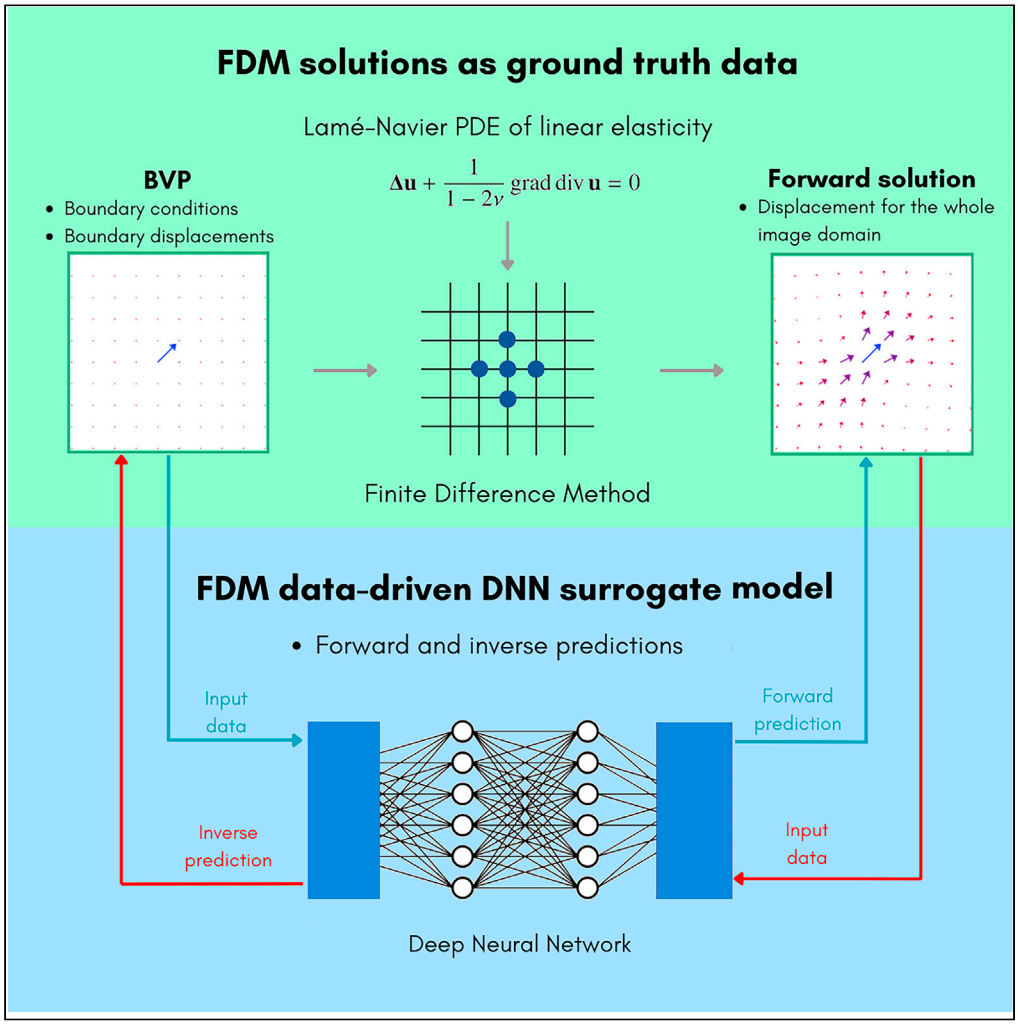
iScience

CellPress
OPEN ACCESS

Article

Linel2D-Net: A deep learning approach to solving 2D linear elastic boundary value problems on image domains



Anto Nivin Maria Antony, Narendra Narisetti, Evgeny Gladilin

gladilin@ipk-gatersleben.de

Highlights

Linel2D-Net accurately reproduces solutions of continuum mechanics BVPs

Linel2D-Net solves both forward and inverse problems

Linel2D-Net is 33x times faster than conventional FDM numerical solver

Linel2D-Net finds applications in deformable modeling and simulation

Maria Antony et al., iScience *27*, 109519
April 19, 2024 © 2024 The Authors. Published by Elsevier Inc.
https://doi.org/10.1016/j.isci.2024.109519

iScience

CellPress
OPEN ACCESS

Article

LineI2D-Net: A deep learning approach to solving 2D linear elastic boundary value problems on image domains

Anto Nivin Maria Antony,[1] Narendra Narisetti,[1] and Evgeny Gladilin[1,2,*]

SUMMARY

Efficient solution of physical boundary value problems (BVPs) remains a challenging task demanded in many applications. Conventional numerical methods require time-consuming domain discretization and solving techniques that have limited throughput capabilities. Here, we present an efficient data-driven DNN approach to non-iterative solving arbitrary 2D linear elastic BVPs. Our results show that a U-Net-based surrogate model trained on a representative set of reference FDM solutions can accurately emulate linear elastic material behavior with manifold applications in deformable modeling and simulation.

INTRODUCTION

The demand for image analysis goes hand in hand with the amount of image data that continuously grows over decades, especially in the field of biomedical imaging. Typically, analysis of biological images is restricted to the derivation of simple descriptors of target structures such as location, volume, color, shape, and their temporal changes. However, with the current technological advancements and the availability of powerful computational platforms, image analysis can be extended by the estimation of the physical behavior and constitutive properties of underlying the biological structures and simulation of their response to variable boundary conditions.[1,2] Image- and physics-based frameworks have been applied for the biomechanical simulation of cell and tissue growth as well as organ and whole organism morphogenesis in a number of previous works.[3–6] A large set of image analysis problems in structural and developmental biology can be traced back to the computational modeling of soft tissue behavior using theoretical and computational frameworks of continuum mechanics[4,7] and non-rigid image registration.[8] Structural changes in biological tissue under the impact of external forces and internal pressures are typically described in terms of displacements of landmark points or the entire set of image pixels (displacement fields).[7,8] The non-rigid displacement field can, in turn, be used for the assessment of local strains/stresses that play a key role in triggering and regulating signaling in developmental processes.[9,10] Efficient algorithms for fully automated physics-based modeling are one of the major prerequisites for shifting the paradigm of biomedical image analysis from simple morphological description to high-throughput functional phenotyping.

Arbitrary boundary value problems (BVPs) of continuum mechanics that are given by constitutive laws in the form of partial differential equations (PDEs) and prescribed boundary conditions can, in general, be solved using numerical solving techniques only. Conventional PDE solving techniques such as finite elements (FEMs),[11] finite difference (FDM),[12] boundary elements [13] or mesh-free methods[14] are known for decades and have been frequently applied in the context of biomedical data analysis and modeling.[15,16] However, conventional discretization techniques typically require laborious adaptation to concrete data and expertise in numerical modeling, which limits their straightforward and broader application in biomedical applications. Furthermore, conventional solving techniques are based on iterative computations, which hampers their application to fast and real-time simulations. To address these limitations, alternative approaches have been developed in the past that focus on reducing the computational demand required for the conventional solvers, such as model reduction techniques.[17–22] With increasing computational power of modern GPUs, a number of GPU-accelerated numerical schemes were developed. GPU-based computational frameworks such as SoFa[23] and NiftySim[24] may principally allow to speed up numerical solutions compared to conventional CPU implementations, however, they still have the same disadvantage of being essentially iterative and laborious by routine application in biomedical sciences.

Recent advances in machine learning and, in particular, deep learning open new possibilities for more efficient computational data analysis, including physics-based modeling. This new family of methods, including surrogate and physics-informed neural networks (PINNs), rely on the idea of training neural networks to resemble exact theoretical or numerical solutions of physical boundary value problems such as fluid and continuum mechanics, diffusion, or heat transfer problems.[25] Based on a representative amount of data (i.e., example solutions obtained from conventional solvers), neural networks can establish a mapping between the input and output data, effectively emulating the physical laws with a sufficiently high accuracy.[25–29] In previous work, neural networks were applied to a number of specific numerical problems, including the simulation of the arterial blood pressure from 4D MRI data,[30] laser-assisted bending[31] and several other numerical modeling

[1]Leibniz Institute of Plant Genetics and Crop Plant Research, OT Gatersleben, Corrensstr. 3, 06466 Seeland, Germany
[2]Lead contact
*Correspondence: gladilin@ipk-gatersleben.de
https://doi.org/10.1016/j.isci.2024.109519

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

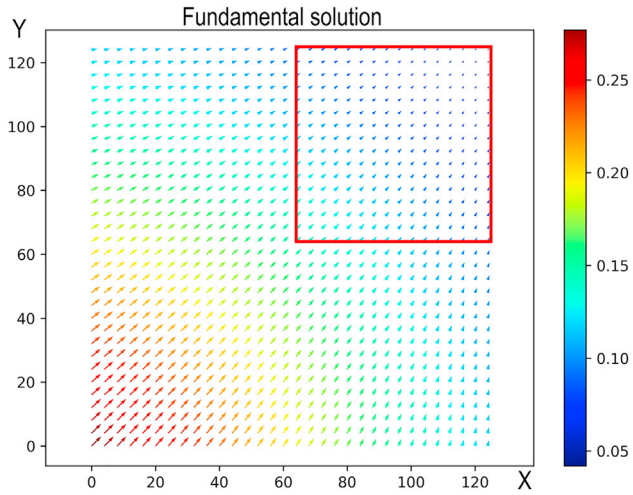


Figure 1. Validation of FDM solution vs. analytical (fundamental) solution of the 2D Lamé-Navier PDE $u(r)$ is performed for the quadratic subdomain $r(x, y) \in \Omega$ (indicated in red) of the infinite medium Ω_{inf}

For the validation of FDM, only inner points (i.e., inner image pixels) $r(x, y) \in \Omega \backslash \Gamma$ were considered, while the boundary values were set equal to the fundamental solution $u(r) = u_{fs}, r(x, y) \in \Gamma$ and used as boundary conditions for computing the FDM solution in the remaining inner subdomain.

and physics-based problems.[32–37] Meanwhile, powerful frameworks to solve physical problems with neural networks such as DeepXDE,[38] Nvidia SimNet,[39] NeuroDiffEq[40] were established. DeepXDE provides the option to include arbitrary boundary conditions, and it works with fully connected neural networks to train the model. The other frameworks do not provide enough flexibility for the consideration of arbitrary BVPs and/or a straightforward interface for implementation in the image domain.

In this feasibility study, we aim to develop a deep learning-based solver for a large set of 2D linear elastic BVPs including arbitrary geometrical and physical boundary conditions using convolutional neural networks (CNN), which are known to be very efficient when compared to fully connected neural networks when applying to image-based problems. The deep neural network used in this work is derived from the conventional U-Net architecture[41] which was widely used for image segmentation. The U-Net architecture was modified in such a way that it can predict the displacement field for the entire image domain from displacements and boundary conditions defined on its parts, - effectively solving a given linear elastic BVP. This CNN (further termed as Linel2D-Net) is trained on a large set of reference solutions to emulate the solutions produced by the conventional methods. Instead of laborious conventional solvers, data-based deep learning (surrogate) models can be used for high-throughput analysis of large, in particular, biomedical image datasets in a fully automated manner. Previously, such surrogate modeling approaches have been applied to different fluid and continuum mechanics problems.[42,43] However, the focus of previous works was predominantly on solving physical BVP for a limited class of geometrical boundary conditions. However, due to the extremely high level of the variability of living systems, biomedical applications demand efficient methods that can be applied to arbitrary boundary values. Consequently, here we aim to train a linear elasticity solver that is not restricted to particular boundary conditions and can be applied to solving BVPs with arbitrary boundary conditions.

Experimental results of this study show that our surrogate Linel2D-Net model, pre-trained on a large amount of "exact" linear elastic FDM simulations is capable of accurately reproducing solutions of various continuum mechanics BVPs with an accuracy sufficiently high for manifold applications on the field of deformable body modeling and biomedical image analysis.

RESULTS

Validation of the finite difference solver using the fundamental solution of the 2D Lamé-Navier partial differential equation

In this work, reference solutions of the 2D Lamé-Navier PDE were generated using the Finite Difference Method on the regular grid of 128x128 image pixels as described in the STAR Methods section. In order to validate the accuracy of the implemented FDM scheme, the fundamental solution of the 2D Lamé-Navier PDE was used (Equation 8). The fundamental solution provides an analytic expression for the displacement of every point of the infinite 2D elastic medium Ω_{inf}. In order to use this theoretically justified solution for the validation of FDM predictions, a finite rectangle subdomain of the infinite elastic medium $\Omega \subset \Omega_{inf}$ was selected.

According to the Gaussian integral theorem, boundary values uniquely define the solution inside the enclosed homogeneous and isotropic domain. Consequently, the comparison of the FDM predicted displacement field vs. the fundamental solution was performed for the inner nodes of the rectangular subdomain of the infinite elastic continuum, whereas the displacements of boundary nodes from the fundamental solution were used as boundary conditions (Figure 1). Comparison between the FDM and fundamental solutions in Figure 2 shows that the difference of the displacement magnitude between FDM predicted and theoretical values amounts up to 1e-3 or 0.75%.

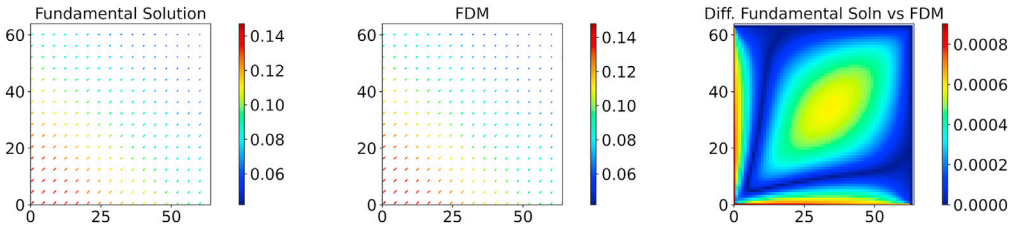


Figure 2. Comparison of the fundamental solution of the Lamé-Navier PDE vs. the FDM solution computed for a 128x128 image domain

From left to right: the fundamental solution, the FDM solution computed for the BVP with the boundary conditions given by displacements from the fundamental solution for all four image boundaries.

Generation of boundary value problems

In order to cover a large spectrum of possible boundary value problems, different geometrical patterns, and prescribed boundary displacements were generated and combined. While (Free) boundary conditions were annotated with "black" image pixels, i.e., image intensity equal to zero, and boundary conditions with prescribed displacement (Dirichlet) were labeled with non-zero values. For the definition of image regions with prescribed (Dirichlet) boundary conditions, different simple geometrical patterns such as circles, lines, polygons, free shapes, and their combinations were used. For each geometrical pattern, several different values of prescribed displacements were defined. Thereby, the magnitude as well as the spatial orientation of displacements were perturbed in order to achieve a possible bias-free set of BVP problems.

Finite difference modeling of material compressibility

In this work, we focus on modeling a homogeneous isotropic medium, which is described by a single material parameter - the Poission's ratio ranging between $\nu \in [0, 0.5]$. To simulate the behavior of materials with different compressibility, FDM solutions were generated for three distinctive values of $\nu = 0, 0.25, 0.49$ that correspond to high, moderate, and low compressibility, respectively. Accordingly, three sets of FDM solutions were used for the training of three different models. Figure 3 shows examples of two FDM solutions corresponding to $\nu = 0$ and $\nu = 0.49$ for a simple BVP given by the displacement of the central image pixel. As one can see that patterns of displacement fields for low and high-compressible materials show distinctive differences. In particular, the low-compressible medium exhibits vortices due to volume preservation that are absent in the deformation pattern of the high-compressible material.

Forward Linel2D-Net models

Forward Linel2D-Net models were trained to predict the displacements of the entire image domain from the displacements that were defined on some partial (Dirichlet) image regions with the prescribed displacement. The architecture of Linel2D-Net is described in Table 4. Figure 4 shows an example of automatically generated BVP, including boundary conditions and displacements, as well as FDM solutions for high ($\nu = 0$) and low ($\nu = 0.49$) compressible material models. For the training of forward Linel2D-Net models, three matrices (i.e., images) including (i) X-, (ii) Y-components of the displacement field as well as (iii) the type of boundary conditions cumulatively defining the BVP were used as the input. The output for training the forward model was defined by two matrices corresponding to (i) X- and (ii) Y-displacement components for

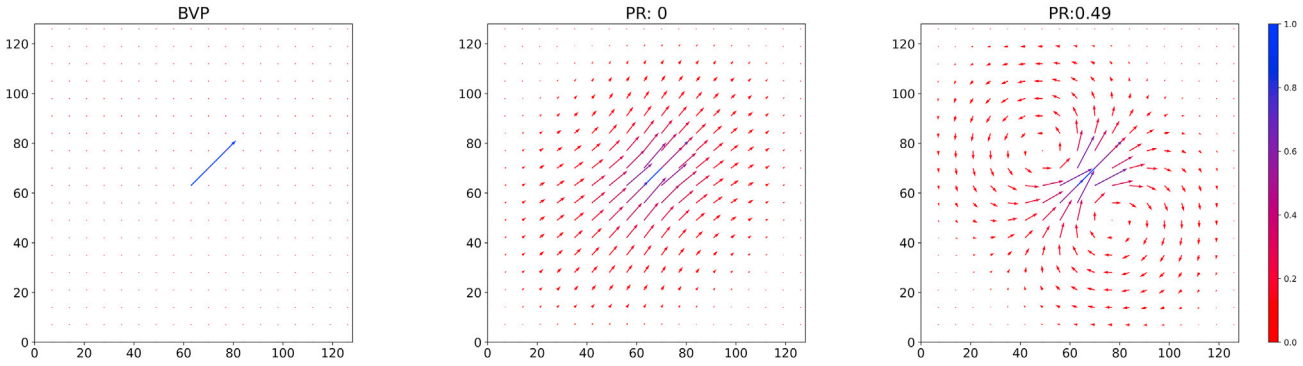


Figure 3. An example of FDM solutions or linear elastic BVP for two different values of the Poissons's ratio $\nu = 0$ and $\nu = 0.49$ corresponding to high- and low-compressible material approximation

From left to right: a BVP with prescribed boundary displacement in the middle of the image, FDM solution for the Poisson's ratio $\nu = 0$ (high compressibility) and $\nu = 0.49$ (low compressibility). As one can see, a higher degree of volume preservation in the case of low-compressible material leads to the emergence of distinctive vortex pattern of the image displacement. For the sake of better visualization, displacement vectors are visualized for every 5th pixel of the 128x128 image only.

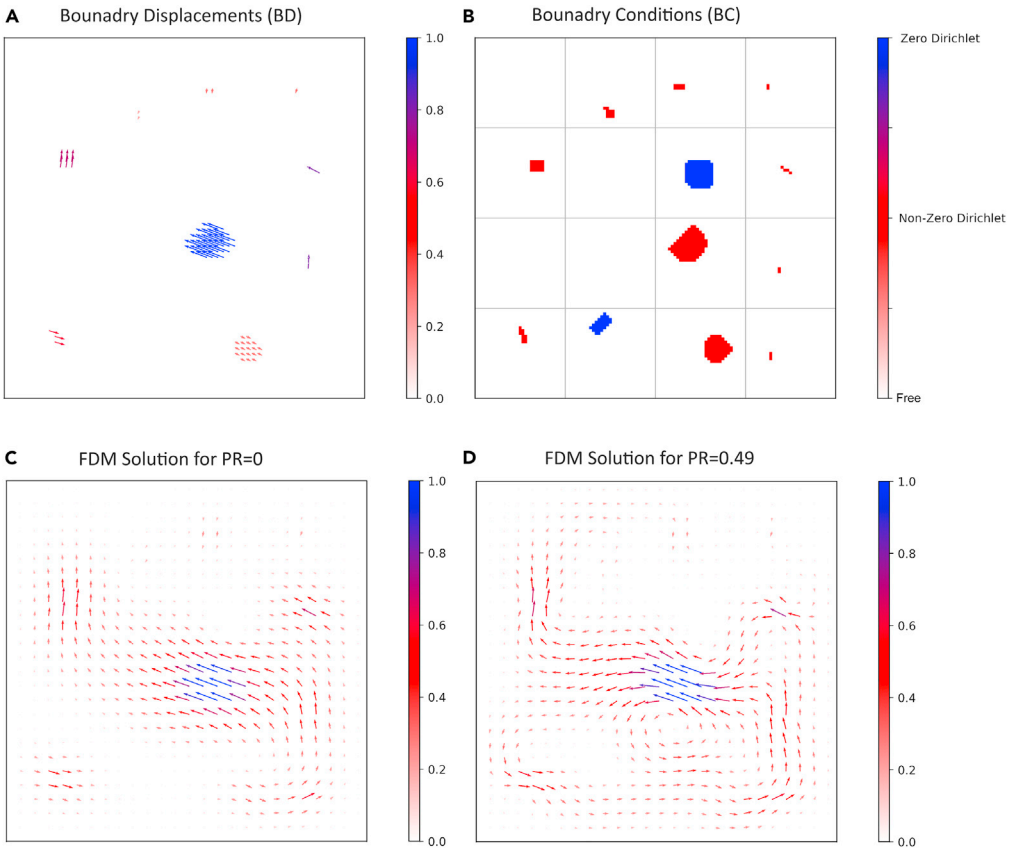


Figure 4. Examples of automatically generated linear elastic BVPs and their numerical solutions using FDM for different values of the Poisson's ratio $\nu = 0$ and $\nu = 0.49$

(A) Boundary displacements defined on Dirichlet pixels of the 128x128 image domain.

(B) Dirichlet regions generated by embedding random shapes and patterns into the grid of 16 32x32 cells (see gray lines).

(C and D) FDM solutions of the BVP (i.e., whole-image displacement) for high ($\nu = 0$) and low ($\nu = 0.49$) compressible material models.

all image pixels, which were then compared vs. the ground truth FDM solution. The models were evaluated against FDM predictions of "unseen" but principally similar BVPs as in the training set, i.e., random geometrical patterns of boundary conditions and corresponding displacements (i.e., patch-wise parallel displacement vectors on every Dirichlet region). Our experimental results show that the model predictions exhibit high similarity with the reference FDM solutions. Table 1 summarizes the performance of forward Linel2D-Net models by a direct comparison of model prediction against 10500 "unseen" FDM solutions, including mean average errors in the magnitude and spatial orientation of predicted displacements. Our results indicate that the low-compressible material model exhibits, in average a slightly higher deviation from the reference FDM solution compared to high-compressible material emulations. Figure 5 shows an example of the displacement field predicted by high- and low-compressible Linel2D-Net models compared to the reference FDM solution from Figure 4.

Inverse Linel2D-Net models

For training inverse Linel2D-Net models, two matrices, i.e., (i) X- and (ii) Y-components of the displacement from FDM solutions (rf. Figures 4C and 4D), were used as input for predicting (i) X- and (ii) Y-components of the boundary displacements as well as (iii) the boundary conditions (rf. Figures 4A and 4B). Consequently, the inverse model is implemented as a consistent reversal of the forward case. Thereby, components of

Table 1. Evaluation of forward Linel2D-Net models on 10500 FDM solutions of "unseen" BVPs

Models	MAE_{mag}	SD_{mag}	MAE_{dir}	SD_{dir}
Forward $\nu = 0.49$	0.007	0.0025	4.84	4.00
Forward $\nu = 0.25$	0.0052	0.0019	3.97	3.49
Forward $\nu = 0$	0.0066	0.0028	5.08	5.02

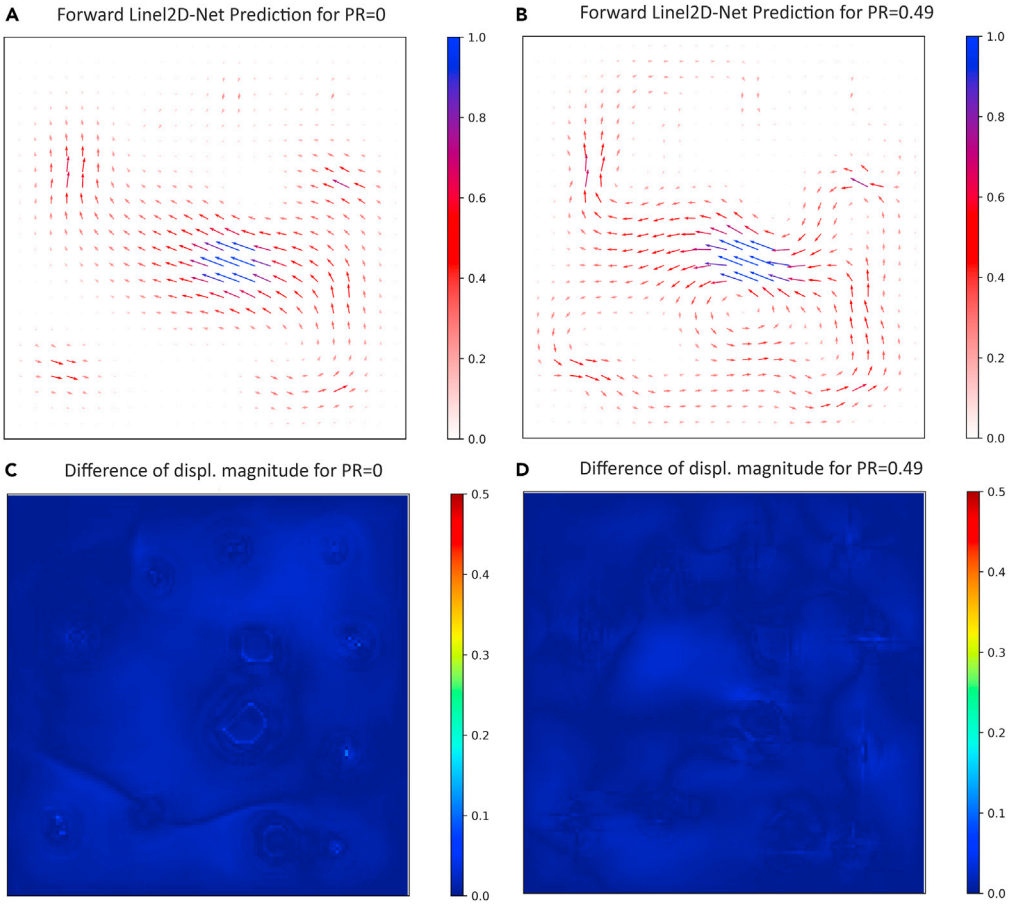


Figure 5. Examples of forward Linel2D-Net predictions for the BVP from the example pair of BVP - FDM solution shown in the previous figure.
(A and B) Whole-image displacements for high ($\nu = 0$) and low ($\nu = 0.49$) compressible material approximations for the BVP from Figure 4.
(C and D) Color map of differences between displacement magnitudes predicted by Linel2D-Net vs. FDM.

boundary displacements are sought as floating point numbers, while the boundary conditions in the output are predicted as probabilistic values ranging between 0 and 1 indicating the confidence score by the classification of two possible categories: prescribed (Dirichlet) or free boundary conditions. Table 2 summarizes the performance of inverse Linel2D-Net models by comparison vs. 10500 "unseen" BVPs including mean average errors in the magnitude and spatial orientation of predicted displacements of Dirichlet regions as well as the accuracy of the prediction of their geometrical location and shape. Compared to the performance of forward models, inverse Linel2D-Net exhibits higher deviations from original displacements, both with regarding to magnitude and spatial orientation. However, the inverse model is capable of consistently predict boundary conditions from the whole-image displacement with a high accuracy of over 99% for types of material models. Figure 6 shows examples of inverse predictions of boundary displacements and conditions from the whole-image displacement field which was previously computed using FDM as a solution of the forward BVP.

Verification of inverse predictions using forward finite difference

The results from Table 2 indicate that the inverse predictions exhibit high deviations from the ground truth BVP. However, it is well known that different boundary conditions can cause similar solutions of physical BVPs, especially with increasing distance from the source. To re-evaluate

Table 2. Evaluation of inverse Linel2D-Net models

Models	MAE_{mag}	SD_{mag}	MAE_{dir}	SD_{dir}	BC Accuracy [%]	BC F1 [%]
Inverse $\nu = 0.49$	0.014	0.013	4.04	6.10	99.87	99.93
Inverse $\nu = 0.25$	0.013	0.007	5.57	11.06	99.89	99.94
Inverse $\nu = 0$	0.0081	0.0046	5.89	9.91	99.89	99.94

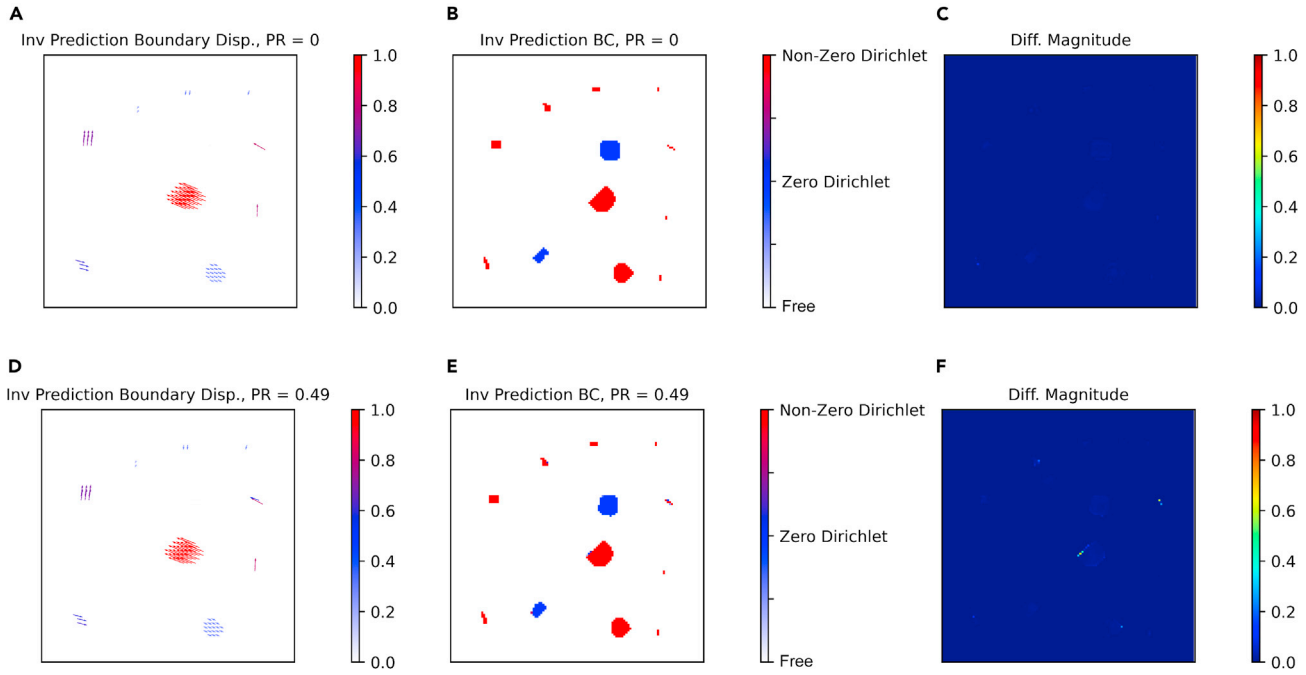


Figure 6. Examples of inverse Linel2D-Net predictions for the inverted BVP from Figure 4

(A and D) Boundary displacements predicted by the inverse Linel2D-Net model for high (ν = 0) and low (ν = 0.49) compressible material approximations.

(B and E) Boundary conditions predicted by the inverse Linel2D-Net model cf. Figure 4.

(C and F) Color maps of differences between displacement magnitudes predicted by Linel2D-Net vs. FDM.

the accuracy of predicted boundary conditions and displacement, they were again used as an input for the forward FDM scheme in order to predict the resulting displacement field for the entire image domain. The resulting displacement field was then compared vs. the FDM prediction using the original BVP. This verification study demonstrated a high similarity in the resulting whole-image displacements from both original and inversely predicted boundary conditions. The statistical analysis of this verification is summarized in Table 3. Figure 7 shows and example of the verification of whole-image displacement computed from inversely predicted vs. original BVP.

Linel2D-Net performance on completely new boundary value problems

In addition to previously "unseen" but principally similar BVPs from the set of automatically generated, random geometrical patterns and boundary displacements, Linel2D-Net was exemplary evaluated on completely different, manually generated and somewhat more realistic BVPs, see Figure 8. Figure 9 shows four examples of such BVPs including the cases of obstacle-free and obstacle containing elastic domains with two different values of the Poisson's ratio ν = 0 and ν = 0.49. Differently from fully automated random BVPs used for model training, these BVPs were drafted manually. As one can see the model is capable of reproducing the distinctive differences in behavior of high- and low-compressible materials (rf. Figure 8). However, predictions of the Linel2D-Net for low-compressible model approximation (ν = 0.49) are less accurate compared to predictions of the high-compressible model (ν = 0). Compared to the performance of Linel2D-Net on totally random BVPs, the predicted displacement field in the case of these structurally very different BVPs exhibits larger deviations from the reference FDM solution (MAE = 0.017, STD = 0.014). Thereby, predictions of the low-compressible material model exhibit larger deviations from the reference FDM solution compared to high-compressible material emulation using Linel2D-Net. Similar to the forward Linel2D-Net inverse prediction, it also exhibits higher deviation in the case of low-compressible material as shown in Figure 10. However, the geometrical pattern of boundary conditions could still be estimated in this example with a very high accuracy of 100% for high (ν = 0) and 99.93% for low (ν = 0.49) compressible material models.

Table 3. Evaluation of FDM scheme of inverse predictions compared against the ground truth FDM solutions

Models	MAE_{mag}	SD_{mag}	MAE_{dir}	SD_{dir}
Inverse ν = 0.49	0.0065	0.0036	2.01	1.37
Inverse ν = 0.25	0.0056	0.0031	1.13	1.36
Inverse ν = 0	0.0034	0.0018	2.21	1.85

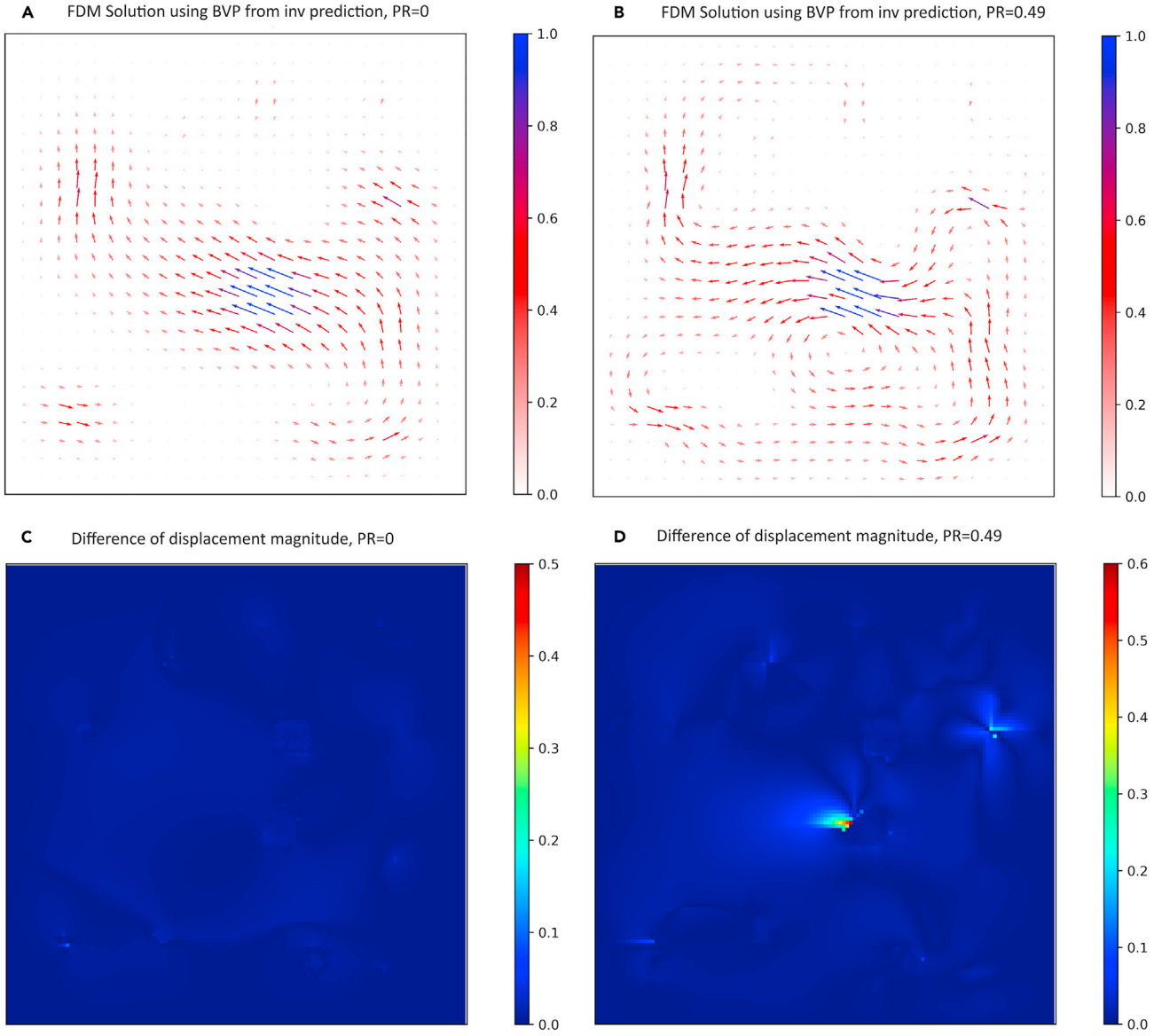


Figure 7. Evaluation of inverse Linel2D-Net predictions

(A and B) Whole-image displacement fields computed using FDM from boundary conditions that were predicted by the inverse Linel2D-Net model (cf. Figure 6). (C and D) Color maps of differences between displacement magnitudes predicted by the Linel2D-Net model vs. FDM for high ($\nu = 0$) and low ($\nu = 0.49$) compressible material approximations.

Performance of Linel2D-Net solver vs. finite difference

The performance of the Linel2D-Net vs. FDM solver was compared using 150 randomly selected BVPs. Generating solutions with the Linel2D-Net solver for 128x128 BVPs is 33x times faster than solving them with the FDM scheme. If the grid size is increased, computational time increases exponentially, which is not the case with the Linel2D-Net solver. The inverse FDM solutions are calculated from the displacement field of the entire image by inverting the stiffness matrix.

Example of inverse analysis of synthetic tissue deformation

To demonstrate the possible biomedical application of Linel2D-Net, an inverse analysis of synthetic 2D tissue deformation was performed. For this purpose, a synthetic pattern of 2D cell tissue layer was used, see Figure 11A. The boundary conditions were defined by the uniform displacements of five selected cells with the same magnitude but random spatial orientation, Figure 11B. Next, the displacement field for the whole image (cell pattern) was computed using the forward FDM simulation, Figure 11C as well as Video S1 in Supplementary Materials.

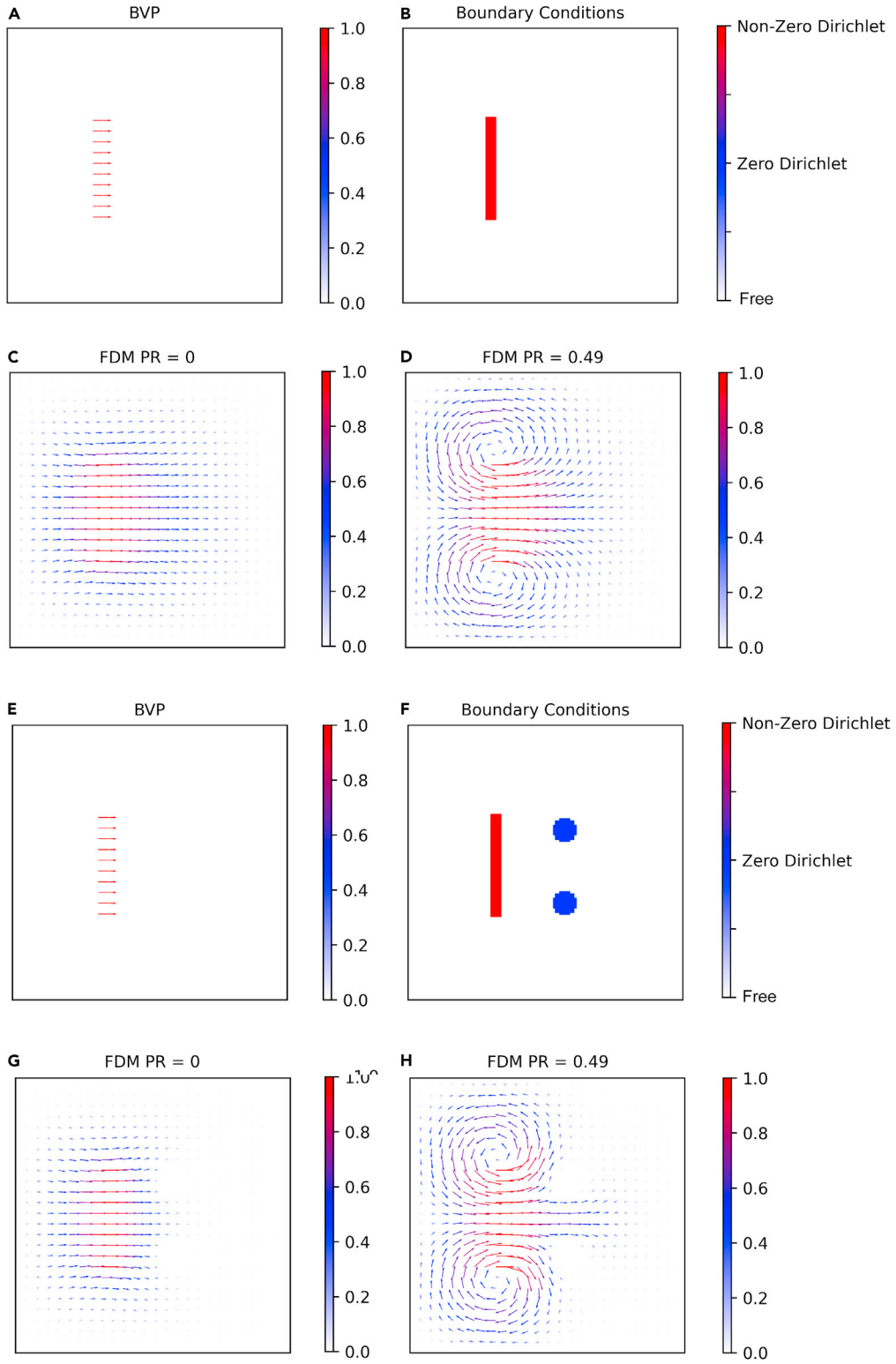


Figure 8. Examples of FDM solutions of linear elastic BVP with mixed boundary conditions for different values of the Poisson's ratio

(A and B) A BVP given by the horizontal displacement of the red bar within an otherwise obstacle-free elastic domain.

(C and D) FDM solutions of the BVP (a, b) for the Poisson's ratio $\nu = 0$ and $\nu = 0.49$.

(E and F) BVP given by the horizontal displacement of the red bar and two circle obstacles.

(G and H) FDM solutions of the BVP (E,F) for the Poisson's ratio $\nu = 0$ and $\nu = 0.49$.

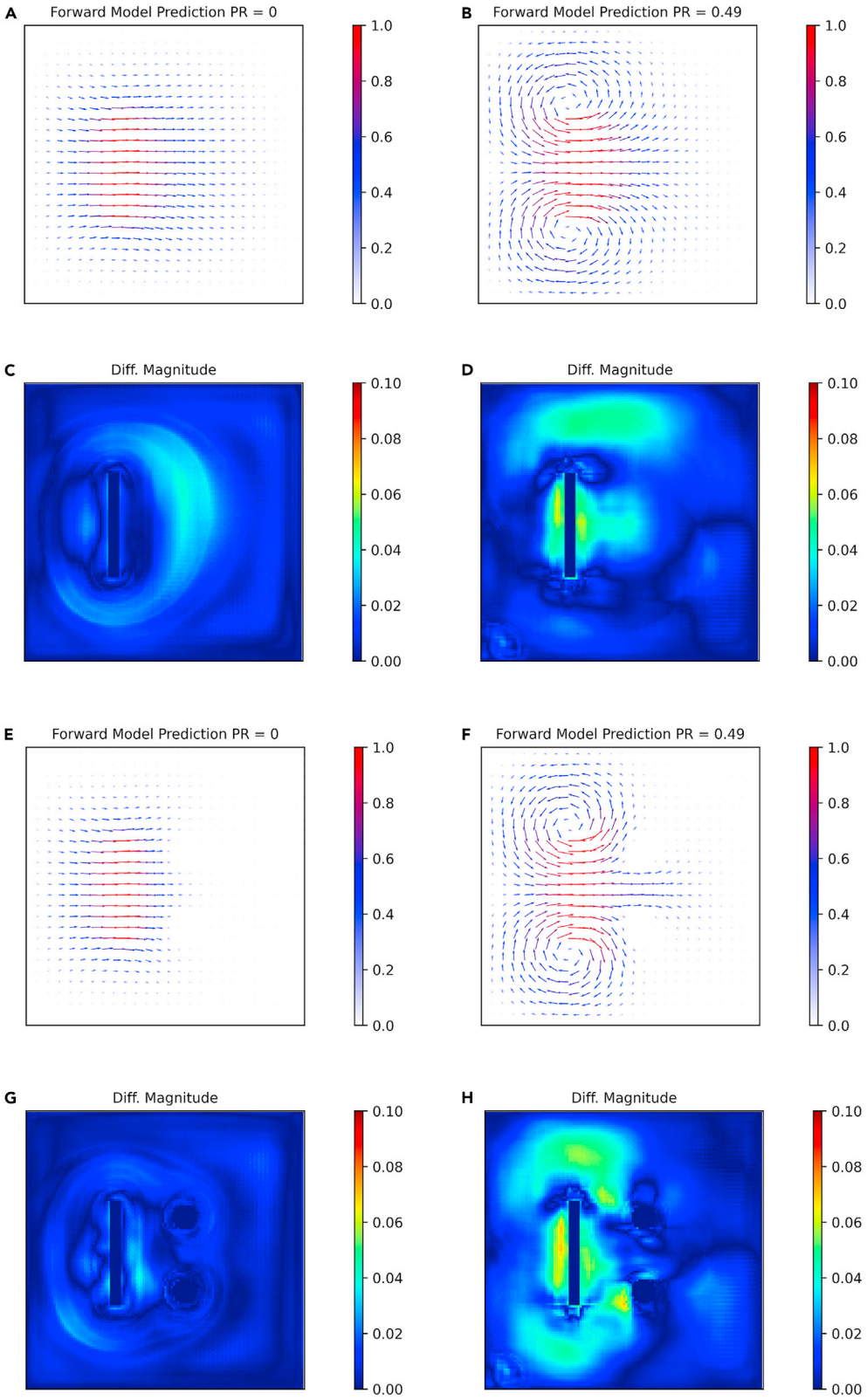

A
Forward Model Prediction PR = 0
1.0
0.8
0.6
0.4
0.2
0.0
B
Forward Model Prediction PR = 0.49
1.0
0.8
0.6
0.4
0.2
0.0
C
Diff. Magnitude
0.10
0.08
0.06
0.04
0.02
0.00
D
Diff. Magnitude
0.10
0.08
0.06
0.04
0.02
0.00
E
Forward Model Prediction PR = 0
1.0
0.8
0.6
0.4
0.2
0.0
F
Forward Model Prediction PR = 0.49
1.0
0.8
0.6
0.4
0.2
0.0
G
Diff. Magnitude
0.10
0.08
0.06
0.04
0.02
0.00
H
Diff. Magnitude
0.10
0.08
0.06
0.04
0.02
0.00

Figure 9. Examples of forward Linel2D-Net solutions for different BVPs and material properties

(A and B) Displacement vectors predicted as a solution of the BVP from Figures 8A and 8B by two Linel2D-Net models trained to emulate the behavior of high ($\nu = 0$) and low ($\nu = 0.49$) compressible elastic medium.

(C and D) Color map of absolute difference in the displacement magnitude between FDM (Figures 8C and 8D) and Linel2D-Net solutions (A and B) for $\nu = 0$ and $\nu = 0.49$.

(E and F) Displacement vectors predicted as a solution of the BVP from Figures 8E and 8F by two Linel2D-Net models trained to emulate the behavior of high ($\nu = 0$) and low ($\nu = 0.49$) compressible elastic medium.

(G and H) Color map of absolute difference in the displacement magnitude between FDM (Figures 8G and 8H) and Linel2D-Net solutions (E, F) for $\nu = 0$ and $\nu = 0.49$. As one can see the high compressibility ($\nu = 0.49$) model exhibits larger deviations from the reference FDM solutions compared to the low compressibility model ($\nu = 0$).

Finally, the inverse Linel2D-Net model was applied for the reconstruction of the boundary conditions from the whole image displacement field. As a result of the inverse Linel2D-Net prediction, the correct location and area of five cells as well as their displacements, were estimated. Figure 11D shows the color map of differences between the magnitude of the original (Figure 11B) and inversely predicted boundary displacements of five selected cells. This feasibility example demonstrates that the inverse Linel2D-Net model can be used for the contact-free estimation of material properties of target structures (for example, biological tissues) by the observation of their deformation in image time-series. The video depicting the deformation of a few selected cells causing the whole medium to deform is provided in Video S1.

Linel2D-Net demo tool

The Linel2D-Net solver is available as a pre-compiled demo tool along with the example data and brief user guidelines from https://ag-ba.ipk-gatersleben.de/Linel2DNet.html.

DISCUSSION

In this study, we presented a surrogate neural network approach to solving arbitrary 2D linear elastic boundary value problems defined on a regular grid (image) domain. Differently from the majority of previously published works, our approach is based on training CNN models on a combination of not only physically but also geometrically variable sets of reference numerical solutions. Our results show that such surrogate model solve forward as well as inverse linear elastic BVPs, which are significantly more efficient but still sufficiently accurate compared to the conventional iterative techniques (here: FDM, in particular). On 128x128 images, the Linel2D-Net achieves a nearly real-time performance and does not require additional domain discretization, which makes it an attractive tool for straightforward application to high-content high-throughput model-based image analysis. In the case of forward BVPs, the Linel2D-Net models can reproduce linear elastic material behavior with a very high accuracy which in the case of deformable imaging applications, is practically indistinguishable from reference FDM solutions. Further, the spectrum of prescribed boundary value includes zero as well as non-zero displacement, which allows the solving of BVPs including combinations of moving and fixed (obstacle) boundary structures. Our strategy of handling forward models is handy in applications which requires high-performance real-time simulations that are highly demanded in machine vision systems, digital-twins, and so forth. In the case of inverse modeling, the Linel2D-Net models are not as accurate as the forward Linel2D-Net models in reconstructing the boundary displacements. However, it is capable of reconstructing the BVPs that reproduce almost the same solution for the entire domain as the original BVPs. While the reconstruction of boundary displacements undergoes stronger deviations from the original BVPs, boundary conditions can be reconstructed with a surprisingly high accuracy of more than 99%. Rapid discovery of boundary displacements and conditions from observed linear elastic deformations of the whole-image domain, could be of great interest for a large manifold of reverse engineering problems, for example, in biomedical imaging, where the problem of the inverse reconstruction of biological triggers frequently emerges. Both forward and the inverse models exhibit high accuracy by comparison to the reference FDM solutions crossover all types of materials, whereby the largest deviations were observed for low-compressible material approximation with the Poisson's ratio $\nu = 0.49$. Our models appear to perform well with examples that are not a part of the training set, however, BVPs with extremely deviating boundary conditions are associated with larger errors in model predictions. The major limitation of the present implementation is the restriction to a relatively small and fixed image size (128x128), even though, it can still be applied to larger image domains by means of subsampling to the 128x128 grid followed by the subsequent interpolation of the solution on higher resolution images. While the focus of the present study was put on modeling linear elastic, quasi-geometric BVPs, extensions to more complex non-linear elastic, non-homogeneous materials and/or a larger set of BVPs including the estimation of domain forces, stress and not only displacements are principally possible and straightforward in implementation using the same data-driven approach. For applications demanding a significantly finer resolution of large spatial domains or irregular domain discretization (e.g., arbitrary point clouds) alternative approaches such as those suggested in[53] should be taken into consideration.

Limitations of the study

Both forward and inverse Linel2D-Net models were trained with and can process only image domain of the size 128x128. The image of any other resolutions has to be rescaled to 128x128 or the FDM ground truth solutions have to be regenerated for retraining the DNN model on the new domain size completely anew including both forward and inverse BVPs. Especially for significantly larger domains, it can become a computationally expensive task. By retraining DNN models on the new domain size, the hyperparameters of the DNN model may have to be readjusted as well as in order to obtain the optimal results of model predictions.

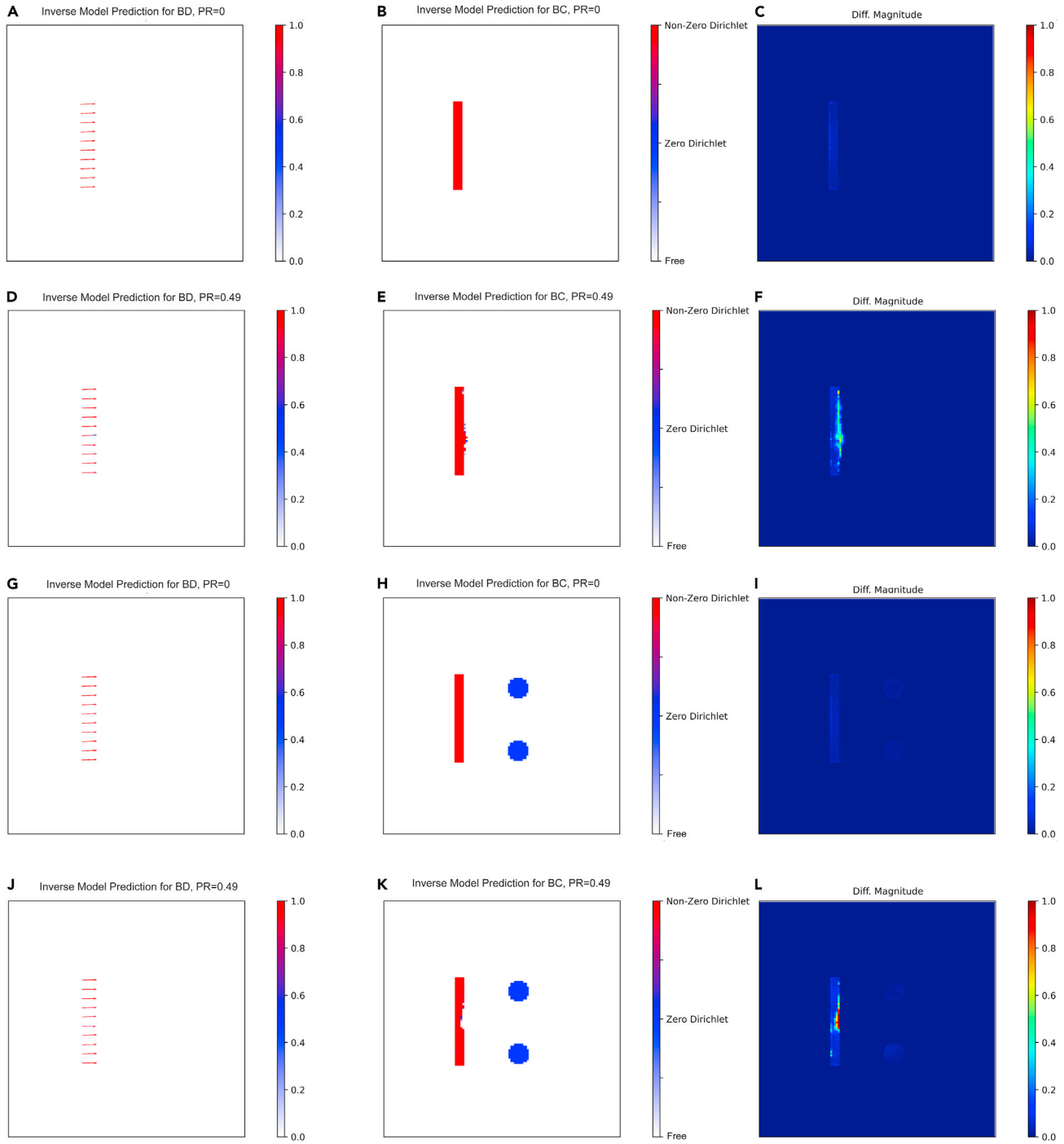


Figure 10. Examples of inverse Linel2D-Net solutions for different BVPs and material properties

Left and middle columns: boundary displacements and boundary conditions predicted by the inverse Linel2D-Net model from the whole-image displacement fields in Figures 8C, 8D, 8G, and 8H for two different values of the Poisson's ratio ($\nu = 0$) and low ($\nu = 0.49$).

(A, D, G, and J) boundary displacements, (B, E, H, and K) boundary condition predicted by Linel2D-Net and (C, F, I, and L) color maps of differences in the absolute displacement magnitude between the original boundary displacements and the boundary displacement predicted by the inverse Linel2D-Net. Similar to observations in Figure 9, the low compressibility ($\nu = 0.49$) inverse model exhibits larger deviations from the reference FDM solutions compared to the high compressibility model ($\nu = 0$).

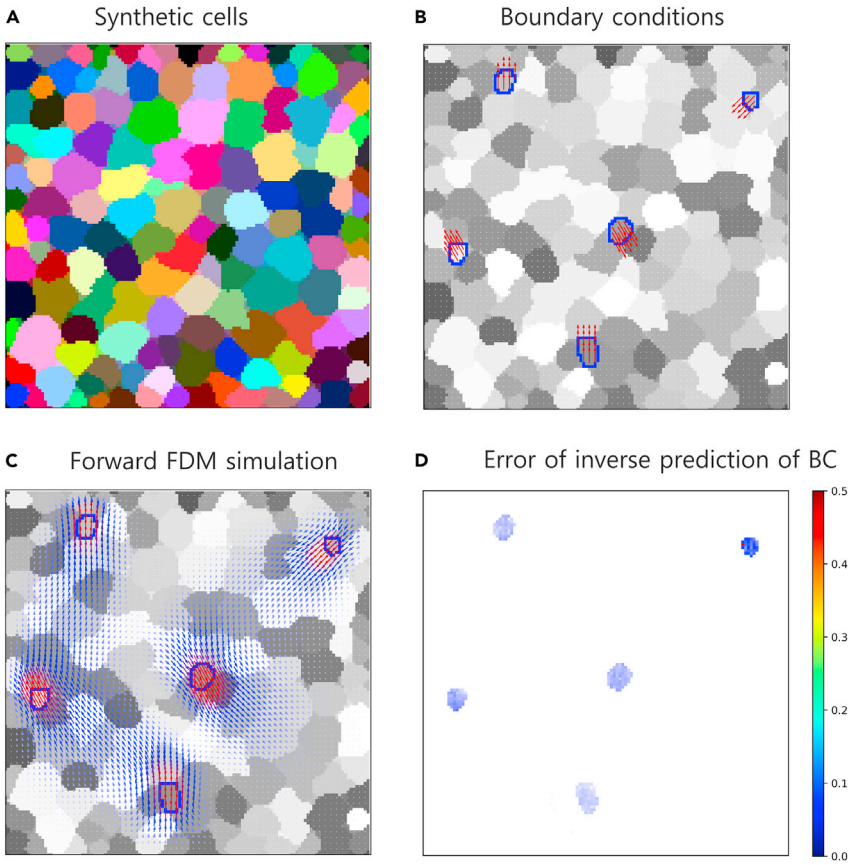


Figure 11. Example of inverse analysis of cell tissue proliferation using Linel2D-Net

(A) 2D pattern of synthetic 2D cell culture, (B) boundary conditions given by displacements of five selected cells, (C) forward simulation of the whole image/tissue displacement using FDM (see also quasi-dynamic visualization in Video S1), (D) color map of differences between the original (B) and Linel2D-Net predicted boundary displacements.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - 2D Lamé-Navier PDE of the linear elasticity

Table 4. Comparison of parameters of Linel2D-Net vs. original U-Net

Parameters	Proposed DNN model	U-Net
Contracting path kernels	7x7	3x3
Expansive path kernels	3x3	2x2
Stride	2x2	1x1
Padding	Unpadded	padded with zeros
Depth	2	4
Filters	(24, 48, 96, 192, 384,768,1536)	(64, 128, 256, 512, 1024)

- Fundamental solution of the 2D Lamé-Navier PDE
- Numerical solution of the 2D Lamé-Navier PDE using FDM
- Generation of BVP image datasets
- Reformulation and implementation of U-Net as a surrogate DNN solver
- Forward and inverse DNN solvers
- Training DNN models
- Validation of model predictions

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2024.109519.

ACKNOWLEDGMENTS

This work was supported by the German Federal Ministry of Education and Research (BMBF) within the scope of the AVATARS project (FKZ 031B0770A).

AUTHOR CONTRIBUTIONS

ANMA performed the computational experiments, analyzed the data, wrote the article, prepared figures, and tables, and reviewed drafts of the article. NN reviewed the drafts of the article. EG conceived and supervised investigations, performed computational analysis, prepared figures, and wrote and reviewed the article. All authors agreed on the article in its present form.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: January 4, 2024
Revised: February 2, 2024
Accepted: March 14, 2024
Published: March 18, 2024

REFERENCES

1. Gladilin, E., Micoulet, A., Hosseini, B., Rohr, K., Spatz, J., and Eils, R. (2007). Finite element analysis of uniaxial cell stretching: From image to insight. Phys. Biol. *4*, 104–113.
2. Gladilin, E., Schulz, M., Kappel, C., and Eils, R. (2010). Contactless determination of nuclear compressibility by means of 3D image- and model-based analysis of drug-induced cellular deformation. J. Microsc. *240*, 216–226.
3. Gonzalez Avalos, P., Reichenzeller, M., Eils, R., and Gladilin, E. (2011). Probing compressibility of the nuclear interior in wild-type and lamin deficient cells using microscopic imaging and computational modeling. J. Biomech. *44*, 2642–2648.
4. Wyczalkowski, M.A., Chen, Z., Filas, B.A., Varner, V.D., and Taber, L.A. (2012). Computational models for mechanics of morphogenesis. Birth Defects Res. C Embryo Today. *96*, 132–152.
5. Gladilin, E., Eils, R., and Peshkin, L. (2015). On the embryonic cell division beyond the contractile ring mechanism: Experimental and computational investigation of effects of vitelline-confinement, temperature and egg size. PeerJ *3*, e1490.
6. Boudon, F., Chopard, J., Ali, O., Gilles, B., Hamant, O., Boudaoud, A., Traas, J., and Godin, C. (2015). A computational framework for 3D mechanical modeling of plant morphogenesis with cellular resolution. PLoS Computat. Biol. *11*, e1003950.
7. Hamant, O., and Traas, J. (2010). The mechanics behind plant development. New Phytol. *185*, 369–385. https://doi.org/10.1111/j.1469-8137.2009.03100.x.
8. Sotiras, A., Davatzikos, C., and Paragios, N. (2013). Deformable medical image registration: A survey. IEEE Trans. Med. Imag. *32*, 1153–1190.
9. Wozniak, M.A., and Chen, C.S. (2009). Mechanotransduction in development: A growing role for contractility. Nat. Rev. Mol. Cell Biol. *10*, 34–43.
10. Wang, N. (2017). Review of cellular mechanotransduction. J. Phys. D Appl. Phys. *50*, 233002. https://doi.org/10.1088/1361-6463/aa6e18.
11. Maas, S.A., Ateshian, G.A., and Weiss, J.A. (2017). FEBio: History and advances. Annu. Rev. Biomed. Eng. *19*, 279–299.
12. Trew, M.L., Smaill, B.H., Bullivant, D.P., Hunter, P.J., and Pullan, A.J. (2005). A generalized finite difference method for modeling cardiac electrical activation on arbitrary, irregular computational meshes. Math. Biosci. *198*, 169–189.
13. Mackerle, J. (1992). Finite and boundary element methods in biomechanics: a bibliography (1976–1991). Eng. Comput. *9*, 403–435.
14. Zhang, L.W., Ademiloye, A.S., and Liew, K.M. (2018). Meshfree and particle methods in biomechanics: Prospects and challenges. Arch. Comput. Methods Eng. *26*, 1547–1576.
15. Mang, A., Bakas, S., Subramanian, S., Davatzikos, C., and Biros, G. (2020). Integrated biophysical modeling and image analysis: Application to neuro-oncology. Annu. Rev. Biomed. Eng. *22*, 309–341.
16. Wu, C., Lorenzo, G., Hormuth, D., Lima, E., Slavkova, K., DiCarlo, J., Virostko, J., Phillips, C., Patt, D., Chung, C., and Yankeelov, T.E. (2022). Integrating mechanism-based modeling with biomedical imaging to build practical digital twins for clinical oncology. Biophys. Rev. *3*.
17. Niroomandi, S., Alfaro, I., Cueto, E., and Chinesta, F. (2010). Model order reduction for hyperelastic materials. Int. J. Numer. Methods Eng. *81*, 1180–1206.
18. Barbič, J., and James, D.L. (2005). Real-time subspace integration for st. Venant-kirchhoff deformable models. ACM Trans. Graph. *24*, 982–990.
19. An, S.S., Kim, T., and James, D.L. (2009). Optimizing cubature for efficient integration of subspace deformations. ACM Trans. Graph. *27*, 165–210.
20. Goury, O., and Duriez, C. (2018). Fast, generic, and reliable control and simulation of soft robots using model order reduction. IEEE Trans. Robot. *34*, 1565–1576.
21. Chaturantabut, S., and Sorensen, D.C. (2009). Discrete empirical interpolation for nonlinear model reduction. In Proceedings of the 48h IEEE conference on decision and control (CDC) held jointly with 2009 28th chinese control conference (IEEE), pp. 4316–4321.
22. Bui-Thanh, T., Willcox, K., and Ghattas, O. (2008). Model reduction for large-scale systems with high-dimensional parametric input space. SIAM J. Sci. Comput. *30*, 3270–3288.
23. Comas, O., Taylor, Z.A., Allard, J., Ourselin, S., Cotin, S., and Passenger, J. (2008). Efficient

nonlinear FEM for soft tissue modelling and its GPU implementation within the open source framework SOFA. In International symposium on biomedical simulation (Springer), pp. 28–39.
24. Johnsen, S.F., Taylor, Z.A., Clarkson, M.J., Hipwell, J., Modat, M., Eiben, B., Han, L., Hu, Y., Mertzanidou, T., Hawkes, D.J., and Ourselin, S. (2015). NiftySim: A GPU-based nonlinear finite element package for simulation of soft tissue biomechanics. Int. J. Comput. Assist. Radiol. Surg. *10*, 1077–1095.
25. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. Nat. Rev. Phys. *3*, 422–440.
26. Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat. (2019). Deep learning and process understanding for data-driven earth system science. Nature *566*, 195–204.
27. Alber, M., Buganza Tepole, A., Cannon, W.R., De, S., Dura-Bernal, S., Garikipati, K., Karniadakis, G., Lytton, W.W., Perdikaris, P., Petzold, L., and Kuhl, E. (2019). Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. NPJ Digit. Med. *2*, 115–211.
28. Iten, R., Metger, T., Wilming, H., Del Rio, L., and Renner, R. (2020). Discovering physical concepts with neural networks. Phys. Rev. Lett. *124*, 010508.
29. Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2017). Physics informed deep learning (part I): Data-driven solutions of nonlinear partial differential equations. CoRR abs/1711.10561.
30. Kissas, G., Yang, Y., Hwuang, E., Witschey, W.R., Detre, J.A., and Perdikaris, P. (2020). Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks. Comput. Methods Appl. Mech. Eng. *358*, 112623.
31. Fetene, B.N., Shufen, R., and Dixit, U.S. (2018). FEM-based neural network modeling of laser-assisted bending. Neural Comput. Appl. *29*, 69–82.
32. Roewer-Després, F., Khan, N., and Stavness, I. (2018). Towards finite-element simulation using deep learning. In 15th international symposium on computer methods in biomechanics and biomedical engineering.
33. Meister, F., Passerini, T., Mihalef, V., Tuysuzoglu, A., Maier, A., and Mansi, T. (2018). Towards fast biomechanical modeling of soft tissue using neural networks. Preprint at arXiv. https://doi.org/10.48550/ARXIV.1812.06186.
34. Luo, R., Shao, T., Wang, H., Xu, W., Chen, X., Zhou, K., and Yang, Y. (2020). NNWarp: Neural network-based nonlinear deformation. IEEE Trans. Vis. Comput. Graph. *26*, 1745–1759.
35. Odot, A., Haferssas, R., and Cotin, S. (2022). DeepPhysics: A physics aware deep learning framework for real-time simulation. Int. J. Numer. Methods Eng. *123*, 2381–2398.
36. Cai, S., Liang, J., Gao, Q., Xu, C., and Wei, R. (2020). Particle image velocimetry based on a deep learning motion estimator. IEEE Trans. Instrum. Meas. *69*, 3538–3554.
37. Mendizabal, A., Márquez-Neila, P., and Cotin, S. (2020). Simulation of hyperelastic materials in real-time using deep learning. Med. Image Anal. *59*, 101569.
38. Lu, L., Meng, X., Mao, Z., and Karniadakis, G.E. (2021). DeepXDE: A deep learning library for solving differential equations. SIAM Rev. *63*, 208–228.
39. Hennigh, O., Narasimhan, S., Nabian, M.A., Subramaniam, A., Tangsali, K., Fang, Z., Rietmann, M., Byeon, W., and Choudhry, S. (2021). NVIDIA SimNet™: An AI-accelerated multi-physics simulation framework. In International conference on computational science (Springer), pp. 447–461.
40. Chen, F., Sondak, D., Protopapas, P., Mattheakis, M., Liu, S., Agarwal, D., and Di Giovanni, M. (2020). Neurodiffeq: A python package for solving differential equations with neural networks. J. Open Source Softw. *5*, 1931.
41. Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention (MICCAI) LNCS (Springer), pp. 234–241.
42. Thuerey, N., Weißenow, K., Prantl, L., and Hu, X. (2020). Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. AIAA J. *58*, 25–36.
43. Kapoor, S., Mianroodi, J.R., Khorrami, M., Siboni, N.S., and Svendsen, B. (2022). Comparison of two artificial neural networks trained for the surrogate modeling of stress in materially heterogeneous elastoplastic solids. Preprint at arXiv. https://doi.org/10.48550/arXiv.2210.16994.
44. Gómez Lera, S., and Alarcón Álvarez, E. (1987). Elastostatics. In Boundary element methods in mechanics Computational methods in mechanics, D.E. Beskos, ed., pp. 107–189.
45. Gladilin, E. (2003). Biomechanical Modeling of Soft Tissue and Facial Expressions for Craniofacial Surgery Planning.
46. Ioffe, S., and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (PMLR), pp. 448–456.
47. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Preprint at arXiv. https://doi.org/10.48550/arXiv.1603.04467.
48. van der Walt, S., Colbert, S.C., and Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. Comput. Sci. Eng. *13*, 22–30.
49. Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., and Yu, T.; scikit-image contributors (2014). Scikit-image: Image processing in python. PeerJ *2*, e453.
50. Bradski, G. (2000). The opencv library. Dr. Dobb's Journal: Software Tools for the Professional Programmer. *25*, 120–123.
51. Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097–1105.
52. Kingma, D.P., and Ba, J. (2015). Adam: A method for stochastic optimization. CoRR abs/1412.6980.
53. Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. (2020). Fourier neural operator for parametric partial differential equations. Preprint at arXiv. https://doi.org/10.48550/arXiv.2010.08895.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Example data & compiled models	This paper	Image Analysis Group, IPK Gatersleben. https://ag-ba.ipk-gatersleben.de/Linel2DNet.html
Software and algorithms		
Tensorflow	Open source package for machine learning and AI from Google	https://www.tensorflow.org/
Keras	Python package	https://keras.io/
Scikit-learn	Python package	https://scikit-learn.org/stable/
Python	Python Software Foundation	https://www.python.org/
Matplotlib	Python package	https://matplotlib.org/
Numpy	Python package	https://numpy.org/

RESOURCE AVAILABILITY

Lead contact

Further Information and requests for resources should be directed to and fulfilled by the lead contact, Evgeny Gladilin (email: gladilin@ipk-gatersleben.de).

Materials availability

This study did not generate new unique materials.

Data and code availability

- This paper does not report original code
- The examples of FDM data and compiled models used in the study have been mentioned in the key resources table. Additional information will be available from the lead contact upon request.

METHOD DETAILS

2D Lamé-Navier PDE of the linear elasticity

In this work, we focus on a linear elastic material approximation which is described by the linear stress-strain relationship

$$\boldsymbol{\sigma}(\boldsymbol{\varepsilon}) = C\boldsymbol{\varepsilon}, \quad \text{(Equation 1)}$$

where σ is the stress tensor, ε is the strain tensors and C is the tensor of elastic constants. In the case of homogeneous and isotropic material, Equation 1 can be explicitly written as

$$\boldsymbol{\sigma}(\boldsymbol{\varepsilon}) = \lambda\, tr(\boldsymbol{\varepsilon})\, I + 2\,\mu\,\boldsymbol{\varepsilon}, \quad \text{(Equation 2)}$$

where λ and μ are the so-called Lamé material constants. In the range of relatively low strains (i.e. typically $\max(\varepsilon) < 0.05$), the strain tensor is the linear function of displacement

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right), \quad \text{(Equation 3)}$$

and Equation 1 takes the form a partial differential equation with respect to the displacement (the so-called Lamé-Navier PDE)

$$\mu\, \Delta\, \boldsymbol{u} + (\lambda + \mu)\, grad\, div\, \boldsymbol{u} = -f \quad \text{(Equation 4)}$$

where $\mathbf{u}$ is the displacement vector, f is the 'right-hand side' vector of force density acting on the 2D spatial domain. Alternatively, to the pair Lamé constants (λ, μ), another two material parameters, - the Young's modulus or E and the Poisson's ratio or ν describing the material stiffness and compressibility, are more frequently used in material science:

$$\nu = \frac{\lambda}{2(\lambda+\mu)}, E = \frac{\mu(3\lambda+2\mu)}{\lambda+\mu} \quad \text{(Equation 5)}$$

In the case of a one-material problem and boundary forces that are given implicitly as prescribed boundary displacements (i.e. $\mathbf{f} = 0$ and $\mathbf{u} = \mathbf{u}_{known} + \mathbf{u}_{unknown}$), Equation 4 takes a quasi-geometric form

$$\Delta \boldsymbol{u} + \frac{1}{1-2\nu}\, grad\ div\ \boldsymbol{u} = 0 \quad \text{(Equation 6)}$$

with the displacement u as a single variable and the Poisson's ratio ν as single material parameter. Non-trivial solutions of Equation 6 exist when prescribed displacements on the inner of outer boundaries of the material domain are defined. Boundary conditions defined by prescribed displacements on some parts of the material domain are known as Dirichlet boundary conditions. Regions with no prescribed boundary displacements are termed as Free boundary conditions. The problem of finding unknown values of target variables (here: XY components of the 2D displacement vector) by solving Equation 4 for the given boundary conditions is known as the Boundary Value Problem (BVP).

Fundamental solution of the 2D Lamé-Navier PDE

Analytical solutions of Equation 4 are known only for some special cases with particularly simple and symmetric spatial domains and boundary conditions. One such special solution is known for the case of an infinite medium under the impact of the (Dirac-impulse shaped) point force density,

$$\boldsymbol{f}(\boldsymbol{r}) = \delta(\boldsymbol{r} - \boldsymbol{r}') \quad \text{(Equation 7)}$$

where $\mathbf{r}'$ is the radius vector pointing to the source of force application and $\mathbf{r}$ is the radius vector of any other point of the infinite elastic medium. The solution of Equation 4 for the case of the Dirac-shaped right hand side is known as the fundamental solution of 2D Lamé-Navier PDE and is given by[44]

$$u_{ij} = \frac{1}{8\pi E(1-\nu)}\left[(3-4\nu)\, ln\left(\frac{1}{r}\right)\delta_{ij} + r_i\, r_j\right] \quad \text{(Equation 8)}$$

where $\mathbf{r} = \mathbf{r}(x, y)$ is the radius vector pointing to 2D coordinates (x, y) of the elastic medium and δ_{ij} denotes the Kroneker delta symbol ($\delta_{ij} = 0$ if $i \neq j$ and $\delta_{ij} = 1$ if $i = j$). The fundamental solution describes an abstract case of an infinite medium, however, the composition of displacements within a finite subdomain of the infinite elastic continuum is of practical interest for this study as it can be used for evaluation of numerical solutions.[45]

Numerical solution of the 2D Lamé-Navier PDE using FDM

Equation 6 with arbitrary boundary conditions can be solved using numerical methods only. In this work, the Finite Different Method on the regular 2D image grid is applied for this purpose. In particular, the FDM approximates the derivatives by differences between the variable values between neighbor grid nodes, i.e.,

$$\begin{gathered}\frac{du}{dx} = \frac{u(i+1) - u(i)}{x(i+1) - x(i)}\,. \\ \frac{d^2u}{dx^2} = \frac{u(i+1) - 2u(i) + u(i-1)}{(x(i+1) - x(i))^2}\,. \\ \frac{d^2u}{dxdy} = \frac{u(i+1,j+1) - u(i+1,j-1) - u(i-1,j+1) + u(i-1,j-1)}{4(x(i+1) - x(i))(y(i+1) - y(i))}\end{gathered} \quad \text{(Equation 9)}$$

where (i, j) are indices of the nodes in the Euclidean system of image pixel coordinates (x,y). In the case of a quasi-geometrical (pure displacement), linear elastic plane strain problem (2D) the FD scheme of the Lamé-Navier PDE takes the following form

$$\begin{aligned}&2(1-\nu)\left[\frac{u_x(i+1,j) - 2u_x(i,j) + u_x(i-1,j)}{\Delta x^2}\right] + (1-2\nu)\left[\frac{u_x(i,j+1) - 2u_x(i,j) + u_x(i,j-1)}{\Delta y^2}\right] \\ &\quad + \left[\frac{u_y(i+1,j+1) - u_y(i+1,j-1) - u_y(i-1,j+1) + u_y(i-1,j-1)}{4\Delta x\Delta y}\right] = 0, \\ &2(1-\nu)\left[\frac{u_y(i,j+1) - 2u_y(i,j) + u_y(i,j-1)}{\Delta y^2}\right] + (1-2\nu)\left[\frac{u_y(i+1,j) - 2u_y(i,j) + u_y(i-1,j)}{\Delta x^2}\right] \\ &\quad + \left[\frac{u_x(i+1,j+1) - u_x(i+1,j-1) - u_x(i-1,j+1) + u_x(i-1,j-1)}{4\Delta x\Delta y}\right] = 0\end{aligned} \quad \text{(Equation 10)}$$

The FDM scheme transforms the continuous second-order 2D Lamé-Navier PDE to the linear system of equations with respect to unknown values of xy-components of the displacement:

$$A\,u = b \quad \text{(Equation 11)}$$

where A is a sparse symmetric and positive definite matrix (the so-called stiffness matrix), b is the right-hand side vector resulting from the implementation of known boundary conditions, and u is the vector of unknown displacements. Further details on the FDM discretization scheme for 2D Lamé-Navier PDE and its numerical implementation can be found in Supplementary Materials. In the case of a single-material problem the Equation 6 contains only one material parameter, - the Poisson's ratio ($\nu \in [0,0.5]$) describing the material compressibility. In this work, three distinctive values of $\nu = 0, 0.25, 0.49$ were chosen for the simulation of high, moderate, and low compressible material behavior, respectively.

Generation of BVP image datasets

A large set of pair-wise BVP and FDM solutions of 2D Lamé-Navier PDE with various boundary conditions is required to train an unbiased DNN model. For this purpose, a set of totally 10500 BVPs with different geometrical patterns of boundary conditions and prescribed displacements values is generated. To avoid potential biases due to particular spatial orientation and location of automatically generated patterns, BVP images are subjected to augmentation by geometrical transformations including image rotations and displacements. In order to normalize different BVPs, the maximum value of the displacement magnitude was set to one. Accordingly, the components of the displacement vector were calculated as follows,

$$\begin{aligned} u_x &= |u| \cos(\alpha) \\ u_y &= |u| \sin(\alpha), \end{aligned}$$

where $|u| = \sqrt{u_x^2+u_y^2}$ is the displacement magnitude and the angle of spatial orientation α takes random values between 0 and 2π which is again assigned uniformly for all displacement in the subdomains. The BVPs are assigned with displacement of variable magnitude. Thereby, the zero displacement magnitude corresponds to the cases of fixed regions (obstacles) that do not move but still have an effect on the surrounding displacement field. Three different datasets of the same size 10500 BVPs corresponding to three distinctive values of the Poisson's ratio were generated for training models.

Reformulation and implementation of U-Net as a surrogate DNN solver

Our approach to the solution of linear elastic BVPs is based on the U-Net architecture which has been widely used for image segmentation tasks.[41] The original U-Net architecture was modified to enable the prediction of 2D displacements within a finite domain where the neural network acts as a surrogate model instead of numerical solvers. For this task, the neural network models have to be trained in such a way that the unknown function q_i should match the ground truth q_i^*. Adapt the original U-Net to the task, a few modifications have been introduced in this work. The batch normalization replaces the dropout layer in our study which fastens the training process and improves stability by standardizing the input distribution for every mini-batch. Batch normalization helps avoid internal covariance shifts caused by the change in the input distribution to the model every time which makes the hidden layers to adapt the new distribution.[46] This shift affects the time taken for the model to reach the global minimum thereby reducing the speed of the training process. The *relu* activation function is used across the hidden layers and in the final layer *tanh* activation function is used to map the values between -1 and 1. U-Net by default works with unsigned integer values, in our work we have changed the setting to floating point values so that the neural network can handle the decimal as well as negative values. Since the problem is defined as a prediction problem, the loss function used is the mean squared error given as

$$L_{\text{MSE}} = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left((q_i;\theta) - q_i^* \right)^2 \quad \text{(Equation 12)}$$

where q_i^* is the ground truth value on the i-th image pixel, q_i is the predicted value for the given network parameter θ and N is the number of image pixels. The modifications of the original U-Net architecture used in this work are summarized in Table 4.

Forward and inverse DNN solvers

Two basic types of problems are known in computational physics. Forward problems consist in finding a solution for the entire domain from some boundary conditions typically given on partial domain boundaries. Accordingly, training of the forward model is performed by passing boundary conditions and prescribed boundary displacements defined on the image grid (i.e. BVP images) as input and FDM solutions of these BVPs to the output of the U-Net. Consequently, our approach is purely data-driven and does not rely on physical equations as a part of the training process.

Inverse problems are posed in opposing direction: boundary conditions have to be traced back from the FDM solution defined on the entire domain. In general, the prediction of inverse solutions represents a more challenging and intrinsically ambiguous problem, since different boundary conditions may, in general, lead to similar or even the same solutions of a BVP. For example, two BVPs with different values inside the Dirichlet domain will lead to the same solution, since only boundary values are principally determining for the solution according to the integral theorems. Numerical solving of inverse problems is also known to be computationally more expensive and requires a substantial reformulation of the solving scheme. In contrast, within the scope of this framework, the task of inverse prediction is trivially addressed by

reversing the input and output from the forward model. That is: the sparse boundary conditions defined on some partial domain boundary has to be predicted from the global solution of 2D Lamé-Navier PDE defined on the entire image domain.

Training DNN models

The training of both the forward and the inverse models were implemented using Python 3.8 and Tensorflow 2.0[47] with Keras API. For loading dataset and other useful utility operations, the Numpy,[48] Scikit-image,[49] and OpenCV[50] were used. The forward models and the inverse models were trained with a GPU system running with Linux Operating System which has the following configurations, NVIDIA A100 SXM4 - 80 GB RAM. The models were trained with a split of 80:20. Initially, the weights of the model were assigned in random with zero mean and standard deviation of 0.05[51] and the Adam optimizer[52] is used. The models were trained for 2000 epochs with a learning rate of 1e-4 and a batch size of 64.

Validation of model predictions

The predictions of both the forward and inverse models are validated against the ground truth (i.e. FDM solution) using the mean average difference (MAE) and standard deviation (SD) calculated for both magnitude and orientation Equation 13,

$$MAE_{mag} = \frac{1}{N}\sum_{i=1}^{N}\left|u^{(i)} - u^{*(i)}\right|$$
$$MAE_{dir} = \frac{1}{N}\sum_{i=1}^{N}\left|\alpha^{(i)} - \alpha^{*(i)}\right|$$
$$SD_{mag} = \sqrt{\frac{\sum_{i=1}^{N}\left(u^{(i)} - \overline{u}\right)^2}{N}} \quad \text{(Equation 13)}$$
$$SD_{dir} = \sqrt{\frac{\sum_{i=1}^{N}\left(\alpha^{(i)} - \overline{\alpha}\right)^2}{N}}$$

In addition to the above validation metrics, inverse models are validated for the prediction of boundary conditions (BC) against the ground truth data. In the ground truth BVPs, the boundary conditions take binary values 0 (Free) or 1 (Dirichlet). In the inverse prediction, the values of boundary conditions are initially assessed in terms of probabilities ranging between 0 to 1 that indicate the level of model confidence for identification of the Dirichlet BC. Binary classification of pixel BC to either Free or Dirichlet BC is done using the probabilistic threshold ≥ 0.5 for the Dirichlet BC. Boundary conditions are validated using the F1 score calculated from the binary classification confusion matrix as Equation 14

$$F1 = \frac{2 * (Precision * Sensitivity)}{Precision + Sensitivity} \quad \text{(Equation 14)}$$

where Precision and Sensitivity are given as,

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive},$$
$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative}. \quad \text{(Equation 15)}$$