Analyses of Internal Structures and Defects in Materials Using Physics-Informed Neural Networks

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

[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
[†]Corresponding authors. E-mail: mingdao@mit.edu (M.D.);
george_karniadakis@brown.edu (G.E.K.);
ssuresh@ntu.edu.sg (S.S.)

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 (PINNs) for identifying unknown geometric and material parameters. By employing 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.

Teaser

Physics-informed deep learning helps detect unknown internal structures and defects with limited non-destructive measurements

Introduction

Deep learning (*1*) approaches play an increasingly significant 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 encode such formulations implicitly 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 Neural Networks (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*). Such 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 new 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 following (*25–27*): safety and failure analysis of civil, mechanical, nuclear and aeronautical structures; land, sea and air transportation; relia-bility analysis in microelectronic devices; non-destructive testing of materials; and processing of engineered materials. In a geometry identification problem, the unknown geometric fea-tures and parameters are determined in a solid material/structure given measured material re-sponse under static or dynamic loading, thereby characterizing unknown structures including internal defects or boundaries such as voids, vacancies or holes (*28–32*), inclusions and rein-forcements (*31, 33–36*), and/or cracks (*30–32, 37*). Traditionally, computational algorithms for geometry identification are established based on 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 parame-ters (*39*) (see Section S1 in Supplementary Materials (SM) 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 in-herently brings about complications in these algorithms. The estimated geometry is updated by repeatedly re-meshing the domain through iterations (*33*). Alternatively, the 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 consti-tutive 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 utilizing the workflow of NNs, our method can automatically update the geometry estimation. 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 SM more systematic studies of additional cases and parametric analyses highlighting the advantages and limitations of the method.

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 an 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 due to 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). Finally, 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 setup 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 (1) Fig. 3A for (compressible) linear elasticity, (2) Fig. 3B for (incompressible) hyperelasticity, and (3) Fig. 3C for (compressible) deformation plasticity. The architectures of the PINNs are slightly different for different material models due to the characteristics of their mathematical expressions. Fig. 3D includes the definitions of the the mechanical quantities of interest in the architectures.

The detailed formulations and relevant governing equations of PINNs are explained in Materials and Methods and in Section S2 in SM. Here we summarize the basic workflow of PINNs as follows. First, we apply a neural network (NN; with trainable parameters $\boldsymbol{\lambda}$) to approximate the primary solution fields (top left panels in Fig. 2A-2C) with respect to the in-plane coordinates $\mathbf{X} = (X_1, X_2)$. Secondly, we integrate the mechanical laws into the PINN architecture (top right panels in Fig. 2A-2C) 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 $\boldsymbol{\theta}_{\text{mat}}$ are involved. Thirdly, we formulate the loss function $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\theta})$, 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. 2A-2C). For example, for linear elasticity in Fig. 2A, the loss function is expressed as

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \alpha_{\text{PDE}}\mathcal{L}_{\text{PDE}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) + \alpha_{\text{BC}}\mathcal{L}_{\text{BC}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) + \alpha_{\text{Data}}\mathcal{L}_{\text{Data}}(\boldsymbol{\lambda}, \boldsymbol{\theta}), \tag{1}$$

where the three loss terms $\mathcal{L}_j(\boldsymbol{\lambda}, \boldsymbol{\theta})$ ($j = \text{PDE}, \text{BC}, \text{Data}$) 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

$$\mathcal{L}_j(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \frac{1}{N_j}\sum_{i=1}^{N_j}\left|\mathbf{r}_j\left(\mathbf{X}_j^{(i)}(\boldsymbol{\theta}_{\text{geo}}); \boldsymbol{\lambda}, \boldsymbol{\theta}_{\text{mat}}\right)\right|^2, \tag{2}$$

where $\mathbf{r}_j$ is the residual of the condition j at the i-th residual point $\mathbf{X}_j^{(i)}(\boldsymbol{\theta}_{\text{geo}})$. The N_j residual points are distributed in the domain of condition j to correctly evaluate $\mathcal{L}_j(\boldsymbol{\lambda}, \boldsymbol{\theta})$. 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 $\mathbf{X}_j^{(i)}$ depends on geometric parameters $\boldsymbol{\theta}_{\text{geo}}$ because of the variable computational domain, which will be explained in detail in the next section. Lastly, we conduct parameter estimation through the training of the PINN (bottom left panels in Fig. 2A-2C), during which the estimated unknown parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\text{mat}}, \boldsymbol{\theta}_{\text{geo}})$ and neural networks parameters $\boldsymbol{\lambda}$ are updated/trained to minimize the loss function. This process can be expressed as

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

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

Formulation for Geometry Identification. Geometric parameters $\boldsymbol{\theta}_{\text{geo}}$ play an essentially different role in the inverse problem compared to material parameters $\boldsymbol{\theta}_{\text{mat}}$. Material parameters parameterize the governing PDEs of mechanics, which are naturally endowed with trainability through automatic differentiation of (physics-informed) neural networks. As a result, material parameters 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 boundary conditions, 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 utilizing 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 $\boldsymbol{\theta}_{\text{geo}}$ are 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 ge-ometric 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 have 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 firstly went through a pre-training 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 towards 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 $\hat{\boldsymbol{\theta}}$. The detailed setups of the prototypical problem, the finite element solver, and the hyperparameters of the PINN are included in Section S3 in SM.

We present the major results for cases 0-5 in the main text. Further results for cases 0-5 are included in Section S4 (Figs. S1 and S2) in SM. 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-S5) in SM. Additionally, we consider other modified setups of our inverse problem in Sections S7-S11 (Figs. S6-S9 and Table S1) in SM 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 SM.

Parameter Estimation Results. The results of parameter estimation for cases 0-5 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 re-train 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-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 captures 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 as well. Specifically, we apply the estimation results $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\theta}}_{\text{geo}}$ (see Eq. 3) to the neural network part of the PINN (top left panels in Figs. 2A-2C) to infer the deformed configuration, where $\hat{\boldsymbol{\theta}}_{\text{geo}}$ determines the reference/undeformed configuration, and $\hat{\boldsymbol{\lambda}}$ 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 $= 10^3$, M $= 10^6$), which from the left to the right correspond to the completion of pre-training (begin-ning 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 another to a high extent. The remaining minor discrepancy gradually diminishes through the remaining iterations. After the training process completes, 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. Fig. 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, the plastic zone of the loaded matrix is also 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 towards the reference values during the training process. In Fig. 7, we consider case 1 (Fig. 7A and 7B) and case 5 (Fig. 7C and 7D) as representative examples and show the convergence process for the estimated parameters (Fig. 7A and 7C) and loss function (Fig. 7B and 7D). The same figures for other cases are included in Section S4 (Fig. S1) in SM. Fig. 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 pre-training (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. Fig. 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 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 significantly slower during the late stage. We find similar tendency for case 5 in Fig. 7C and 7D on the evolution of estimated parameters and loss function.

Notably, we find that both the convergence rate and estimation accuracy differ among the unknown parameters. For cases 1 and 5, the location of the void/inclusion described by $(X_1^{(\mathrm{c})}$, $X_2^{(\mathrm{c})})$ 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 $(A,\, B,\, \Gamma)$ in case 1 and $(R,\, \mu_i)$ in case 5. We attribute these two phenomena to the sensitivity of $\mathcal{L}_{\mathrm{data}}$ with respect to estimated unknown parameters, or the identifiability of unknown parameters. $(X_1^{(\mathrm{c})},\ X_2^{(\mathrm{c})})$ remarkably influence the displacement pattern at the outer boundary. Hence, a small deviation of estimated $(X_1^{(\mathrm{c})},\ X_2^{(\mathrm{c})})$ from their reference values causes a large increase of $\mathcal{L}_{\text{data}}$. In fact, by examining the displacement data, one may even roughly estimate $(X_1^{(\mathrm{c})},\ X_2^{(\mathrm{c})})$ by intuition. On the other hand, a significantly smaller increase of $\mathcal{L}_{\text{data}}$ is rendered for certain combinations of perturbation on $(A,\ B,\ \Gamma)$ in case 1 and $(R,\ \mu_i)$ 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 $(R^*,\ \mu_i^*)$ 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 $(R^*,\ \mu_i^*)$ 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 two-fold 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; secondly, 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 in SM 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 possesses 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 neural network. 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 re-mesh 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 significantly 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 utilizing 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 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 L-BFGS optimizer (*48*) to achieve similar accuracy within much less computational time (around 30 minutes; see Section S11 and Table S1 in SM for detailed results). Recently, parallel PINNs (*49*) have been proposed to accelerate the learning process of PINNs by utilizing multiple CPUs and 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 significantly 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 decays 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 utilized 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

Physics-Informed Neural Networks 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 in SM 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 $\widetilde{\mathbf{u}}(\mathbf{X};\boldsymbol{\lambda})$ and the pressure field $\widetilde{p}(\mathbf{X};\boldsymbol{\lambda})$, 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 neural network. 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.

Secondly, 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 $\widetilde{\mathbf{F}}(\mathbf{X};\boldsymbol{\lambda})$ and the first Piola-Kirchhoff stress $\widetilde{\mathbf{P}}(\mathbf{X};\boldsymbol{\lambda},\mu)$ are calculated by:

$$\widetilde{\mathbf{F}}(\mathbf{X};\boldsymbol{\lambda}) = \mathbf{I} + \frac{\partial \widetilde{\mathbf{u}}}{\partial \mathbf{X}}(\mathbf{X};\boldsymbol{\lambda}), \tag{A1}$$

$$\widetilde{\mathbf{P}}(\mathbf{X};\boldsymbol{\lambda},\mu) = -\widetilde{p}(\mathbf{X};\boldsymbol{\lambda})\widetilde{\mathbf{F}}^{-\text{T}}(\mathbf{X};\boldsymbol{\lambda}) + \mu\widetilde{\mathbf{F}}(\mathbf{X};\boldsymbol{\lambda}), \tag{A2}$$

where $\mathbf{I}$ is the identity tensor, Eq. A1 is kinematics, and Eq. A2 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:

$$\widetilde{\mathbf{r}}_{\text{PDE}}(\mathbf{X};\boldsymbol{\lambda},\mu) = \text{Div}\,\widetilde{\mathbf{P}}(\mathbf{X};\boldsymbol{\lambda},\mu), \qquad \mathbf{X} \in \Omega(\boldsymbol{\theta}_{\text{geo}}), \tag{A3}$$

$$\widetilde{r}_{\text{inc}}(\mathbf{X};\boldsymbol{\lambda}) = \det(\widetilde{\mathbf{F}}(\mathbf{X};\boldsymbol{\lambda})) - 1, \qquad \mathbf{X} \in \Omega(\boldsymbol{\theta}_{\text{geo}}). \tag{A4}$$

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

$$\widetilde{\mathbf{r}}_{\mathrm{D}}(\mathbf{X};\boldsymbol{\lambda}) = \widetilde{\mathbf{u}}(\mathbf{X};\boldsymbol{\lambda}) - \overline{\mathbf{u}}(\mathbf{X}), \qquad \mathbf{X} \in \partial\Omega_{\mathrm{D}}(\boldsymbol{\theta}_{\text{geo}}), \tag{A5}$$

$$\widetilde{\mathbf{r}}_{\mathrm{N}}(\mathbf{X};\boldsymbol{\lambda},\mu) = \widetilde{\mathbf{P}}(\mathbf{X};\boldsymbol{\lambda},\mu)\mathbf{N}(\mathbf{X}) - \overline{\mathbf{T}}(\mathbf{X}), \qquad \mathbf{X} \in \partial\Omega_{\mathrm{N}}(\boldsymbol{\theta}_{\text{geo}}), \tag{A6}$$

where $\mathbf{N}$ is the outward unit normal vector on the boundary, and $\overline{\mathbf{u}}$ and $\overline{\mathbf{T}}$ are the specified displacement and traction on the boundary, respectively. $\partial\Omega_{\mathrm{D}}(\boldsymbol{\theta}_{\text{geo}})$ and $\partial\Omega_{\mathrm{N}}(\boldsymbol{\theta}_{\text{geo}})$ refer to the domains for Dirichlet/displacement and Neumann/traction boundary conditions, 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 i-th displacement observation is:

$$\widetilde{\mathbf{r}}_u^{(i)}(\boldsymbol{\lambda}) = \widetilde{\mathbf{u}}(\mathbf{X}_u^{(i)};\boldsymbol{\lambda}) - \mathbf{u}^{*(i)}. \tag{A7}$$

Thirdly, 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 $\partial\Omega_{\mathrm{D}}$ and $\partial\Omega_{\mathrm{N}}$, denoted as $\mathbf{X}_\Omega^{(i)}$ ($i \in \{1, 2, ..., N_\Omega\}$), $\mathbf{X}_{\mathrm{D}}^{(i)}$ ($i \in \{1, 2, ..., N_{\mathrm{D}}\}$), and $\mathbf{X}_{\mathrm{N}}^{(i)}$ ($i \in \{1, 2, ..., N_{\mathrm{N}}\}$), respectively. Since 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 boundary conditions, and data, respectively. Each loss term is defined by

$$\mathcal{L}_{\text{PDE}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} \left| \widetilde{\mathbf{r}}_{\text{PDE}} \left(\mathbf{X}_\Omega^{(i)}(\boldsymbol{\theta}_{\text{geo}}); \boldsymbol{\lambda}, \mu \right) \right|^2 \tag{A8}$$

$$\mathcal{L}_{\text{inc}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} \left| \widetilde{r}_{\text{inc}} \left(\mathbf{X}_\Omega^{(i)}(\boldsymbol{\theta}_{\text{geo}}); \boldsymbol{\lambda} \right) \right|^2 \tag{A9}$$

$$\mathcal{L}_{\text{D}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \frac{1}{N_{\text{D}}} \sum_{i=1}^{N_{\text{D}}} \left| \widetilde{\mathbf{r}}_{\text{D}} \left(\mathbf{X}_{\text{D}}^{(i)}(\boldsymbol{\theta}_{\text{geo}}); \boldsymbol{\lambda} \right) \right|^2 \tag{A10}$$

$$\mathcal{L}_{\text{N}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \frac{1}{N_{\text{N}}} \sum_{i=1}^{N_{\text{N}}} \left| \widetilde{\mathbf{r}}_{\text{N}} \left(\mathbf{X}_{\text{N}}^{(i)}(\boldsymbol{\theta}_{\text{geo}}); \boldsymbol{\lambda}, \mu \right) \right|^2 \tag{A11}$$

$$\mathcal{L}_u(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \frac{1}{N_u} \sum_{i=1}^{N_u} \left| \widetilde{\mathbf{r}}_u^{(i)}(\boldsymbol{\lambda}) \right|^2, \tag{A12}$$

and the loss function is:

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \alpha_{\text{PDE}} \mathcal{L}_{\text{PDE}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) + \alpha_{\text{inc}} \mathcal{L}_{\text{inc}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) + \alpha_{\text{D}} \mathcal{L}_{\text{D}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) \\ + \alpha_{\text{N}} \mathcal{L}_{\text{N}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) + \alpha_u \mathcal{L}_u(\boldsymbol{\lambda}, \boldsymbol{\theta}), \tag{A13}$$

where $\alpha_{\text{PDE}}, \alpha_{\text{inc}}, \alpha_{\text{D}}, \alpha_{\text{N}}, \alpha_u$ 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 boundary conditions are simplified into $\mathcal{L}_{\text{BC}}$ in Eq. 1 in Results.

Lastly, 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:

$$\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\theta}}_{\text{unk}} = \underset{\boldsymbol{\lambda}, \boldsymbol{\theta}_{\text{unk}} \subseteq \boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\theta}). \tag{A14}$$

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 towards their respective target values. In this way, the PINN is able to solve inverse problems.

In Section S2 in SM, 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.

Pre-training Procedure

We find it necessary to pre-train the model before using the model to characterize unknown geometry. If we directly apply the model without pre-training, 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 pre-training 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 pre-training process, we initiates the parameter estimation process by making both $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ trainable. Such a pre-training procedure induces $\boldsymbol{\lambda}$ to converge towards 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 pre-trained until there emerges a qualitative pattern indicating the existence of a stress concentration around the void or the soft inclusion.

References

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. *Advances in neural information processing systems* **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 Processing Magazine* **29** (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. *Proceedings of the National Academy of Sciences* **116**, 4117–4122 (2019).

6. Z. Shi, M. Dao, E. Tsymbalov, A. Shapeev, J. Li, S. Suresh, Metallization of diamond. *Proceedings of the National Academy of Sciences* **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. *Proceedings of the National Academy of Sciences* **117**, 7052–7062 (2020).

8. Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering* **32**, 361–378 (2017).

9. H. Adeli, Neural networks in civil engineering: 1989–2000. *Computer-Aided Civil and Infrastructure Engineering* **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 preprint arXiv:2108.11985* (2021).

11. H. Jin, Big-data-driven multi-scale experimental study of nanostructured block copolymer's dynamic toughness, Ph.D. thesis, Brown University (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. *Journal of Computational Physics* **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. *Proceedings of the National Academy of Sciences* **118** (2021).

15. S. Cai, Z. Mao, Z. Wang, M. Yin, G. E. Karniadakis, Physics-informed neural networks (pinns) for fluid mechanics: A review. *arXiv preprint arXiv:2105.09506* (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. *Computer Methods in Applied Mechanics and Engineering* **362**, 112790 (2020).

17. C. Rao, H. Sun, Y. Liu, Physics-informed deep learning for computational elastodynamics without labeled data. *Journal of Engineering Mechanics* **147**, 04021043 (2021).

18. E. Zhang, M. Yin, G. E. Karniadakis, Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging. *arXiv preprint arXiv:2009.04525* (2020).

19. J. N. Fuhg, N. Bouklas, The mixed deep energy method for resolving concentration features in finite strain hyperelasticity. *arXiv preprint arXiv:2104.09623* (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. *Journal of Nondestructive Evaluation* **39**, 1–20 (2020).

21. M. Yin, X. Zheng, J. D. Humphrey, G. E. Karniadakis, Non-invasive inference of thrombus material properties with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering* **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 preprint arXiv:2108.06905* (2021).

23. L. Yang, D. Zhang, G. E. Karniadakis, Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing* **42**, A292–A317 (2020).

24. G. Pang, L. Lu, G. E. Karniadakis, fPINNs: fractional physics-informed neural networks. *SIAM Journal on Scientific Computing* **41**, A2603–A2626 (2019).

25. T. L. Anderson, *Fracture mechanics: fundamentals and applications* (CRC Press, 2017).

26. S. Suresh, *Fatigue of materials* (Cambridge University Press, 1998).

27. L. B. Freund, S. Suresh, *Thin film materials: stress, defect formation and surface evolution* (Cambridge University Press, 2004).

28. H. B. Ameur, M. Burger, B. Hackl, Level set methods for geometric inverse problems in linear elasticity. *Inverse Problems* **20**, 673 (2004).

29. H. Sun, H. Waisman, R. Betti, Nondestructive identification of multiple flaws using xfem and a topologically adapting artificial bee colony algorithm. *International Journal for Numerical Methods in Engineering* **95**, 871–900 (2013).

30. R. Gallego, G. Rus, Identification of cracks and cavities using the topological sensitivity boundary integral equation. *Computational Mechanics* **33**, 154–163 (2004).

31. J. Jung, E. Taciroglu, Modeling and identification of an arbitrarily shaped scatterer using dynamic xfem with cubic splines. *Computer Methods in Applied Mechanics and Engineering* **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. *International Journal for Numerical Methods in Engineering* **82**, 303–328 (2010).

33. D. Schnur, N. Zabaras, An inverse method for determining elastic material properties and a material interface. *International Journal for Numerical Methods in Engineering* **33**, 2039–2057 (1992).

34. Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation* **4**, 1–32 (1996).

35. S. Chaabane, M. Masmoudi, H. Meftahi, Topological and shape gradient strategy for solving geometrical inverse problems. *Journal of Mathematical Analysis and Applications* **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. *International Journal of Solids and Structures* **83**, 73–80 (2016).

37. S. Amstutz, I. Horchani, M. Masmoudi, Crack detection by the topological gradient method. *Control and Cybernetics* **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. *Computer methods in applied mechanics and engineering* **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. *Computer Methods in Applied Mechanics and Engineering* **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. *Optics 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 systems (2015).

44. S. Wang, P. Perdikaris, Deep learning of free boundary and stefan problems. *Journal of Computational Physics* **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 preprint arXiv:2012.07938* (2020).

46. Dassault Systèmes, *Abaqus 2020 documentation* (2020).

47. D. P. Kingma, J. Ba, Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

48. D. C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization. *Mathematical programming* **45**, 503–528 (1989).

49. K. Shukla, A. D. Jagtap, G. E. Karniadakis, Parallel physics-informed neural networks via domain decomposition. *arXiv preprint arXiv:2104.10013* (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 preprint arXiv:2004.01806* (2020).

51. S. Wang, X. Yu, P. Perdikaris, When and why pinns fail to train: A neural tangent kernel perspective. *arXiv preprint arXiv:2007.14527* (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. *Computer Methods in Applied Mechanics and Engineering* **384**, 113938 (2021).

53. M. M. Doyley, Model-based elastography: a survey of approaches to the inverse elasticity problem. *Physics in Medicine & Biology* **57**, R35 (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 International* **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 Physics & Technology* **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. *Journal of Nondestructive Evaluation* **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 Journal* **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. *Optics Letters* **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 and Wireless Propagation Letters* **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 preprint arXiv:2102.04626* (2021).

61. J. Céa, S. Garreau, P. Guillaume, M. Masmoudi, The shape and topological optimizations connection. *Computer Methods in Applied Mechanics and Engineering* **188**, 713–726 (2000).

62. S. Goenezen, P. Barbone, A. A. Oberai, Solution of the nonlinear elasticity imaging inverse problem: the incompressible case. *Computer Methods in Applied Mechanics and Engineering* **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. *Journal of Computational Physics* **404**, 109136 (2020).

64. X. Glorot, Y. Bengio, *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (2010), pp. 249–256.

Acknowledgements

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 co-inventors on a provisional patent related to this work to be filed at the United States Patent and Trademark Office. The authors declare 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. Additional data available from authors upon request.

Figures and Tables

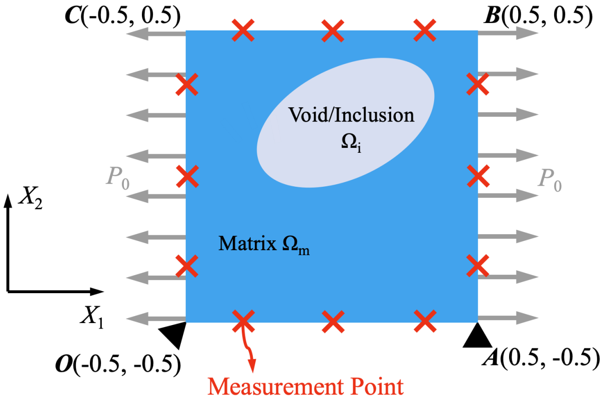


Fig. 1. General setup of the prototypical problem on geometry and material identification in this study. We consider a plane-strain problem in the $X_1 - X_2$ plane about a square-shaped matrix specimen Ω_{m} with a void/inclusion Ω_{i}. Displacements are measured on the outer boundary of the matrix 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.

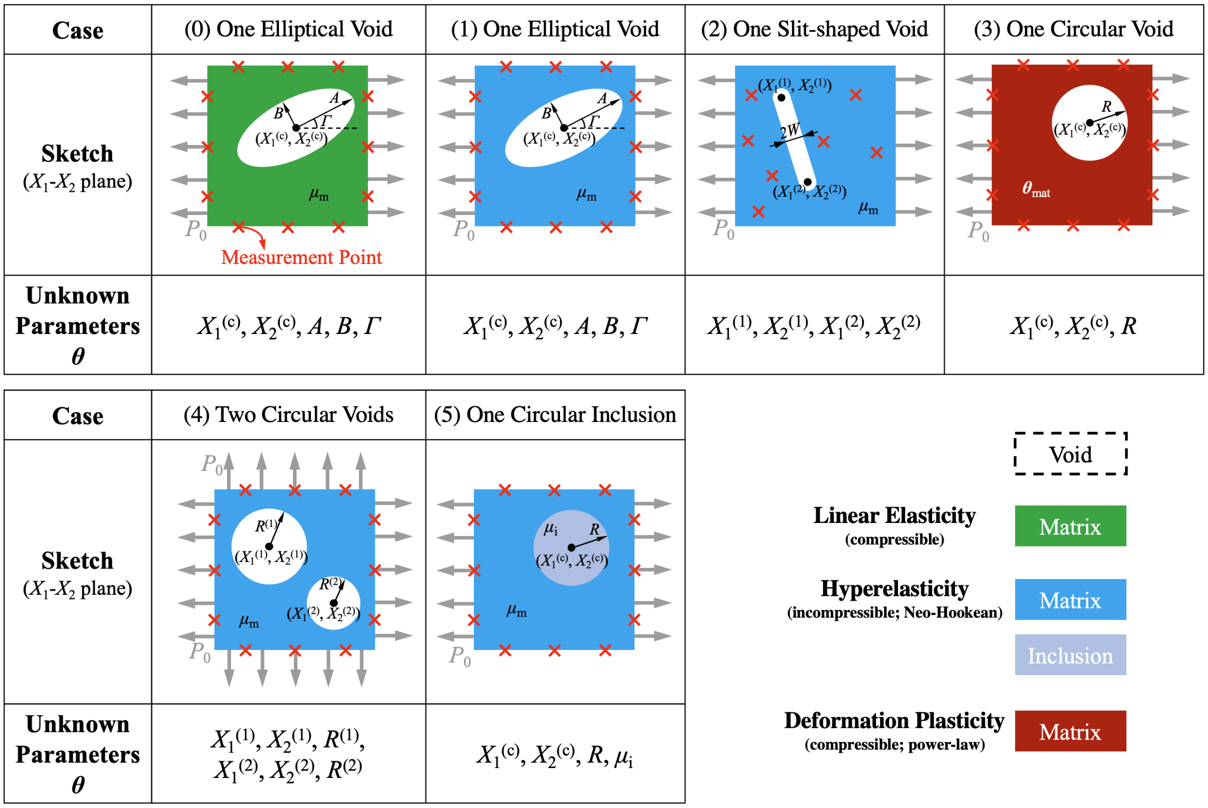
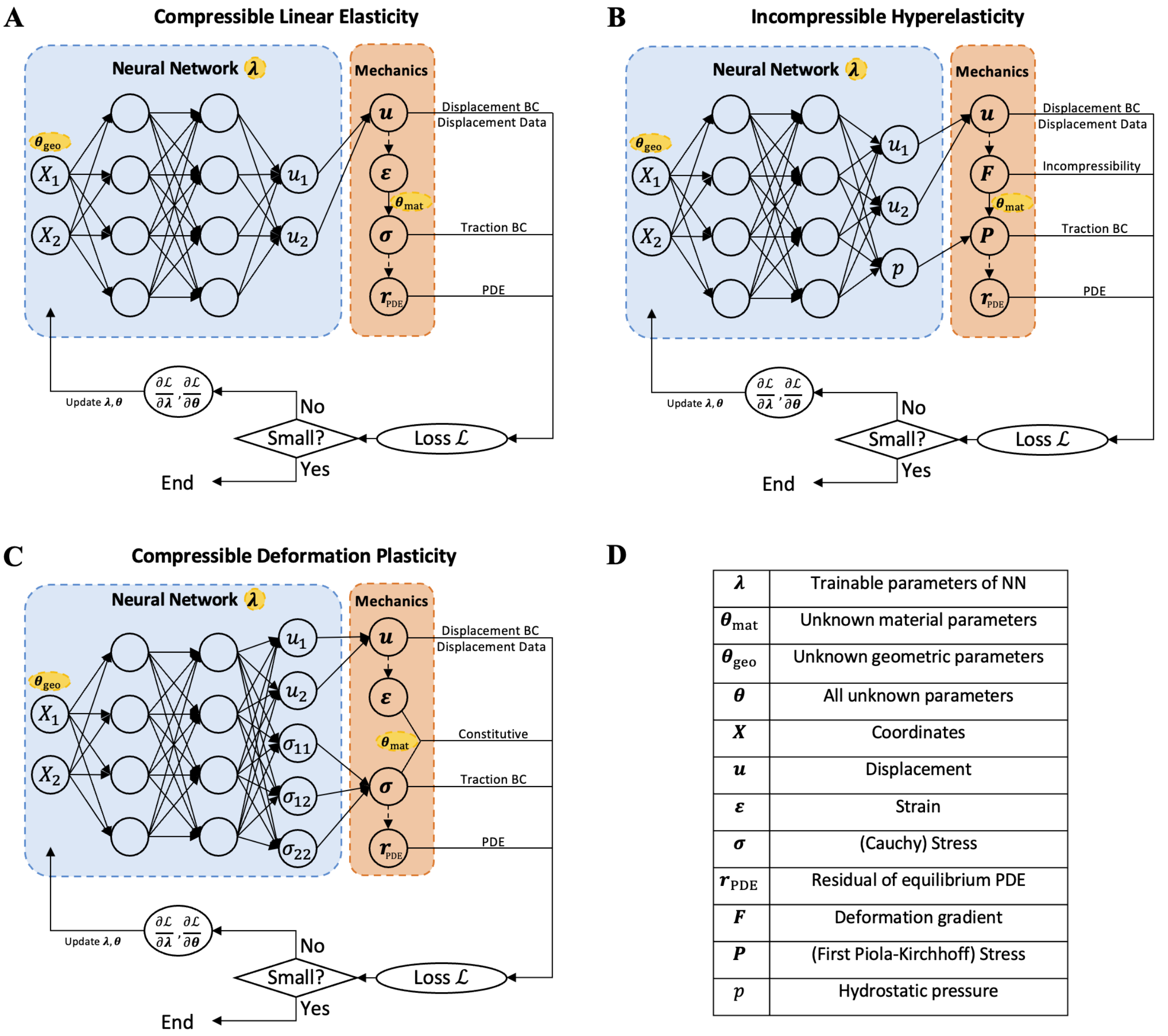


Fig. 2. Setup of cases 0-5 of the prototypical problem. All the illustrations and geometric parameters are given in the reference (undeformed) configuration. The target of each case is to estimate the unknown parameters $\boldsymbol{\theta}$ given the displacement data on the measurement points. All 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 SM.



$\boldsymbol{\lambda}$	Trainable parameters of NN
$\boldsymbol{\theta}_{\text{mat}}$	Unknown material parameters
$\boldsymbol{\theta}_{\text{geo}}$	Unknown geometric parameters
$\boldsymbol{\theta}$	All unknown parameters
$\boldsymbol{X}$	Coordinates
$\boldsymbol{u}$	Displacement
$\boldsymbol{\varepsilon}$	Strain
$\boldsymbol{\sigma}$	(Cauchy) Stress
$\boldsymbol{r}_{\text{PDE}}$	Residual of equilibrium PDE
$\boldsymbol{F}$	Deformation gradient
$\boldsymbol{P}$	(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**-**C**) We apply neural networks with trainable parameters $\boldsymbol{\lambda}$ 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 represents 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 due to the variable computational domain. Finally, parameter estimation is conducted through the minimization of loss function. In this process, $\boldsymbol{\lambda}$ 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**)-(**C**).

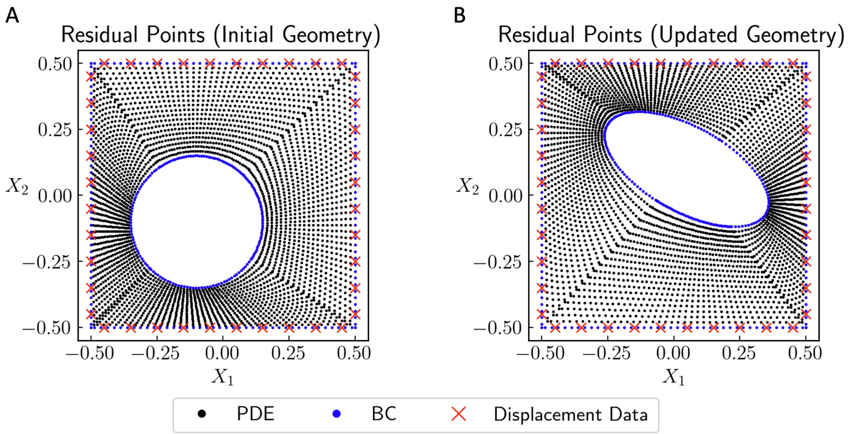
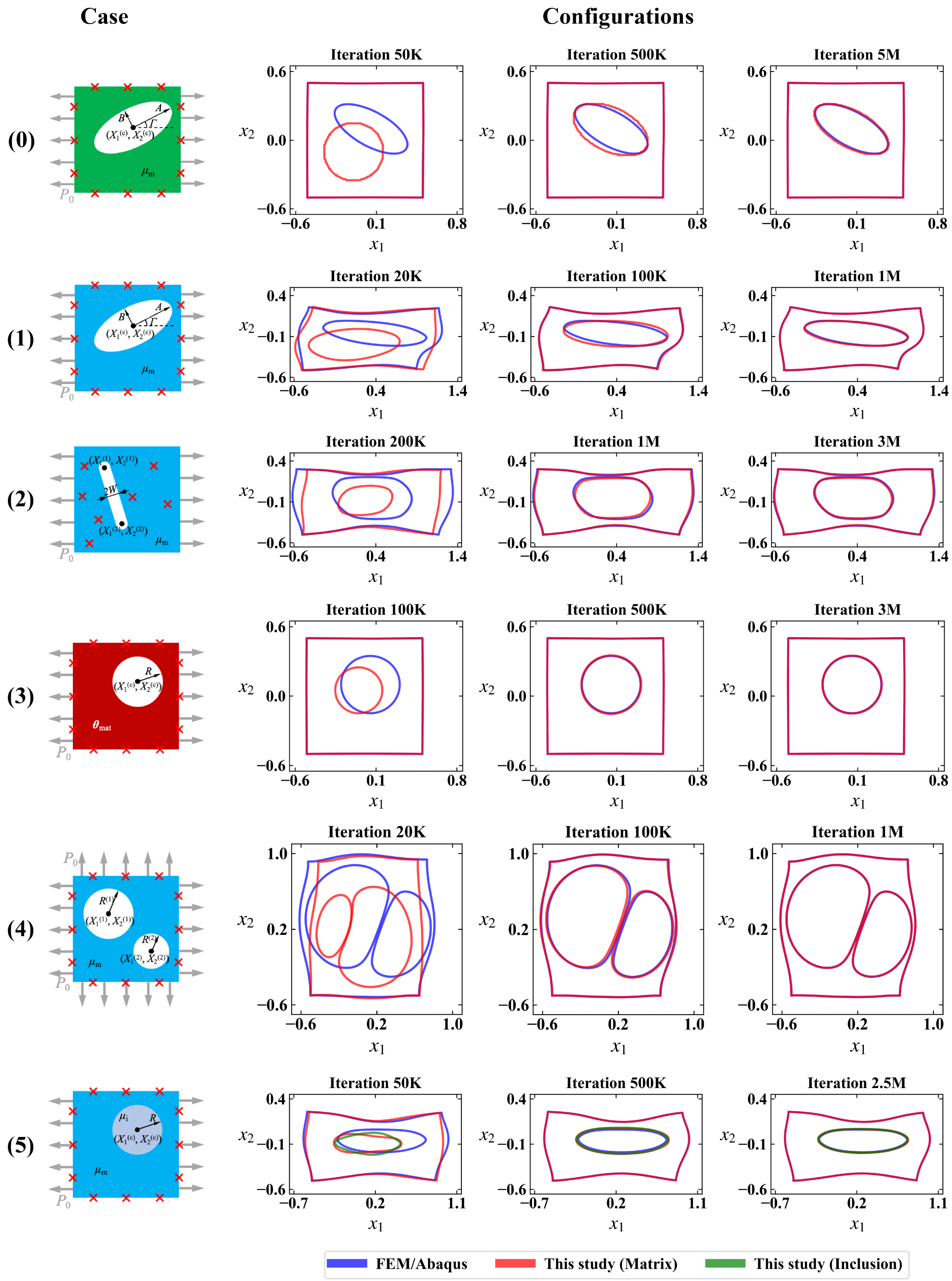


Fig. 4. Residual points for the initial geometry and updated geometry in Case 0. Different residual terms (PDEs, BCs, and data) require different residual points. We propose the geometry-parameterized residual points, so that the locations of the residual points automatically change as the geometric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



Iteration 50K | Iteration 500K | Iteration 5M

Iteration 20K | Iteration 100K | Iteration 1M

Iteration 200K | Iteration 1M | Iteration 3M

Iteration 100K | Iteration 500K | Iteration 3M

Iteration 20K | Iteration 100K | Iteration 1M

Iteration 50K | Iteration 500K | Iteration 2.5M

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 the left to right) correspond to the completion of pre-training (beginning of parameter estimation), amid the training, and the completion of training, respectively ($\mathrm{K} = 10^3$, $\mathrm{M} = 10^6$).

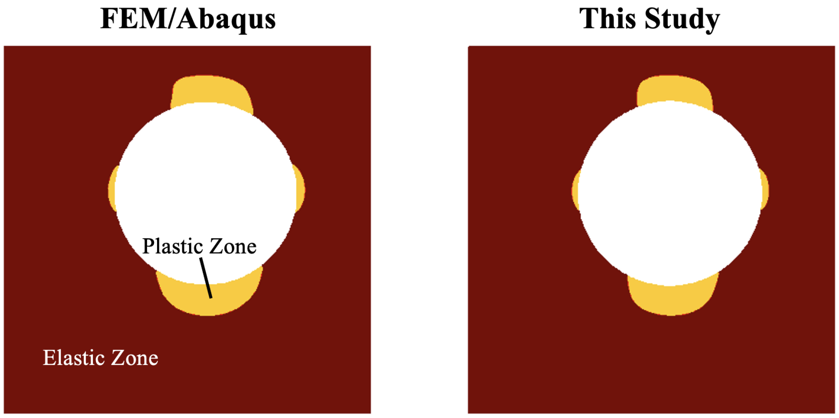


Fig. 6. Inference of the plastic zone compared with FEM ground truth in case 3 shown in Fig. 2. We mark the plastic zone by yellow and the void by white.

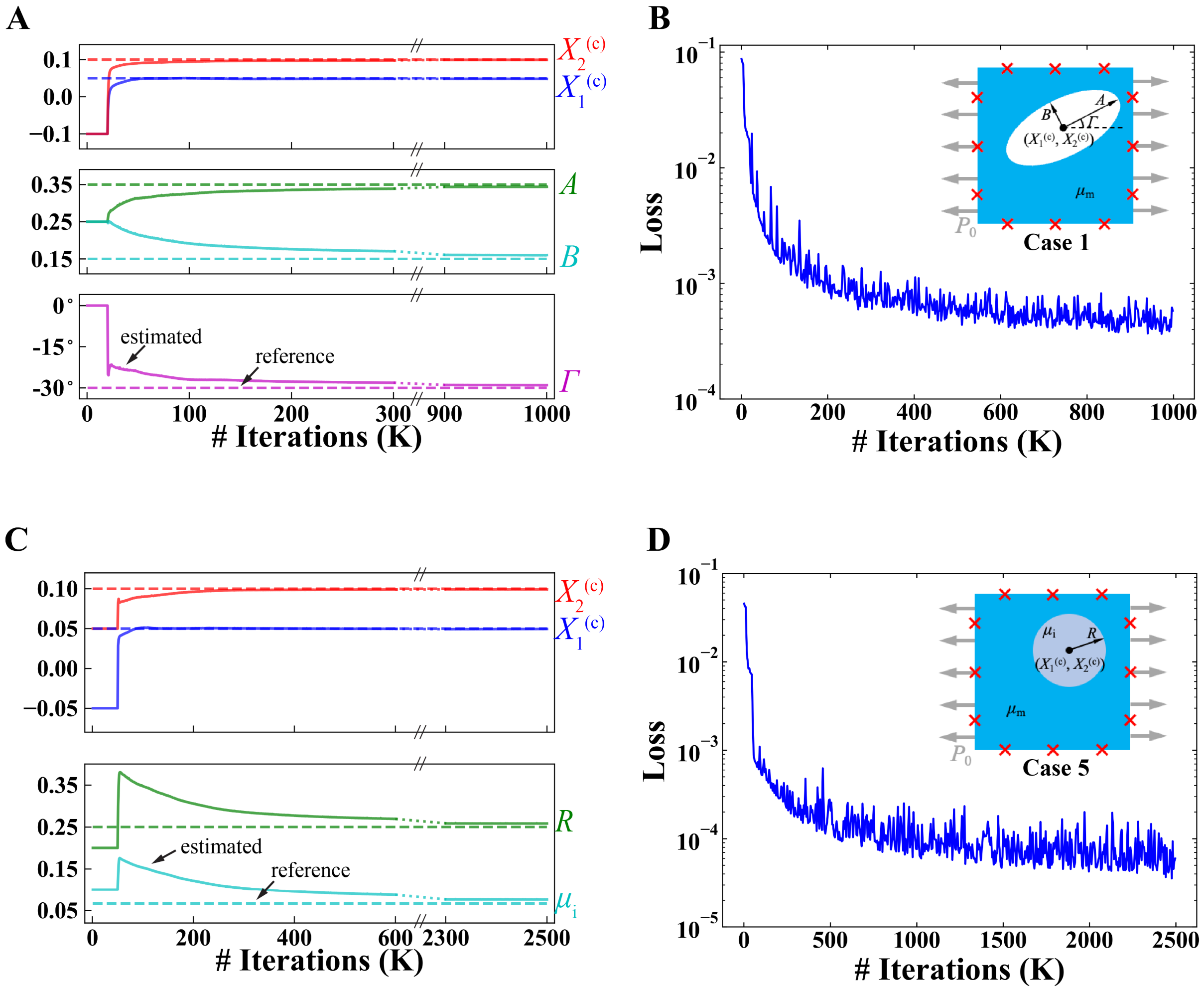


Fig. 7. Evolution of estimated unknown parameters and loss function in case 1 and case 5 during the training process. Case 1 (A and B) involves geometry identification, while case 5 (C and D) involves both material and geometry identification. See Fig. 2 for the definitions of the cases. (**A** and **C**) 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. S2) in SM.

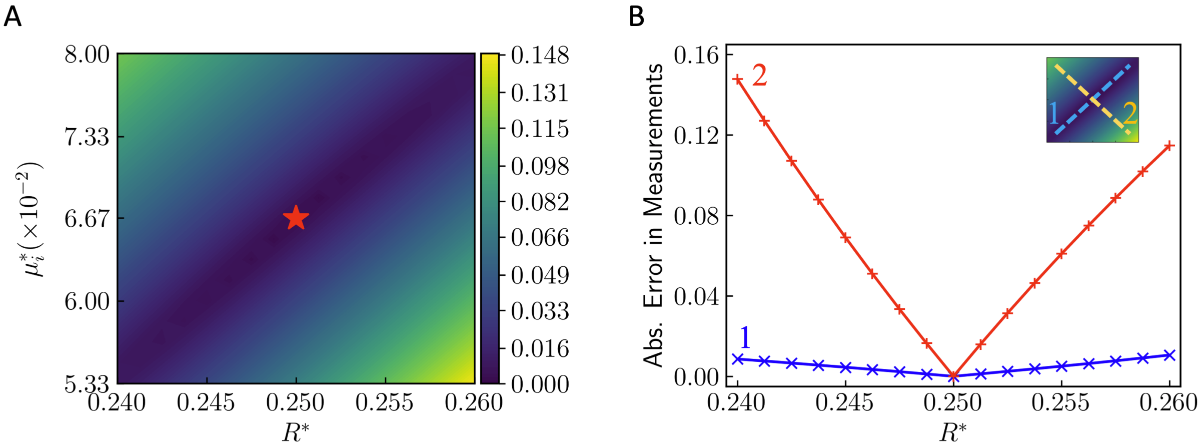


Fig. 8. Perturbation analysis of unknown parameters in case 5. Using the finite element solver, we calculate the root-mean-squared error of the displacement data on the measurement points caused by the perturbation $(\Delta R^*, \Delta\mu_i^*)$ on the reference values of the shear modulus (μ_{i}^*) and 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 $\Delta R^*\Delta\mu_i^* < 0$.

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 ($\times 10^{-2}$, 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 ($\times 10^{-2}$, 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($\times 10^{-2}$)	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($\times 10^{-2}$)	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($\times 10^{-2}$)	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 ($\times 10^{-2}$)	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 ($\times 10^{-2}$)	0.05	0.02	0.24	0.20
Relative Error (%)	0.05	0.02	0.96	3.0

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 re-train the PINN, which is shown in the table as “Case 5 (With Internal Data)”.

Supplementary Materials for

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

Enrui Zhang, Ming Dao*, George Em Karniadakis*, Subra Suresh*

*Corresponding author. Email: mingdao@mit.edu (M.D.); george_karniadakis@brown.edu (G.E.K.); ssuresh@ntu.edu.sg (S.S.)

This PDF file includes:

Sections S1 to S11
Figs. S1 to S9
Table S1

Section S1. A Brief Review on the Inverse Algorithms for Solid Mechanics

To determine how to update the estimations of unknown parameters, the most widely-used family of methods is based on the calculation of gradients, where the differentiable loss function provides guidance on the increments of estimations. For geometry identification problems, however, the geometry is not naturally differentiable. To address this issue, the shape/topological gradient is proposed by some researchers (*30, 35, 37, 61*), with some studies combined with the level set method (*28*). With the gradient with respect to geometry calculated, the remaining issue is to apply an optimization method to minimize the loss function. Examples of commonly applied optimization algorithms include the L-BFGS algorithm (*36,48,62*), a quasi-Newton iterative method for nonlinear optimization, and the Levenberg-Marquardt method (*33*), a nonlinear least square fitting algorithm. In some other studies, non-gradient methods such as evolutionary algorithms (*32, 34*), have also been applied for updating the geometry.

Section S2. Additional Information on the Formulation of PINNs

Mechanics of Incompressible Hyperelastic Materials. Here we summarize the mechanics of incompressible hyperelastic materials. The reference (undeformed) and current (deformed) configurations of the solid are described by the vectors $\mathbf{X}$ and $\mathbf{x}$, respectively. Then, we may define the displacement vector $\mathbf{u}(\mathbf{X}) = \mathbf{x}(\mathbf{X}) - \mathbf{X}$, the deformation gradient tensor $\mathbf{F}(\mathbf{X}) = \partial\mathbf{x}/\partial\mathbf{X} = \mathbf{I} + \partial\mathbf{u}/\partial\mathbf{X}$ (with the F_{iJ} component being $\partial x_i/\partial X_J$, $i \in \{1, 2, 3\}$, $J \in \{1, 2, 3\}$; $\mathbf{I}$ is the identity tensor), and the right Cauchy-Green tensor $\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$.

For isotropic, incompressible materials, the three invariants of $\mathbf{C}$ are $I_1 = \mathrm{tr}(\mathbf{C})$, $I_2 = [\mathrm{tr}(\mathbf{C})^2 - \mathrm{tr}(\mathbf{C}^2)]/2$, and $I_3 = \det(\mathbf{C})(= 1$ because of incompressibility). The strain energy density can be expressed as $W(I_1, I_2; \boldsymbol{\theta}_{\mathrm{mat}})$. Specifically, for incompressible Neo-Hookean materials, it takes the form $W(I_1; \mu) = \mu(I_1 - 3)/2$, where the shear modulus μ is the only material parameter ($\boldsymbol{\theta}_{\mathrm{mat}} = \mu$).

The first Piola-Kirchhoff (PK) stress for incompressible materials is expressed as

$$\mathbf{P} = -p\mathbf{F}^{-\mathrm{T}} + 2\mathbf{F}\frac{\partial W}{\partial \mathbf{C}}, \tag{S1}$$

where p is the hydraulic pressure serving as the Lagrange multiplier for incompressible materials to uniquely determine the stress field. For incompressible Neo-Hookean materials,

$$\mathbf{P} = -p\mathbf{F}^{-\mathrm{T}} + \mu\mathbf{F}. \tag{S2}$$

Given the first PK stress tensor $\mathbf{P}$, the equilibrium equation without body force is

$$\mathrm{Div}\ \mathbf{P} = 0, \tag{S3}$$

where the corresponding component form is $\partial P_{iJ}/\partial X_J = 0$ ($i \in \{1, 2, 3\}$; the Einstein summation convention is applied for J).

Now we define the computational domain and boundary conditions. In our current work, we parameterize the domain by $\boldsymbol{\theta}_{\mathrm{geo}}$. Hence, the domain of the PDEs follows as:

$$\mathbf{X} \in \Omega(\boldsymbol{\theta}_{\mathrm{geo}}) \subset \mathbb{R}^d, \tag{S4}$$

where Ω is the parameterized domain of the reference configuration of the problem, and d is the dimension of the problem ($d = 2$ in our cases). The displacement/Dirichlet and traction/Neumann boundary conditions can be expressed as:

$$\mathbf{u} = \overline{\mathbf{u}}, \mathbf{X} \in \partial\Omega_{\mathrm{D}}(\boldsymbol{\theta}_{\mathrm{geo}}), \tag{S5}$$

$$\mathbf{P}\mathbf{N} = \overline{\mathbf{T}}, \mathbf{X} \in \partial\Omega_{\mathrm{N}}(\boldsymbol{\theta}_{\mathrm{geo}}), \tag{S6}$$

where $\mathbf{N}$ is the outward unit normal vector in the reference configuration, and $\partial\Omega_{\mathrm{D}}(\boldsymbol{\theta}_{\mathrm{geo}})$ and $\partial\Omega_{\mathrm{N}}(\boldsymbol{\theta}_{\mathrm{geo}})$ are the displacement and traction boundaries, respectively. The right-hand-side terms $\overline{\mathbf{u}}$ and $\overline{\mathbf{T}}$ are the specified values of displacement and traction at the boundary, respectively. The component form corresponding to Eq. S6 is $P_{iJ}N_J = \overline{T}_i$.

In the case where there is more than one material, the continuity of displacement and traction along the interface $\Gamma_{\text{int}}(\boldsymbol{\theta}_{\text{geo}})$ (between material 1 and 2) must also be satisfied:

$$\mathbf{u}^{(1)} = \mathbf{u}^{(2)}, \mathbf{X} \in \Gamma_{\text{int}}(\boldsymbol{\theta}_{\text{geo}}), \tag{S7}$$

$$\mathbf{P}^{(1)}\mathbf{N} = \mathbf{P}^{(2)}\mathbf{N}, \mathbf{X} \in \Gamma_{\text{int}}(\boldsymbol{\theta}_{\text{geo}}), \tag{S8}$$

where the superscripts refer to the materials 1 and 2, respectively, and $\mathbf{N}$ is the unit normal vector of the interface Γ_{int} in the reference configuration.

PINNs for Linear Elasticity and Deformation Plasticity We summarize the differences in the architecture of PINNs for compressible linear elasticity (Fig. 2A) and deformation plasticity (Fig. 2C) compared to incompressible hyperelasticity (Fig. 2B). For these two constitutive relations, the solid undergoes infinitesimal deformation, so that one does not need to distinguish reference (undeformed) configuration (X_1, X_2) and deformed configuration (x_1, x_2). In this section, the coordinates are written as (x_1, x_2) to keep consistent with the conventional notations in solid mechanics community. Due to the compressibility, the hydrostatic pressure p is no longer an independent state variable, so that p is not needed as a primary output of the NN.

To build the PINN for linear elasticity, kinematics in Eq. 4 in the main text is replaced by the definition of the infinitesimal strain tensor $\varepsilon_{ij} = \partial u_i/\partial x_j$, where $\mathbf{u} = (u_1, u_2)$ is the displacement. Stress-deformation relationship in Eq. 5 is replaced by the stress-strain relationship

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij}, \tag{S9}$$

where λ and μ are Lamé constants. The equilibrium equation similar to Eq. 6 is $\partial \sigma_{ij}/\partial x_j = 0$. Note that the loss function for linear elasticity does not require the term for incompressibility (see Eq. 1 and Eq. 16).

The PINN for the power-law deformation plasticity can be constructed based on the PINN for linear elasticity. In this case, the nonlinear stress-strain relation is expressed as

$$E\boldsymbol{\varepsilon} = \boldsymbol{\sigma} + \frac{3}{2}\alpha\left(\frac{\sigma_{\mathrm{e}}}{\sigma_{\mathrm{Y}}}\right)^{n-1}\boldsymbol{s}, \tag{S10}$$

where E (Young's modulus), α, σ_{Y} (yield stress), and n (hardening exponent) are material parameters. $\boldsymbol{s}$ is the deviatoric stress tensor ($s_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}$), and σ_{e} ($= \sqrt{\frac{3}{2}s_{ij}s_{ij}}$) is the von Mises stress.

According to Eq. S10, stress cannot be explicitly expressed by strain for power-law deformation plasticity, unlike linear elasticity (Fig. 2A) and hyperelasticity (Fig. 2B). Therefore, the integration of constitutive relation through the analytical expression of stress tensor no longer works for deformation plasticity. As an alternative approach, we include the stress field $\boldsymbol{\sigma} = (\sigma_{11}, \sigma_{12}, \sigma_{22})$ as primary solution field of the neural network in addition to the displacement field (u_1, u_2) (see Fig. 2C). After calculating the strain field, we bridge the strain and stress fields with an additional loss term according to Eq. S10. In this way, we integrate constitutive relation as an additional penalty term in the loss function for deformation plasticity.

Multiple materials. In the case of inverse problems for multiple materials as in case 5 in the main paper, we apply two independent NNs to approximate the displacement and pressure fields of the matrix and inclusion materials, respectively. Such approximation can be expressed by

$$\text{Matrix: } (\widetilde{\mathbf{u}}(\mathbf{X};\boldsymbol{\lambda}_{\mathrm{m}}), \widetilde{p}(\mathbf{X};\boldsymbol{\lambda}_{\mathrm{m}})), \mathbf{X} \in \Omega_{\mathrm{m}}(\boldsymbol{\theta}_{\mathrm{geo}}) \tag{S11}$$

$$\text{Inclusion: } (\widetilde{\mathbf{u}}(\mathbf{X};\boldsymbol{\lambda}_{\mathrm{i}}), \widetilde{p}(\mathbf{X};\boldsymbol{\lambda}_{\mathrm{i}})), \mathbf{X} \in \Omega_{\mathrm{i}}(\boldsymbol{\theta}_{\mathrm{geo}}) \tag{S12}$$

In addition to the loss terms for each material, we need an additional loss term $\mathcal{L}_{\mathrm{int}}$ to force the continuity of displacement and traction on the interface $\Gamma_{\mathrm{int}}(\boldsymbol{\theta}_{\mathrm{geo}})$ of two materials. To do this, we first write the residuals on a single point for the two conditions of continuity. The residual of displacement continuity is

$$\widetilde{\mathbf{r}}_{\mathrm{Dint}}(\mathbf{X};\boldsymbol{\lambda}_{\mathrm{m}},\boldsymbol{\lambda}_{\mathrm{i}}) = \widetilde{\mathbf{u}}(\mathbf{X};\boldsymbol{\lambda}_{\mathrm{m}}) - \widetilde{\mathbf{u}}(\mathbf{X};\boldsymbol{\lambda}_{\mathrm{i}}), \mathbf{X} \in \Gamma_{\mathrm{int}}(\boldsymbol{\theta}_{\mathrm{geo}}), \tag{S13}$$

and the residual of stress continuity is

$$\widetilde{\mathbf{r}}_{\text{Nint}}(\mathbf{X};\boldsymbol{\lambda}_{\text{m}},\boldsymbol{\lambda}_{\text{i}},\mu_{\text{m}},\mu_{\text{i}}) = \widetilde{\mathbf{P}}(\mathbf{X};\boldsymbol{\lambda}_{\text{m}},\mu_{\text{m}})\mathbf{N}(\mathbf{X}) - \widetilde{\mathbf{P}}(\mathbf{X};\boldsymbol{\lambda}_{\text{i}},\mu_{\text{i}})\mathbf{N}(\mathbf{X}), \mathbf{X} \in \Gamma_{\text{int}}(\boldsymbol{\theta}_{\text{geo}}). \quad \text{(S14)}$$

We place residual points $\mathbf{X}_{\Gamma}^{(i)}(\boldsymbol{\theta}_{\text{geo}})$ ($i \in \{1, 2, ..., N_{\Gamma}\}$) on $\Gamma(\boldsymbol{\theta}_{\text{geo}})$, and construct the loss component on the interface of two materials by taking a weighted sum as

$$\mathcal{L}_{\text{int}}(\boldsymbol{\lambda}_{\text{m}},\boldsymbol{\lambda}_{\text{i}},\boldsymbol{\theta}) = \alpha_{\text{Dint}}\mathcal{L}_{\text{Dint}}(\boldsymbol{\lambda}_{\text{m}},\boldsymbol{\lambda}_{\text{i}},\boldsymbol{\theta}) + \alpha_{\text{Nint}}\mathcal{L}_{\text{Nint}}(\boldsymbol{\lambda}_{\text{m}},\boldsymbol{\lambda}_{\text{i}},\boldsymbol{\theta}), \quad \text{(S15)}$$

where

$$\mathcal{L}_{\text{Dint}}(\boldsymbol{\lambda}_{\text{m}},\boldsymbol{\lambda}_{\text{i}},\boldsymbol{\theta}) = \frac{1}{N_{\Gamma}}\sum_{i=1}^{N_{\Gamma}}\left|\widetilde{\mathbf{r}}_{\text{Dint}}\left(\mathbf{X}_{\Gamma}^{(i)}(\boldsymbol{\theta}_{\text{geo}});\boldsymbol{\lambda}_{\text{m}},\boldsymbol{\lambda}_{\text{i}}\right)\right|^2 \quad \text{(S16)}$$

$$\mathcal{L}_{\text{Nint}}(\boldsymbol{\lambda}_{\text{m}},\boldsymbol{\lambda}_{\text{i}},\boldsymbol{\theta}) = \frac{1}{N_{\Gamma}}\sum_{i=1}^{N_{\Gamma}}\left|\widetilde{\mathbf{r}}_{\text{Nint}}\left(\mathbf{X}_{\Gamma}^{(i)}(\boldsymbol{\theta}_{\text{geo}});\boldsymbol{\lambda}_{\text{m}},\boldsymbol{\lambda}_{\text{i}},\mu_{\text{m}},\mu_{\text{i}}\right)\right|^2. \quad \text{(S17)}$$

Hence, the loss function in the case of multiple materials is the summation of the loss function for each single material and the additional loss $\mathcal{L}_{\text{int}}$ for the material interface.

PINNs for forward problems. For forward problems with no unknown parameter ($\boldsymbol{\theta}_{\text{unk}} = \emptyset$, hence $\boldsymbol{\theta} = \emptyset$ for convenience), we do not have measurement data. For incompressible materials, for example, the loss function can be written as a weighted sum of all the four loss terms that correspond to PDEs, the incompressibility condition, displacement boundary conditions, and traction boundary conditions, respectively:

$$\mathcal{L}_{\text{for}}(\boldsymbol{\lambda}) = \alpha_{\text{PDE}}\mathcal{L}_{\text{PDE}}(\boldsymbol{\lambda}) + \alpha_{\text{inc}}\mathcal{L}_{\text{inc}}(\boldsymbol{\lambda}) + \alpha_{\text{D}}\mathcal{L}_{\text{D}}(\boldsymbol{\lambda}) + \alpha_{\text{N}}\mathcal{L}_{\text{N}}(\boldsymbol{\lambda}). \quad \text{(S18)}$$

Each loss term is defined by

$$\mathcal{L}_{\text{PDE}}(\boldsymbol{\lambda}) = \frac{1}{N_\Omega}\sum_{i=1}^{N_\Omega}\left|\widetilde{\mathbf{r}}_{\text{PDE}}\left(\mathbf{X}_\Omega^{(i)};\boldsymbol{\lambda}\right)\right|^2 \tag{S19}$$

$$\mathcal{L}_{\text{inc}}(\boldsymbol{\lambda}) = \frac{1}{N_\Omega}\sum_{i=1}^{N_\Omega}\left|\widetilde{r}_{\text{inc}}\left(\mathbf{X}_\Omega^{(i)};\boldsymbol{\lambda}\right)\right|^2 \tag{S20}$$

$$\mathcal{L}_{\text{D}}(\boldsymbol{\lambda}) = \frac{1}{N_{\text{D}}}\sum_{i=1}^{N_{\text{D}}}\left|\widetilde{\mathbf{r}}_{\text{D}}\left(\mathbf{X}_D^{(i)};\boldsymbol{\lambda}\right)\right|^2 \tag{S21}$$

$$\mathcal{L}_{\text{N}}(\boldsymbol{\lambda}) = \frac{1}{N_{\text{N}}}\sum_{i=1}^{N_{\text{N}}}\left|\widetilde{\mathbf{r}}_{\text{N}}\left(\mathbf{X}_{\text{N}}^{(i)};\boldsymbol{\lambda}\right)\right|^2. \tag{S22}$$

The solution of forward problems using PINNs can be expressed as:

$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmin}}\, \mathcal{L}_{\text{for}}(\boldsymbol{\lambda}), \tag{S23}$$

The displacement solved by the PINN is $\widetilde{\mathbf{u}}(\mathbf{X};\hat{\boldsymbol{\lambda}})$ for $\mathbf{X} \in \Omega$.

Section S3. Technical Details

Details of the Prototypical Problem. For case 0, the Young's modulus of the matrix is $E = 1.0$, the Poisson's ratio is $\nu = 0.3$, and the loading is $P_0 = 0.003$. For cases 1, 2, 4 and 5, the shear modulus of the matrix (incompressible Neo-Hookean material) is $\mu_{\text{i}} = 0.333$, and the external load is $P_0 = 0.3$. For case 3, the external load is $P_0 = 0.002$, and the material parameters include $E = 1.0$, $\alpha = 0.1$, $n = 10$ and $\sigma_{\text{Y}} = 0.005$. For cases 0, 1, 3, 4, 5, there are $N_u = 10$ measurement points of on each edge of the matrix. For case 2, measurement points are inside the solid, with 10 of them uniformly placed on the line $X_1 = -0.15$, $X_1 = 0.15$, $X_2 = -0.45$, $X_2 = 0.45$, respectively. For the modified case 5 with additional measurements, the five additional measurement points are located at $(-0.3, 0.05)$, $(-0.3, 0.10)$, $(-0.3, 0.15)$ in the matrix and $(-0.05, 0.10)$, $(0.15, 0.10)$ in the inclusion.

Details of the PINN. We use TensorFlow 1.14 (*43*) to build up our PINN. Each NN in the PINN has 4 hidden layers, each with 30 neurons. For cases 0, 1, 2 and 4, there is one single

NN in the PINN. There are two NNs in the PINN in case 3 (one NN for (u_1, u_2), the other NN for $(\sigma_{11}, \sigma_{22}, \sigma_{12})$) and case 5 (one NN for each material). We adopt the layer-wise adaptive "tanh" function (*63*) as the activation function for the NNs. Among the trainable parameters of the NN, weights are initialized with Xavier initialization (*64*), biases are initialized as zeros, and the variable of adaptive activation are initialized as ones. We use the Adam optimizer (*47*) to train the network. The learning rate is 0.001 for all cases except cases 2 and 3.

For case 2, the unknown geometric parameters directly defined in the code are the location of the center of the slit $(X_1^{(\mathrm{c})}, X_2^{(\mathrm{c})})$, half length of the slit L, and tilting angle of the slit Γ, which are post-processed to be the coordinates of the locations of the centers of the slit tips ($X_1^{(1)} = X_1^{(\mathrm{c})} - L\sin\Gamma$, $X_2^{(1)} = X_2^{(\mathrm{c})} + L\cos\Gamma$, $X_1^{(2)} = X_1^{(\mathrm{c})} + L\sin\Gamma$, $X_2^{(2)} = X_2^{(\mathrm{c})} - L\cos\Gamma$). The learning rate is 0.001 for Γ and 0.0002 for $(X_1^{(\mathrm{c})}, X_2^{(\mathrm{c})}, L)$. For case 3, the learning rate is 0.00005.

For cases 0, 1, 3 and 5, the assignments of residual points are similar. We assign 3200 internal points for the PDE and incompressibility (800 points for each of the four sub-regions; 40 along the circumferential direction and 20 along the radial direction). For case 5 only, 400 residual points are assigned in the inclusion material. We place 160 residual points on the outer boundary for enforcing traction boundary conditions. Another 160 residual points are placed on the inner boundary of the matrix, which enforce the traction-free boundary conditions for cases 1 and 3 and the interface conditions for case 5. The points of displacement measurements are uniformly placed on the outer boundary of the matrix.

For case 2, the smallest length scale is determined by the W, which poses a limitation for the interval of residual points. To accurately resolve the displacement field, especially around the slit tip where the displacement changes drastically and the stress concentration exists, we need a large density of residual points in the computational domain. For this case, we have 49000 internal points, 700 points on the outer boundary, and 700 points on the inner boundary.

For case 4, we divide the square region into two parts, each with one void. For each part, residual points are assigned in a similar way to case 1.

For all the cases, we choose the weights of the loss terms in Eq. 16 to be $\alpha_{\text{PDE}} = \alpha_{\text{inc}} = \alpha_{\text{D}} = \alpha_u = 1$ when applicable. The weight for traction boundary conditions α_{N} is also set to be 1, but we evaluate the this part separately for each of the five boundaries (left, right, top, bottom, inner), leading to five loss terms related to traction boundaries, each with weight 1, in the total loss. In case 2, the calculation of the mean squared error of the PDE loss term $\mathcal{L}_{\text{PDE}}(\boldsymbol{\lambda}, \boldsymbol{\theta})$ is further weighed inversely by the local density of residual points, so that different spatial regions contribute equally to the loss term of PDEs.

For cases 0 and 3, the small values of loading (roughly $\mathcal{O}(10^{-3})$) cause difficulties in training the PINN. To mitigate this issue, we scale up the external loading (and $\sigma_{\text{Y}} = 0.005$ in case 3) 100 times and feed the scaled loading into the PINN. According to Eqs. S9 and S10, such scaling only results in 100 times the original displacement (and hence strain) and does not essentially changes the characteristics of the solution. After obtaining the solution from the PINN, we scale the displacement/strain solution back (0.01 times) to its original value.

Details of the FEM. We use Abaqus as the FEM solver to generate the displacement data $\{\mathbf{u}^{*(i)}\}_{i=1}^{N_u}$ and the entire displacement field, given the reference values of unknown parameters $\boldsymbol{\theta}^*_{\text{unk}}$. The linear density of meshes is 120 per unit length, which is sufficiently dense so that the FEM/Abaqus solution is accurate enough to serve as the reference solution. Plane strain quadratic elements with hybrid formation (CPE8H elements) are applied for hyperelasticity.

Section S4. Additional Results for Cases in the Main Text

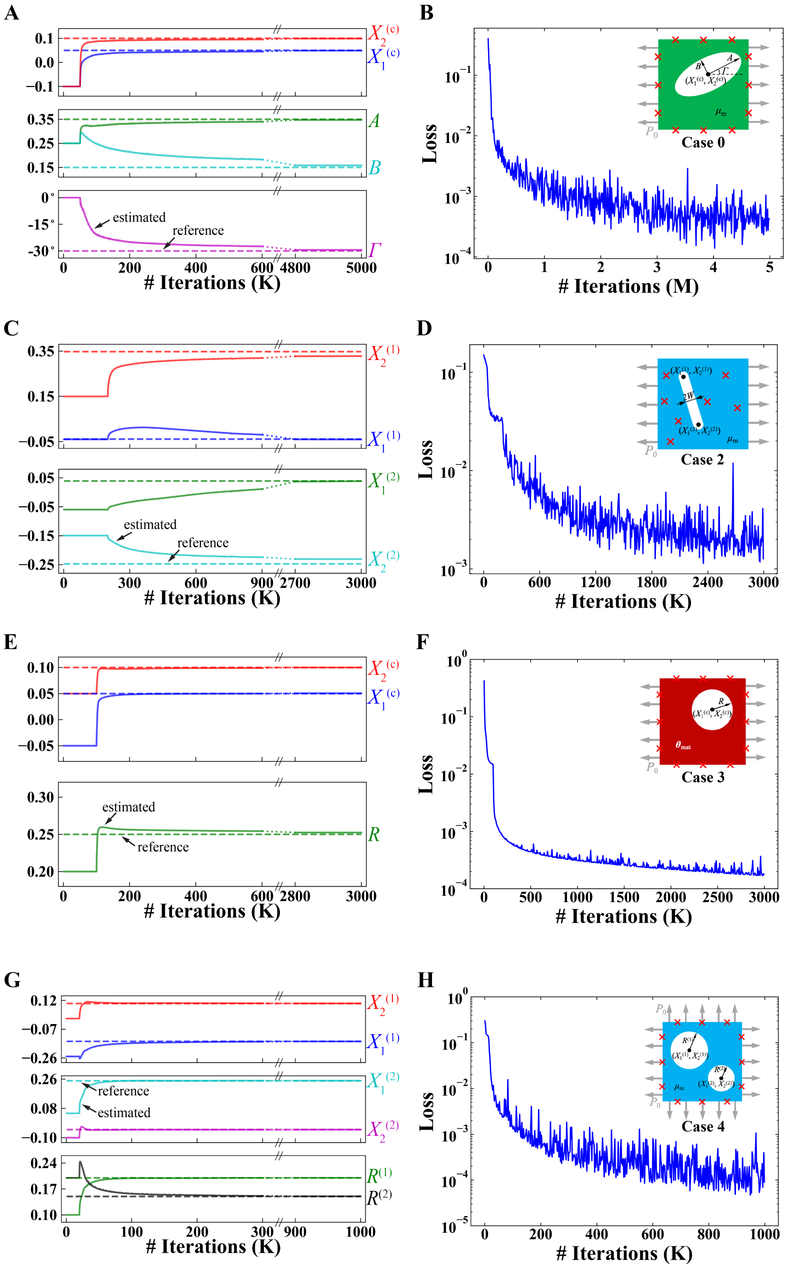


Fig. S1. Evolution of estimated unknown parameters and loss function in cases 0, 2, 3 and 4. See Fig. 2 for the definitions of the cases. (**A**, **C**, **E**, **G**) The dashed lines and solid lines represent the reference value and estimated value of unknown variables. Unknown parameters are not trainable in the pre-training process during the first 50K (case 0), 200K (case 2), 100K (case 3), and 20K (for case 4) iterations, respectively. (**B**, **D**, **F**, **H**) Loss function during the training process. See Fig. 7 in the main text for the results for cases 1 and 5.

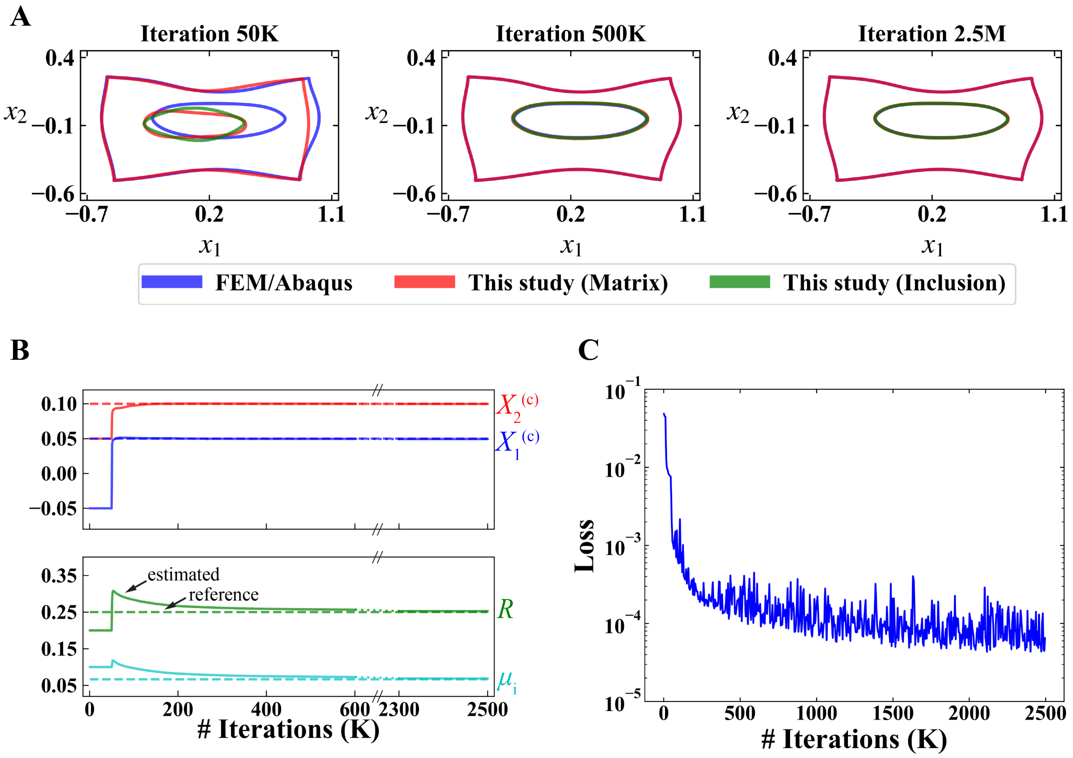


Fig. S2. Results of case 5 with additional internal measurement points. (**A**) Comparison of the visual outlines (including the material interface) of the deformed configurations between FEM/Abaqus (blue) and PINN (red for the matrix; green for the inclusion) results after 50K, 500K, and 2.5M iterations. (**B**) Evolution of the estimated unknown parameters $(X_1^{(c)}, X_2^{(c)}, R, \mu_{\rm i})$ during the training process. The dashed lines and solid lines are reference values and estimated values, respectively. (**C**) The loss function during the training process.

Section S5. Forward Problem on One Centered Circular Void

In this section, we study on a forward problem to verify the ability of PINNs in solving boundary value problems of hyperelastic solids (see Section S2 for details of the formulation of PINNs for forward problems). For forward problems, we have $\boldsymbol{\theta}_{\rm unk} = \emptyset$ (hence $\boldsymbol{\theta} = \emptyset$ for convenience), and we do not have any observation on the displacement field. The setup is a simplified, forward version of case 1: the matrix includes a circular void located at the center $(0.0, 0.0)$ with radius 0.1; all the other setup details are the same as the foregoing description for case 1. The target is to solve the displacement field under the specified loading condition.

The results are shown in Fig. S3. We trained the PINN over 1M iterations in total. Fig. S3A shows the comparison of deformed configurations of the solid between the FEM solver and the PINN solver after 20K, 200K and 1M iterations, with the absolute L^2 error of the displacement field marked on each subfigure. For clarity of presentation, this figure shows the outer and inner boundaries of the specimen visualized from the FEM and PINN analyses. Although we trained the PINN for 1M iterations in total, the visual pattern of the PINN result almost fully overlaps with the pattern of the FEM result after 200K iterations. The L^2 error finally decreases to as small as $\mathcal{O}(10^{-4})$, compared to the typical displacement of $\mathcal{O}(10^{-1})$. Fig. S3B shows the distribution of the true stress component σ_{11} along the central line of the solid ($X_1 = 0$) after 20K, 200K and 1M iterations. After training over 20K iterations, the stress pattern of the PINN result is already qualitatively similar to that of the FEM result. The remaining difference is gradually diminishing through the following training iterations. Fig. S3C shows the value of the loss function throughout the training process. The total loss is decreased to $\mathcal{O}(10^{-5})$ after 1M iterations. According to all the results presented in Fig. S3, the PINN has accurately solved this forward boundary value problem of hyperelastic solids. This example of a forward problem serves as a basis for the inverse problem where the introduction of trainable unknown parameters slightly increases the complexity of the problem. In Section S6, we present a parametric study based on this case.

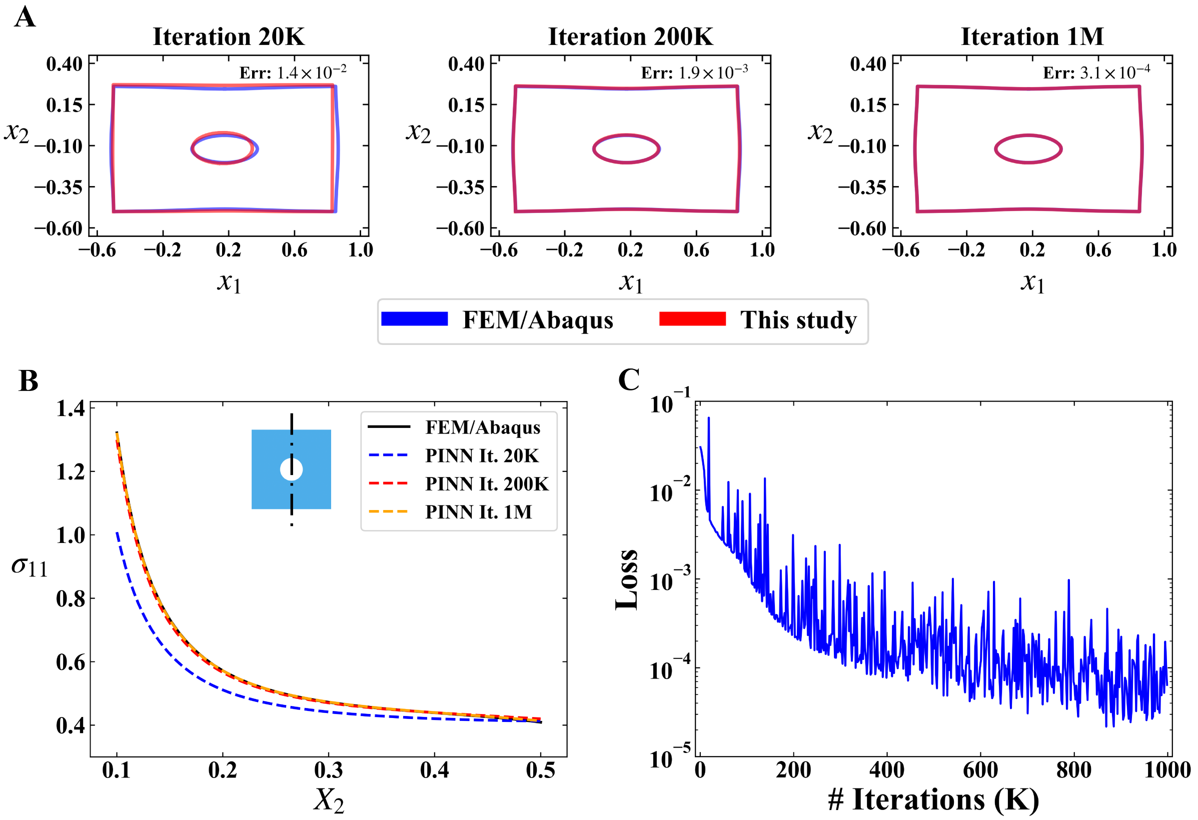


Fig. S3. Results of the forward problem on one centered circular void. (**A**) Comparison of the visual outlines of deformed configurations between FEM/Abaqus (blue) and PINN (red) results after 20K, 200K, and 1M iterations. The absolute L^2 error of the displacement field between the two methods is marked in each subfigure. (**B**) Stress component σ_{11} along the vertical axis of symmetry ($X_1 = 0$) after 20K, 200K, and 1M iterations (dashed lines), with the FEM/Abaqus result as a comparison (solid line). (**C**) The loss function during the training process.

Section S6. Parametric Study of the Forward Problem: Influence of NN Architecture, Density of Residual Points, and Loss Weights

Here we conduct a parametric study on the forward problem in Section S5. We study the influence of hyperparameters on the simulation results to justify choice of hyperparameters in this paper. We consider changing the width and the depth of the NN from the reference architecture 2-30-30-30-30-3 ($d = 4$ hidden layers each with width $w = 30$, called 4-30 for short and similarly d-w for other architectures) in the main paper to be wider (4-45), narrower (4-15), deeper (6-30), and shallower (2-30). In the forward problem in Section S5 (and also case 1 in main text), we have placed 40×20 residual points in each of the four sub-regions (see Section S3 for details). Here we change the number of residual points into variables: $\rho \times (\rho/2)$ for each sub-region, where ρ is the linear density of residual points. With this definition, the number of residual points on the boundary is proportional to ρ, and the number of residual points inside the solid is proportional to ρ^2. Here we consider changing ρ to range from 10 to 80. For each combination of ρ and d-w, we use the PINN to solve the forward problem. The results of the absolute L^2 error of the displacement field compared to the FEM solution are shown in Fig. S4. We consider the displacement fields after 200K (averaged by taking the geometric mean value after 180K, 190K and 200K iterations, to prevent the influence of fluctuation) and 1M iterations (averaged similarly by 980K, 990K and 1M iterations) . For any architecture, as ρ increases to around 30, the L^2 error does not significantly depend on ρ, indicating that the chosen values of ρ in the main paper are sufficiently large for achieving a high accuracy. From 200K to 1M iterations, the L^2 errors of the shallower (2-30) and the narrower (4-15) architectures do not further decrease significantly, while the errors of the reference (4-30), deeper (6-30), and wider (4-45) architectures continue decreasing significantly. This indicates that the insufficient depth for 2-30 and width for 4-15 limits the approximation ability of the NN and hence restricts the further descent of the error. The reference architecture (4-30) is deep and wide enough for reaching a small L^2 error, because its error is close to the error from the deeper (6-30) and wider (4-45) architectures. This simple parametric study indicates that our choice of the architecture of the NN (4-30) and the linear density of the residual points (40) is suitable for solving our prototypical problem.

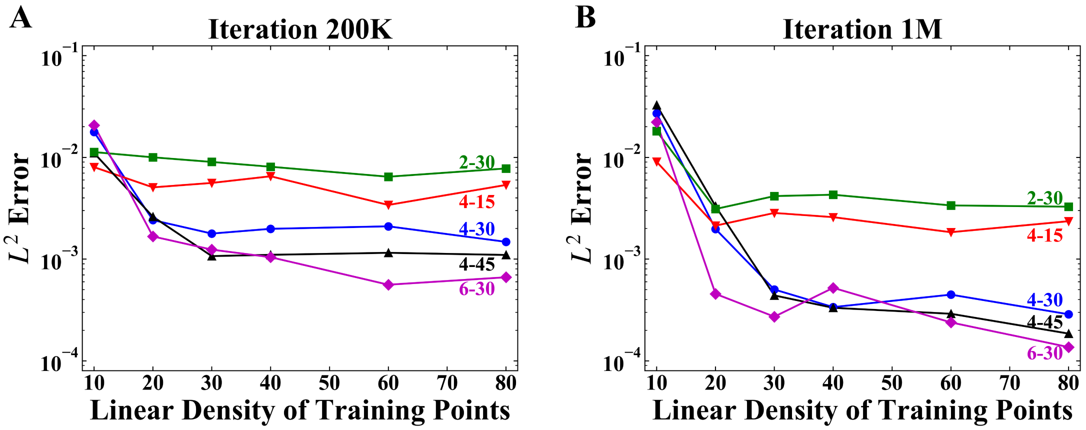


Fig. S4. Parametric study of the forward problem in Section S5 on the NN architecture and density of residual points. We compare the L^2 error of displacement fields obtained by the PINN and the FEM/Abaqus after training over (**A**) 200K and (**B**) 1M iterations. We consider different depths (2, 4, and 6 hidden layers) and widths (15, 30, and 45 neurons for each hidden layer) of the NN, as well as different linear densities of residual points. The results of the network with d hidden layers and w neurons for each hidden layer are marked with "d-w". The linear density of residual points ρ means that the number of residual points is proportional to ρ for the boundary conditions and proportional to ρ^2 for the PDEs. The displayed value of L^2 error is obtained by taking the geometric mean value after 180K, 190K and 200K iterations for (**A**) and 980K, 990K and 1M iterations for (**B**), in order to prevent the fluctuation during the training process.

We also consider changing the weights of the loss terms (α's with various subscripts) and studying their influence on the convergence histories of the loss function and the accuracy of the displacement field. Again, we conduct the study using the forward problem in Section S5. For simplicity, we assume $\alpha_{\text{PDE}} = \alpha_{\text{inc}} = 1$ and $\alpha_{\text{D}} = \alpha_{\text{N}} = \alpha$, where α is a variable. With this assumption, we group the loss weights by whether the corresponding residual points are inside the domain or on the boundary. Note that in Section S5, the choice is $\alpha = 1$. Here we train the PINN with $\alpha = 1, 3, 10$ and all the other setup details the same as Section S5. The results are shown in Fig. S5. Figs. S5A-C show the evolution of the total loss ($\mathcal{L}_{\text{for}}$), the internal loss ($\mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{inc}}$), and the boundary loss ($\mathcal{L}_{\text{D}} + \mathcal{L}_{\text{N}}$) during the training process for $\alpha = 1, 3, 10$, respectively. As α increases, the boundary loss tends to descend faster, while the evolution of the internal loss and the total loss does not change significantly. Fig. S5D shows the evolution of the absolute L^2 errors of the displacement fields for $\alpha = 1, 3, 10$ compared to the FEM/Abaqus solution. With a larger α, the displacement field converges faster. On the other hand, in the late stage the training process (after 500K iterations), the three values of L^2 error reach the same plateau, meaning that the value of α does not significantly influence the final accuracy of the displacement field for the forward problem. Since this paper focuses on the fundamental method and the prototypical problem as a proof of concept rather than tuning the hyperparameters to achieve optimal practical efficiency, we simply fix all the weights to be one in all the cases in the main text.

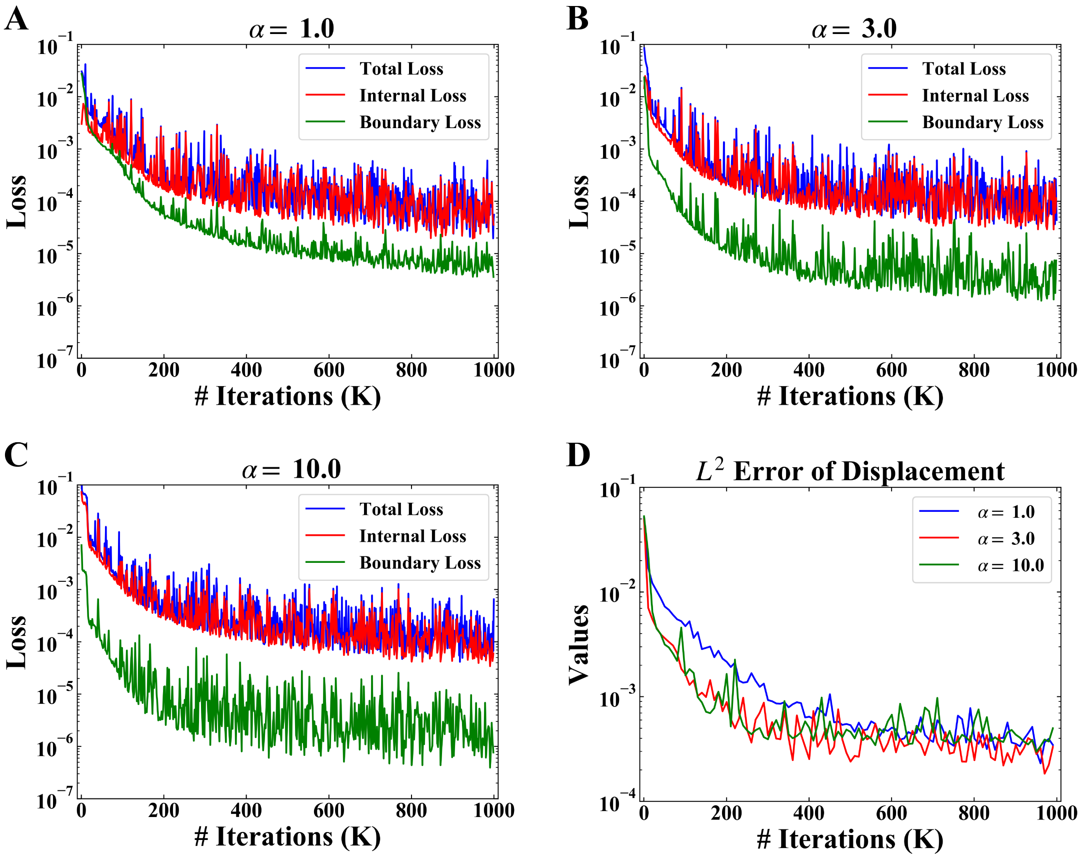


Fig. S5. Parametric study on the weights of loss components. We studied the influence of loss weights on the evolution of the loss function and the accuracy of the displacement using the forward problem in Section S5. We fix the weights of the loss terms inside the PDE domain to be $\alpha_{\text{PDE}} = \alpha_{\text{inc}} = 1$, and set the loss terms on the boundary to be one variable $\alpha_{\text{D}} = \alpha_{\text{N}} = \alpha$. (**A-C**) The evolution of the total loss ($\mathcal{L}_{\text{for}}$), the internal loss ($\mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{inc}}$), and the boundary loss ($\mathcal{L}_{\text{D}} + \mathcal{L}_{\text{N}}$) during the training process for $\alpha = 1, 3, 10$, respectively. (**D**) The absolute L^2 error of the displacement field by the PINN compared to the FEM/Abaqus result for $\alpha = 1, 3, 10$ during the training process.

Section S7. Parametric Study of Simplified Case 1: Influence of Locations of Measurement Points on the Outer Boundary

In this section, we conduct a parametric study on the influence of locations of measurement points on the estimation results based on case 1. We alter the locations of displacement measurements on the outer boundary, while keeping all the other setup details unchanged. The six setups of displacement measurements (sensors) are illustrated in Fig. S6. In all the setups, 10 points are placed on the left and right boundaries, respectively. To create a relatively tough condition, there is no measurement is on the top and bottom boundaries. The difference among six setups comes from the specific locations of measurement points. The left and right edges are equally partitioned into 3 segments, with different segments equipped with measurement points in different setups: the bottom segment for setup 1, the middle segment for setup 2, the top segment for setup 3, the bottom and middle segments for setup 4, the middle and top segments for setup 5, and all segments for setup 6 (see Fig. S6A).

The tilting angle Γ is assumed to be the only unknown parameter for the study in this section. The evolution of Γ (normalized by 180°) in the six setups over 1M training iterations is shown in Fig. S6B. The reference value Γ^* is plotted with dashed line. In setups 3, 5 and 6, the estimated values of Γ gradually approach the reference value, despite the existence of errors which may be caused by the removal of measurement points on the top and bottom boundaries. In setups 1, 2 and 4, the estimated Γ do not approach the target value, indicating that the parameter estimation fails. We further show the deformed configuration of each setup after 870K iterations in Fig. S6C, where the fluctuating values of Γ in setup 2 and 4 are accidentally close to the reference value Γ^*. According to Fig. S6C, setups 3, 5 and 6 infer the displacement field accurately. For setups 2 and 4, although Γ happens to be close to Γ^*, their deformed configurations are significantly different from the FEM results. Consequently, setups 3, 5 and 6 successfully estimate the unknown tilting angle and infer the displacement field, while setups 1, 2 and 4 fail to do so.

For the specific example we consider, a successful characterization of geometry requires that the measurement points are on the top segment, while the middle and bottom segments do not contribute much. This feature may be caused by the fact that the actual location of the void is slightly above the center ($X_2^{(\mathrm{c})*} > 0$). Intuitively, the closer the measurement points are to the void, the more informative the data is. This conforms with the Saint-Venant's principle in elasticity – if the measurements are located too far from the void, then the non-homogeneity of the displacement field is diminished, so that these measurements do not contain enough information regarding the void geometry. Therefore, solving geometry identification problems with PINNs poses requirements on the location of measurement points. For the current problem, without prior knowledge of the location and mechanical properties of the void, a safe way is to place the measurement points uniformly in the admissible region (e.g., on the outer boundary) to acquire as much information as possible.

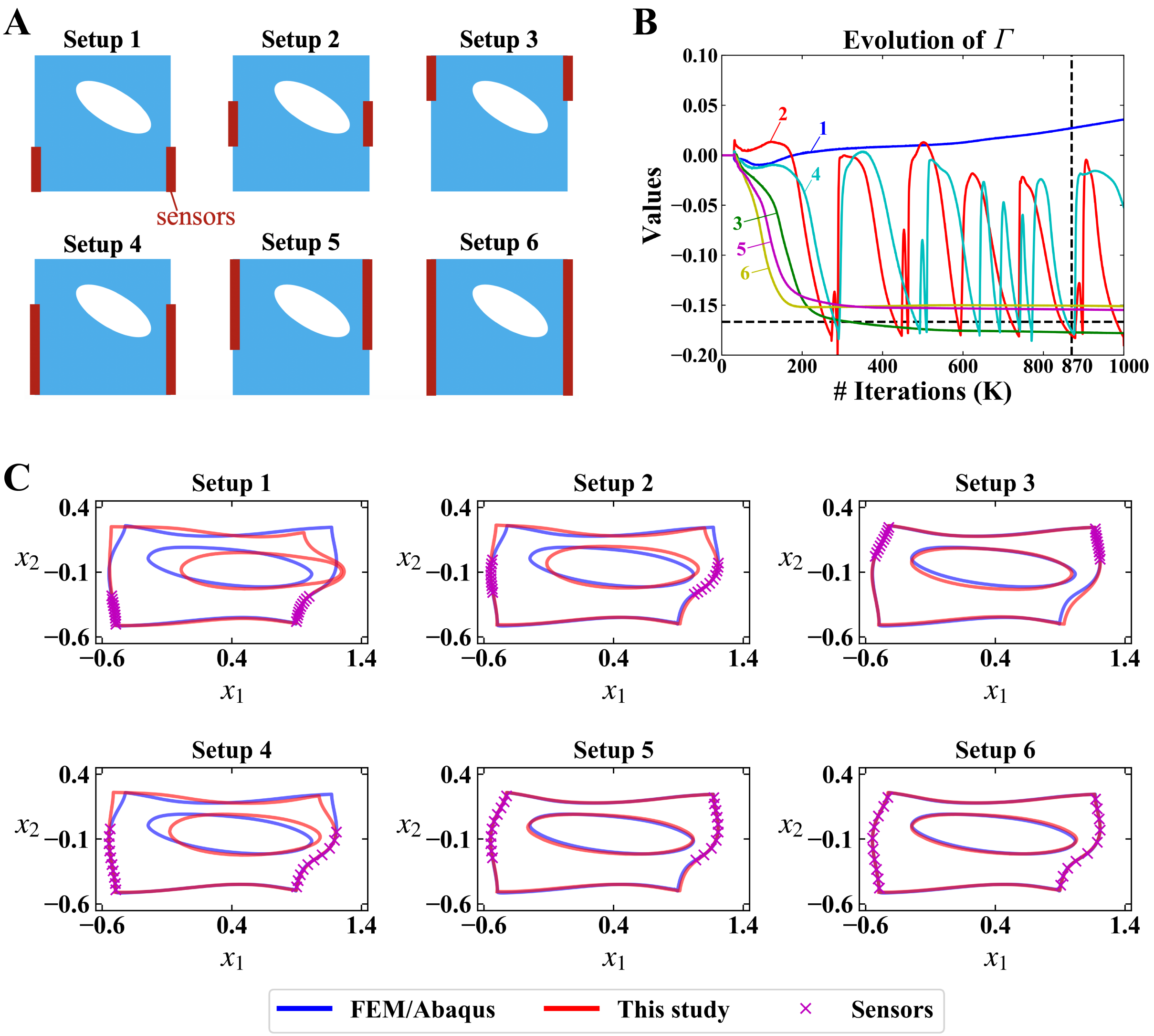


Fig. S6. Parametric study on the locations of measurement points. We show the dependence of parameter estimation results on the locations of measurement points. (**A**) In the 6 different setups, we place 10 points uniformly at different segments of the left and right boundaries, respectively. The left and right edges are equally partitioned into 3 segments, and then measurement points (sensors) are placed on: the bottom segment for setup 1, the middle segment for setup 2, the top segment for setup 3, the bottom and middle segments for setup 4, the middle and top segments for setup 5, and all segments for setup 6. (**B**) Parameter estimation results of the tilting angle Γ (normalized by 180°) in the 6 setups. The curves for different setups are marked with the setup numbers. The horizontal dashed line is the reference value Γ^*. The vertical dashed line marks the location of 870K training iterations. (**C**) The deformation patterns of all setups after 870K iterations.

Section S8. Parametric Study on Simplified Case 1: Influence of the Length Scale of the Void

We in this section study the influence of the length scale of the void on the estimation accuracy. For the purpose of simplicity, we assume that the void is circular and is placed at the center of the matrix, with the only unknown parameter being its radius ($\boldsymbol{\theta}_{\text{unk}} = R$). We set the radius of the void R^* to range from 0.025 to 0.2, and initialize the radius estimation to be $R^0 = 1.5R^*$ for each case. For each R^*, we run the training process until the estimated value converges evidently. Other technical details are the same as case 1 of the main paper.

The results of the relative error of estimated radius are shown in Fig. S7. Note that the side length of the matrix is 1.0. As R^* decreases to around 0.1, the relative error starts to increase beyond 10^{-2}. As the radius decreases to $R^* = 0.05$, the identification error is around 20%, indicating that the method starts to become inaccurate. Consequently, for this specific case where displacement is measured only on the boundary, our approach is valid for voids within around one order of magnitude smaller than the size of the entire domain.

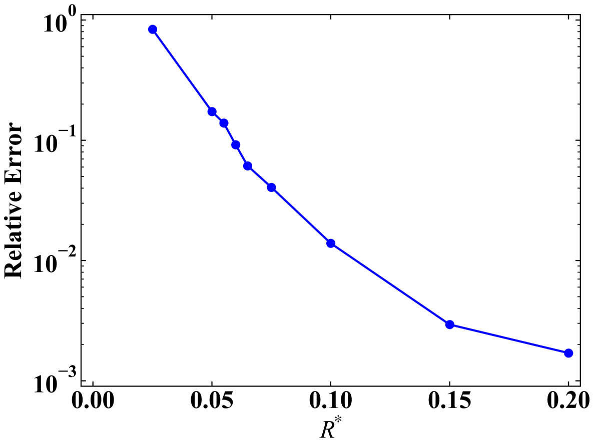


Fig. S7. Parametric study on the length scale of the void. We show the relative error of the estimated void radius for different values of R^* (radius of the void).

Section S9. Parametric Study of Simplified Case 1: Influence of the Location of the Void

A natural question following Section S8 is how the location of the void influences the estimation accuracy. Here we consider another simplified setup of Case 1. We suppose that only $(X_1^{(\mathrm{c})}, X_2^{(\mathrm{c})}, R)$ are unknown to the PINN. We fix $R^* = 0.1$ and change the location of the void by choosing $X_1^{(\mathrm{c})*}, X_2^{(\mathrm{c})*} \in \{-0.3, -0.2, -0.1, 0.0, 0.1, 0.2, 0.3\}$, which essentially moves the void throughout the square matrix. The absolute error of parameter estimation is shown in Fig. S8, with the three panels for $X_1^{(\mathrm{c})}$, $X_2^{(\mathrm{c})}$, and R, respectively. Despite some difference in the value of error, the PINN can accurately estimate the three unknown parameters for all the locations of the void. Consequently, our method is robust about the location of the void.

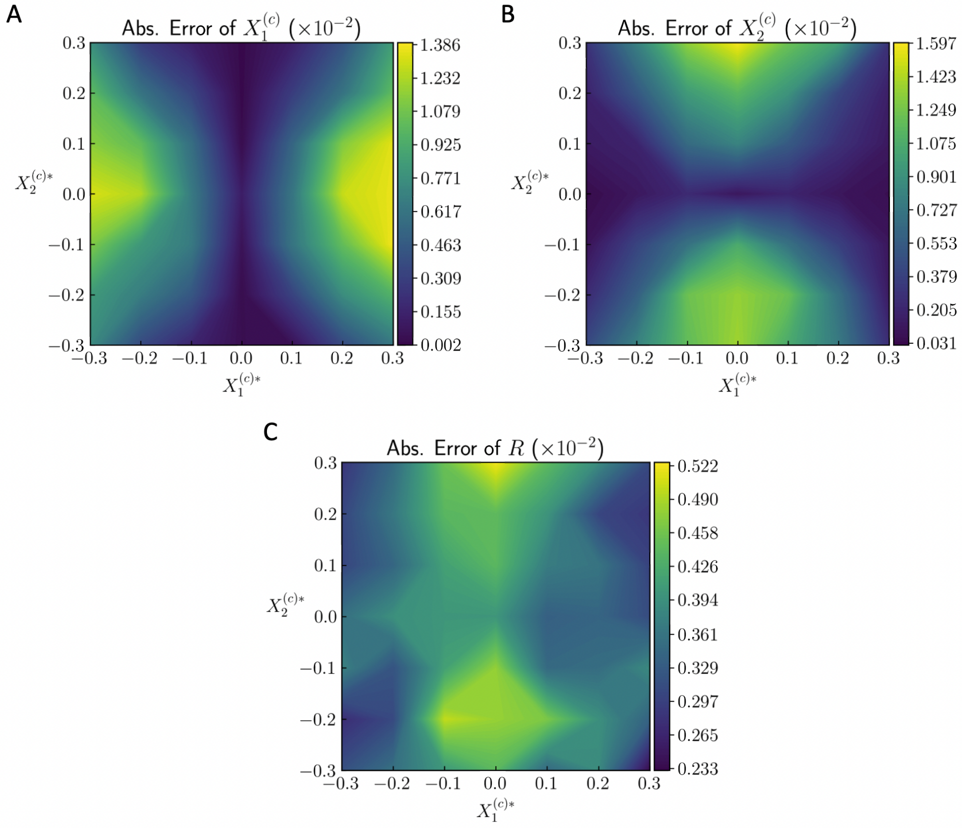


Fig. S8. Parametric study on the location of the void. We show the absolute error of parameter estimation for the three unknown parameters ($X_1^{(\mathrm{c})}$, $X_2^{(\mathrm{c})}$, R) for different true locations of the void ($X_1^{(\mathrm{c})*}$, $X_2^{(\mathrm{c})*}$).

Section S10. Parametric study on Simplified Case 5: Influence of the Moduli Ratio

Here we study the influence of the moduli ratio of the inclusion to the matrix (μ_i/μ_m) on the estimation accuracy. The circular inclusion with known radius $R = 0.2$ is placed at the center of the matrix. The modulus of the matrix $\mu_m = 0.333$ is known to the PINN, and the only unknown parameter is the modulus of the inclusion ($\boldsymbol{\theta}_{\text{unk}} = \mu_i$). We set the moduli ratio μ_i^*/μ_m to range from 0.2 to 5.0, and initialize the modulus of the inclusion to be $\mu_i^0 = 1.5\mu_i^*$ for each case. For each μ_i^*/μ_m, we run the training process until the estimated value converges evidently. Other technical details are the same as case 5 in the main text.

The results of the (signed) relative error for each μ_i^*/μ_m are shown in Fig. S9. Within the range we consider ($0.2 \leq \mu_i^*/\mu_m \leq 5.0$), all cases produce very accurate results, with relative error no larger than 10^{-2}. Therefore, for the prototypical problem, our approach provides accurate estimation results for the moduli ratio ranging from 0.2 to 5.0, which is reasonably large and covers more than one order of magnitude.

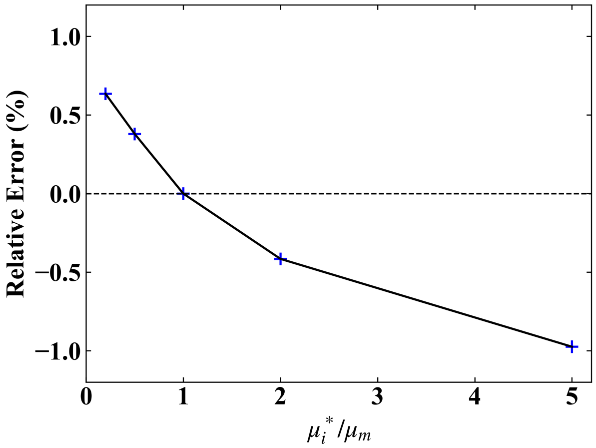


Fig. S9. Parametric study on the moduli ratio of the matrix to the inclusion. We show the (signed) relative error of the estimation of the moduli ratio for different values of μ_i^*/μ_m.

Section S11. L-BFGS Optimizer

Here we do a simple comparison on the model performance in terms of accuracy and efficiency with different strategies on optimizers based on case 4. In the main text, we adopt the Adam optimizer as the only optimizer throughout the entire training process with 1M iterations (called strategy 1A in this section), after which both the parameter estimations and loss function reach a relative plateau. Such a strategy gives high accuracy and helps us study the convergence history as a fundamental characteristic of our method. To achieve a reasonable accuracy practically, one may not need as many as 1M iterations. Here, we consider training the PINN over 200K iterations only (called strategy 1B). To further improve the computational efficiency, we also consider using Adam for the first few iterations (40K iterations in our case), and then switching to L-BFGS (*48*) (called strategy 2), which is common for training PINNs practically. We compare the results of the three strategies (1A, 1B and 2) in Table S1 in terms of accuracy of parameter estimation and computational efficiency. The computational time is measured by running the code on typical machines using CPU only. The results of strategy 1B indicates that training the PINN with 200K iterations provides reasonably high accuracy. Strategy 2 using L-BFGS performs even better – accuracy similar to strategy 1A is achieved, while the computational cost is significantly reduced.

Case 4	$X_1^{(1)}$	$X_2^{(1)}$	$R^{(1)}$	$X_1^{(2)}$	$X_2^{(2)}$	$R^{(2)}$
Reference Value	-0.15	0.10	0.20	0.25	-0.05	0.15
Strategy 1A: Adam 1M (around 667 minutes)						
Estimated Value	-0.15089	0.10018	0.20007	0.25045	-0.05008	0.15019
Absolute Error($\times 10^{-2}$)	0.09	0.02	0.01	0.05	0.01	0.02
Relative Error(%)	0.09	0.02	0.04	0.05	0.01	0.13
Strategy 1B: Adam 200K (around 133 minutes)						
Estimated Value	-0.15517	0.10191	0.19911	0.24975	-0.05033	0.15298
Absolute Error($\times 10^{-2}$)	0.52	0.19	0.09	0.03	0.03	0.30
Relative Error(%)	0.52	0.19	0.45	0.10	0.03	1.99
Strategy 2: Adam 40K and L-BFGS (around 35 minutes)						
Estimated Value	-0.15110	0.10002	0.19998	0.24993	-0.04953	0.15076
Absolute Error($\times 10^{-2}$)	0.11	0.00	0.00	0.01	0.05	0.08
Relative Error(%)	0.11	0.00	0.01	0.01	0.05	0.51

Table S1. Parameter estimation results for case 4 with different strategies on optimizers.