

<https://doi.org/10.1038/s41524-025-01718-y>

# Digitalizing metallic materials from image segmentation to multiscale solutions via physics informed operator learning



Shahed Rezaei<sup>1</sup> ✉, Kianoosh Taghikhani<sup>1</sup>, Alexandre Viardin<sup>1</sup>, Reza Najian Asl<sup>2</sup>, Ali Harandi<sup>3</sup>, Nikhil Vijay Jagtap<sup>4</sup>, David Bailly<sup>4</sup>, Hannah Naber<sup>5</sup>, Alexander Gramlich<sup>5</sup>, Tim Brepols<sup>3</sup>, Mustapha Abouridouane<sup>6</sup>, Ulrich Krupp<sup>5</sup>, Thomas Bergs<sup>6</sup> & Markus Apel<sup>1</sup>

Fast prediction of microstructural responses based on realistic material topology is vital for linking process, structure, and properties. This work presents a digital framework for metallic materials using microscale features. We explore deep learning for two primary goals: (1) segmenting experimental images to extract microstructural topology, translated into spatial property distributions; and (2) learning mappings from digital microstructures to mechanical fields using physics-informed operator learning. Loss functions are formulated using discretized weak or strong forms, and boundary conditions-Dirichlet and periodic-are embedded in the network. Input space is reduced to focus on key features of 2D and 3D materials, and generalization to varying loads and input topologies are demonstrated. Compared to FEM and FFT solvers, our models yield errors under 1–5% for averaged quantities and are over 1000× faster during 3D inference.

Multiscale material modeling is essential for optimizing metallic materials and linking process parameters to product performance. Integrated computational materials engineering (ICME) combines approaches across scales, from atomic interactions and phase diagrams (CALPHAD) to phase field models and finite element simulations. While these methods excel in material design, their real-time application during production is limited by computational demands. This challenge is critical as the shift towards a circular steel economy introduces variability in compositions and process parameters<sup>1</sup>.

Digital twins<sup>2,3</sup> and digital shadows (DS)<sup>4</sup> have been explored as solutions for real-time feedback. However, these concepts are unsuitable for describing materials during production due to their multi-layered nature. To address this, the digital material twin (DMT) and digital material shadow (DMS) were introduced<sup>5</sup>. The DMT integrates the nano-, micro-, and macroscopic material structures, enabling comprehensive material description during processing. This framework accelerates material innovation and improves predictions for properties like strength and durability, bridging microstructural design to large-scale applications.

A digital material shadow framework<sup>6</sup> is crucial for capturing data from machines and simulations, enabling cross-domain collaboration and defining real-world applications. This is especially relevant when microstructures evolve through distinct processes, as in drive shaft

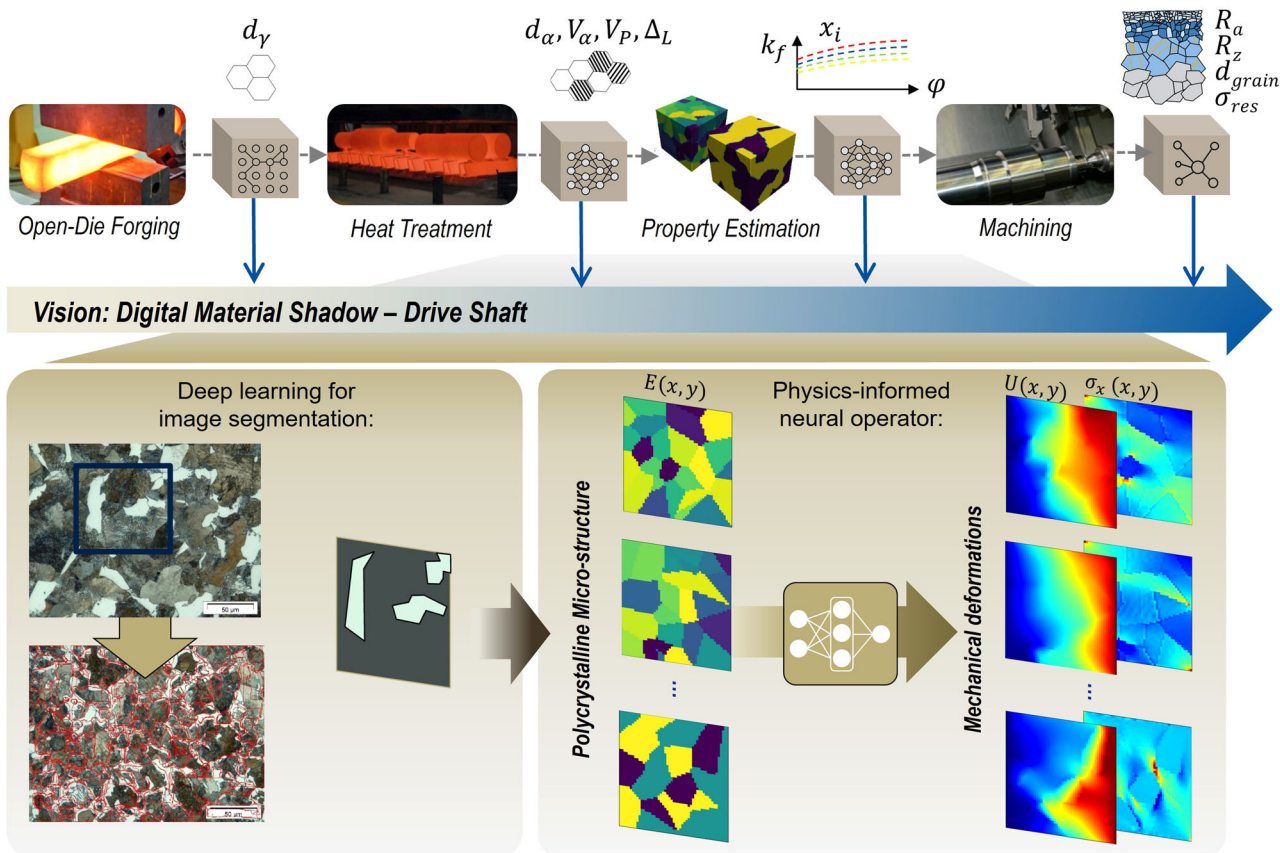
manufacturing<sup>7</sup>, involving forging, heat treatment, and machining. Using an ICME approach, the DMS tracks production data to predict microstructural changes at each stage, as shown in Fig. 1. The final microstructure analysis supports evaluations of local properties like surface integrity and component service life.

The microstructure of the workpiece, material 42CrMo4 (1.7225), through the forging process undergoes static and dynamic recrystallization, resulting into an austenitic microstructure at the end. As this workpiece cools during heat treatment, the two-phase ferrite perlite microstructure is formed, which determines the properties of the material. The local properties are thus to be analyzed from the microstructure for the DMS of the shaft component.

The first challenge to address is the automatic detection of different microstructural features from experimental images, creating a digitalized version of the material for further analysis. The application of artificial intelligence (AI), particularly deep learning and convolutional neural networks (CNNs), has recently garnered significant attention for automating the identification, categorization, and analysis of material properties from two-dimensional experimental images<sup>8,9</sup>. Agbozo and Jin<sup>10</sup> utilized Mask R-CNN<sup>11</sup> to segment carbide particles, achieving an accuracy exceeding 90%. Similarly, Fu et al.<sup>12</sup> employed Faster R-CNN<sup>13</sup> with transfer learning to detect dendrite cores in

<sup>1</sup>ACCESS e.V., Aachen, Germany. <sup>2</sup>Technical University of Munich, München, Germany. <sup>3</sup>Institute of Applied Mechanics, RWTH Aachen University, Aachen, Germany. <sup>4</sup>Institute for Metal Forming, Aachen, Germany. <sup>5</sup>Steel Institute, Aachen, Germany. <sup>6</sup>Manufacturing Technology Institute, Aachen, Germany.

✉ e-mail: [s.rezaei@access-technology.de](mailto:s.rezaei@access-technology.de)



**Fig. 1 | Data pipeline of the drive shaft digital material shadow.** The production data and simulation results in each step of process chain is used in reduced models to predict the local microstructure and the material properties. The reduced models can be AI based or analytical, all linked by a data pipeline. By employing physics-

informed operator learning, we aim to map the microstructure property to its mechanical deformation field. Consequently, this approach eliminates the necessity of data generation in forward problems.

Ni-based superalloys, demonstrating high precision. Studies by Mulewicz et al.<sup>14</sup> and Elbana et al.<sup>15</sup> validated the utility of CNNs for classifying microstructural features in steels and ultrahigh carbon steel, while Banerjee and Sparks<sup>16</sup> applied CNNs to categorize dendritic microstructures. Liotti et al.<sup>17</sup> developed a supervised method that used real-time X-ray imaging to monitor crystal nucleation events during aluminum alloy solidification, achieving remarkable accuracy with automated image analysis algorithms. Further advancements by Breumier et al.<sup>18</sup>, Martinez-Ostomujof et al.<sup>19</sup>, and Germain et al.<sup>20</sup> demonstrated that U-Net-based deep learning models<sup>21</sup> surpassed conventional thresholding techniques for tasks such as graphite shape characterization and phase discrimination in steels. The reviewed articles and advancements clearly demonstrate the potential of deep learning to automate the digitalization of experimental images, enabling the next level of progress in computational aspects.

In order to link macroscopic properties directly to the microscale without any ad-hoc assumptions, one needs to solve the microscale with appropriate numerical methods. Notable examples of simulation techniques include finite difference, finite volume, spectral methods<sup>22</sup>, boundary element methods<sup>23–25</sup>, virtual element method<sup>26–28</sup> and finite element methods (FEM)<sup>29</sup>. See also examples literature on FE<sup>2</sup> methods<sup>30–32</sup>, a review of FE-FFT-based two-scale methods<sup>22</sup> and hybrid DL-FE based ones for multiscaling<sup>33,34</sup>.

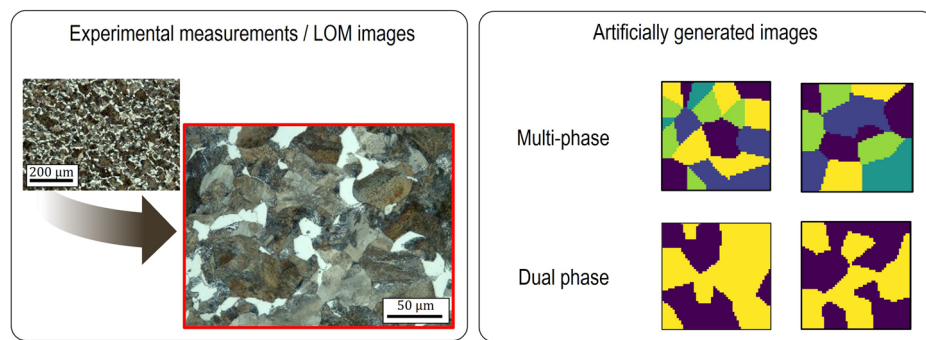
Despite their predictive power, obtaining the solution from these methods can easily become time-consuming, which is problematic for many upcoming design applications. Secondly, as soon as any parameter changes (e.g., the morphology of the microstructure), one has to recompile the adapted model and redo the computation to obtain the new solution. In

other words, the standard solvers are limited to one particular boundary value problem (BVP) and do not seem to be a sustainable and green choice, since they are designed to be used only once.

Deep learning (DL) methods provide solutions to the stated problem by leveraging their interpolation capabilities. For an overview of the potential of deep learning methods in the field of computational material mechanics see refs. 35–37. The interpolation power of deep learning models is raised to a level that encourages researchers to train the neural network to learn the solution to a given boundary value problem in a parametric way. This idea is now pressed as operator learning, which involves the mapping of two infinite spaces or functions to each other. In the ideal case, the evaluation of a single forward pass of a neural network is similar to analytical solutions for partial differential equations, which provide the solution in a very fast way and under any given physical parameters. Some well-established methods for operator learning include but are not limited to, DeepOnet<sup>38</sup>, Fourier Neural Operator<sup>39</sup>, Graph Neural Operator<sup>40</sup> and Laplace Neural Operator<sup>41,42</sup>.

In this category, the data for the training is obtained from the available resources or by performing offline computations and/or experimental measurements for a set of parameters of interest. The idea can also be combined with different architectures of convolutional neural networks (CNN), recurrent neural networks, etc., depending on the application. Winovich et al.<sup>43</sup> proposed and trained networks that are capable of predicting the solutions to linear and nonlinear elliptic problems with heterogeneous source terms. Yang et al.<sup>44</sup> employed a DL method to predict complex stress and strain fields in composites. Mohammadzadeh and Lejeune<sup>45</sup> provided a dataset for mechanical tests under various conditions and heterogeneities and predicted full-field solutions with MultiRes-WNet

**Fig. 2 | Polycrystalline-based microstructures of 42CrMo4.** Left: LOM image of a ferritic-pearlitic steel grade. Right: Artificially generated random samples for unsupervised training of the finite operator learning framework.



architecture. Koric et al.<sup>46</sup> applied the DeepONet formulation to solve the stress distribution in homogeneous elastoplastic solids with variable loads and material properties. Lu et al.<sup>47</sup> employed a deep neural operator to learn the transient response of interpenetrating phase composites under dynamic loading. He et al.<sup>48</sup> introduced a deep operator network with a residual U-Net to predict elastic-plastic stress response for complex geometries under variable loads. To further explore this topic, one can find comparative studies on available operator learning algorithms<sup>49,50</sup>. Despite their benefits, deep learning methods face challenges: high training costs, limited performance beyond training data, and ensuring physical consistency in predictions.

Another mainstream goes in the direction of integrating the model equations into the loss function of the neural network<sup>51,52</sup>. It is known as Physics-Informed Neural Networks (PINNs) as introduced by Raissi et al.<sup>53</sup>. When the underlying physics of the problem is completely known, we can train the neural network without any initial data. Among many contributions, see its applications in fluid dynamics<sup>54,55</sup>, solid mechanics<sup>56–59</sup>, and constitutive material behavior<sup>60,61</sup>. On the other hand, applying PINNs to forward problems without any training data remains an active area of research. To date, PINNs generally may match classical numerical methods (e.g., FEM) in terms of accuracy and computational efficiency for fixed boundary value problems<sup>57</sup>.

An emerging area of research explores combining operator learning with PINNs. For example, Li et al.<sup>62</sup> used Fourier neural operators to integrate training data and physical constraints. Wang et al.<sup>63</sup> introduced physics-informed DeepONets, enforcing physical laws through soft penalty constraints to improve prediction accuracy and reduce training data needs. Other approaches include combining CNNs with physical constraints<sup>64–66</sup>, and Zhang and Gu<sup>67</sup> trained PINNs on digitalized materials without labeled data, using minimum energy criteria. Zhang and Garikipati<sup>68</sup> proposed encoder-decoder models to solve differential equations in weak form across varying conditions. In these methods, automatic differentiation is mainly used to build the loss function from physical knowledge.

Despite the current active progress reviewed above, key challenges remain. Accurately capturing high-frequency oscillations and sharp discontinuities is still difficult, and the generalizability of such models to out-of-distribution cases or different resolutions is not yet fully understood especially in the context of materials micromechanics. Moreover, eliminating the need for data generation by directly combining deep learning with classical numerical methods opens a promising research direction. Finally, an end-to-end pipeline—from image segmentation to solution prediction—remains largely unexplored (see also Fig. 1).

In this paper, we propose a novel method combining ideas from numerical methods (e.g., finite element and fast Fourier transform) for loss construction and neural operators to efficiently predict mechanical behavior in heterogeneous polycrystalline solids. By discretizing the loss function, we handle complex geometries and boundary conditions while improving training efficiency. The approach maps elasticity parameters to deformation states. Additionally, automated image segmentation is integrated to create digital microstructure representations, allowing

direct use of light optical as well as scanning electron microscopy (LOM and SEM) for material property prediction (see e.g.,<sup>69,70</sup>). This streamlines the workflow and extends applicability to complex microstructures, such as bainitic steels.

## Results

### Preparation of collocation fields

The neural network introduced in Section “Method” should be trained using initial input samples representing the Young’s modulus distribution. On the other hand, these samples should align with the realistic LOM images. It is important to note that the training process is entirely unsupervised (i.e., no FE solution is required) and random fields are sufficient to initiate the training by having the loss function based on discretized FEM residual vector.

In this work, we focus on the study of dual phase polycrystalline materials. Additionally, to ensure the completeness of the proposed methodology and the provided study strategies, we will also explore training the model for multiphase heterogeneous materials. These types of materials are commonly found in many alloys and composite materials<sup>71,72</sup>. An overview of the LOM-based investigations, along with examples of randomly generated two-phase and multiphase polycrystalline structures, is shown in Fig. 2.

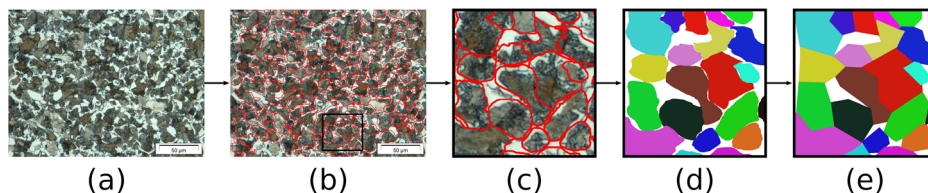
The samples that were examined were from shafts made of 42CrMo4 (1.7225). In order to be able to examine the sample under the light microscope, small pieces were first removed from the shaft, which were then heat-embedded with ATM Duroplast black with graphite. The embedded sample was then grinded with silicon carbide paper up to 2500 grit and polished with a diamond paste. To make the microstructure visible, the surface was etched with three percent nitric acid. Microstructural images with a 100x or 500x magnification were then taken using a Leica DM 2500 reflected-light microscope and SIS software. The samples show a ferritic-pearlitic microstructure. The majority of the microstructure is pearlitic, with some ferritic grains on the prior austenite grain boundaries.

On a typical production line, after forging one end of the shaft, the gripping end is forged after reheating. This produces two distinct microstructures in two different parts of the workpiece. Thus during experimentation, one shaft was forged without reheating, the other was cooled and reheated. The different process steps resulted in different grain sizes. The samples with reheating were coarser than the samples without reheating.

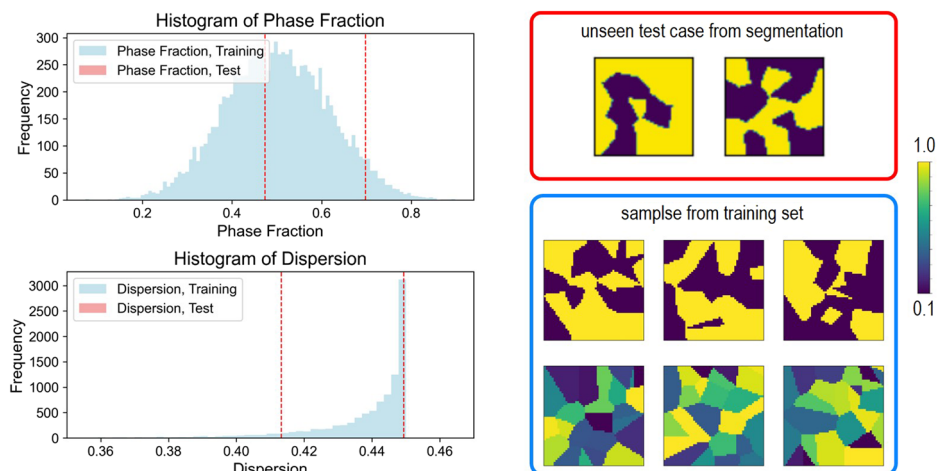
Our objective is to modify the morphology of these microstructures by controlling the number and distribution of grains. We have developed a straightforward yet effective algorithm to generate random input files, allowing control over the shape, quantity, and spatial distribution of grains. Although inspired by LOM images, the algorithm is adaptable and can be customized for different applications as needed. A detailed description of the algorithm is described in Supplementary Information.

For the LOM images, we converted them to grayscale and performed inference using the same trained network. The ferrite appears white in the original images, and the conversion to grayscale enhances its brightness. As a result, the Mask R-CNN model, trained on solidification results, effectively

**Fig. 3 | An example of the inference results.** **a** Initial image provided to the Mask R-CNN model (before conversion to grayscale); **b** Segmentation results using Mask R-CNN; **c** Specific area from the region outlined by the black square in **(b)**; **d** Isolated grains based on the contours in **(c)**; **e** Voronoi construction derived from **(d)**.



**Fig. 4 | The histogram of training and some of unseen test samples.** These samples are used for the unsupervised training of FOL in a 2D setting and are generated using Voronoi-based parameterization.



segments grains surrounded by ferrite and large ferrite grains. This provides a reasonable approximation of the grain distributions and sizes.

An example of the inference results is shown in Fig. 3. In Fig. 3a, the initial microstructure image, used as input for the Mask R-CNN model, is displayed. The segmentation results are presented in Fig. 3b, where the contours primarily align with the ferrite precipitates. From Fig. 3b, we selected a specific area, as shown in Fig. 3c. Based on the contours in Fig. 3c, we isolated each grain individually, with each grain represented in different colors in Fig. 3d. Finally, we propagated these grain contours isotropically to create a Voronoi structure, as shown in Fig. 3e.

This image will serve as input for the next model by assigning different material properties to each identified phase. Given a segmented microstructure image with distinct phases, the spatial distribution of a material property  $E(\mathbf{X})$ , such as Young's modulus, can be defined as:

$$E(\mathbf{X}) = E_i \text{ if } \mathbf{X} \in \Omega_i, \quad (1)$$

where  $\mathbf{X} \in \mathbb{R}^d$  is the spatial position (2D or 3D),  $\Omega_i$  denotes the region occupied by phase  $i$ ,  $E_i$  is the assigned material property for phase  $i$ . This expression assigns a constant material property  $E_i$  to each spatial point  $\mathbf{X}$  based on the recognized phase from the segmented image.

We present a statistical analysis of the synthetic samples used to train the FOL and SPiFOL models, highlighting differences between training and test set topologies. For the 2D FOL model, 8000 dual-phase samples are analyzed based on phase fraction and dispersion, as shown in Fig. 4.

The unseen test cases derived from segmentation exhibit similar patterns to the artificially generated training samples. Their positions in the histograms, marked by red dashed lines, indicate that while some fall within the typical sample distribution, others lie further from the training data, suggesting a degree of generalization challenge.

The setup for the 2D and 3D SPiFOL frameworks presents a more challenging test for the deep learning models. Specifically, training was performed solely on Fourier-based samples at a resolution of  $32^3$ , while testing involved higher-resolution samples (up to  $128^3$ ) with entirely different topologies inspired by polycrystalline microstructures.

Unlike the 2D case-where test samples were obtained via image segmentation-the 3D microstructures were synthetically generated to represent the target material (See Figure. 5). In future work, one could explore generative models to reconstruct 3D structures directly from 2D images. For more details on sample generation, see<sup>73,74</sup> and “Method” section. To quantitatively demonstrate that the test samples are out of distribution, we compute the Wasserstein distance between the training and test images, which shows a clear deviation. Additional details are provided in the Supplementary Information.

### Network parameters

The algorithms developed in this study are implemented using JAX<sup>75</sup> software and the methodology can be adapted to other programming platforms as well. A comprehensive summary of the network's hyperparameters and configurations for all the models is provided in Table 1.

The hyperparameters listed in Table 1 were selected based on extensive experimentation and represent configurations with some of the best performance. We explored the effects of various activation functions, network architectures, batch sizes, and even multi-stage training strategies to improve network initialization. Model selection was guided by the average error over 1000 reserved test samples in the final stage. For the FOL framework, we found that Swish and LeakyReLU activation functions performed best. Other activation functions, such as *tanh*, often led to trivial solutions dominated primarily by the boundary conditions.

Increasing the number of layers and neurons generally improved accuracy, but at the cost of higher training time. The final architecture balances performance and efficiency, based on systematic trial and error rather than global optimization (see Supplementary Information). Regarding batch size, we recommend using ~10% of the training dataset. Smaller batches lead to slower training but can yield better accuracy, while larger batches accelerate training but may be limited by GPU memory. Showing the network moderate-sized batches may also help avoid convergence to trivial solutions dominated by boundary conditions. For the SPiFOL model, increasing the number of Fourier layers and modes tends to improve performance, as illustrated in Supplementary Information.

**Table 1 | Summary of the network parameters**

FOL 2D	Learning on heterogeneity
Inputs, Outputs	$[X_j, Y_j, E_j], \{[U_i] = \{U_1, \dots, U_N\}, \{V_i\} = \{V_1, \dots, V_N\}\}$
Activation function	Swish (initial training) and Sigmoid/LeakyRelu (for re-training)
Layers and neurons	[2000, 2000]
Number of samples	1, 10, 100, 200, 500, 1000, $\dots$ , 8000
Batch size	10% of Number of samples
Learning rate	$10^{-4}$
Number of epochs	2000 for each step
SPiFOL 2D	Learning on heterogeneity
Inputs, Outputs	$[E(\mathbf{X})], [\varepsilon_{xx}(\mathbf{X}), \varepsilon_{yy}(\mathbf{X}), \varepsilon_{xy}(\mathbf{X})]$
Latent size	32
Fourier layers and modes	3, 16
Number of samples	8000
Batch size	20
Activation function	<i>Gelu</i>
FOL 3D	Learning on BCs
Inputs, Outputs	$[U_{b,j}, V_{b,j}, W_{b,j}], \{[U_i], \{V_i\}, \{W_i\}\}$
Activation function	Swish
Layers and neurons	[100, 100]
Number of samples	1, 10, 100, 200, 400, 600, 800
Batch size	10% of Number of samples
Learning rate	$10^{-4}$
Number of epochs	2000 for each step
SPiFOL 3D	Learning on heterogeneity
Inputs, Outputs	$[E(\mathbf{X})], [\varepsilon_{xx}(\mathbf{X}), \varepsilon_{yy}(\mathbf{X}), \varepsilon_{zz}(\mathbf{X}), \varepsilon_{xy}(\mathbf{X}), \varepsilon_{xz}(\mathbf{X}), \varepsilon_{yz}(\mathbf{X})]$
Latent size	32
Fourier layers and modes	3, 16
Number of samples	8000
Batch size	20
Activation function	<i>Gelu</i>

The first and second sections pertain to the 2D examples via FOL model and SPiFOL models, while the third and fourth sections summarize the network details for the 3D setup using FOL and SPiFOL model. For all the models, Adam Optimizer<sup>48</sup> is used.

The loss function for training all models follows a standard decay over epochs, as shown in Supplementary Information. We stopped the training when there is no significant improvement by increasing epochs.

Quadro RTX 6000s with 24 GB of RAM is utilized for the training of all models. An important aspect of parametric training for the proposed network is the proper initialization of its free parameters. From a mathematical perspective, we are dealing with a multi-objective optimization problem with a highly complex loss function. This loss function aims to map microstructural features to the physical solution space while strictly satisfying boundary conditions. Consequently, it is highly challenging to avoid local minima and trivial solutions that fail to capture the complex deformation patterns in heterogeneous material systems. This challenge is commonly associated with capturing high frequencies and sharp jumps in the solution response, as well as ensuring the network remains unbiased toward these features. To address this issue, we drew inspiration from transfer learning. Specifically, we gradually train the system using an increasing number of samples, rather than training the network all at once with the entire set of randomly generated samples. This stepwise approach enables the network to progressively refine its performance, benefiting from improved initialization at each stage.

For the material parameters, we primarily focus on varying the Young's modulus. The normalized values range from 0.05 to 1.0, depending on each example. We intentionally examine a wide range of phase contrasts to demonstrate the applicability of the proposed approach to higher phase contrast values, which are common in various material engineering applications. In other words, for the training we go beyond the typical properties of 42CrMo4 (1.7225). In the "Result" section, we specify the corresponding values for each test case. Furthermore, for the multiphase material system, random values are assigned to each grain. These values are normalized so that the normalized Young's modulus values remain between 0 and 1.

Finally, we address the applied boundary conditions. In the first set of studies with the 2D setup, we focus on an arbitrary set of boundary conditions (BCs), which consist of a mix of Dirichlet and Neumann conditions. The applied displacement on the right edge is  $(U_R, V_R) = (0.05, 0.05)$  mm, while the upper and lower edges remain traction-free.

For the SPiFOL setup, where periodic boundary conditions (PBCs) are enforced, the macroscopic strain tensor, denoted by  $\bar{\varepsilon}$ , is fixed for all reported results, as

$$\bar{\varepsilon}_{2D} = \begin{bmatrix} 0.05 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}, \quad \text{and} \quad \bar{\varepsilon}_{3D} = \begin{bmatrix} 0.05 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}. \quad (2)$$

For the 3D setup within the FOL framework, we aim to learn the solution for randomly applied boundary conditions, while the microstructure topology remains fixed. In this setup, the back surface is fixed, arbitrary boundary conditions are applied to the front surface, and all other surfaces are traction-free.

#### FOL prediction for 2D test cases: learning on different topologies

We train the network using the parameters listed in Table 1 and the artificially generated samples discussed in Fig. 4 and Supplementary Information. Once training is complete, we evaluate the performance of the deep learning model on unseen test cases derived from segmented LOM images, as described in Section "Method" and illustrated in Fig. 3. The evaluation results for the 2D cases are presented in the following sections.

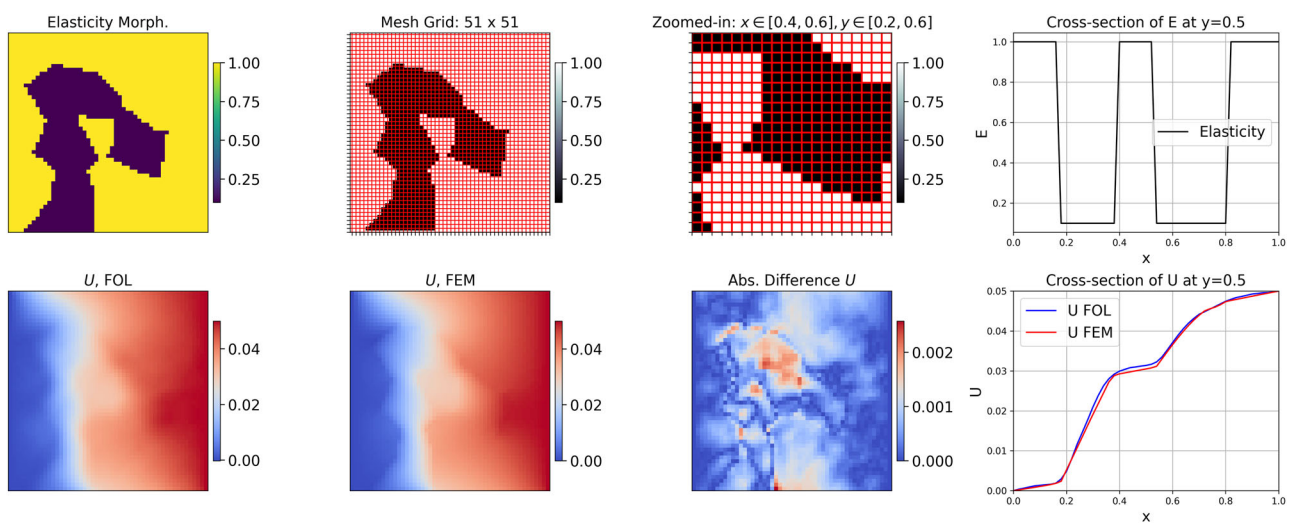
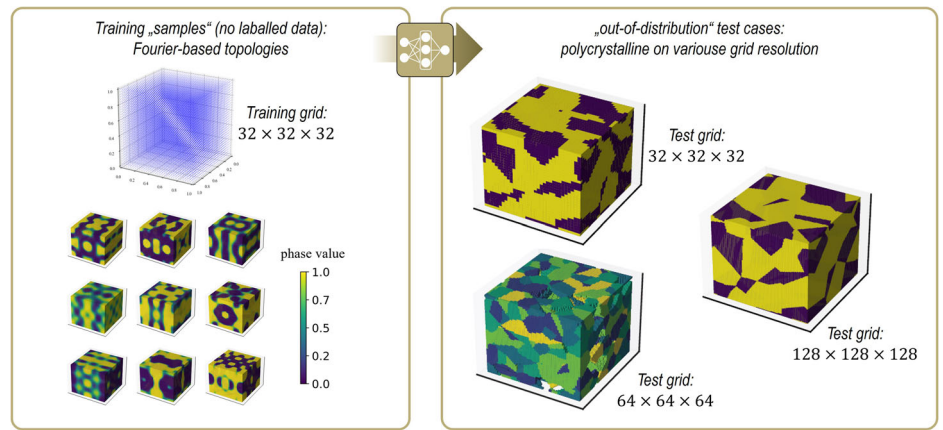
The results presented in the following are selected examples from the test set, consisting of either segmented images or synthetically generated samples with similar characteristics such as grain size and phase fraction. More comprehensive statistical analysis and systematic error evaluations of the trained models are provided later. Note that most results show the absolute pointwise error between the DL model and classical FEM or FFT solvers. While this is common in the operator learning literature, relative errors are less meaningful here due to potential division by zero. Instead, we later report relative errors of homogenized (averaged) quantities, which offer more useful insights for future multiscale applications. For convenience, all the reported values are presented in normalized notation. Further error analysis and comparisons of computational costs are provided later on.

In Figs. 6 and 7, the solutions obtained from FOL and FEM for the deformation component  $U$  are compared, showing an excellent match between the two methods. Similar results are observed for the deformation component  $V$ , although these are not shown for brevity.

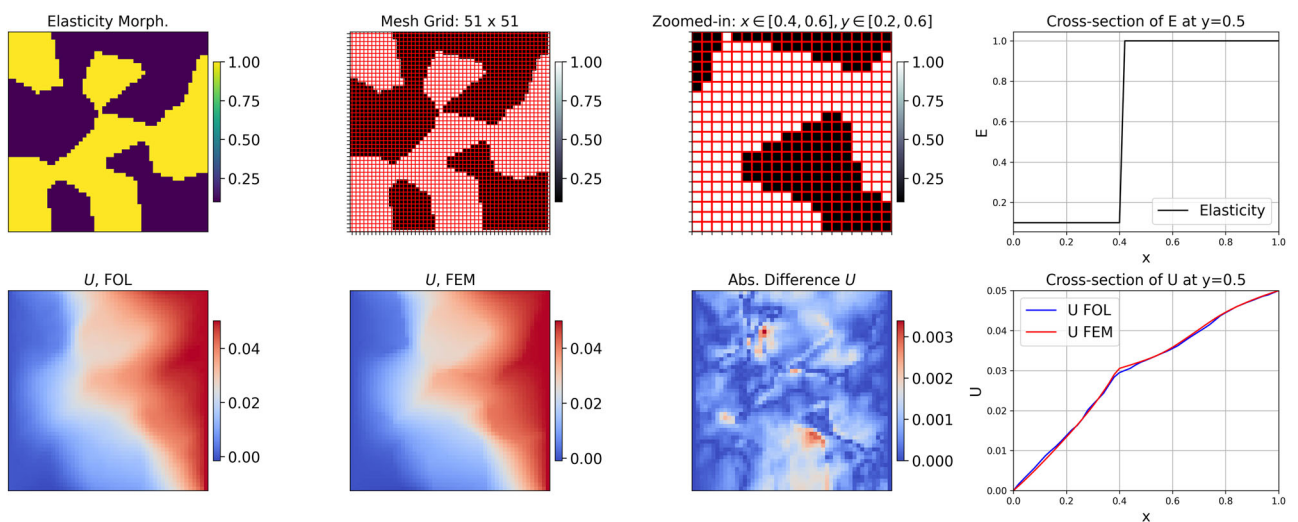
In Fig. 8, we repeat the same study for a multiphase polycrystalline material, where similar observations confirm the effectiveness of the proposed approach for approximating solutions solely based on governing equations. Next, we use the predicted deformation components from the trained network, apply spatial gradients based on standard FEM routines, and calculate the stress components.

In Fig. 9, the comparison is presented for the stress components  $\sigma_{xx}$  and  $\sigma_{yy}$ . While the overall behavior matches very well, small fluctuations in the predicted solution lead to slightly more oscillatory behavior in the DL model. Nevertheless, the peak values and averages are captured with less than 5%. Here, FOL is trained to predict displacement, not stress directly.

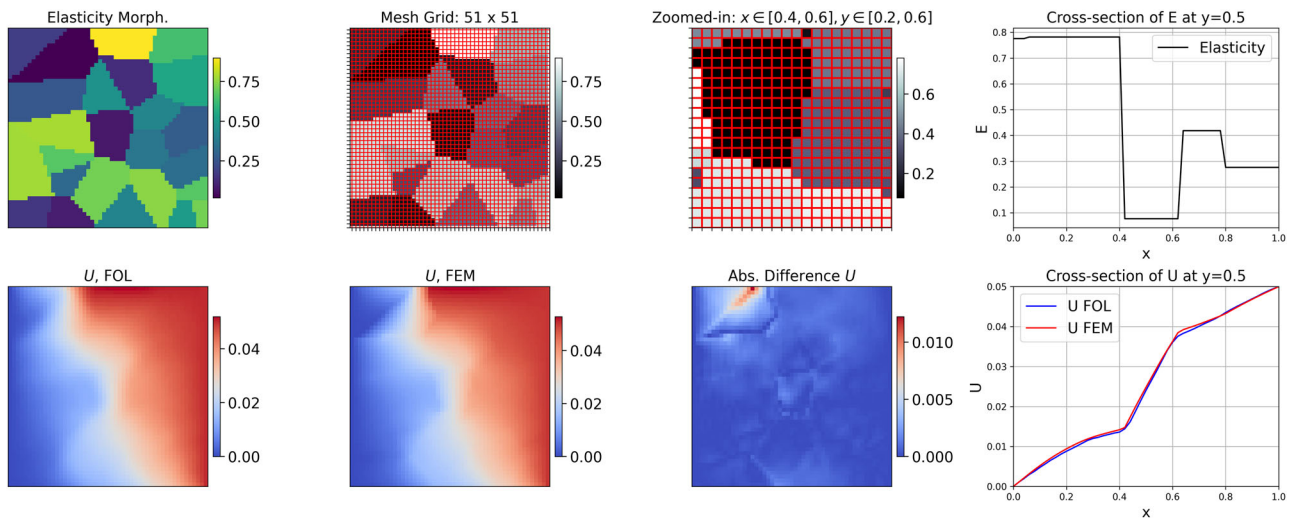
**Fig. 5 | Unsupervised training of SPiFOL in a 3D setting.** Training samples are based on Fourier parametrization. For testing, we use extreme out-of-distribution samples derived from polycrystalline microstructures.



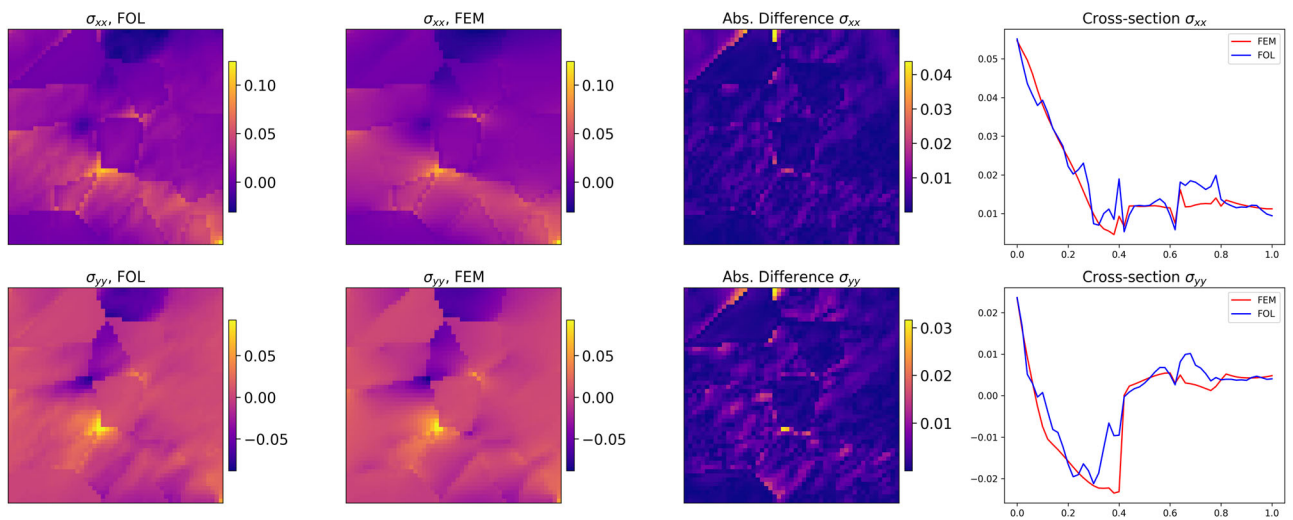
**Fig. 6 | Results of the FOL model.** Prediction of the displacement field for an unseen test case (Case 1).



**Fig. 7 | Results of the FOL model.** Prediction of the displacement field for an unseen test case (Case 2).



**Fig. 8 | Results of the FOL model.** Prediction of the displacement field for an unseen test case (Case 3).



**Fig. 9 | Results of the FOL model.** Prediction of the stress field for an unseen multi-phase test case (Case 3).

Although displacement errors are low, stress errors can be amplified due to sensitivity to derivatives, which explains the higher errors. A potential solution is to train the model to predict stress directly, as demonstrated in the next section. An alternative not explored here would be to enhance FOL using a mixed formulation to directly learn stress fields, as suggested in ref. 57.

### SPiFOL prediction for 2D test cases: learning on different topologies

The SPiFOL model is now evaluated and tested on unseen periodic samples, and the results of the predicted stress components are shown in Fig. 10 for one unseen test case. Similar to the previous section, the results shown here are a few representative cases for demonstration purposes. A more comprehensive statistical error analysis over larger test sets and varying training sample sizes will be presented later.

The stress profile errors are lower in the SPiFOL model compared to the FOL model. This improvement arises because the SPiFOL model's loss function directly targets the strain components, which can be used to evaluate the stress. In contrast, the FOL framework, while accurately capturing the deformation field, shows increased errors in predicting spatial derivatives due to their higher sensitivity to deviations. It is important to note that SPiFOL, based on the FNO architecture and FFT algorithm, is restricted to regular grids with periodic topologies.

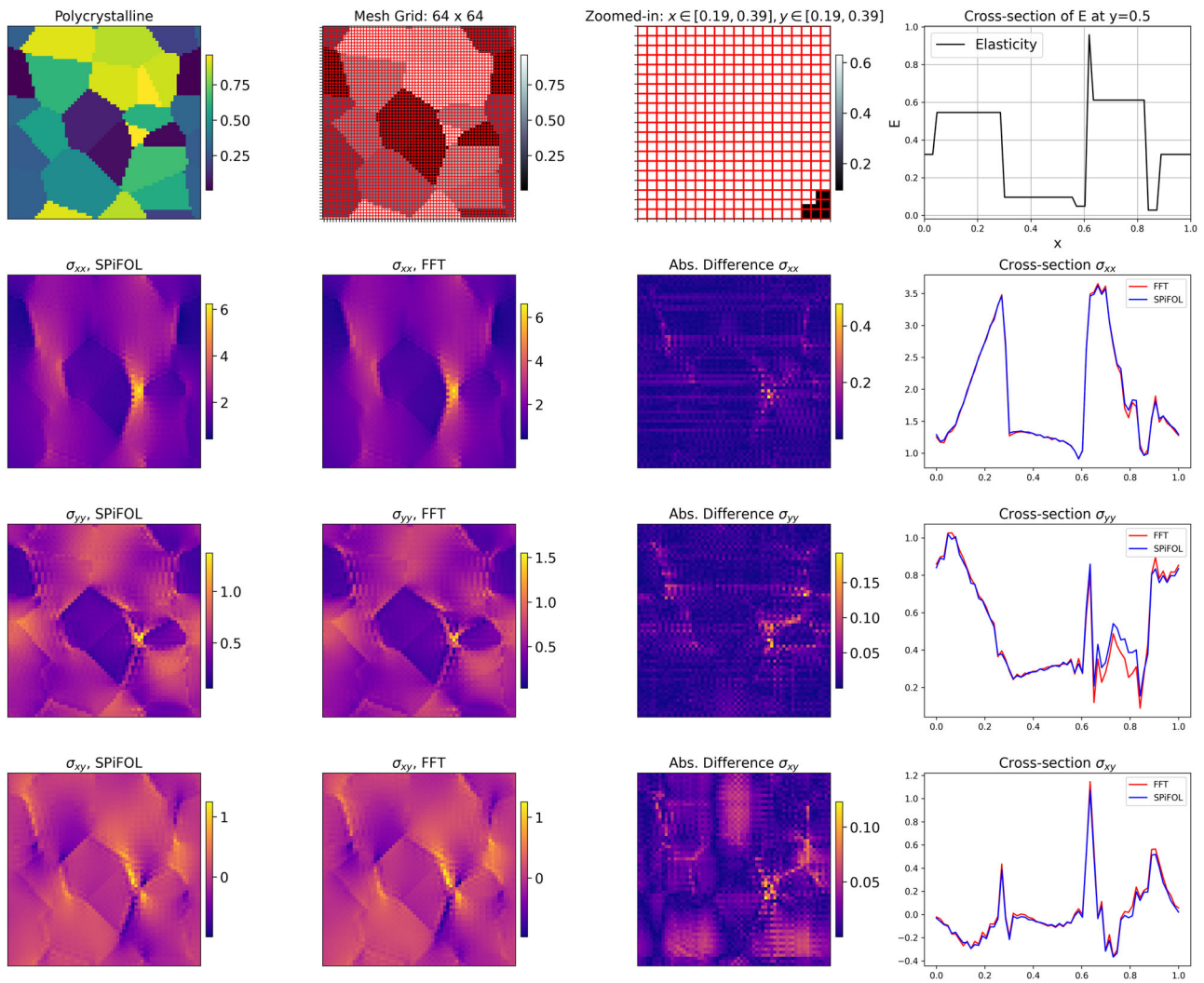
It is important to note that the test sample topologies shown above satisfy geometric periodicity and may not exactly replicate experimental microstructures. Nevertheless, they preserve key statistical features such as grain count, phase fraction, and phase contrast. Additionally, these test cases represent challenging scenarios, as the SPiFOL model was trained exclusively on Fourier-based samples without prior exposure to any Voronoi-based structures.

### Extension to 3D: learning on different boundary conditions

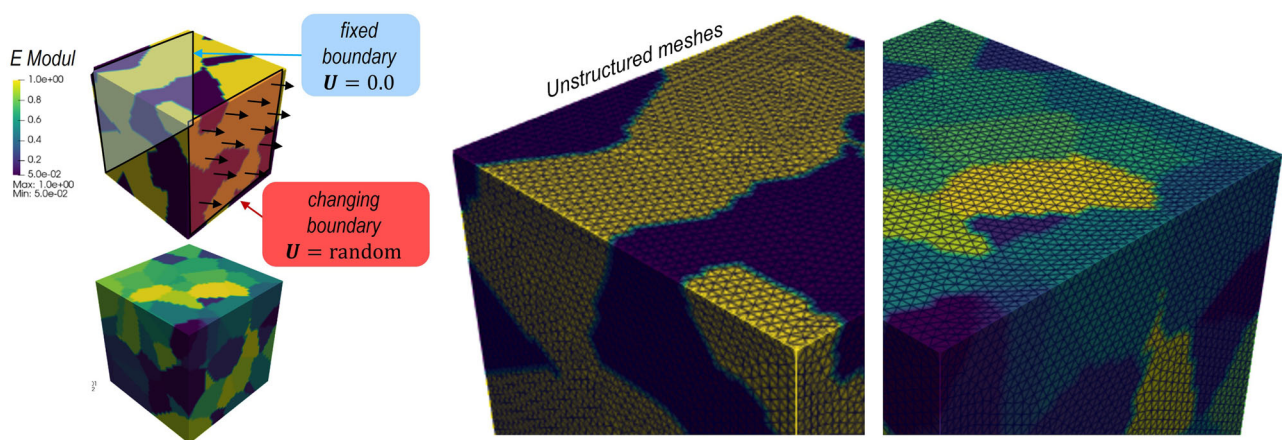
Here, the model learns with respect to the applied Dirichlet boundary conditions on the front surface. The morphology of the microstructure is kept constant, representing a chosen volume element, while we investigate how deformation and stress components develop under various boundary conditions.

The details of the boundary value problem are illustrated in Fig. 11, where two different cases with two-phase and multiphase polycrystalline materials are analyzed. For the meshing process, the approach is highly adaptable, allowing the use of either structured or unstructured meshes with any element type, seamlessly integrated into the proposed FOL framework.

For more details on the type and distribution of the training samples as well as the NN hyperparameters, see Supplementary Information. According to Fig. 12 where distribution plots of the Dirichlet boundary condition components are shown, the selected test



**Fig. 10 | Results of the SPiFOL model.** Prediction of the stress field for an unseen test case (Case 4).



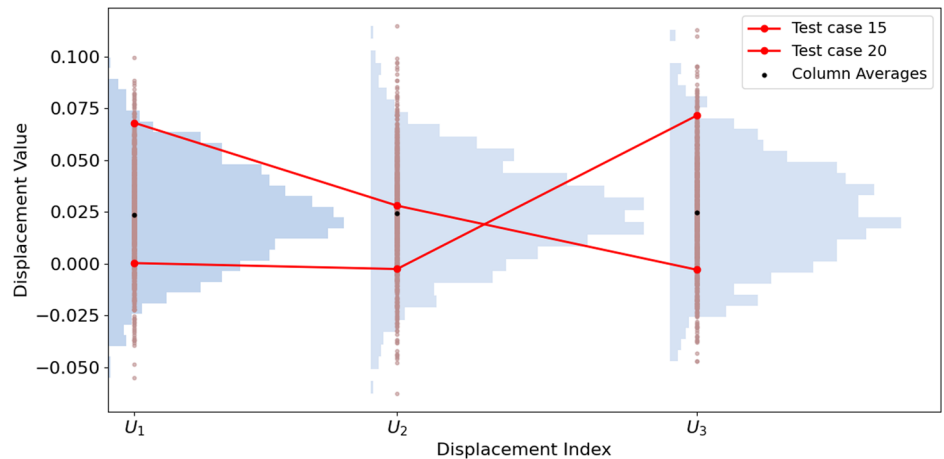
**Fig. 11 | 3D microstructure and its corresponding mesh.** This test is used to learn the system response under arbitrary loading directions.

cases are intentionally chosen outside the typical range to evaluate the interpolation capability of the trained network.

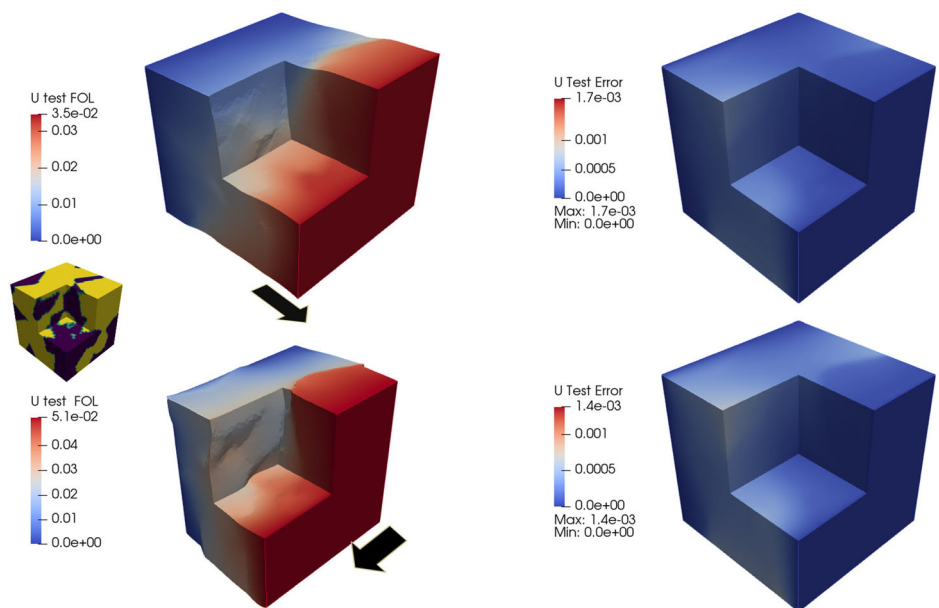
Deformation magnitude ( $\sqrt{U^2 + V^2 + W^2}$ ), as well as the  $\sigma_x$  stress component are reported. For the two-phase 3D volume element, we present two entirely different test cases for the applied deformation: one

predominantly in tensile mode and the other primarily involving shear and compression, as illustrated in Fig. 13. The corresponding predicted values for the stress component  $\sigma_x$  are shown in Fig. 14. Due to the strong heterogeneity, we observe sharp jumps in the deformation and stress fields, which are both captured very accurately.

**Fig. 12 | Distribution of the applied displacement components.** The colored markers represent selected test cases and the histograms depict the distribution of 1000 training samples. The test cases are intentionally chosen outside the typical range.



**Fig. 13 | Results of the FOL model.** Prediction of the displacement for two different unseen loading directions.



### Extension to 3D: learning on different topologies using SPiFOL

Finally, we examine the performance of the trained 3D SPiFOL model. The test cases are selected to feature topologies similar to experimental images and yet very different from those in the training set, and the evaluation is carried out at various resolutions to assess the model's zero-shot super-resolution capability. Figures 15 and 16 compare the SPiFOL predictions with those of the FFT solver for two selected stress components and two representative morphologies: one dual-phase and one multi-phase polycrystalline microstructure.

In Fig. 17, we evaluate the same dual-phase polycrystalline test topology at different resolutions. Importantly, no re-training is performed; the model trained on a  $32^3$  grid is directly evaluated on finer grids of  $64^3$  and  $128^3$ . As expected, the maximum pointwise error increases with resolution, primarily due to fine-scale features in the microstructure that may be missed when training on lower-resolution data. This highlights the challenge of applying models across significantly different resolutions. Therefore, we recommend conducting a proper mesh or grid convergence study, similar to the one in Supplementary Information, before selecting the training resolution.

A potential remedy is to integrate the model with microstructure-embedded autoencoders, as proposed in ref. 76, to retain critical details lost during downsampling. Nevertheless, when considering the error in

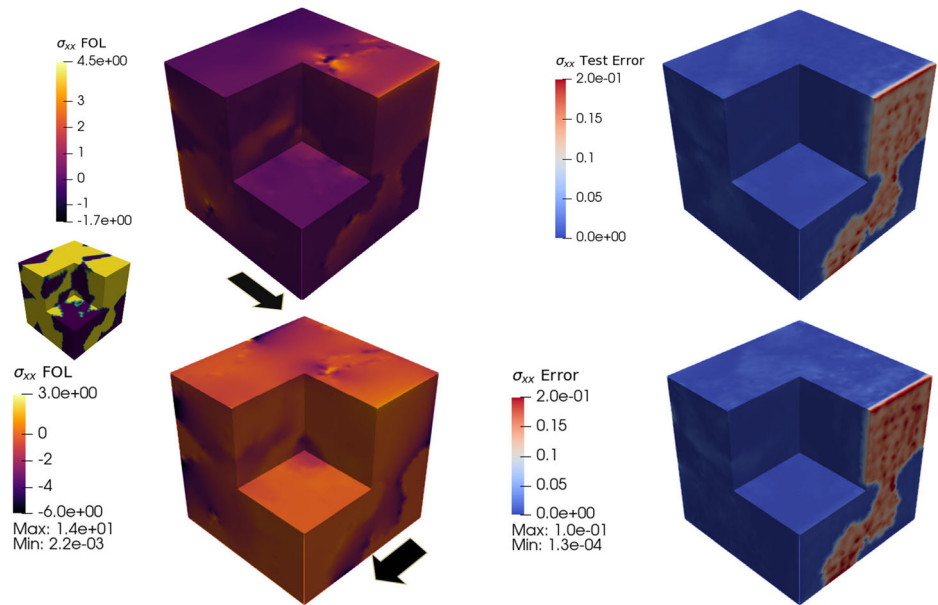
homogenized (averaged) stress values, the results remain low and within acceptable limits. This suggests that SPiFOL effectively reduces the dimensionality of the problem, enabling training at low resolution while still capturing meaningful averaged quantities when evaluated at higher resolutions.

### Computational costs and error analysis

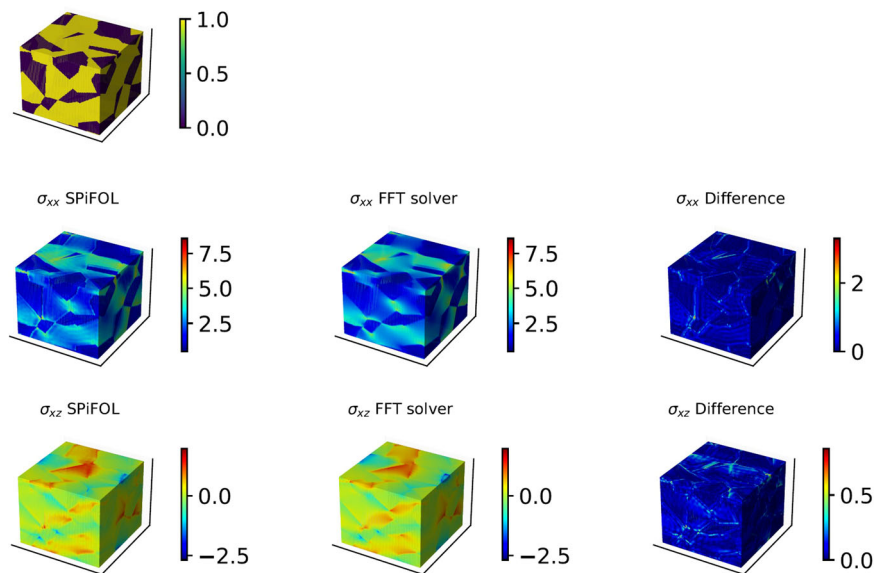
Here, we evaluate the current methodology by comparing its performance in training and evaluation against classical numerical solvers. The results of such comparisons are summarized in Table 2. For brevity, FEM and FOL are used for comparison, with similar trends also observed when comparing SPiFOL to the FFT solver.

The test cases presented above demonstrate the potential of the trained model to generalize to unseen scenarios, even those outside the training patterns. In this section, we critically evaluate the performance of the trained models using a larger set of test cases. Three main aspects are explored: first, we analyze the distribution of errors across a broader range of test samples, examining variations in topology; second, we investigate the impact of the number of training samples on model performance; and third, we compare the purely physics-informed training (i.e., label-free) with a data-driven approach, where we first solve for all training samples using classical numerical solvers and then train the network using these labeled data. The

**Fig. 14 | Results of the FOL model.** Prediction of the stress field for two different unseen loading directions.



**Fig. 15 | Results of SPiFOL model.** A 3D dual-phase polycrystalline is used as an extreme unseen test case.



results of these investigations are summarized in the following figures. Note that, due to the presence of two models and numerous possible variations, we alternate between them for the sake of brevity in these studies. However, similar results and trends are expected, unless stated otherwise.

In order to report the relative errors in percentage, we introduce the following measure, which applies to an arbitrary quantity.

$$\text{Relative Error}(\%) = \frac{|\hat{\bullet} - \hat{\bullet}_{\text{numerical}}|}{|\hat{\bullet}_{\text{numerical}}|} \times 100, \quad \hat{\bullet} = \frac{1}{V} \int_V \bullet dV$$

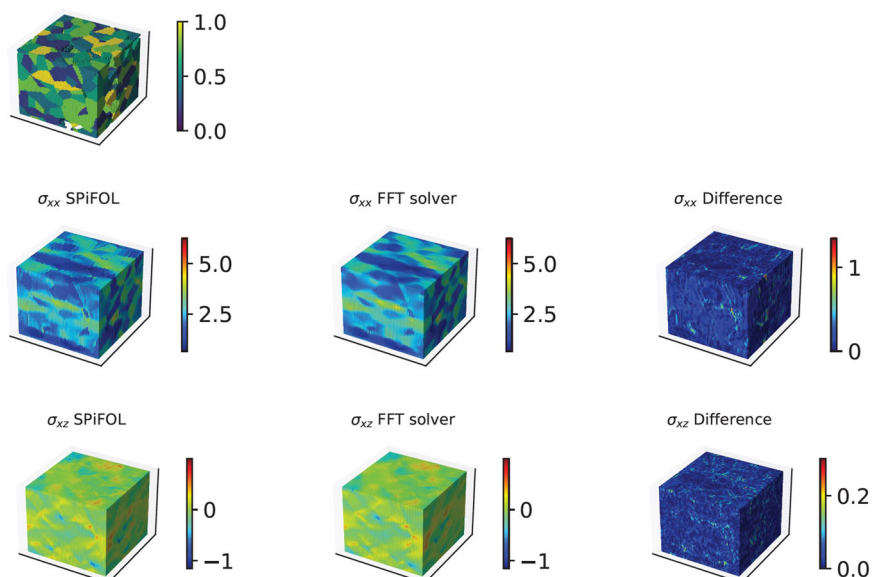
Here,  $\hat{\bullet}$  represents the volume-averaged value predicted by the model,  $\hat{\bullet}_{\text{numerical}}$  represents the volume-averaged value obtained from the numerical method (e.g., classical FEM, FFT solver, etc.). Moreover, the volume-averaged value is defined as  $\hat{\bullet}$  is the volume-averaged quantity,  $V$  is the volume over which the averaging is performed,  $\bullet$  is the quantity of interest (e.g., stress, deformation, etc.).

In the left panel of Fig. 18, the relative averaged error in displacement components (primary field) over 500 unseen samples is shown. The trained

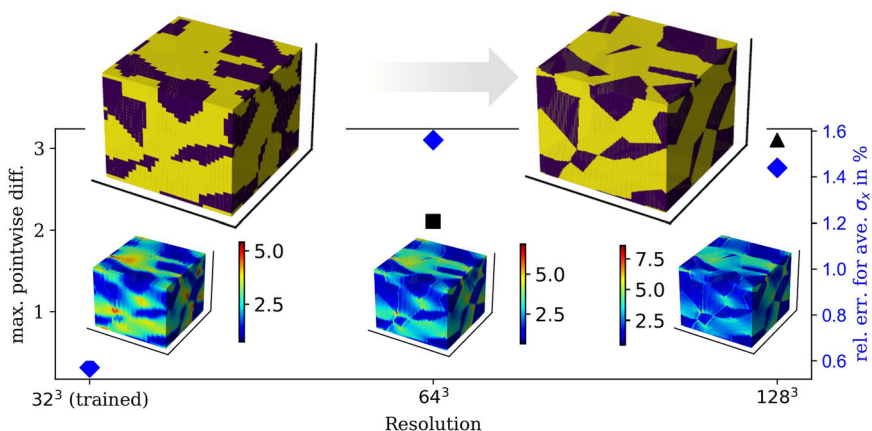
FOL model demonstrates excellent performance, with overall average errors consistently below 4%. Dual-phase topologies exhibit slightly higher and more widely distributed errors, reflecting the increased difficulty due to sharp field variations. A similar level of accuracy is observed for stress components, as discussed earlier, since local fluctuations have minimal influence on averaged quantities. The overall relative error in homogenized stress remains below 3%. Same pattern for the results also holds for the 3D FOL model. In the 3D case, the reported errors are even lower, as the training was based solely on varied boundary conditions while maintaining a constant topology. This observation suggests that the solution's sensitivity to topology is significantly higher than to the applied boundary conditions, which aligns with intuitive expectations.

We now turn to the SPiFOL model. In Fig. 19, two key analyses are presented. First, we observe how the error decreases as the number of training samples increases. Second, we compare the error distributions between two models: one trained with physics-informed constraints in Fourier space, and another trained purely on data from an FFT solver. Notably, the physics-informed model consistently outperforms the data-

**Fig. 16 | Results of SPiFOL model.** A 3D multi-phase polycrystalline used as an extreme unseen test case.



**Fig. 17 | Results of SPiFOL model.** Upscaling the results of SPiFOL (from  $32^3$  to  $128^3$  grid points) using zero shot super resolution.



**Table 2 | Comparison of the computational costs**

	Averaged run-time
Training 2D (dual or multi phase)	≈40 min
Training 3D	≈30 min
Data generation (2D or 3D)	Does not apply (method is data-free)
Network evaluation, 2D	≈0.1 ms
Network evaluation, 3D	≈0.6 ms
Finite element, 2D	≈0.3 s
Finite element, 3D	≈1.5 s

driven counterpart, showing both lower average errors and reduced standard deviation. Similar trends are also observed for the FOL model. In some cases, the data-driven model captures only trivial solutions, missing critical gradients arising from material heterogeneity, whereas the physics-informed model better resolves such complexities. Same pattern of results also repeat itself for the 3D SPiFOL model.

## Discussion

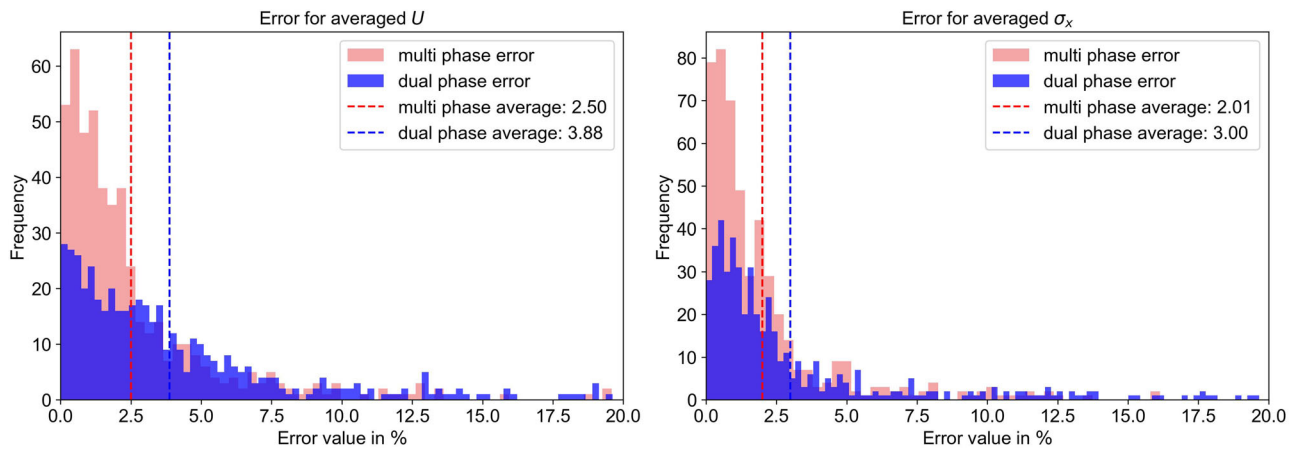
This work discusses and aims to further motivate the application of advanced deep learning algorithms for the chain of metallic material

engineering. We address two main challenges: one is translating experimental images into digitized images for numerical analysis, and the second is solving the system of equations via again DL techniques in a complex heterogeneous domain.

To address the first challenge, we utilize Mask R-CNN deep learning models to segment the LOM images of the metallic microstructure. The backbone of the proposed model builds on established literature, particularly<sup>77</sup> and references therein. To address the second challenge, we utilize the Finite Operator Learning method. This approach is data-free, physics-informed, and notably supports parametric learning of partial differential equations. The key novelty lies in formulating the loss function using either the parametric discretized weak form or the strong form in Fourier space. Furthermore, the proposed framework inherently satisfies Dirichlet and PBCs through its architectural design. See also refs. 73,74,78.

We demonstrate the performance of DL models in 2D and 3D setting with multi- and dual-phase distributions of properties, where we feed different topologies of the microstructure (elasticity distribution) as input. The results of the analysis show acceptable agreement compared to the classical numerical method, making it an attractive surrogate model to replace costly calculations for this particular set of materials.

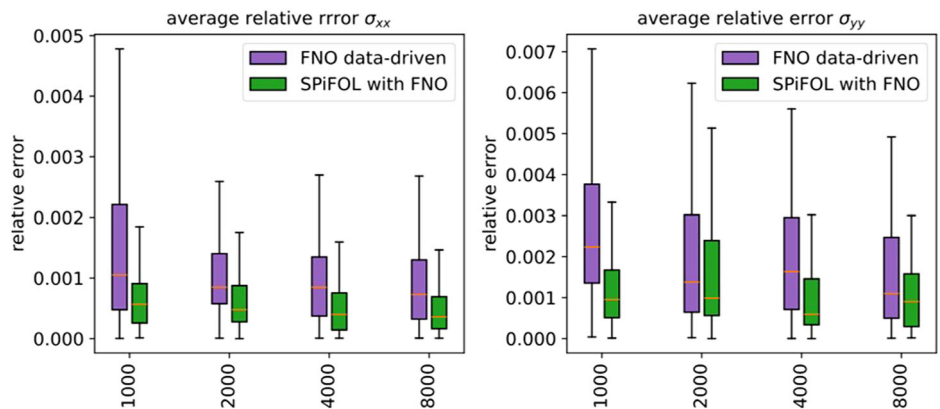
While the proposed framework shows promising generalization—especially with embedded physical constraints—it shares common limitations of deep learning methods. Performance may decline when inputs deviate significantly from the training distribution or when low-resolution



**Fig. 18 | Error analysis of the FOL model.** About 500 test samples for dual- and multi-phase polycrystalline microstructures are evaluated using the trained 2D FOL models. Higher errors are observed for the dual-phase case which is likely due to the

increased complexity and presence of high-frequency solution modes caused by abrupt phase transitions.

**Fig. 19 | Error analysis of the trained 2D SPiFOL model.** About 100 unseen test cases are evaluated where we also compare its performance against a purely data-driven method and examining how the error decreases as more training samples are provided<sup>74</sup>.



models are applied to much finer grids, leading to local errors. However, integral quantities remain accurate, underscoring the potential of neural operators for zero-shot super-resolution in computational material mechanics (see also ref. 79).

The current work can be extended in several important directions. Future efforts should incorporate anisotropic behavior and nonlinear material responses such as plasticity and damage. Promising contributions in this direction have been reported in refs. 60,80, where information on yield surfaces and evolution laws is embedded into the physical loss terms.

The concept of FOL can be extended to other numerical methods such as the Boundary Element Method<sup>24</sup> and the Virtual Element Method<sup>27</sup>, both of which are well-suited for simulating material micromechanics. VEM provides enhanced meshing flexibility, broadening the applicability of traditional FEM approaches, while BEM offers significant mesh reduction advantages for specific classes of problems.

The presented framework demonstrates strong potential for advancing computational materials engineering. In particular, the development of multiscale models and homogenized datasets lays a foundation for improved machinability predictions and optimization of manufacturing processes. These data-driven insights, when integrated with Industry 4.0 technologies, offer a path toward real-time monitoring and adaptive control. Furthermore, the generated data can support Digital Twin systems for predictive simulation and optimization. This approach is broadly applicable to machining, and processes such as metal forming and additive manufacturing, and future work will explore these directions in more detail.

## Methods

In this work, we primarily focus on two-phase and multi-phase material systems, addressing a broad range of potential applications through two distinct models. The first model integrates ideas from FEM and is designed to handle arbitrary combinations of Neumann and Dirichlet boundary conditions, accommodating complex shapes and geometries. The second model is tailored for PBCs and microstructures, leveraging spectral methods for efficient analysis. Both models are constructed based on physical domain knowledge and do not require any labeled data for training. Moreover, they are designed to learn the governing equations in a parametric manner, enabling them to generalize and solve a class of problems once the training stage is complete. In what follows, we refer to the 2D formulation for simplicity; however, extension to 3D is straightforward, and the results of the 3D calculations are reported later.

## Continuum model and elastic theory

Here we shall summarize the mechanical problem in a 2D heterogeneous solid where the position of material points is denoted by  $\mathbf{X}^T = [x, y]$ . We denote the displacement components by  $U$  and  $V$  in the  $x$  and  $y$  directions, respectively. The kinematic relation defines the strain tensor  $\boldsymbol{\varepsilon}$  in terms of the deformation vector  $\mathbf{U}^T = [U, V]$  and reads:

$$\boldsymbol{\varepsilon} = \text{sym}(\text{grad}(\mathbf{U})) = \nabla^s \mathbf{U} = \frac{1}{2}(\nabla \mathbf{U} + \nabla \mathbf{U}^T). \quad (3)$$

In the context of linear elasticity, we define the elastic energy of the solid as  $\psi_{lin} = \frac{1}{2} \boldsymbol{\varepsilon} : \mathbb{C}(\mathbf{X}) : \boldsymbol{\varepsilon}$ , where  $\mathbb{C}$  is the fourth-order elasticity tensor. Through the constitutive relation, one relates the stress tensor to the strain tensor via  $\boldsymbol{\sigma} = \mathbb{C}(\mathbf{X}) \boldsymbol{\varepsilon}$ , where we have  $\mathbb{C}(\mathbf{X}) = \Lambda(\mathbf{X}) \mathbf{I} \otimes \mathbf{I} + 2\mu(\mathbf{X}) \mathbb{I}^s$ . Defining  $\mathbf{I}$  as the second-order identity tensor and  $\mathbb{I}^s$  as the symmetric fourth-order identity tensor, the above relation can also be written in the following form

$$\boldsymbol{\sigma} = \Lambda(\mathbf{X}) \operatorname{tr}(\boldsymbol{\varepsilon}) \mathbf{I} + 2\mu(\mathbf{X}) \boldsymbol{\varepsilon}. \quad (4)$$

Here, we have position-dependent Lamé constants which can be written in terms of Young's modulus  $E$  and Poisson's ratio  $\nu$  as  $\Lambda = E\nu/[(1-2\nu)(1+\nu)]$  and  $\mu = E/[2(1+\nu)]$ . Here, the elastic properties are phase-dependent and vary throughout the microstructure. Finally, the mechanical equilibrium in the absence of body force, as well as the Dirichlet and Neumann boundary conditions, are written as:

$$\operatorname{div}(\boldsymbol{\sigma}) = \operatorname{div}(\mathbb{C}(\mathbf{X}) \nabla^s \mathbf{U}) = \mathbf{0} \quad \text{in } \Omega \quad (5)$$

$$\mathbf{U} = \bar{\mathbf{U}} \quad \text{on } \Gamma_D \quad (6)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \Gamma_N \quad (7)$$

In the above relations,  $\Omega$  and  $\Gamma$  denote the material points in the body and on the boundary area, respectively. Moreover, the Dirichlet and Neumann boundary conditions are introduced in Eq. (6) and Eq. (7), respectively. Rewriting in the Voigt notation, we have  $\hat{\boldsymbol{\sigma}} = \mathbf{C}(\mathbf{X}) \hat{\boldsymbol{\varepsilon}}$ . Considering the plane stress assumption in 2D, we write:

$$\mathbf{C}(\mathbf{X}) = \frac{E(\mathbf{X})}{1-\nu^2(\mathbf{X})} \begin{bmatrix} 1 & \nu(\mathbf{X}) & 0 \\ \nu(\mathbf{X}) & 1 & 0 \\ 0 & 0 & \frac{1-\nu(\mathbf{X})}{2} \end{bmatrix}. \quad (8)$$

## Finite element method

By introducing  $\delta \mathbf{U}^T = [\delta \mathbf{U}, \delta \mathbf{V}]$  as standard test functions and performing integration by parts, the weak form of the mechanical equilibrium problem reads:

$$\int_{\Omega} \delta \hat{\boldsymbol{\varepsilon}}^T \mathbf{C}(\mathbf{X}) \hat{\boldsymbol{\varepsilon}} dV - \int_{\Gamma_N} \delta \mathbf{U}^T \bar{\mathbf{t}} dA = 0. \quad (9)$$

The corresponding linear shape functions  $\mathbf{N}$  and the deformation matrix  $\mathbf{B}$  used to discretize the mechanical weak form. To compute these derivatives, we utilize the Jacobian matrix  $\mathbf{J} = \partial \mathbf{X} / \partial \boldsymbol{\xi}$ .

Utilizing the standard finite element method, for each element the deformation field, stress tensor  $\hat{\boldsymbol{\sigma}}$  as well as elasticity field  $E, \nu$  are approximated as

$$\mathbf{U} = \mathbf{N} \mathbf{U}_e, \mathbf{V} = \mathbf{N} \mathbf{V}_e, E = \mathbf{N} E_e, \nu = \mathbf{N} \nu_e, \hat{\boldsymbol{\sigma}} = \mathbf{C} \mathbf{B} \mathbf{U}_e. \quad (10)$$

Here,  $\mathbf{U}_e^T, \mathbf{V}_e^T, \nu_e$  and  $E_e^T$  are the nodal values of the deformation field and elastic properties of element  $e$ . One can write the so-called *discretized* residual vector for one element as

$$\mathbf{r}_e = - \int_{\Omega_e} [\mathbf{B}]^T \mathbf{C} [\mathbf{B}] \mathbf{U}_e dV + \int_{\Omega_e} [\mathbf{N}]^T \underbrace{\mathbf{C} [\mathbf{B} \mathbf{U}_e]^T}_{\hat{\mathbf{t}}} \mathbf{n} dS. \quad (11)$$

## FFT-based homogenization

Building upon the approach proposed by<sup>81</sup>, the FFT-based homogenization method begins with an additive decomposition of the strain field  $\boldsymbol{\varepsilon}$  into the

macroscopic averaged strain field  $\bar{\boldsymbol{\varepsilon}}$  and a fluctuation contribution  $\tilde{\boldsymbol{\varepsilon}}$  as  $\boldsymbol{\varepsilon} = \bar{\boldsymbol{\varepsilon}} + \tilde{\boldsymbol{\varepsilon}}$ .

The constitutive relationship, we consider a homogeneous medium with stiffness  $\mathbb{C}^0$ , leading to the expression for the stress field as  $\boldsymbol{\sigma} = \mathbb{C}^0 : (\bar{\boldsymbol{\varepsilon}} + \tilde{\boldsymbol{\varepsilon}}) + \boldsymbol{\tau}$ . Here,  $\boldsymbol{\tau}$  is the polarization stress, defined as  $\boldsymbol{\tau} = (\mathbb{C} - \mathbb{C}^0) : (\bar{\boldsymbol{\varepsilon}} + \tilde{\boldsymbol{\varepsilon}})$ . The balance of linear momentum will then take the form

$$\operatorname{div}(\mathbb{C}^0 : \tilde{\boldsymbol{\varepsilon}}) + \operatorname{div} \boldsymbol{\tau} = \mathbf{0}. \quad (12)$$

By employing the Green's function approach, Eq. (12) is converted to the integral equation

$$\tilde{\boldsymbol{\varepsilon}}(\mathbf{X}) = - \int_{\Omega} \mathbb{I}^0(\mathbf{X} - \mathbf{x}') : \boldsymbol{\tau}(\mathbf{X}') d\mathbf{X}', \quad (13)$$

which is known as the Lippmann-Schwinger equation. In Eq. (13),  $\mathbb{I}^0$  is the Lippmann-Schwinger or Green's operator, which is expressed by the following closed form in the Fourier space:

$$\hat{\mathbb{I}}_{ijkl}^0(\boldsymbol{\xi}) = \frac{1}{4\mu^0 |\boldsymbol{\xi}|^2} \left( \delta_{ki} \xi_j \xi_l + \delta_{li} \xi_k \xi_j + \delta_{kj} \xi_l \xi_i + \delta_{lj} \xi_k \xi_i \right) - \frac{(\lambda^0 + \mu^0)}{\mu^0 (\lambda^0 + 2\mu^0)} \frac{\xi_i \xi_j \xi_k \xi_l}{|\boldsymbol{\xi}|^4}. \quad (14)$$

Here,  $\boldsymbol{\xi}$  is the frequency vector,  $\delta$  denotes the Kronecker delta function, and  $\lambda^0$  and  $\mu^0$  are the Lamé constants of the homogeneous reference medium. Moreover,  $\hat{(\cdot)}$  is used to represent a quantity in the Fourier space. Finally, the total strain field in the real space is obtained via

$$\boldsymbol{\varepsilon}(\mathbf{x}) = \bar{\boldsymbol{\varepsilon}} - \mathcal{F}^{-1} \left( \hat{\mathbb{I}}^0(\boldsymbol{\xi}) : \hat{\boldsymbol{\tau}}(\boldsymbol{\xi}) \right). \quad (15)$$

## Physics-informed finite operator learning

This model relies solely on standard feed-forward neural networks. The computation of each component of the vector  $\mathbf{z}$  is expressed as follows:

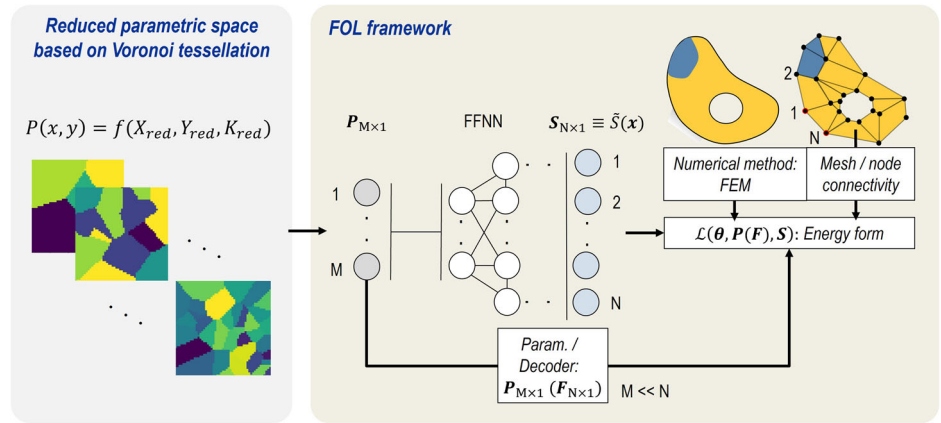
$$\mathbf{z}_m^l = a \left( \sum_{n=1}^{N_l} w_{mn}^l \mathbf{z}_n^{l-1} + b_m^l \right), \quad l = 1, \dots, L. \quad (16)$$

The component  $w_{mn}^l$  shows the weight between the  $n$ -th neuron of the layer  $l-1$  and the  $m$ -th neuron of the layer  $l$ . Every neuron in the  $l$ -th hidden layer owns a bias variable  $b_m^l$ . The number  $N_l$  corresponds to the number of neurons in the  $l$ -th hidden layer. The total number of hidden layers is  $L$ . The letter  $a$  stands for the activation function.

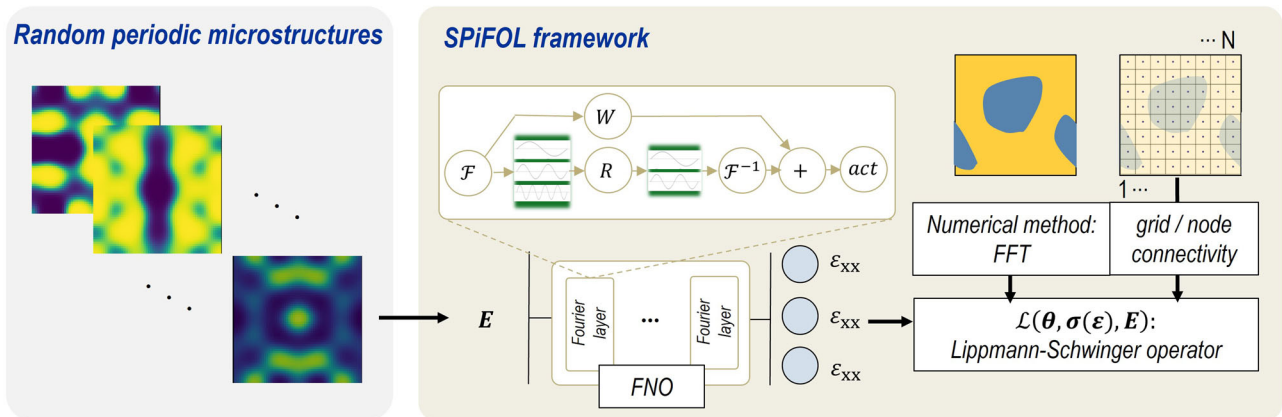
In the context of the current work which we coined as Finite Operator Learning (FOL)<sup>78</sup>, we utilize the so-called *collocation fields*, which constitute randomly generated and admissible parametric spaces used to train the neural network. For the current work, collocation fields represent possible choices for the elastic properties. More details on how to generate samples are provided in Supplementary Information. Therefore, the input layer consists of information on Young's modulus  $\{E_i\} = \{E_1, \dots, E_N\}$  and Poisson's ratios  $\{\nu_i\} = \{\nu_1, \dots, \nu_N\}$  at all the nodes, and the output layer consists of the components of the mechanical deformation at each discretization node  $i$ , i.e.,  $\{U_i\} = \{U_1, \dots, U_N\}$  and  $\{V_i\} = \{V_1, \dots, V_N\}$ . The model is summarized in the following steps which are also depicted on the right-hand side of Fig. 20.

$$\mathbf{X} = [\{E_i\}, \{\nu_i\}], \quad \mathbf{Y} = [\{U_i\}, \{V_i\}], \quad i = 1 \dots N, \quad (17)$$

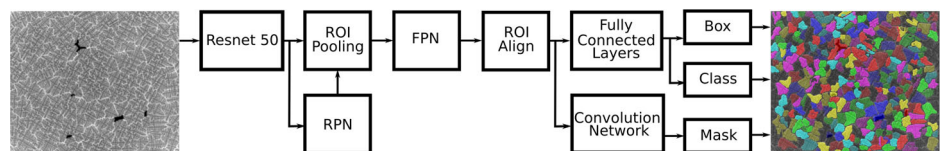
**Fig. 20 | Network architecture for finite operator learning (FOL).** Information about the Young's modulus distribution goes in and the unknown displacement components are evaluated utilizing deep neural networks.



**Fig. 21 | Network architecture for spectral-based physics-informed finite operator learning (SPiFOL).** Information about the Young's modulus distribution goes in and the unknown strain components are evaluated. Here, the Lippmann-Schwinger operator is built based on the fixed finite output space and FFT based physical loss is defined.



**Fig. 22 | Mask R-CNN Diagram with input and output images.** For more information, readers are referred to the supplementary information and ref. 77.



$$Y = \mathcal{N}(X; \theta), \quad \theta = \{W, b\}. \quad (18)$$

$$Y = \mathcal{N}(P; \theta), \quad \theta = \{W, b\}. \quad (20)$$

Here, the trainable parameters are represented as  $W$  and  $b$ , and are collectively denoted by  $\theta$ . The domain of interest is inspired by micro-mechanical examples and is thus represented as a square shape. However, this formulation is inherently flexible and can be readily adapted to accommodate irregular geometries and unstructured meshes<sup>78,82</sup>.

Given the polycrystalline nature of the samples, modeled using the concept of Voronoi tessellation, the input layer can be significantly simplified. The input now consists of the spatial coordinates of the seed points and the material properties associated with each seed point, effectively representing individual grains within the structure. We denote this as the so-called Voronoi parametrization,  $P_i = \{X_i, Y_i, E_i\}$ . Accordingly, we rewrite the expressions in Eq. (17) as follows:

$$P = \{X_j, Y_j, E_j\}, \quad Y = [U_i, V_i], \quad i = 1 \cdots N, \quad j = 1 \cdots M, \quad M \ll N, \quad (19)$$

In Fig. 20 and throughout this work, we assume a constant Poisson's ratio within the domain, focusing solely on variations in Young's modulus. This assumption does not limit the generality of the methodology. Moreover, the output of the model consists of the deformation components, while the stress components can be subsequently derived using standard FEM procedures.

Next, we introduce the loss function. The loss term simply combines the elemental energy form of the governing equation. Note that, thanks to the weak formulation, the Neumann boundary conditions are automatically included and the Dirichlet boundary terms are satisfied in a hard way exactly following the FEM routines (see also Rezaei et al.<sup>73</sup>). The total loss function involves the integration of element residual vectors using Gaussian integration, resulting in

$$\mathcal{L}_{FOL} = \sum_{e=1}^{n_d} U_e^T(\theta) [K_e U_e(\theta)], \quad (21)$$

$$K_e = \sum_{n=1}^{n_{\text{int}}} \frac{w_n}{2} \det(J) [\mathbf{B}(\xi_n)]^T \mathbf{C}_e(\xi_n) \mathbf{B}(\xi_n), \quad (22)$$

In the above set of equations,  $n_{\text{int}} = 4$  represents the number of Gaussian integration points and  $\xi_n$  and  $w_n$  denote the coordinates and weight of the  $n$ -th integration point. The determinant of the Jacobian matrix is denoted by  $\det(J)$ . The final loss function reads as

$$\mathcal{L} = \sum_{i=1}^{N_n} \left( (-\bar{\varepsilon} + \varepsilon(\mathbf{X}_i) + \mathcal{F}^{-1} [\hat{\Gamma}^0(\xi) \cdot \mathcal{F} [\mathbb{C}(\mathbf{X}_i) - \mathbb{C}^0] \cdot \varepsilon(\mathbf{X}_i)]) \right)^2, \quad (25)$$

$$\mathcal{L}_{\text{FOL}} = a_e \sum_{e=1}^{n_d} \mathbf{U}_e^T \left[ \sum_{n=1}^{n_{\text{int}}} \mathbf{B}^T \mathbf{C}_e \mathbf{B} \right]. \quad (23)$$

Note that we assume here that the determinant of the Jacobian matrix remains constant which holds only for quadrilateral elements. Additionally, in the four integration points we have  $w_n = 1$ . Based on the energy formulation, we obtain  $a_e = \frac{1}{2} \det(J)$ .

### Spectral-based physics-informed finite operator learning

For many applications in the multiscale modeling of materials, PBCs and periodic microstructure topologies are often preferred. The approach involves solving microscale problems and homogenizing the results to derive macroscopic properties. However, this process can become com-

square error (MSE) of Eq. (15). Due to the varying scales of strain field components, influenced by the applied macroscopic strain  $\bar{\varepsilon}$ , a weighting scheme is applied to normalize these components, ensuring comparable magnitudes in the loss function. This normalization enhances model accuracy. The final loss is constructed as the sum of weighted losses over all spatial points, defined as: where  $N_n$  is the total number of points and  $\mathbf{X}_i$  stands for the coordinates at that point. The MSE can be calculated for each strain component to ensure that the network learns the solution for each

component and that their corresponding loss functions are in the same order.

$$\mathcal{L} = w_1 \underbrace{\text{MAE}(\mathbf{L}(1, 1))}_{\text{corresponds to } \varepsilon_{xx}} + w_2 \underbrace{\text{MAE}(\mathbf{L}(1, 2) + \mathbf{L}(2, 1))}_{\text{corresponds to } \varepsilon_{xy} + \varepsilon_{yx}} + w_3 \underbrace{\text{MAE}(\mathbf{L}(2, 2))}_{\text{MAE}_{\varepsilon_{yy}}}. \quad (26)$$

In Eq. (26), MAE denotes the mean absolute error and  $w_1$ ,  $w_2$ , and  $w_3$  show the weighting factor for strain field components. These weighting factors can be adjusted by neural tangent kernels<sup>74,84</sup>. See also Figure 21. By considering Eq. (25), the final loss in Eq. (26) can be interpreted as MSE. When the weighting factors are equal, one can simply write the total loss as

$$\mathcal{L}_{\text{SPiFOL}} = \text{MSE} \left( \sum_{i=1}^{N_n} \left( (-\bar{\varepsilon} + \varepsilon(\mathbf{X}_i) + \mathcal{F}^{-1} [\hat{\Gamma}^0(\xi) \cdot \mathcal{F} [\mathbb{C}(\mathbf{X}_i) - \mathbb{C}^0] \cdot \varepsilon(\mathbf{X}_i)]) \right) \right). \quad (27)$$

putationally expensive, particularly for repetitive calculations in FE<sup>2</sup> methods. To address this challenge, we designed a novel deep learning model that operates on similar principles to the one introduced earlier, but with adaptations tailored to periodic systems and their unique demands. In this section, we employ Fourier Neural Operator (FNO) architectures<sup>83</sup> to map various microstructures to their corresponding strain fields ( $\varepsilon_{xx}$ ,  $\varepsilon_{yy}$ ,  $\varepsilon_{xy}$ ). The physical loss functions in this framework are formulated based on the strong form of mechanical equilibrium expressed in Fourier space. This approach is coined as spectral-based physics-informed finite operator learning (SPiFOL) according to the developments presented by Harandi et al.<sup>74</sup>.

In the SPiFOL framework, the FNO architecture begins with a dense layer that projects the input function into a higher-dimensional latent space, enabling sufficient channels for Fourier layers. Each Fourier layer transforms the input into the frequency domain using the Fourier transform ( $\mathcal{F}$ ), truncates higher frequencies via a weight tensor  $R$ , applies convolution in the truncated space, and then returns to the spatial domain using the inverse transform ( $\mathcal{F}^{-1}$ ). A non-linear activation (e.g., GeLU) is applied, and a final dense layer maps the output to the target strain fields. The computation of the output at the  $l + 1$ -th Fourier layer is expressed as:

$$\mathbf{z}^{l+1} = \text{GeLU}(\mathcal{F}^{-1}(R \cdot \mathcal{F}(\mathbf{z}^l)) + \mathbf{W}_l \cdot \mathbf{z}^l + \mathbf{b}_l). \quad (24)$$

In Eq. (24),  $\mathbf{b}_l$  is the bias,  $\mathbf{W}_l$  is the weight matrix, and  $\mathbf{z}^l$  refers to the output of the previous Fourier layer. The loss function is formulated using the output and the computed Lippmann-Schwinger operator, following the fixed-point scheme for FFT-based mechanical methods. It is expressed as the mean

### Digitalization of LOM images through deep learning-based image segmentation

To test the trained networks from the previous sections, it is essential to automatically delimit grains in LOM images. This enables either direct input of the images into the trained neural operators or the design of additional test cases that mimic similar topologies.

A useful methodology is presented and discussed in ref. 77 where a deep learning model called Mask R-CNN<sup>11,77</sup> is utilized to automatically detect and segment closely spaced dendrites in CT tomography sections of polycrystalline Al-Cu alloys. UNet performs semantic segmentation by grouping all grains of the same class together, while Mask R-CNN achieves instance segmentation, identifying and isolating each grain individually, even if they belong to the same class. This model was chosen due to the difficulty of separating dendrites that are closely packed or interconnected. Mask R-CNN is a deep learning model that performs three tasks simultaneously: object detection, bounding box prediction, and mask generation for each object in the image. The model starts by using a convolutional neural network to extract features from the input image. Then, a Region Proposal Network (RPN) identifies regions of interest. For each region, the model predicts the object class, refines the bounding box, and generates a pixel-wise mask, allowing for precise segmentation of even overlapping objects. This makes Mask R-CNN highly effective for tasks where detailed, instance-level segmentation is required, such as detecting and segmenting dendrites or grains in microstructures.

Here is a short description of the steps of the Mask R-CNN algorithm for alloy microstructure segmentation illustrated on

Fig. 22. The input image, typically obtained through a microscope, displays the alloy's microstructure, revealing various phases and grains within the sample. This image is processed through a feature extraction network, commonly ResNet50, which identifies hierarchical features such as edges, textures, and patterns relevant to the microstructure. A RPN then generates region proposals that may contain distinct microstructural features like grains or precipitates. These proposed regions are standardized using RoI Pooling, allowing consistent processing across the network. The extracted features are further refined using a Feature Pyramid Network, which creates feature maps at multiple scales to detect both small and large structures effectively. RoI Align follows, ensuring proposed regions are uniformly resized while maintaining spatial accuracy. Fully connected layers handle classification and bounding box regression based on these standardized features. Additionally, a convolutional network generates spatial masks for each region, enabling instance segmentation to precisely identify object shapes. The final output includes an image annotated with masks and bounding boxes that delineate and highlight the various microstructural components of the alloy.

Details on the data generation and training of the Mask R-CNN model are provided and summarized in Supplementary Information.

The results from applying Mask R-CNN achieving a detection rate of more than 90% as illustrated in Fig. 22, where the input image for inference and mask results of grain segmentation are shown. The model successfully segmented most dendrites, even in areas where they were densely packed or strongly interconnected. It segments dendrites based on the white boundaries delimiting the grains, where high segregation is present. However, some grains at the edges were not well detected, as they were cropped from a larger original experimental image and therefore do not share the same characteristics as the traditional dendrites in the training/test dataset.

#### Fourier-based parametrization

The Fourier-based approach combines specific frequencies with random amplitudes to generate diverse microstructures. To enhance variability, the *sigmoid* function is applied as a transformation, enabling the creation of smoother and more intricate patterns.

$$\phi = \frac{\text{sigmoid}(t_1(\phi^* - t_2)) + 0.05}{1 + 0.05}, \quad (28)$$

where  $\phi^*$ , the summation value of frequencies and random amplitudes, is defined as

$$\phi^* = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} a_i a_j \cos(2\pi f_{x_i} x) \cos(2\pi f_{y_j} y). \quad (29)$$

In Eq. (28),  $\phi$  devotes to the final distribution of phases.  $t_1$  and  $t_2$  are the tuning parameters of the *sigmoid* function. Moreover,  $a_i$  and  $a_j$  denote the normalized random amplitudes, while  $f_{x_i}$  and  $f_{y_j}$  represent the Fourier frequencies in  $x$  and  $y$  directions and are integers.  $x$  and  $y$  show the coordinates of mesh points.

#### Data availability

Data are provided within the manuscript or supplementary information files. Any additional raw data that support the findings in this paper are available from the corresponding author upon reasonable request. No labeled data are used for training the operator learning models; the code for generating training samples is provided in the corresponding section.

#### Code availability

The data supporting the findings of this study are openly available and can be accessed via the following links: [FOL framework](#) and [SPiFOL framework](#).

Received: 21 December 2024; Accepted: 27 June 2025;

Published online: 13 August 2025

#### References

1. Raabe, D. et al. Circular steel for fast decarbonization: thermodynamics, kinetics, and microstructure behind upcycling scrap into high-performance sheet steel. *Annu. Rev. Mater. Res.* **54**, 247–297 (2024).
2. Grieves, M. Digital twin: manufacturing excellence through virtual factory replication. *White paper* **1**, 1–7 (2014).
3. Tao, F., Zhang, H., Liu, A. & Nee, A. Y. C. Digital twin in industry: state-of-the-art. *IEEE Trans. Ind. Inform.* **15**, 2405–2415 (2019).
4. Bergs, T. et al. The concept of digital twin and digital shadow in manufacturing. *Procedia CIRP* **101**, 81–84 (2021).
5. Wang, J. et al. Concept for databased material property description along the process chain press hardening: implementing the digital material twin and the digital material shadow. *Comput. Mater. Sci.* **249**, 113666 (2025).
6. Becker, F. et al. A conceptual model for digital shadows in industry and its application. In *Conceptual Modeling* (eds Ghose, A., Horkoff, J., Silva Souza, V. E., Parsons, J. & Evermann, J.) 271–281 (Springer International Publishing, 2021).
7. Rajaei, A. et al. *Materials in the Drive Chain—Modeling Materials for the Internet of Production* 1–21 (Springer International Publishing, 2023).
8. Holm, E. A. et al. Overview: computer vision and machine learning for microstructural characterization and analysis. *Metall. Mater. Trans. A* **51**, 5985–5999 (2020).
9. Alrfou, K., Zhao, T. & Kordijazi, A. Deep learning methods for microstructural image analysis: the state-of-the-art and future perspectives. *IMMI* **13**, 703–731 (2024).
10. Agbozo, R. & Jin, W.-Y. Quantitative metallographic analysis of GCr15 microstructure using mask R-CNN. *JKSPE* **37**, 361–369 (2020).
11. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. <https://arxiv.org/abs/1703.06870> (2018).
12. Fu, L., Yu, H., Shah, M., Simmons, J. & Wang, S. Crystallographic symmetry for data augmentation in detecting dendrite cores. *J. Electron. Imaging* **32**, 1–7 (2020).
13. Ren, S., He, K., Girshick, R. B. & Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. <https://doi.org/10.48550/arXiv.1506.01497> (2015).
14. Mulewicz, B., Korpala, G., Kusiak, J. & Prah, U. Autonomous interpretation of the microstructure of steels and special alloys. *Mater. Sci. Forum* **949**, 24–31 (2019).
15. Elbana, R., Mostafa, R. & Elkeran, A. Data processing for automatic classification of spheroidite microstructure using deep learning based on FNNs. *Int. J. Mech. Mechatron. Eng.* **20**, 18–31 (2020).
16. Banerjee, D. & Sparks, T. D. Comparing transfer learning to feature optimization in microstructure classification. *iScience* **25**, 103774 (2022).
17. Liotti, E. et al. Crystal nucleation in metallic alloys using X-ray radiography and machine learning. *Sci. Adv.* **4**, eaar4004 (2018).
18. Breumier, S. et al. Leveraging EBSD data by deep learning for bainite, ferrite and martensite segmentation. *Mater. Charact.* **186**, 111805 (2022).
19. Martinez Ostomujof, T. et al. Deep learning for automated phase segmentation in EBSD maps: a case study in dual phase steel microstructures. *Mater. Charact.* **184**, 111638 (2022).
20. Germain, L., Sertucha, J., Hazotte, A. & Lacaze, J. Classification of graphite particles in metallographic images of cast irons: Quantitative image analysis versus deep learning. *Mater. Charact.* **217**, 114333 (2024).
21. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. <https://doi.org/10.48550/arXiv.1505.04597> (2015).
22. Gierden, C., Kochmann, J., Waimann, J., Kabel, M. & Reese, S. A review of FFT-based two-scale methods for computational modeling of microstructure evolution and macroscopic material behavior. *Arch. Comput. Methods Eng.* **29**, 4115–4135 (2022).

23. Benedetti, I. An integral framework for computational thermo-elastic homogenization of polycrystalline materials. *Comput. Methods Appl. Mech. Eng.* **407**, 115927 (2023).
24. Galvis, A. F., Rodríguez, R. Q. & Sollero, P. Analysis of three-dimensional hexagonal and cubic polycrystals using the boundary element method. *Mech. Mater.* **117**, 58–72 (2018).
25. Benedetti, I. & Aliabadi, M. Multiscale modeling of polycrystalline materials: a boundary element approach to material degradation and fracture. *Comput. Methods Appl. Mech. Eng.* **289**, 429–453 (2015).
26. Liu, T.-R., Aldakheel, F. & Aliabadi, M. Virtual element method for phase field modeling of dynamic fracture. *Comput. Methods Appl. Mech. Eng.* **411**, 116050 (2023).
27. Lo Cascio, M., Milazzo, A. & Benedetti, I. Virtual element method for computational homogenization of composite and heterogeneous materials. *Compos. Struct.* **232**, 111523 (2020).
28. Böhm, C. et al. Virtual elements for computational anisotropic crystal plasticity. *Comput. Methods Appl. Mech. Eng.* **405**, 115835 (2023).
29. Liu, W. K., Li, S. & Park, H. S. Eighty years of the finite element method: birth, evolution, and future. *Arch. Comput. Methods. Eng.* **29**, 4431–4453 (2022).
30. Feyel, F. & Chaboche, J.-L. Fe2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials. *Comput. Methods Appl. Mech. Eng.* **183**, 309–330 (2000).
31. Geers, M., Kouznetsova, V. & Brekelmans, W. Multi-scale computational homogenization: trends and challenges. *J. Comput. Appl. Math.* **234**, 2175–2182 (2010).
32. Jörg Schröder. *A Numerical Two-Scale Homogenization Scheme: The FE2-Method* 1–64. ISBN 978-3-7091-1625-8. [https://doi.org/10.1007/978-3-7091-1625-8\\_1](https://doi.org/10.1007/978-3-7091-1625-8_1) (Springer Vienna, 2014).
33. Kalina, K. A., Linden, L., Brummund, J. & Kästner, M. Fe<sup>ann</sup>: An efficient data-driven multiscale approach based on physics-constrained neural networks and automated data mining. *Comput. Mech.* **71**, 827–851 (2023).
34. Mianroodi, J. et al. Lossless multi-scale constitutive elastic relations with artificial intelligence. *npj Comput. Mater.* **8**, 67 (2022).
35. Faroughi, S. A. et al. Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: fluid and solid mechanics. *J. Comput. Inf. Sci. Eng.* **24**, 040802 (2024).
36. Herrmann, L. & Kollmannsberger, S. Deep learning in computational mechanics: a review. *Comput. Mech.* **74**, 281–331 (2024).
37. Kim, D. & Lee, J. A review of physics informed neural networks for multiscale analysis and inverse problems. *Multiscale Sci. Eng.* **6**, 1–11 (2024).
38. Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **3**, 218–229 (2021).
39. Li, Z. et al. Fourier neural operator for parametric partial differential equations. <https://doi.org/10.48550/arXiv.2010.08895> (2021).
40. Li, Z. et al. Neural operator: Graph kernel network for partial differential equations. <https://doi.org/10.48550/arXiv.2003.03485> (2020).
41. Chen, G. et al. Learning neural operators on Riemannian manifolds. <https://doi.org/10.48550/arXiv.2302.08166> (2023).
42. Cao, Q., Goswami, S. & Karniadakis, G. E. Laplace neural operator for solving differential equations. *Nat. Mach. Intell.* **6**, 631–640 (2024).
43. Winovich, N., Ramani, K. & Lin, G. Convpde-ug: convolutional neural networks with quantified uncertainty for heterogeneous elliptic partial differential equations on varied domains. *J. Comput. Phys.* **394**, 263–279 (2019).
44. Yang, Z., Yu, C.-H. & Buehler, M. J. Deep learning model to predict complex stress and strain fields in hierarchical composites. *Sci. Adv.* **7**, eabd7416 (2021).
45. Mohammadzadeh, S. & Lejeune, E. Predicting mechanically driven full-field quantities of interest with deep learning-based metamodelling. *Extrem. Mech. Lett.* **50**, 101566 (2022).
46. Koric, S., Viswanta, A., Abueidda, D. W., Sobh, N. A. & Khan, K. Deep learning operator network for plastic deformation with variable loads and material properties. *Eng. Comput.* **40**, 917–929 (2023).
47. Lu, M. et al. Deep neural operator for learning transient response of interpenetrating phase composites subject to dynamic loading. *Comput. Mech.* **72**, 563–576 (2023).
48. He, J. et al. Novel deepnet architecture to predict stresses in elastoplastic structures with variable complex geometries and loads. *Comput. Methods Appl. Mech. Eng.* **415**, 116277 (2023).
49. Lu, L. et al. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Comput. Methods Appl. Mech. Eng.* **393**, 114778 (2022).
50. Rashid, M. M., Chakraborty, S. & Krishnan, N. A. Revealing the predictive power of neural operators for strain evolution in digital composites. *J. Mech. Phys. Solids* **181**, 105444 (2023).
51. Lagaris, I., Likas, A. & Fotiadis, D. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.* **9**, 987–1000 (1998).
52. Sirignano, J. & Spiliopoulos, K. DGM: a deep learning algorithm for solving partial differential equations. *J. Comput. Phys.* **375**, 1339–1364 (2018).
53. Raissi, M., Perdikaris, P. & Karniadakis, G. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).
54. Sun, L., Gao, H., Pan, S. & Wang, J.-X. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Comput. Methods Appl. Mech. Eng.* **361**, 112732 (2020).
55. Geneva, N. & Zabarar, N. Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. *J. Comput. Phys.* **403**, 109056 (2020).
56. Haghighat, E., Raissi, M., Moure, A., Gomez, H. & Juanes, R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput. Methods Appl. Mech. Eng.* **379**, 113741 (2021).
57. Rezaei, S., Harandi, A., Moeineddin, A., Xu, B.-X. & Reese, S. A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: comparison with finite element method. *Comput. Methods Appl. Mech. Eng.* **401**, 115616 (2022).
58. Wu, J., Jiang, J., Chen, Q., Chatzigeorgiou, G. & Meraghni, F. Deep homogenization networks for elastic heterogeneous materials with two- and three-dimensional periodicity. *Int. J. Solids Struct.* **284**, 112521 (2023).
59. Roy, A. M., Bose, R., Sundararaghavan, V. & Arróyave, R. Deep learning-accelerated computational framework based on physics informed neural network for the solution of linear elasticity. *Neural Netw.* **162**, 472–489 (2023).
60. Rezaei, S., Moeineddin, A. & Harandi, A. Learning solutions of thermodynamics-based nonlinear constitutive material models using physics-informed neural networks. *Comput. Mech.* **74**, 333–366 (2024).
61. Haghighat, E., Abouali, S. & Vaziri, R. Constitutive model characterization and discovery using physics-informed deep learning. *Eng. Appl. Artif. Intell.* **120**, 105828 (2023).
62. Li, Z. et al. Physics-informed neural operator for learning partial differential equations. <https://doi.org/10.48550/arXiv.2111.03794> (2023).
63. Wang, S., Wang, H. & Perdikaris, P. Learning the solution operator of parametric partial differential equations with physics-informed deepnets. *Sci. Adv.* **7**, eabi8605 (2021).
64. Zhu, Y., Zabarar, N., Koutsourelakis, P.-S. & Perdikaris, P. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *J. Comput. Phys.* **394**, 56–81 (2019).
65. Gao, H., Sun, L. & Wang, J.-X. Phygeonet: Physics-informed geometry-adaptive convolutional neural networks for solving

- parameterized steady-state pdes on irregular domain. *J. Comput. Phys.* **428**, 110079 (2021).
66. Liu, X.-Y., Zhu, M., Lu, L., Sun, H. & Wang, J.-X. Multi-resolution partial differential equations preserved learning framework for spatiotemporal dynamics. *Commun. Phys.* **7**, 31 (2024).
  67. Zhang, Z. & Gu, G. X. Physics-informed deep learning for digital materials. *Theor. Appl. Mech. Lett.* **11**, 100220 (2021).
  68. Zhang, X. & Garikipati, K. Label-free learning of elliptic partial differential equation solvers with generalizability across boundary value problems. *Comput. Methods Appl. Mech. Eng.* **417**, 116214 (2023).
  69. Ackermann, M., Iren, D., Wesselmecking, S., Shetty, D. & Krupp, U. Automated segmentation of martensite-austenite islands in bainitic steel. *Mater. Charact.* **191**, 112091 (2022).
  70. Ackermann, M., Iren, D. & Yao, Y. Explainable machine learning for predicting the mechanical properties in bainitic steels. *Mater. Des.* **230**, 111946 (2023).
  71. Laschet, G., Abouridouane, M., Fernández, M., Budnitski, M. & Bergs, T. Microstructure impact on the machining of two gear steels. Part 1: derivation of effective flow curves. *Mater. Sci. Eng. A* **845**, 143125 (2022).
  72. Vogiatzief, D., Evirgen, A., Pedersen, M. & Hecht, U. Laser powder bed fusion of an Al-Cr-Fe-Ni high-entropy alloy produced by blending of prealloyed and elemental powder: process parameters, microstructures and mechanical properties. *J. Alloy. Compd.* **918**, 165658 (2022).
  73. Rezaei, S. et al. A finite operator learning technique for mapping the elastic properties of microstructures to their mechanical deformations. *Int. J. Numer. Methods Eng.* **126**, e7637 (2025).
  74. Harandi, A., Danesh, H., Linka, K., Reese, S. & Rezaei, S. Spifol: a spectral-based physics-informed finite operator learning for prediction of mechanical behavior of microstructures. *J. Mech. Phys. Solids* **203**, 106219 (2025).
  75. Bradbury, J. et al. JAX: composable transformations of Python +NumPy programs <http://github.com/google/jax> (2018).
  76. Koopas, R. N., Rezaei, S., Rauter, N., Ostwald, R. & Lammering, R. Introducing a microstructure-embedded autoencoder approach for reconstructing high-resolution solution field from reduced parametric space. *Comput. Mech.* **75**, 1377–1406 (2024).
  77. Viardin, A., Nöth, K., Pickmann, C. & Sturz, L. Automatic detection of dendritic microstructure using computer vision deep learning models trained with phase field simulations. *Integr. Mater. Manuf. Innov.* **14**, 89–105 (2025).
  78. Rezaei, S. et al. Finite operator learning: bridging neural operators and numerical methods for efficient parametric solution and optimization of PDEs. <https://arxiv.org/abs/2407.04157> (2024).
  79. Asl, R. N. et al. A physics-informed meta-learning framework for the continuous solution of parametric PDEs on arbitrary geometries <https://arxiv.org/abs/2504.02459> (2025).
  80. Niu, S., Zhang, E., Bazilevs, Y. & Srivastava, V. Modeling finite-strain plasticity using physics-informed neural network and assessment of the network performance. *J. Mech. Phys. Solids* **172**, 105177 (2023).
  81. Moulinec, H. & Suquet, P. A numerical method for computing the overall response of nonlinear composites with complex microstructure. *Comput. Methods Appl. Mech. Eng.* **157**, 69–94 (1998).
  82. Yamazaki, Y. et al. A finite element-based physics-informed operator learning framework for spatiotemporal partial differential equations on arbitrary domains. *Eng. Comput.* **41**, 1–29 (2025).
  83. Li, Z. et al. Fourier neural operator for parametric partial differential equations. <https://doi.org/10.48550/arXiv.2010.08895> (2020).
  84. Wang, S., Yu, X. & Perdikaris, P. When and why pinns fail to train: a neural tangent kernel perspective. *J. Comput. Phys.* **449**, 110768 (2022).
  85. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. <https://doi.org/10.48550/arXiv.1412.6980> (2017).

## Acknowledgements

The authors would like to thank the Deutsche Forschungsgemeinschaft (DFG) for the funding support provided to develop the present work in the project Cluster of Excellence “Internet of Production” (project: 390621612).

## Author contributions

S.R.: Conceptualization, Supervision, Software, Writing—Review and Editing. K.T.: Investigation, Software. A.V.: Software, Writing—Review and Editing. R.N.A.: Methodology, Software. A.H.: Investigation, Software. N.V.J.: Investigation, Writing—Review and Editing. D.B.: Review and Editing. H.N.: Experiment, Investigation, Writing—Review. A.G.: Review and Editing. T.B.: Review and Editing. M.A.: Writing—Review and Editing. Th.B.: Funding, Supervision, Review, and Editing. U.K.: Funding, Supervision, Review, and Editing. M.A.: Funding, Supervision, Review, and Editing.

## Funding

Open Access funding enabled and organized by Projekt DEAL.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41524-025-01718-y>.

**Correspondence** and requests for materials should be addressed to Shahed Rezaei.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher’s note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2025