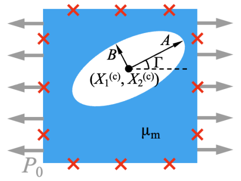
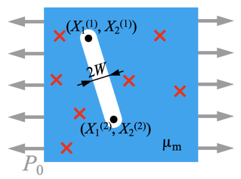
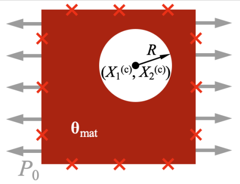
Setup of the prototypical inverse problem

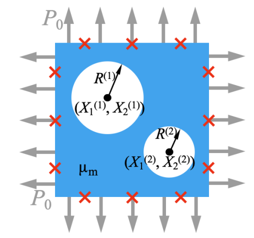
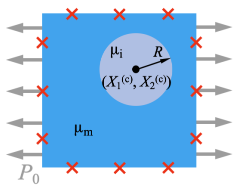
The general setup of the prototypical inverse problem has been presented in the introduction and in Fig. 1. We consider a plane-strain problem in the $X_1 - X_2$ plane about a square-shaped matrix specimen with a void/inclusion. The goal of the inverse problem is to estimate the geometric parameters $\boldsymbol{\theta}_{\text{geo}}$ (and material parameters $\boldsymbol{\theta}_{\text{mat}}$ in the constitutive model) of the void (inclusion) Ω_i inside the matrix Ω_m, by applying uniaxial/biaxial loading P_0 and collecting displacement data on the matrix boundary. We designed six specific plane-strain problems as shown in Fig. 2. For each case, we specify the type of the inhomogeneity (void/inclusion), the unknown parameters [$\boldsymbol{\theta}_{\text{geo}}$ for void, or $\boldsymbol{\theta}_{\text{geo}}$ and $\boldsymbol{\theta}_{\text{mat}}$ for inclusion; denoted together as $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\text{mat}}, \boldsymbol{\theta}_{\text{geo}})$], the material model (compressible linear elasticity, incompressible Neo-Hookean hyperelasticity, or compressible deformation plasticity), type of the loading (uniaxial/biaxial), and the location of displacement measurements (uniformly on the outer boundary/inside the solid). All unknown parameters describe the geometry of the void/inclusion except μ_i in case 5, which is a material parameter representing the shear modulus of the inclusion. The sketch and all the geometric parameters are shown in the reference (undeformed) configuration. The material properties of the matrix are known for all the cases.

The solution of the six cases will provide a proof of concept for our method under different practical scenarios, demonstrating the wide applicability of the method. The three material models (cases 0, 1, and 3 as the baseline cases) cover a wide range of mechanical behavior patterns of natural and engineered materials in a vast array of practical applications. We place the displacement measurement points only on the outer boundary of the matrix, to mimic the real-world situation where the internal details are not available. Case 2 explores the scenario of engineering application where the void has a large aspect ratio (such as a crack), which we approximate by a slender slit. For this case only, we allow the displacement measurements to be inside the solid because of the relative insensitivity of the boundary displacement with respect to the slit geometry. Case 4 demonstrates the applicability of the method for materials with multiple voids (such as porous materials or those with multiple cracks/slits). Last, for case 5, we estimate the material and geometric parameters for a soft circular inclusion, to show that our method can handle combined material and geometry identification problems.

Summary of PINN architecture for continuum solid mechanics

We set up the general formulation of PINNs in continuum solid mechanics involving both material and geometry identification. Corresponding to our computational examples, we design the architectures of the PINNs for plane-strain problems for the three material models, as shown in (i) Fig. 3A for (compressible) linear elasticity, (ii) Fig. 3B for (incompressible) hyperelasticity, and (iii) Fig. 3C for (compressible) deformation plasticity. The architectures of the PINNs are slightly different for different material models because of the characteristics of their mathematical expressions. Figure 3D includes the definitions of the mechanical quantities of interest in the architectures.

Case	(0) One elliptical void	(1) One elliptical void	(2) One slit-shaped void	(3) One circular void
Sketch (X_1-X_2 plane)	 B A Γ $(X_1^{(c)}, X_2^{(c)})$ μ_m P_0 Measurement point	 B A Γ $(X_1^{(c)}, X_2^{(c)})$ μ_m P_0	 $(X_1^{(1)}, X_2^{(1)})$ $2W$ $(X_1^{(2)}, X_2^{(2)})$ μ_m P_0	
Unknown parameters θ	$X_1^{(c)}, X_2^{(c)}, A, B, \Gamma$	$X_1^{(c)}, X_2^{(c)}, A, B, \Gamma$	$X_1^{(1)}, X_2^{(1)}, X_1^{(2)}, X_2^{(2)}$	$X_1^{(c)}, X_2^{(c)}, R$

Case	(4) Two circular voids	(5) One circular inclusion
Sketch (X_1-X_2 plane)	 P_0 $R^{(1)}$ $(X_1^{(1)}, X_2^{(1)})$ $R^{(2)}$ $(X_1^{(2)}, X_2^{(2)})$ μ_m P_0	 μ_i R $(X_1^{(c)}, X_2^{(c)})$ μ_m P_0
Unknown parameters θ	$X_1^{(1)}, X_2^{(1)}, R^{(1)},$ $X_1^{(2)}, X_2^{(2)}, R^{(2)}$	$X_1^{(c)}, X_2^{(c)}, R, \mu_i$

	Void
Linear elasticity (compressible)	🟩 Matrix
Hyperelasticity (incompressible; Neo-Hookean)	🟦 Matrix Inclusion
Deformation plasticity (compressible; power law)	🟥 Matrix

Fig. 2. Setup of cases 0 to 5 of the prototypical problem. All the illustrations and geometric parameters are given in the reference (undeformed) configuration. The target of each case is to estimate the unknown parameters θ given the displacement data on the measurement points. All unknown parameters describe the geometry of the void/inclusion except μ_i in case 5, which is the shear modulus of the inclusion. For each case, we specify the type of the inhomogeneity (void/inclusion), unknown parameters, the material model (linear elasticity/hyperelasticity/deformation plasticity), type of the loading (uniaxial/biaxial), and the location of displacement measurements (uniformly on the outer boundary/inside the solid). Additional cases are summarized in the main text and presented in detail in the Supplementary Materials.

The detailed formulations and relevant governing equations of PINNs are explained in Materials and Methods and in section S2. Here, we summarize the basic workflow of PINNs as follows. First, we apply an NN (with trainable parameters λ) to approximate the primary solution fields (top left panels in Fig. 2, A to C) with respect to the in-plane coordinates $\mathbf{X} = (X_1, X_2)$. Second, we integrate the mechanical laws into the PINN architecture (top right panels in Fig. 2, A to C) by deriving relevant mechanical quantities of interest from the NN outputs, such as strain, stress, and the residual of equilibrium PDEs. In this process, unknown material parameters θ_{mat} are involved. Third, we formulate the loss function $\mathcal{L}(\lambda, \theta)$, which measures the discrepancy between the predicted mechanical quantities of interest and their respective true values provided by mechanical laws and measured data (bottom right panels in Fig. 2, A to C). For example, for linear elasticity in Fig. 2A, the loss function is expressed as

$$\mathcal{L}(\lambda, \theta) = \alpha_{PDE}\mathcal{L}_{PDE}(\lambda, \theta) + \alpha_{BC}\mathcal{L}_{BC}(\lambda, \theta) + \alpha_{data}\mathcal{L}_{data}(\lambda, \theta) \quad (1)$$

where the three loss terms $\mathcal{L}_j(\lambda, \theta)$ (j = PDE, BC, and data) weighted by α_j correspond to PDEs, boundary conditions (BCs), and data, respectively. Each loss term is the mean squared error evaluated on N_j residual points

$$\mathcal{L}_j(\lambda, \theta) = \frac{1}{N_j}\sum_{i=1}^{N_j} |\mathbf{r}_j(\mathbf{X}_j^{(i)}(\theta_{geo});\ \lambda, \theta_{mat})|^2 \quad (2)$$

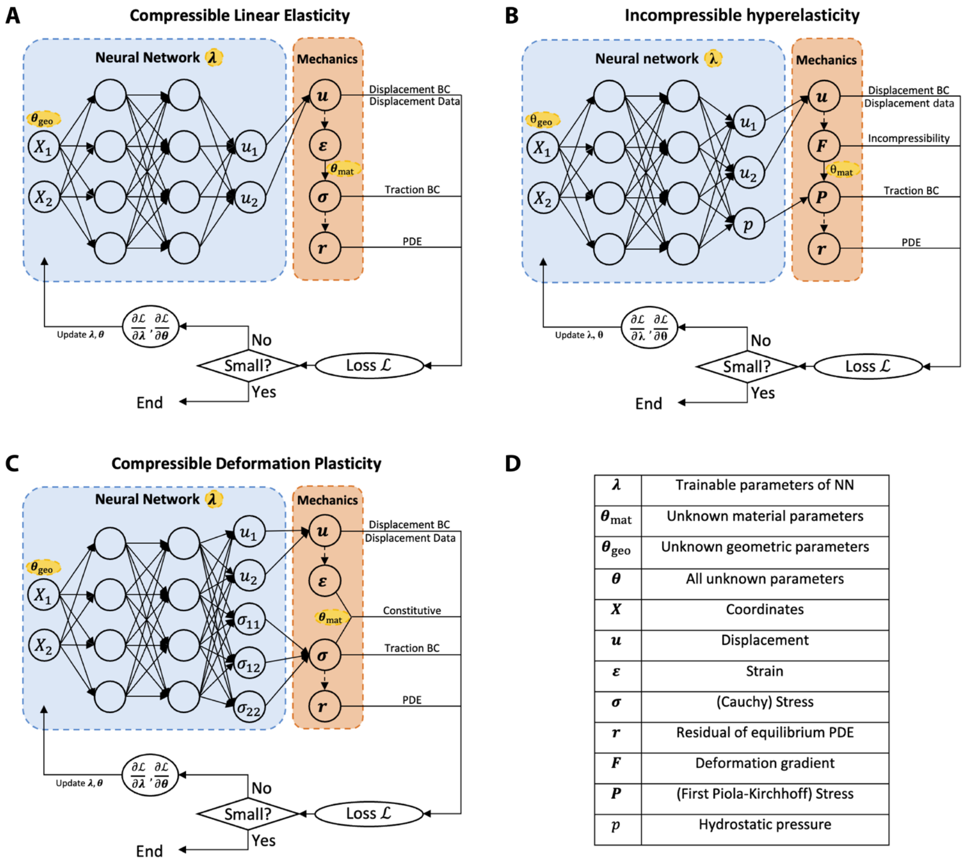
where $\mathbf{r}_j$ is the residual of the condition j at the ith residual point $\mathbf{X}_j^{(i)}(\theta_{geo})$. The N_j residual points are distributed in the domain of condition j to correctly evaluate $\mathcal{L}_j(\lambda, \theta)$. As an example, we show the residual points for each condition in case 0 before and during the simulation in Fig. 4. The coordinates of the residual points $\mathbf{X}_j^{(i)}$ depend on geometric parameters θ_{geo} because of the variable computational domain, which will be explained in detail in the next section. Last, we conduct parameter estimation through the training of the PINN (bottom left panels in Fig. 2, A to C), during which the estimated unknown parameters $\theta = (\theta_{mat}, \theta_{geo})$ and NN parameters λ are updated/trained to minimize the loss function. This process can be expressed as

$$\hat{\lambda}, \hat{\theta} = \underset{\lambda, \theta}{\operatorname{argmin}}\, \mathcal{L}(\lambda, \theta) \quad (3)$$

where the hat symbol refers to the value of these trainable parameters after the training process is completed. As the solution to the inverse problem, the estimation of the unknown parameters is $\hat{\theta}$.

Formulation for geometry identification

Geometric parameters θ_{geo} play an essentially different role in the inverse problem compared to material parameters θ_{mat}. Material parameters parameterize the governing PDEs of mechanics, which are naturally endowed with trainability through automatic differentiation of (physics-informed) NNs. As a result, material parameters



λ	Trainable parameters of NN
θ_{mat}	Unknown material parameters
θ_{geo}	Unknown geometric parameters
θ	All unknown parameters
X	Coordinates
u	Displacement
ε	Strain
σ	(Cauchy) Stress
r	Residual of equilibrium PDE
F	Deformation gradient
P	(First Piola-Kirchhoff) Stress
p	Hydrostatic pressure

Fig. 3. Architectures of PINNs for continuum solid mechanics. We established the PINNs for plane-strain problems involving geometry and material identification. Three material models are considered, including (**A**) compressible linear elasticity, (**B**) incompressible hyperelasticity, and (**C**) deformation plasticity. (A to C) We apply NNs with trainable parameters λ to approximate primary solution fields with respect to the in-plane coordinates (X_1, X_2). Mechanical laws are integrated to derive relevant mechanical quantities of interest from the NN outputs, such as strain, stress, and the residual of equilibrium PDEs, during which unknown material parameters $\boldsymbol{\theta}_{\text{mat}}$ are involved. The loss function $\mathcal{L}$ is formulated to represent the prediction error of each condition in the problem, such as PDEs, BCs, and data in (A), during which unknown geometric parameters $\boldsymbol{\theta}_{\text{geo}}$ are involved because of the variable computational domain. Last, parameter estimation is conducted through the minimization of loss function. In this process, λ and $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\text{mat},} \boldsymbol{\theta}_{\text{geo}})$ are iteratively updated. The final solution of the identification problem is the updated value of $\boldsymbol{\theta}$ after iterations. (**D**) Definitions of the notations in (A) to (C).

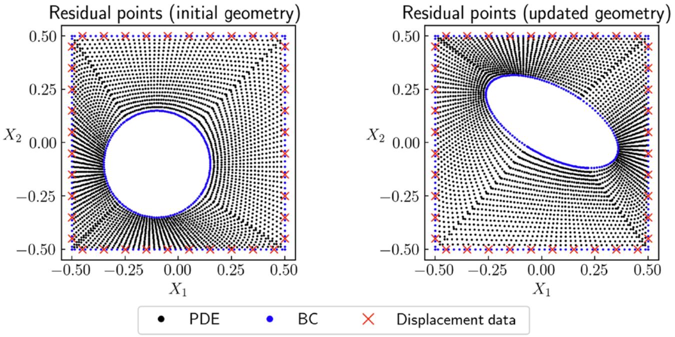


Fig. 4. Residual points for the initial geometry and updated geometry in case 0. Different residual terms (PDEs, BCs, and data) require different residual points. We propose the geometry-parameterized residual points, so that the locations of the residual points automatically change as the geometric parameters $\boldsymbol{\theta}_{\text{geo}}$ are updated.

can be directly estimated using the standard formulation of PINNs for inverse problems (*12*, *18*, *41*, *42*). Geometric parameters $\boldsymbol{\theta}_{\text{geo}}$, on the other hand, parameterize the computational domains of the PDEs and BCs, which do not naturally serve as trainable parameters in the framework of PINNs. To make the geometric parameters $\boldsymbol{\theta}_{\text{geo}}$ differentiable and hence trainable in a similar way to material parameters $\boldsymbol{\theta}_{\text{mat}}$, we propose to parameterize the coordinates of residual points by geometric parameters $\boldsymbol{\theta}_{\text{geo}}$. Technically, such parameterization can be implemented by using the definition of trainable variables in deep learning libraries such as TensorFlow (*43*): We first define the geometric parameters $\boldsymbol{\theta}_{\text{geo}}$ as trainable variables; then, we express the locations of residual points as functions of these trainable variables. As a result, the coordinates of residual points are automatically updated as the estimation of $\boldsymbol{\theta}_{\text{geo}}$ is updated throughout the iterative training process (see Fig. 4). In this way, we ensure that the residual points for different conditions are always located in their correct domains. Furthermore, this implementation allows us to capture the gradient of the loss function $\mathcal{L}$ with respect to the geometric parameters $\boldsymbol{\theta}_{\text{geo}}$, which otherwise could not be realized using the standard formulations of PINNs (*44*, *45*). With the geometry-parameterized residual points, the PINN can correctly update the geometric parameters $\boldsymbol{\theta}_{\text{geo}}$ throughout the training process, thereby characterizing the unknown geometry. To the best of our knowledge, such form of parameterization invoking PINNs to solve geometry identification problems has hitherto not been addressed.

Procedure of simulation

We adopted Abaqus (*46*) as the finite element solver to generate the computational examples. Specifically, we preset reference values of unknown parameters to be $\boldsymbol{\theta}^*$ and conducted forward simulations, which generated the displacement data provided to the PINN and ground-truth full-field solution for assessing the performance of the PINN. The PINN initialized the estimation of unknown parameters to be $\boldsymbol{\theta}^0$. The PINN first went through a pretraining procedure for stabilizing the forward prediction, where the estimated parameters were fixed to be $\boldsymbol{\theta}^0$ (see Materials and Methods for details). As the PINN initiated parameter estimation through the iterative training process, we expected that the estimated parameters $\boldsymbol{\theta}$ would migrate toward the correct value $\boldsymbol{\theta}^*$. The training process terminated after the loss function and the estimated parameters reached a relative plateau, yielding the parameter estimation results $\widehat{\boldsymbol{\theta}}$. The detailed setups of the prototypical problem, the finite element solver, and the hyperparameters of the PINN are included in section S3.

We present the major results for cases 0 to 5 in the main text. Further results for cases 0 to 5 are included in section S4 (figs. S1 and S2). To justify our choice of hyperparameters of the PINN (width and depth of the NN, weights of loss components, and number of residual points), we show the results of a parametric study for a forward problem in sections S5 and S6 (figs. S3 to S5). In addition, we consider other modified setups of our inverse problem in sections S7 to S11 (figs. S6 to S9 and table S1) for illustrating the applicability, characteristics, and limitations of our method for the prototypical problems. For these additional cases, we summarize the objectives and major findings in the following sections and present the detailed results in the Supplementary Materials.

Parameter estimation results

The results of parameter estimation for cases 0 to 5 are shown in Table 1. For each case, we compare the estimated and reference values of the unknown parameters by presenting absolute errors and relative errors. To calculate the relative error, we normalize the coordinates, the lengths and the modulus, and the tilting angle by the domain size (side length of the matrix), respective reference values, and 180°, respectively. Table 1 indicates that the PINN estimates unknown parameters with high accuracy, with relative error $\mathcal{O}(10^{-2})$ on most parameters and as small as $\mathcal{O}(10^{-4})$ for some parameters.

It is worth noting that the estimated shear modulus of the inclusion μ_i in case 5 has an error slightly more than 10%. We provide a discussion on this issue in a following section ("Interpreting the convergence histories"). To improve the accuracy of case 5, we suppose that five additional data points inside the solid are available as in case 2. We retrain the PINN with the expanded measurement data and append the results in Table 1 as the modified case 5. With the additional data, the relative error of estimated parameters decreases to $\mathcal{O}(10^{-2})$, similar to other cases. In summary, given scattered displacement measurements, the PINN can accurately characterize the geometry (and material properties) of the internal void(s)/inclusion for various problem setups, including different constitutive relations, shapes of voids, and numbers of voids. The result indicates the generality of our method for solving a broad spectrum of inverse problems in mechanics of materials.

In sections S7 to S10, we provide additional parametric studies based on simplified cases 1 and 5 for demonstrating the influence of various factors on the estimation accuracy of unknown parameters, including the locations of measurement points (S7), the size of the void (S8), the location of the void (S9), and the moduli ratio of matrix and inclusion (S10). These studies show that our method is robust against various choices of true values $\boldsymbol{\theta}^*$, including different locations and different sizes [no smaller than $\mathcal{O}(10^{-1})$ of the matrix geometry] of the void, and moduli ratio of matrix and inclusion spanning within roughly $\mathcal{O}(10^{1})$. In addition, without prior knowledge on the location of the void, the measurement points should be uniformly placed on the matrix boundary, to make sure that the displacement data effectively capture the key information related to the void/inclusion.

Inference of deformed patterns

Our method not only is capable of estimating unknown parameters but also provides quantitative measures of the deformed patterns of the solid. Specifically, we apply the estimation results $\widehat{\boldsymbol{\lambda}}$ and $\widehat{\boldsymbol{\theta}}_{\text{geo}}$ (see Eq. 3) to the NN part of the PINN (top left panels in Figs. 2, A to C) to infer the deformed configuration, where $\widehat{\boldsymbol{\theta}}_{\text{geo}}$ determines the reference/undeformed configuration and $\widehat{\boldsymbol{\lambda}}$ determines the mapping from the reference/undeformed configuration to the deformed configuration. In Fig. 5, we display the comparison of the deformed configurations between the FEM ground truth (blue) and the PINN inference results (red for matrix; green for inclusion in case 5) for the six cases. Three snapshots are shown for each case after different numbers of training iterations ($k = 10^3$ and $M = 10^6$), which from the left to the right correspond to the completion of pretraining (beginning of parameter estimation), amid the training, and the completion of training, respectively. For clarity of presentation, this figure shows the outer and inner boundaries of the specimen visualized from the FEM and PINN analyses. In the snapshots in the second column, the two outlines match each other to a high extent. The remaining minor discrepancy gradually diminishes through the remaining iterations. After the training process is completed, the deformed configurations from the PINN are almost identical to those from the FEM ground truth. For case 5, specially, the inner boundary of the matrix (red) and the boundary of the inclusion (green) predicted by the PINN also overlap well with each other, indicating that the continuity of the material surfaces in the matrix-inclusion system is preserved in the inference of the PINN.

For case 3 where plasticity is involved, we also examined the inference of the plastic zone. Figure 6 shows the comparison of the plastic zone between the PINN prediction and the FEM ground truth. Not only is the geometry of the void characterized correctly (white region within the matrix) as previously verified in Table 1 and Fig. 5 but also the plastic zone of the loaded matrix is inferred with high accuracy.

Interpreting the convergence histories

Besides the final results obtained for parameter estimation and inference of deformation, we also address how the estimated values evolve toward the reference values during the training process. In

Table 1. Parameter estimation for all cases shown in Fig. 2. We compare the estimated values ($\hat{\boldsymbol{\theta}}$) and reference values ($\boldsymbol{\theta}^*$) of unknown parameters. To calculate the relative error, we normalize the coordinates, the lengths and the modulus, and the tilting angle by the domain size (side length of the matrix), their respective reference values, and 180°, respectively. To improve the accuracy of case 5, we provide the PINN with additional displacement measurement points inside the solid and then retrain the PINN, which is shown in the table as "Case 5 (with internal data)".

Case 0	$X_1^{(c)}$	$X_2^{(c)}$	A	B	Γ	
Estimated value	0.0488	0.0987	0.3475	0.1582	−29.42°	
Reference value	0.05	0.10	0.35	0.15	−30°	
Absolute error (× 10^{-2}, except Γ)	0.12	0.13	0.25	0.82	0.58°	
Relative error (%)	0.12	0.13	0.71	5.47	0.32	
Case 1	$X_1^{(c)}$	$X_2^{(c)}$	A	B	Γ	
Estimated value	0.0479	0.0991	0.3440	0.1602	−29.02°	
Reference value	0.05	0.10	0.35	0.15	−30°	
Absolute error (× 10–2, except Γ)	0.21	0.09	0.60	1.02	0.98°	
Relative error (%)	0.21	0.09	1.7	6.8	0.54	
Case 2	$X_1^{(1)}$	$X_2^{(1)}$	$X_1^{(2)}$	$X_2^{(2)}$		
Estimated value	−0.0399	0.3273	0.0396	−0.2315		
Reference value	−0.0392	0.3474	0.0392	−0.2474		
Absolute error (× 10^{-2})	0.07	2.01	0.04	1.59		
Relative error (%)	0.07	2.01	0.04	1.59		
Case 3	$X_1^{(c)}$	$X_2^{(c)}$	R			
Estimated value	0.0506	0.0999	0.2525			
Reference value	0.05	0.10	0.25			
Absolute error (× 10^{-2})	0.06	0.01	0.25			
Relative error (%)	0.06	0.01	1.00			
Case 4	$X_1^{(1)}$	$X_2^{(1)}$	$R^{(1)}$	$X_1^{(2)}$	$X_2^{(2)}$	$R^{(2)}$
Estimated value	−0.15089	0.10018	0.20007	0.25045	−0.05008	0.15019
Reference value	−0.15	0.10	0.20	0.25	−0.05	0.15
Absolute error (× 10^{-2})	0.089	0.018	0.007	0.045	0.008	0.019
Relative error (%)	0.089	0.018	0.04	0.045	0.008	0.13
Case 5	$X_1^{(c)}$	$X_2^{(c)}$	R	μ_i		
Estimated value	0.0496	0.0991	0.2583	0.0760		
Reference value	0.05	0.10	0.25	0.0667		
Absolute error (× 10^{-2})	0.04	0.09	0.83	0.93		
Relative error (%)	0.04	0.09	3.3	13.9		
Case 5 (with internal data)	$X_1^{(c)}$	$X_2^{(c)}$	R	μ_i		
Estimated value	0.0495	0.0998	0.2524	0.0687		
Reference value	0.05	0.10	0.25	0.0667		
Absolute error (× 10^{-2})	0.05	0.02	0.24	0.20		
Relative error (%)	0.05	0.02	0.96	3.0		

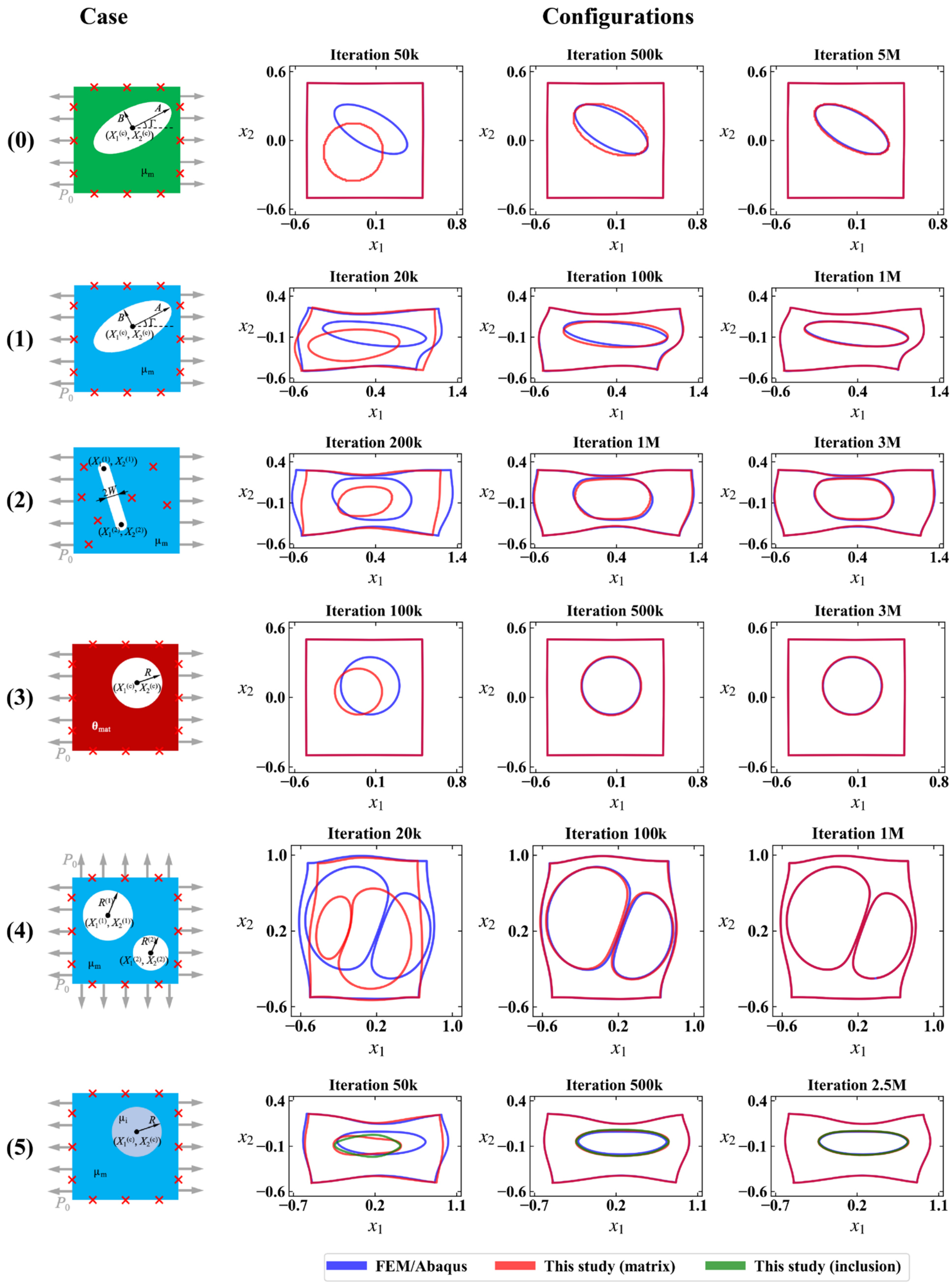


Fig. 5. Inference of the deformed patterns compared with FEM ground truth for all cases shown in Fig. 2. We displayed the visual outlines of deformed configurations of FEM/Abaqus (blue) and PINN results (red for matrix; green for inclusion in case 5). Three snapshots are shown for each case after different numbers of training iterations, which (from left to right) correspond to the completion of pretraining (beginning of parameter estimation), amid the training, and the completion of training, respectively (k = 10^3 and M = 10^6).

Fig. 7, we consider case 1 (Fig. 7, A and B) and case 5 (Fig. 7, C and D) as representative examples and show the convergence process for the estimated parameters (Fig. 7, A and C) and loss function (Fig. 7, B and D). The same figures for other cases are included in section S4 (fig. S1). Figure 7A shows the evolution of the estimated values of unknown parameters (solid lines) of case 1 during the training process. As the estimated values become trainable after the initial 20k iterations of pretraining (see Materials and Methods), they rapidly deviate from their respective initial guesses and gradually approach the reference values (dashed lines). After around 300k iterations, the estimations already approach the reference value. Within the remaining 700k iterations, the estimated values further approach the reference values slowly, reaching a high estimation accuracy at the end. Figure 7B shows the evolution of the loss function during the training process. The loss decreases from $\mathcal{O}(10^{-1})$ to $\mathcal{O}(10^{-4})$. Such a small value of the loss function indicates that all the conditions involved in the loss function are approximately satisfied by the PINN prediction. Similar to the evolution of estimated

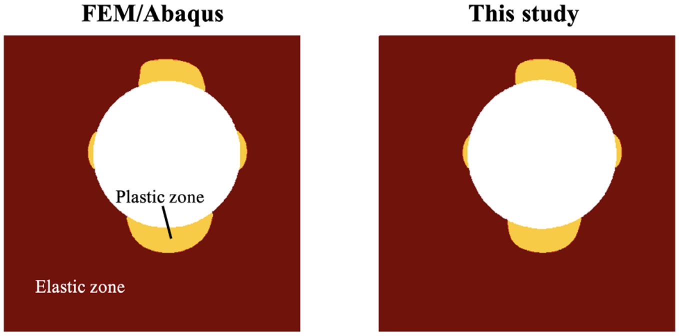


Fig. 6. Inference of the plastic zone compared with FEM ground truth in case 3 shown in Fig. 2. We mark the plastic zone by yellow and the void by white.

parameters, the loss decays rapidly at the early stage of training [from $\mathcal{O}(10^{-1})$ to $\mathcal{O}(10^{-3})$ within 200k iterations]. The rate of decrease turns to be substantially slower during the late stage. We find similar tendency for case 5 in Fig. 7 (C and D) on the evolution of estimated parameters and loss function.

Notably, we find that both the convergence rate and the estimation accuracy differ among the unknown parameters. For cases 1 and 5, the location of the void/inclusion described by $(X_1^{(c)}, X_2^{(c)})$ almost converges with high accuracy after around 150k/300k iterations, while the remaining unknown parameters exhibit worse behavior in terms of both convergence rate and estimation accuracy, including (A, B, Γ) in case 1 and (R, μ_i) in case 5. We attribute these two phenomena to the sensitivity of $\mathcal{L}_{\text{data}}$ with respect to estimated unknown parameters, or the identifiability of unknown parameters. $(X_1^{(c)}, X_2^{(c)})$ remarkably influence the displacement pattern at the outer boundary. Hence, a small deviation of estimated $(X_1^{(c)}, X_2^{(c)})$ from their reference values causes a large increase of $\mathcal{L}_{\text{data}}$. By examining the displacement data, one may even roughly estimate $(X_1^{(c)}, X_2^{(c)})$ by intuition. On the other hand, a significantly smaller increase of $\mathcal{L}_{\text{data}}$ is rendered for certain combinations of perturbation on (A, B, Γ) in case 1 and (R, μ_i) in case 5.

To support our statement on the cause of different convergence rates and estimation accuracies, we use the FEM solver to analyze how a perturbation $(\Delta R^*, \Delta \mu_i^*)$ on the reference values (R^*, μ_i^*) influences the displacement data collected on the measurement points in case 5. We show the root mean squared error of the displacement data caused by various combinations of $(\Delta R^*, \Delta \mu_i^*)$ in Fig. 8. From Fig. 8A, we observe that the error is significantly smaller for certain combinations of $\Delta R^* \Delta \mu_i^* > 0$ than for $\Delta R^* \Delta \mu_i^* < 0$. A detailed comparison along the two diagonal lines of the (R^*, μ_i^*) domain in Fig. 8A is displayed in Fig. 8B. The error along line 1 is roughly $\mathcal{O}(10^{-1})$ the error along line 2. Such a phenomenon indicates that $\mathcal{L}_{\text{data}}$ is insensitive to perturbations satisfying $\Delta R^* \Delta \mu_i^* > 0$. As the PINN estimates unknown parameters by minimizing the loss function, there exists intrinsically poor identifiability due to the coexistence of R and μ_i as unknown parameters and the placement of measurement points on the outer boundary. Such an interaction of R and μ_i has a twofold effect: First, the accurate estimation of R and μ_i is postponed to rather late stages of the training process when the total loss has been relatively small; second, the estimation error of R and μ_i is notably larger than other unknown parameters. Such an analysis matches our observation in Fig. 7C, where many more iterations are needed to estimate R and μ_i. The issue of poor identifiability in case 5 may be mitigated by providing a small number of additional internal data points as shown in Table 1 (see section S4 and fig. S2 for complete results). We conclude that the interplay between the unknown parameters and available data measurements renders relatively poor identifiability for some unknown parameters.

DISCUSSION

Inverse problems, especially those related to geometry identification, are notoriously difficult to solve for solids with nonlinear constitutive relations. With the hyperelastic solid undergoing severe distortion, it is hard to recover the unknown reference geometry with limited data. By examining the deformed pattern of the void in case 2 (see Fig. 5), it is intuitively not a straightforward task to trace back to the original slit in the reference configuration. In this work, we have demonstrated the capability of PINNs in effectively solving the geometry and material identification problems for engineering solids that incorporate large deformation response and plasticity through our computational examples for the matrix-void/inclusion system. We have shown that the present framework is able to accurately estimate the unknown geometric and material parameters with a relative error $\mathcal{O}(10^{-2})$ when proper displacement data are supplied to ensure identifiability.

The approach presented here has some unique characteristics, endowing this method with some distinct advantages. It provides a unified framework for solving forward problems and inverse problems with unknown parameters in PDEs (material identification) and/or domains (geometry identification), by combining the underlying mechanical principles and data into a deep NN. Unlike traditional methods based on FEM, there is neither the need to design problem-specific algorithms to update estimated unknown parameters beyond the forward solver nor the need to repeatedly remesh the computational domain throughout the iterations. In our method, the update of geometry is realized by the automated process built in the deep learning algorithms. In particular, the estimation of geometric parameters is automatically updated as the PINN seeks to minimize the loss function through the iterative training process. With deep learning libraries such as TensorFlow (*43*), the entire length of our PINN code for the current work is merely a few hundred lines. From the perspective of both design and implementation of the algorithm, PINNs reduce the human effort and related costs in setting up algorithms for inverse problems. On the other hand, compared to typical data-driven deep learning approaches, PINNs have the advantage of using well-established mechanics formulations as training guidelines, thereby requiring data only for the current instance of the problem setup and ensuring data efficiency.

Throughout this work, we adopt the Adam optimizer (*47*) as the optimization algorithm to achieve best accuracy and to study the convergence history as a fundamental characteristic of our method. The PINN is trained until both loss function and the estimated parameters reach a relative plateau. With such a setup, the computational time for case 4, for example, is around 11 hours on a typical machine [with central processing unit (CPU) only] to complete the entire 1M iterations and achieve high accuracy. We note that reasonable accuracy has been achieved within the first 200k iterations. One may further combine Adam and the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimizer (*48*) to achieve similar accuracy within much less computational time (around 30 min; see section S11 and table S1 for detailed results). Recently, parallel PINNs

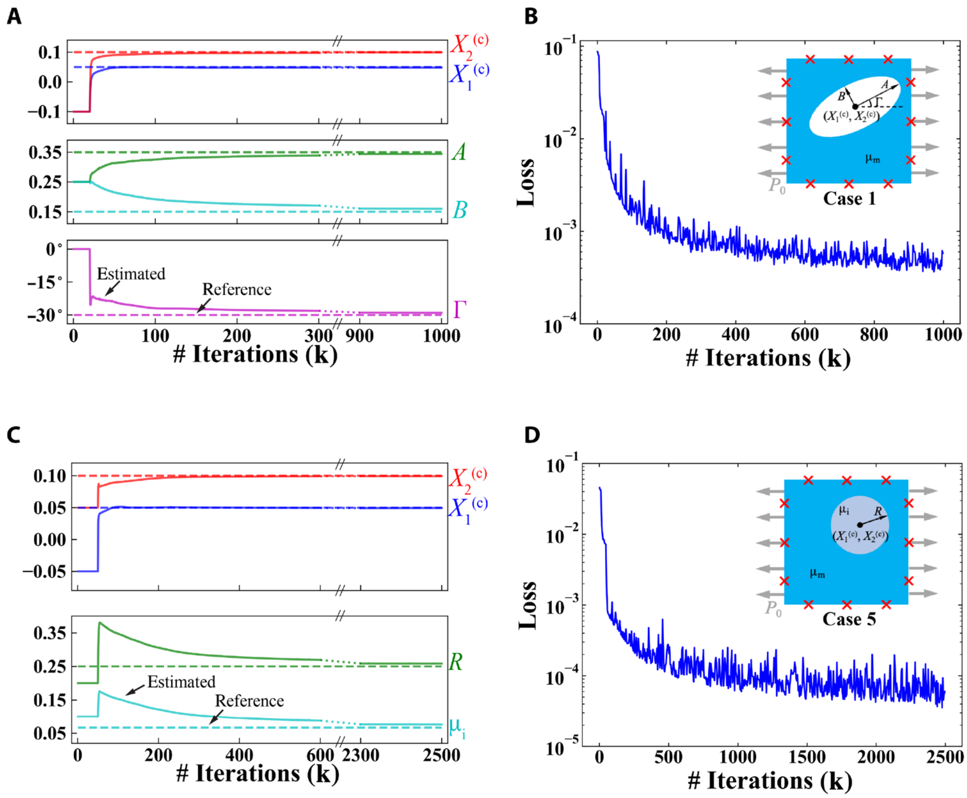


Fig. 7. Evolution of estimated unknown parameters and loss function in case 1 and case 5 during the training process. Case 1 (**A** and **B**) involves geometry identification, while case 5 (**C** and **D**) involves both material and geometry identification. See Fig. 2 for the definitions of the cases. (A and C) The dashed lines and solid lines represent the reference value and estimated value of unknown parameters. Unknown parameters are not updated in the pretraining process during the first 20k (for case 1) and 50k (for case 5) iterations, respectively. (B and D) The value of the loss function during the training process. The results for cases 0, 2, 3, and 4 are shown in section S4 (fig. S1).

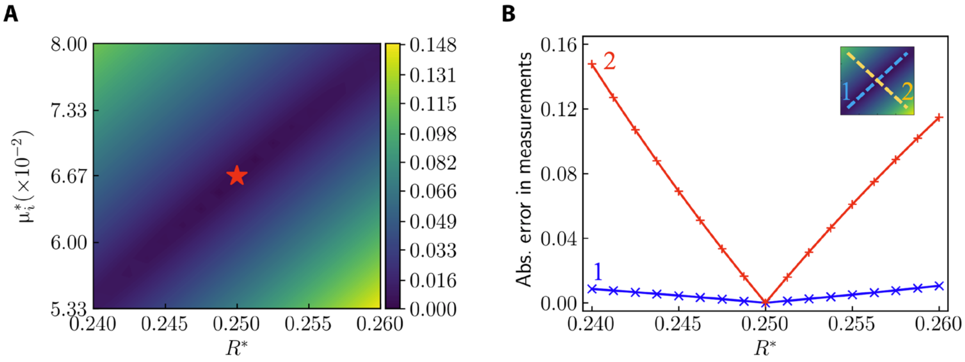


Fig. 8. Perturbation analysis of unknown parameters in case 5. Using the finite element solver, we calculate the root mean squared error of the displacement data on the measurement points caused by the perturbation ($\Delta R^*, \Delta\mu_i^*$) on the reference values of the shear modulus (μ_i^*) and the radius of the inclusion (R^*). (**A**) Absolute error of displacement data for various perturbations on (R^*, μ_i^*). The red star at the center corresponds to the unperturbed state ($\Delta R^* = \Delta\mu_i^* = 0$). (**B**) Absolute error of measurements along the diagonal lines of the (R^*, μ_i^*) domain in (A). Perturbation states on line 1 satisfy $\Delta R^* \Delta\mu_i^* > 0$. Perturbation states on line 1 satisfy $\Delta R^* \Delta\mu_i^* < 0$.

(*49*) have been proposed to accelerate the learning process of PINNs by using multiple CPUs and graphics processing units (GPUs) and introducing parallel algorithms. In addition, other studies have focused on analyzing convergence rate of PINNs and proposing practical techniques for accelerating convergence (*50*–*52*). With the ongoing efforts to improve the original formulation of PINNs, the computational efficiency is expected to be substantially enhanced over time.

We have focused on the prototypical problem as a simple proof of concept, seeking to characterize the internal structures with static loading on outer boundaries. According to Saint-Venant's principle, under static loading, the inhomogeneous stress and deformation states caused by the internal void/inclusion decay as the distance from the void/inclusion increases. Subsequently, the measurements on outer boundaries essentially provide the PINN with limited amount of information regarding the internal void/inclusion. Modern experimental techniques have adopted dynamic external loading such as ultrasound (*53*) to acquire time-dependent measurements, through which we anticipate that the performance of our method will benefit from more information provided by measurements.

Our method can be applied to a wide range of engineering problems. Defect detection represents a broad class of practical engineering needs in various fields, where identification and characterization of internal structures and defects in materials are essential. Experimental techniques have so far been developed for different materials based on ultrasound (*54*), active thermography (*55*), eddy current (*56*, *57*), optical coherent tomography (*58*), and microwave (*59*). By integrating the respective physical principles in these problems, our approach can potentially be combined with these techniques for dealing with unknown and moving geometries, which extends our method beyond continuum solid mechanics. Notably, one may need to carefully consider the applicability of governing PDEs for practical problems. For instance, continuum solid mechanics does not take into consideration the length scale of microstructures of materials, so that continuum mechanics is accurate only when the key dimensions in the problem (e.g., void size) are much larger than these intrinsic length scales of materials. Our method can also be used for structure design/optimization problems, where typically a mechanical structure is designed with optimized stiffness within volume constraints. For these problems, PINNs can incorporate the design target as a loss term, aspects of which have been preliminarily explored in (*60*).

MATERIALS AND METHODS

PINNs for continuum solid mechanics

We introduce the detailed formulation of PINNs for inverse problems in continuum solid mechanics. Here, we focus on the PINN for hyperelasticity (specifically, incompressible Neo-Hookean material) as most of our computational examples adopt this material model (see section S2 for the mechanics of hyperelastic materials). To better clarify the quantitative formulation, here, we denote all the material and geometric parameters of interest as $\boldsymbol{\theta}_{\text{mat}}$ and $\boldsymbol{\theta}_{\text{geo}}$, respectively. For incompressible Neo-Hookean materials, the only material parameter is the shear modulus μ so that $\boldsymbol{\theta}_{\text{mat}} = \mu$. The unknown part of $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\text{mat}}, \boldsymbol{\theta}_{\text{geo}})$ in the inverse problem is denoted as $\boldsymbol{\theta}_{\text{unk}}$.

As summarized in Results, the workflow of PINNs comprises four steps. First, we apply a NN to approximate the primary solution fields (top left panel in Fig. 2B) in domain $\Omega(\boldsymbol{\theta}_{\text{geo}})$, including the displacement field $\tilde{\mathbf{u}}(\mathbf{X};\boldsymbol{\lambda})$ and the pressure field $\tilde{p}(\mathbf{X};\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ represents trainable parameters of the NN, $\mathbf{X} = (X_1, X_2)$ is the in-plane coordinates in the reference/undeformed configuration, and the quantities with tilde represent the approximation from the NN. For incompressible materials, we need the hydrostatic pressure field p as a Lagrange multiplier accompanying the displacement field $\mathbf{u}$ to uniquely determine the stress field.

Second, we integrate mechanical laws into the PINN architecture (top right panel in Fig. 2B) by deriving relevant mechanical quantities of interest from the NN outputs. During this calculation process, partial derivatives are handled by automatic differentiation. The deformation gradient $\tilde{\mathbf{F}}(\mathbf{X};\boldsymbol{\lambda})$ and the first Piola-Kirchhoff stress $\tilde{\mathbf{P}}(\mathbf{X};\boldsymbol{\lambda},\mu)$ are calculated by

$$\tilde{\mathbf{F}}(\mathbf{X};\boldsymbol{\lambda}) = \mathbf{I} + \frac{\partial \tilde{\mathbf{u}}}{\partial \mathbf{X}}(\mathbf{X};\boldsymbol{\lambda}) \tag{4}$$

$$\tilde{\mathbf{P}}(\mathbf{X};\boldsymbol{\lambda},\mu) = -\tilde{p}(\mathbf{X};\boldsymbol{\lambda})\,\tilde{\mathbf{F}}^{-\mathrm{T}}(\mathbf{X};\boldsymbol{\lambda}) + \mu\tilde{\mathbf{F}}(\mathbf{X};\boldsymbol{\lambda}) \tag{5}$$

where $\mathbf{I}$ is the identity tensor, Eq. 4 is kinematics, and Eq. 5 is the constitutive relation for incompressible Neo-Hookean materials. The residuals of the equilibrium PDE and the incompressibility condition at $\mathbf{X}$ are expressed by

$$\tilde{\mathbf{r}}_{\text{PDE}}(\mathbf{X};\boldsymbol{\lambda},\mu) = \text{Div}\,\tilde{\mathbf{P}}(\mathbf{X};\boldsymbol{\lambda},\mu), \quad \mathbf{X} \in \Omega(\boldsymbol{\theta}_{\text{geo}}) \tag{6}$$

$$\tilde{r}_{\text{inc}}(\mathbf{X};\boldsymbol{\lambda}) = \det(\tilde{\mathbf{F}}(\mathbf{X};\boldsymbol{\lambda})) - 1, \quad \mathbf{X} \in \Omega(\boldsymbol{\theta}_{\text{geo}}) \tag{7}$$

The residuals of Dirichlet/displacement and Neumann/traction BCs at $\mathbf{X}$ are

$$\tilde{\mathbf{r}}_{\text{D}}(\mathbf{X};\boldsymbol{\lambda}) = \tilde{\mathbf{u}}(\mathbf{X};\boldsymbol{\lambda}) - \bar{\mathbf{u}}(\mathbf{X}), \quad \mathbf{X} \in \partial\Omega_{\text{D}}(\boldsymbol{\theta}_{\text{geo}}) \tag{8}$$

$$\tilde{\mathbf{r}}_{\text{N}}(\mathbf{X};\boldsymbol{\lambda},\mu) = \tilde{\mathbf{P}}(\mathbf{X};\boldsymbol{\lambda},\mu)\,\mathbf{N}(\mathbf{X}) - \bar{\mathbf{T}}(\mathbf{X}), \quad \mathbf{X} \in \partial\Omega_{\text{N}}(\boldsymbol{\theta}_{\text{geo}}) \tag{9}$$

where $\mathbf{N}$ is the outward unit normal vector on the boundary, and $\bar{\mathbf{u}}$ and $\bar{\mathbf{T}}$ are the specified displacement and traction on the boundary, respectively. $\partial\Omega_{\text{D}}(\boldsymbol{\theta}_{\text{geo}})$ and $\partial\Omega_{\text{N}}(\boldsymbol{\theta}_{\text{geo}})$ refer to the domains for Dirichlet/displacement and Neumann/traction BCs, respectively. For inverse problems, we have displacement data $\{\mathbf{u}^{*(i)}\}_{i=1}^{N_u}$ at $\{\mathbf{X}_u^{(i)}\}_{i=1}^{N_u}$. The residual of the ith displacement observation is

$$\tilde{\mathbf{r}}_u^{(i)}(\boldsymbol{\lambda}) = \tilde{\mathbf{u}}(\mathbf{X}_u^{(i)};\boldsymbol{\lambda}) - \mathbf{u}^{*(i)} \tag{10}$$

Third, we formulate the loss function according to the foregoing residuals from mechanics and data (bottom right panel in Fig. 2B). To define the loss terms corresponding to the problem definition, we place N_Ω, N_{D}, and N_{N} residual points in Ω, on $\partial\Omega_{\text{D}}$ and $\partial\Omega_{\text{N}}$, denoted as $\mathbf{X}_\Omega^{(i)}$ ($i \in \{1,2, \ldots, N_\Omega\}$), $\mathbf{X}_{\text{D}}^{(i)}$ ($i \in \{1,2, \ldots, N_{\text{D}}\}$), and $\mathbf{X}_{\text{N}}^{(i)}$ ($i \in \{1,2, \ldots, N_{\text{N}}\}$), respectively. Because we parameterize the coordinates of residual points by $\boldsymbol{\theta}_{\text{geo}}$, these residual points are all parameterized by $\boldsymbol{\theta}_{\text{geo}}$. We evaluate the mean squared residuals of the PDEs, the incompressibility condition, Dirichlet and Neumann BCs, and data, respectively. Each loss term is defined by

$$\mathcal{L}_{\text{PDE}}(\boldsymbol{\lambda},\boldsymbol{\theta}) = \frac{1}{N_\Omega}\sum_{i=1}^{N_\Omega} |\tilde{\mathbf{r}}_{\text{PDE}}(\mathbf{X}_\Omega^{(i)}(\boldsymbol{\theta}_{\text{geo}});\ \boldsymbol{\lambda},\mu)|^2 \tag{11}$$

$$\mathcal{L}_{\text{inc}}(\boldsymbol{\lambda},\boldsymbol{\theta}) = \frac{1}{N_\Omega}\sum_{i=1}^{N_\Omega} |\tilde{r}_{\text{inc}}(\mathbf{X}_\Omega^{(i)}(\boldsymbol{\theta}_{\text{geo}});\ \boldsymbol{\lambda})|^2 \tag{12}$$

$$\mathcal{L}_{\text{D}}(\boldsymbol{\lambda},\boldsymbol{\theta}) = \frac{1}{N_{\text{D}}}\sum_{i=1}^{N_{\text{D}}} |\tilde{\mathbf{r}}_{\text{D}}(\mathbf{X}_{\text{D}}^{(i)}(\boldsymbol{\theta}_{\text{geo}});\ \boldsymbol{\lambda})|^2 \tag{13}$$

$$\mathcal{L}_{\text{N}}(\boldsymbol{\lambda},\boldsymbol{\theta}) = \frac{1}{N_{\text{N}}}\sum_{i=1}^{N_{\text{N}}} |\tilde{\mathbf{r}}_{\text{N}}(\mathbf{X}_{\text{N}}^{(i)}(\boldsymbol{\theta}_{\text{geo}});\ \boldsymbol{\lambda},\mu)|^2 \tag{14}$$

$$\mathcal{L}_u(\boldsymbol{\lambda},\boldsymbol{\theta}) = \frac{1}{N_u}\sum_{i=1}^{N_u} |\tilde{\mathbf{r}}_u^{(i)}(\boldsymbol{\lambda})|^2 \tag{15}$$

and the loss function is

$$\begin{aligned}\mathcal{L}(\boldsymbol{\lambda},\boldsymbol{\theta}) = {} & \alpha_{\text{PDE}}\mathcal{L}_{\text{PDE}}(\boldsymbol{\lambda},\boldsymbol{\theta}) + \alpha_{\text{inc}}\mathcal{L}_{\text{inc}}(\boldsymbol{\lambda},\boldsymbol{\theta}) + \alpha_{\text{D}}\mathcal{L}_{\text{D}}(\boldsymbol{\lambda},\boldsymbol{\theta}) \\ & + \alpha_{\text{N}}\mathcal{L}_{\text{N}}(\boldsymbol{\lambda},\boldsymbol{\theta}) + \alpha_u\mathcal{L}_u(\boldsymbol{\lambda},\boldsymbol{\theta})\end{aligned} \tag{16}$$

where α_{PDE}, α_{inc}, α_{D}, α_{N}, and α_u are the weights of the loss terms. Note that the two loss terms $\mathcal{L}_{\text{D}}$ and $\mathcal{L}_{\text{N}}$ for the two types of BCs are simplified into $\mathcal{L}_{\text{BC}}$ in Eq. 1 in Results.

Last, we conduct parameter estimation through training/loss minimization (bottom left panel in Fig. 2B). The trainable parameters of the PINN include the trainable parameters of the NN, $\boldsymbol{\lambda}$, and the unknown parameters of the inverse problem, $\boldsymbol{\theta}_{\text{unk}}$ ($\subseteq \boldsymbol{\theta}$). Using the notations in this section, this process can be expressed as

$$\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\theta}}_{\text{unk}} = \underset{\boldsymbol{\lambda}, \boldsymbol{\theta}_{\text{unk}} \subseteq \boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\theta}) \tag{17}$$

With the PINN adjusting $\boldsymbol{\lambda}$ to minimize the loss function, we anticipate that all the mechanical laws will be approximately satisfied, making the NN serve as an approximation to the primary solution fields. Furthermore, the residual of displacement observations in the loss function guides the estimated unknown parameters to evolve toward their respective target values. In this way, the PINN is able to solve inverse problems.

In section S2, we provide additional information regarding the formulation of PINNs. This includes the formulation for forward problems, for linear elasticity and deformation plasticity, and for multiple materials, which is related to case 5 in our main text.

Pretraining procedure

We find it necessary to pretrain the model before using the model to characterize unknown geometry. If we directly apply the model without pretraining, then the estimated geometric parameters rapidly depart from physically admissible values (e.g., void located outside the matrix) after a few iterations. Inspired by the transfer learning technique, we propose to maintain all the estimated unknown parameters fixed (not trainable) and only update the trainable parameters of the NN $\boldsymbol{\lambda}$ for the first few iterations. During this pretraining process, the PINN essentially solves a forward problem, seeking to roughly capture the qualitative pattern of the displacement field and the stress field. After this pretraining process, we initiate the parameter estimation process by making both $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ trainable. Such a pretraining procedure induces $\boldsymbol{\lambda}$ to converge to the desired local minimum, hence serving as a good initialization for the geometry identification problem. For our prototypical problem, technically, the PINN needs to be pretrained until there emerges a qualitative pattern indicating the existence of a stress concentration around the void or the soft inclusion.

SUPPLEMENTARY MATERIALS

Supplementary material for this article is available at https://science.org/doi/10.1126/sciadv.abk0644

REFERENCES AND NOTES

1. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
2. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Proces. Syst.* **25**, 1097–1105 (2012).
3. G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process. Mag.* **29**, 82–97 (2012).
4. K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, A. Walsh, Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
5. Z. Shi, E. Tsymbalov, M. Dao, S. Suresh, A. Shapeev, J. Li, Deep elastic strain engineering of bandgap through machine learning. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 4117–4122 (2019).
6. Z. Shi, M. Dao, E. Tsymbalov, A. Shapeev, J. Li, S. Suresh, Metallization of diamond. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 24634–24639 (2020).
7. L. Lu, M. Dao, P. Kumar, U. Ramamurty, G. E. Karniadakis, S. Suresh, Extraction of mechanical properties of materials through deep learning from instrumented indentation. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 7052–7062 (2020).
8. Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks. *Comput. Aided Civ. Inf. Eng.* **32**, 361–378 (2017).
9. H. Adeli, Neural networks in civil engineering: 1989–2000. *Comput. Aided Civ. Inf. Eng.* **16**, 126–142 (2001).
10. M. Yin, E. Ban, B. V. Rego, E. Zhang, C. Cavinato, J. D. Humphrey, G. E. Karniadakis, Simulating progressive intramural damage leading to aortic dissection using an operator-regression neural network. arXiv:2108.11985 [cs.CE] (25 August 2021).
11. H. Jin, Big-data-driven multi-scale experimental study of nanostructured block copolymer's dynamic toughness, Ph.D. thesis, Brown University, Providence, RI (2021).
12. M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).
13. M. Raissi, A. Yazdani, G. E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* **367**, 1026–1030 (2020).
14. S. Cai, H. Li, F. Zheng, F. Kong, M. Dao, G. E. Karniadakis, S. Suresh, Artificial intelligence velocimetry and microaneurysm-on-a-chip for three-dimensional analysis of blood flow in physiology and disease. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2100697118 (2021).
15. S. Cai, Z. Mao, Z. Wang, M. Yin, G. E. Karniadakis, Physics-informed neural networks (pinns) for fluid mechanics: A review. arXiv:2105.09506 [physics.flu-dyn] (20 May 2021).
16. 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. *Comput. Methods Appl. Mech. Eng.* **362**, 112790 (2020).
17. C. Rao, H. Sun, Y. Liu, Physics-informed deep learning for computational elastodynamics without labeled data. *J. Eng. Mech.* **147**, 04021043 (2021).
18. E. Zhang, M. Yin, G. E. Karniadakis, Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging. arXiv:2009.04525 [cs.LG] (2 September 2020).
19. J. N. Fuhg, N. Bouklas, The mixed deep energy method for resolving concentration features in finite strain hyperelasticity. arXiv:2104.09623 [cs.CE] (15 April 2021).
20. K. Shukla, P. C. Di Leoni, J. Blackshire, D. Sparkman, G. E. Karniadakis, Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. *J. Nondestruct. Eval.* **39**, 61 (2020).
21. M. Yin, X. Zheng, J. D. Humphrey, G. E. Karniadakis, Non-invasive inference of thrombus material properties with physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.* **375**, 113603 (2021).
22. S. Goswami, M. Yin, Y. Yu, G. Karniadakis, A physics-informed variational deeponet for predicting the crack path in brittle materials. arXiv:2108.06905 [cs.LG] (16 August 2021).
23. L. Yang, D. Zhang, G. E. Karniadakis, Physics-informed generative adversarial networks for stochastic differential equations. *SIAM J. Sci. Comput.* **42**, A292–A317 (2020).
24. G. Pang, L. Lu, G. E. Karniadakis, fPINNs: Fractional physics-informed neural networks. *SIAM J. Sci. Comput.* **41**, A2603–A2626 (2019).
25. T. L. Anderson, *Fracture Mechanics: Fundamentals and Applications* (CRC Press, 2017).
26. S. Suresh, *Fatigue of Materials* (Cambridge Univ. Press, 2012).
27. L. B. Freund, S. Suresh, *Thin Film Materials: Stress, Defect Formation and Surface Evolution* (Cambridge Univ. Press, 2010).
28. H. B. Ameur, M. Burger, B. Hackl, Level set methods for geometric inverse problems in linear elasticity. *Inverse Probl.* **20**, 673–696 (2004).
29. H. Sun, H. Waisman, R. Betti, Nondestructive identification of multiple flaws using xfem and a topologically adapting artificial bee colony algorithm. *Int. J. Numer. Methods Eng.* **95**, 871–900 (2013).
30. R. Gallego, G. Rus, Identification of cracks and cavities using the topological sensitivity boundary integral equation. *Comput. Mech.* **33**, 154–163 (2004).
31. J. Jung, E. Taciroglu, Modeling and identification of an arbitrarily shaped scatterer using dynamic xfem with cubic splines. *Comput. Methods Appl. Mech. Eng.* **278**, 101–118 (2014).
32. H. Waisman, E. Chatzi, A. W. Smyth, Detection and quantification of flaws in structures by the extended finite element method and genetic algorithms. *Int. J. Numer. Methods Eng.* **82**, 303–328 (2010).
33. D. Schnur, N. Zabaras, An inverse method for determining elastic material properties and a material interface. *Int. J. Numer. Methods Eng.* **33**, 2039–2057 (1992).
34. Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems. *Evol. Comput.* **4**, 1–32 (1996).
35. S. Chaabane, M. Masmoudi, H. Meftahi, Topological and shape gradient strategy for solving geometrical inverse problems. *J. Math. Anal. Appl.* **400**, 724–742 (2013).
36. Y. Mei, R. Fulmer, V. Raja, S. Wang, S. Goenezen, Estimating the non-homogeneous elastic modulus distribution from surface deformations. *Int. J. Solids Struct.* **83**, 73–80 (2016).
37. S. Amstutz, I. Horchani, M. Masmoudi, Crack detection by the topological gradient method. *Control. Cybern.* **34**, 81–101 (2005).
38. T. J. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis* (Courier Corporation, 2012).
39. C. R. Vogel, *Computational Methods for Inverse Problems* (SIAM, 2002).
40. A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics. *Comput. Methods Appl. Mech. Eng.* **197**, 3768–3782 (2008).
41. A. D. Jagtap, E. Kharazmi, G. E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Comput. Methods Appl. Mech. Eng.* **365**, 113028 (2020).

42. Y. Chen, L. Lu, G. E. Karniadakis, L. Dal Negro, Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Opt. Express* **28**, 11618–11633 (2020).
43. 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. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous distributed systems (2016).
44. S. Wang, P. Perdikaris, Deep learning of free boundary and Stefan problems. *J. Comput. Phys.* **428**, 109914 (2021).
45. O. Hennigh, S. Narasimhan, M. A. Nabian, A. Subramaniam, K. Tangsali, M. Rietmann, J. d. A. Ferrandis, W. Byeon, Z. Fang, S. Choudhry, NVIDIA SimNet: An AI-accelerated multi-physics simulation framework. arXiv:2012.07938 [physics.flu-dyn] (14 December 2020).
46. Abaqus, *Abaqus 2020 Documentation* (Dassault Systèmes, 2020).
47. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization. arXiv:1412.6980 [cs.LG] (22 December 2014).
48. D. C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization. *Math. Program.* **45**, 503–528 (1989).
49. K. Shukla, A. D. Jagtap, G. E. Karniadakis, Parallel physics-informed neural networks via domain decomposition. arXiv:2104.10013 [cs.DC] (20 April 2021).
50. Y. Shin, J. Darbon, G. E. Karniadakis, On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type pdes. arXiv:2004.01806 [math.NA] (2 April 2020).
51. S. Wang, X. Yu, P. Perdikaris, When and why PINNS fail to train: A neural tangent kernel perspective. *J. Comput. Phys.* **449**, 110768 (2020).
52. S. Wang, H. Wang, P. Perdikaris, On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDES with physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.* **384**, 113938 (2021).
53. M. M. Doyley, Model-based elastography: A survey of approaches to the inverse elasticity problem. *Phys. Med. Biol.* **57**, R35–R73 (2012).
54. T. D'orazio, M. Leo, A. Distante, C. Guaragnella, V. Pianese, G. Cavaccini, Automatic ultrasonic inspection for internal defect detection in composite materials. *NDT E Int.* **41**, 145–154 (2008).
55. B. Lahiri, S. Bagavathiappan, P. Reshmi, J. Philip, T. Jayakumar, B. Raj, Quantification of defects in composites and rubber materials using active thermography. *Infrared Phys. Technol.* **55**, 191–199 (2012).
56. D.-G. Park, C. S. Angani, B. Rao, G. Vértesy, D.-H. Lee, K.-H. Kim, Detection of the subsurface cracks in a stainless steel plate using pulsed eddy current. *J. Nondestruct. Eval.* **32**, 350–353 (2013).
57. L. Cheng, G. Y. Tian, Surface crack detection for carbon fiber reinforced plastic (cfrp) materials using pulsed eddy current thermography. *IEEE Sensors J.* **11**, 3261–3268 (2011).
58. M. Bashkansky, M. Duncan, M. Kahn, D. Lewis, J. Reintjes, Subsurface defect detection in ceramics by high-speed high-resolution optical coherent tomography. *Opt. Lett.* **22**, 61–63 (1997).
59. M. Klemm, J. Leendertz, D. Gibbins, I. Craddock, A. Preece, R. Benjamin, Microwave radar-based breast cancer detection: Imaging in inhomogeneous breast phantoms. *IEEE Antennas Wirel. Propag. Lett.* **8**, 1349–1352 (2009).
60. L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, S. G. Johnson, Physics-informed neural networks with hard constraints for inverse design. arXiv:2102.04626 [physics.comp-ph] (9 February 2021).
61. J. Céa, S. Garreau, P. Guillaume, M. Masmoudi, The shape and topological optimizations connection. *Comput. Methods Appl. Mech. Eng.* **188**, 713–726 (2000).
62. S. Goenezen, P. Barbone, A. A. Oberai, Solution of the nonlinear elasticity imaging inverse problem: The incompressible case. *Comput. Methods Appl. Mech. Eng.* **200**, 1406–1420 (2011).
63. A. D. Jagtap, K. Kawaguchi, G. E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *J. Comput. Phys.* **404**, 109136 (2020).
64. X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (PMLR, 2010), pp. 249–256.

Acknowledgments

Funding: The work was supported by the Department of Energy PhILMs project DE-SC001954 and OSD/AFOSR MURI grant FA9550-20-1-0358. M.D. was supported by the National Science Foundation (NSF) award 2004556. S.S. was supported by Nanyang Technological University, Singapore, through the Distinguished University Professorship (S.S.). **Author contributions:** E.Z. developed the method, implemented the computer code, and performed computations. E.Z., M.D., G.E.K., and S.S. analyzed data. E.Z., M.D., G.E.K., and S.S. wrote the paper. M.D. and G.E.K. conceived the project. G.E.K., M.D., and S.S. supervised the project. **Competing interests:** All authors are coinventors on a provisional patent related to this work to be filed at the U.S. Patent and Trademark Office. The authors declare that they have no other competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials.

Submitted 26 June 2021
Accepted 22 December 2021
Published 16 February 2022
10.1126/sciadv.abk0644