

A physics-informed 3D surrogate model for elastic fields in polycrystals

Lucas Monteiro Fernandes^{a,d,*}, Samy Blusseau^a, Philipp Rieder^b, Matthias Neumann^c, Volker Schmidt^b,
Henry Proudhon^d, François Willot^{a,d}

^a*Center for Mathematical Morphology (CMM), Mines Paris, PSL University, Fontainebleau, France*

^b*Institute of Stochastics, Ulm University, Ulm, Germany*

^c*Institute of Statistics, Graz University of Technology, Graz, Austria*

^d*Center of Materials (CMAT), Mines Paris, PSL University, Evry Cedex, France*

Abstract

We develop a physics-informed neural network pipeline for solving linear elastic micromechanics in three dimensions, on a statistical volume element (SVE) of a polycrystalline material with periodic geometry. The presented approach combines a convolutional neural network containing residual connections with physics-informed non-trainable layers. The latter are introduced to enforce the strain field admissibility and the constitutive law in a way consistent with so-called fast Fourier transform (FFT) algorithms. More precisely, differential operators are discretized by finite differences in accordance with the Green operator used in FFT computations and treated as convolutions with fixed kernels. The deterministic relationship between crystalline orientations and stiffness tensors is transferred to the network by an additional non-trainable layer. A loss function dependent on the divergence of the predicted stress field allows for updating the neural network's parameters without further supervision from ground truth data. The surrogate model is trained on untextured synthetic polycrystalline SVEs with periodic boundary conditions, realized from a stochastic 3D microstructure model based on random tessellations. Once trained, the network is able to predict the periodic part of the displacement field from the crystalline orientation field (represented as unit quaternions) of an SVE. The proposed self-supervised pipeline is compared to a similar one trained with a data-driven loss function instead. Further, the accuracy of both models is analyzed by applying them to microstructures larger than the training inputs, as well as to SVEs generated by the stochastic 3D microstructure model, utilizing various different parameters. We find that the self-supervised pipeline yields more accurate predictions than the data-driven one, at the expense of a longer training. Finally, we discuss how the trained surrogate model can be used to solve certain inverse problems on polycrystalline domains by gradient descent.

Keywords: Physics-informed, surrogate model, polycrystalline materials, statistical volume element, periodic boundary conditions, linear elasticity.

1. Introduction

The determination of the mechanical response of polycrystalline materials is a long-standing subject of research that involves complex physical phenomena at different scales [1, 2, 3]. Homogenization theory [4, 5] and computational tools such as the finite element method (FEM) [6] and fast Fourier transform (FFT) [7] techniques have commonly been employed to account for both microstructural and mechanical aspects, through constitutive equations that model the material's behavior. In polycrystalline materials, microscopic local fields emerge from the interaction between grains, i.e. small anisotropic monocrystalline domains. To understand how the morphology of grains and their crystalline orientations impact mechanical properties, a high-dimensional space must be explored, which limits the use of finite element computations due to their high computational cost [8]. Whenever periodic boundary conditions are appropriate, numerical computations

*Corresponding author. E-mail address: lucas.monteiro_fernandes@minesparis.psl.eu

on representative volume elements (RVEs) can be carried out instead with FFT solvers, which are based on Green’s function [9] and allow for cheaper computations [10, 11].

More recently, machine learning (ML) techniques have been developed as well in order to address mechanical problems in materials engineering [12]. These methods include forward problems, i.e. the discovery of solutions to partial differential equations (PDEs) as well as inverse design problems, that is, finding structures or materials with prescribed properties. In particular, specialized neural networks have been developed to predict the effective mechanical properties of composite materials [13]. Notably, graph neural networks (GNN) have been proposed to predict effective properties of polycrystalline microstructures by mathematically treating them as a graph [14]. Additionally, ML techniques have been used for parametric multi-objective optimization of crystallographic orientation distribution functions employing Rodrigues axis-angle parametrization [15]. While that method improves the interpretability of microstructure descriptors, defining the optimization objectives require pre-processing steps. Recent works have combined CNNs with image data of heterogeneous materials, including modeling of nonlinear behavior [16, 17].

While the data-driven or supervised approach for surrogate models of differential systems has been the state-of-the-art for a long time, the physics-informed perspective not only improved generalization capability, but also reduced the amount of training data required. This paradigm, pioneered by Lagaris *et al.* [18] and rediscovered with modern implementations by Raissi *et al.* [19] enables the injection of conservation and evolution laws into loss functions of artificial neural networks, resulting in ML algorithms that better capture the underlying laws between initial or boundary conditions and simulation outputs. Such neural networks are commonly called physics-informed neural networks (PINNs). Neural networks that minimize the potential energy [20] were also proposed to alleviate numerical instabilities frequently occurring when solving physical PDEs. More generally, the PINN framework was applied to linear elastic micromechanics [21], where a multilayer perceptron takes the coordinates of each point in the undeformed material as input features and outputs both 2D displacement and stress fields. While succeeding in achieving learning in a self-supervised manner and for different types of boundary conditions, the model is used as an exact solver rather than as a surrogate model and, consequently, cannot be applied to unseen geometries without retraining. Therefore, the PINN approach can benefit from architectures that take spatial correlations into account such as CNNs or GNNs. Convolutional PINNs have been applied to self-supervised segmentation of images of composite materials [22], in which the approach of discretizing differential operators such as gradient and divergence of tensor fields as non-trainable convolutions is particularly useful. More recently, transformer neural networks have been applied to predicting elastic fields of 2D composite materials, with improvements in computational costs compared to FFT algorithms [23] for high Young’s modulus ratio. Other applications of PINNs to micromechanics concern the combined use of genetic algorithms and deep learning techniques to tune the mechanical properties of polycrystalline materials [24], in which efforts are made towards the optimization of that class of materials, however with a fully-supervised approach applied to 2D images.

Standard solvers remain the preferred way to solve systems of PDEs in complex heterogeneous media because of their reliability and determinism. However, the PINN framework is proven to be particularly convenient for solving inverse problems [25] and optimization problems [26], which cannot be tackled with simple micromechanical solvers and require additional algorithms. For instance, some of such non-trivial problems could be addressed with neural networks and gradient descent schemes, by means of a fully differentiable neural network that estimates linear elastic full-fields of SVEs defined on polycrystalline domains. As a step towards that, we apply concepts from PINNs and stochastic geometry to train a neural network that maps the discretized field of crystalline orientations of an SVE to the solution field of the linear elastic problem with periodic boundary conditions. The rest of this paper is organized as follows. The stochastic 3D microstructure model and the physical problem to be solved are described in Sections 2 and 3, respectively. Both surrogate models, one trained with a self-supervised learning strategy, and another trained with a data-driven approach, are detailed in Section 4. The predictions of the two methods are compared with respect to performance and computational cost in Section 5. The generalizability for larger input microstructures and for slightly different grain morphologies is also analyzed. Additionally, two potential applications of the proposed surrogate model are formulated, namely: finding a crystalline orientation field that matches a given strain field, and optimizing the set of orientations for an objective based on the strain energy field

spatial maximum and penalization on the macroscopic anisotropy. Finally, Section 6 concludes.

2. Stochastic 3D model of virtual polycrystals

In this section, a stochastic 3D microstructure model to generate virtual polycrystals is introduced. Realizations of this model are employed as training input for the surrogate models introduced in Section 4. The present stochastic 3D microstructure model is a special case of the one proposed in [27], which allows for modeling crystallographic twin relationships between neighboring grains. Hereafter, all realizations of the virtual polycrystals are defined on a bounded cubic sampling window $\Omega \subset \mathbb{R}^3$ representing an elementary cell of a periodic material.

To model the polycrystalline grain architecture, a Voronoi tessellation (see [28, Section 9.2]) was employed. Based on a set of so-called seed points $\mathbf{s}_1, \dots, \mathbf{s}_N \in \Omega$ for some integer $N \in \mathbb{N}$, the Voronoi tessellation subdivides the sampling window Ω into N subsets with non-overlapping interiors, called grains. For $1 \leq n \leq N$, the n -th grain G_n consists of all points $\mathbf{x} \in \Omega$, whose distance to the seed point $\mathbf{s}_n \in \Omega$ is smaller than or equal to the distances to all other seed points. Mathematically, G_n is defined by

$$G_n = \{\mathbf{x} \in \Omega : d_{\text{per}}(\mathbf{x}, \mathbf{s}_n) \leq d_{\text{per}}(\mathbf{x}, \mathbf{s}_m) \text{ for all } m \neq n\}, \quad (1)$$

where $d_{\text{per}} : \Omega \times \Omega \rightarrow [0, \infty)$ denotes the Euclidean distance in Ω with respect to periodic boundary conditions. Note that all grains of the Voronoi tessellation G_1, \dots, G_N are convex polytopes.

The seed points $\mathbf{s}_1, \dots, \mathbf{s}_N$ of the Voronoi tessellation are modeled by a Matérn hardcore point process, with some intensity $\lambda > 0$ and hardcore radius $r > 0$, see Section 5.4 in [28]. The hardcore condition is understood with respect to the periodic distance in Ω . This point process is a thinning of a homogeneous Poisson point process, such that the distance $d_{\text{per}}(\cdot, \cdot)$ between an arbitrary pair of points is larger than the hardcore radius r .

Furthermore, each grain G of the Voronoi tessellation G_1, \dots, G_N is assigned a crystallographic orientation O , represented as an element of SO_3 , the space of rotations in \mathbb{R}^3 around the origin. Each orientation O is sampled independently from the uniform distribution on SO_3 , according to the algorithm provided in [29]. This procedure mimics an untextured polycrystal. We follow [30] and represent orientations $O \in \text{SO}_3$ as unit quaternions. Each unit quaternion, in turn, can be expressed as an element on the 3-sphere $\mathbb{S}^3 = \{\mathbf{q} \in \mathbb{R}^4 : |\mathbf{q}| = 1\} \subset \mathbb{R}^4$.

The surrogate models described in Section 4 are trained on realizations of the stochastic 3D model on a cubic sampling window $\Omega = [0, 32]^3$, whose discretization contains 32^3 voxels. The underlying point patterns were generated by a Matérn hardcore process with intensity $\lambda = 15/32^3 \approx 4.58 \cdot 10^{-4}$ and a hardcore radius $r = 5$. The expected number of grains in Ω is equal to 15, whereas each grain is represented by approximately 2200 voxels on average. This is sufficient to represent each grain as well as the fields within a crystal at a sufficiently high resolution, while limiting the computational cost for training the neural network.

To validate the performance of the models introduced in Section 4, they are applied to microstructures exhibiting statistically different properties compared to the ones used for model training. For this purpose, in Section 5, a larger sampling window $\Omega = [0, 64]^3$ is considered, containing approximately 120 grains at the same resolution. Furthermore, the two surrogate models were applied to microstructures drawn from the stochastic microstructure model, where the parameters λ and r of the Matérn hardcore process were systematically varied. Specifically, all combinations of intensities $\lambda \in \{12.5/32^3, 15/32^3, 17.5/32^3\}$ and hardcore radii $r \in \{2.5, 5, 7.5\}$ were taken into account. Realizations of the stochastic 3D model for different combinations of parameter values are presented on Figure 1.

3. Computational homogenization in linear elasticity

This section details the set of partial differential equations solved in linear elastic micromechanics. In statics, the strong form of the balance of momentum equation is given by

$$\text{div}(\boldsymbol{\sigma}) + \mathbf{f} = \mathbf{0}, \quad (2)$$

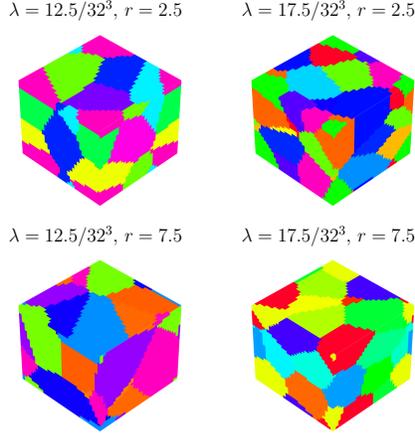


Figure 1: Realizations of the stochastic microstructure model for different combinations of parameter values (coloring based on grain labels).

where $\mathbf{0}$ denotes the null vector, $\boldsymbol{\sigma}$ the rank-2 Cauchy stress tensor, \mathbf{f} the body forces vector and $\text{div}(\cdot)$ the divergence operator, which is defined by $\text{div}(\boldsymbol{\sigma}) = \nabla \cdot \boldsymbol{\sigma}$. We assume that in the micromechanical scale (around several tens of micrometers), body forces arising from gravitational effects can be neglected: $\mathbf{f} = \mathbf{0}$. $\boldsymbol{\sigma}$ verifies

$$\mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n}, \quad (3)$$

where \mathbf{t} denotes a vector of surface forces, the symbol “ \cdot ” tensor contraction and \mathbf{n} the normal to a surface. In infinitesimal strain theory, a strain rank-2 tensor $\boldsymbol{\varepsilon}$ can be expressed as

$$\boldsymbol{\varepsilon} = \text{grad}_{\text{sym}}(\mathbf{u}), \quad (4)$$

where \mathbf{u} is the vector of displacements and $\text{grad}_{\text{sym}}(\cdot)$ denotes the symmetrized gradient operator given by $\text{grad}_{\text{sym}}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^\top)$. Furthermore, we assume a local linear elastic behavior, i.e.,

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}. \quad (5)$$

where \mathbb{C} denotes the local rank-4 stiffness tensor. Eqs. (2), (4) and (5) are assumed to be fulfilled everywhere in \mathbb{R}^3 . We also assume periodic boundary conditions, such that

$$\boldsymbol{\sigma} \cdot \mathbf{n} - \#, \quad \mathbf{u} = \mathbf{E} \cdot \mathbf{x} + \mathbf{u}^* \text{ for } \mathbf{x} \in \Omega, \quad \mathbf{u}^* \#, \quad (6)$$

where \mathbf{E} denotes the macroscopic strain tensor, \mathbf{u}^* the periodic part of the displacement vector, $\#$ the periodicity and $-\#$ anti-periodicity. By the divergence theorem [31], also referred to as Gauss theorem, it can be shown that $\langle \boldsymbol{\varepsilon} \rangle = \mathbf{E}$, where $\langle \cdot \rangle$ denotes spatial average over Ω . Note that

$$\langle u_{i,j}^* \rangle = \frac{1}{V_0} \int_{\Omega} u_{i,j}^* dV = \frac{1}{V_0} \int_{\partial\Omega} u_i^* n_j dS = 0, \quad (7)$$

where V_0 refers to the volume of Ω , dV to a volume element, dS to a surface element, and u_i^* to the i -th component of \mathbf{u}^* , while $u_{i,j}^* = \partial u_i^* / \partial x_j$. The work of internal forces yields

$$W_{int} = \int_{\Omega} \frac{1}{2} \boldsymbol{\sigma} : \boldsymbol{\varepsilon} dV = \frac{1}{2} \langle \boldsymbol{\sigma} : \boldsymbol{\varepsilon} \rangle = \frac{1}{2} \boldsymbol{\Sigma} : \mathbf{E}, \quad (8)$$

where the macroscopic stress tensor is defined as $\boldsymbol{\Sigma} = \langle \boldsymbol{\sigma} \rangle$. The effective stiffness tensor of the considered unit cell can be numerically obtained by imposing \mathbf{E} and computing $\boldsymbol{\sigma}$, for a variety of loads, such as

extension, compression and shear. In the following, we use FFT numerical schemes to estimate the tensor fields $\mathbf{u}^*(\mathbf{x}) \in \mathbb{R}^3$ and $\boldsymbol{\varepsilon}(\mathbf{x})$, $\boldsymbol{\sigma}(\mathbf{x}) \in \mathbb{R}^{3 \times 3}$ for $\mathbf{x} \in \Omega$, which depend on the full microstructure and crystal orientations. Furthermore, the divergence and gradient operators are discretized in the Green operator as backward and forward finite differences, respectively [32]. In Section 4, this same problem, including the discretization, is solved by training a surrogate neural network.

4. Physics-informed surrogate model for linear elastic computations of polycrystals

In this section, the surrogate models' architectures are described, as well as the motivations for the choice of their layers, both trainable and non-trainable. In Section 4.1, the complete framework is explained, with a highlight on the non-trainable physics-informed layers that introduce favorable inductive bias into the learning process. Subsequently, in Section 4.2, the trainable layers of the surrogate model are described, along with some intuition behind the choice of hyperparameters.

4.1. General approach

We now introduce a framework for surrogate modeling computations of the linear elastic behavior of polycrystalline materials. It makes combined use of non-trainable physics-informed layers and trainable layers. The update of the trainable parameters were done with two different approaches. The first one makes use of a self-supervised loss function while the second one employs a standard data-driven loss function for regression problems, i.e. the mean square error (MSE), between the predictions and the ground truth derived from FFT computations.

Recall from Section 3, that the stress and strain fields follow directly from the periodic part of the displacement field $\mathbf{u}^*(\mathbf{x})$. Therefore, for a given set of grainwise crystalline orientations represented by a unit quaternion field $\mathbf{q}(\mathbf{x}) \in \mathbb{R}^4$ for a $\mathbf{x} \in \Omega$ and a fixed macroscopic strain loading $\mathbf{E} \in \mathbb{R}^{3 \times 3}$, the predictions of the strain and stress fields follow from that of the periodic part of the displacement, denoted by

$$\hat{\mathbf{u}}^*(\mathbf{x}) = \mathcal{N}(\Lambda; \mathbf{q}(\mathbf{x}), \mathbf{E}), \quad (9)$$

where Λ is a set of learned parameters and \mathcal{N} a function representing the action of a neural network. The choice of the architecture of the neural network and related hyperparameters directly determines its number of trainable parameters and consequently the number of degrees of freedom. Figure 2 shows the structure of the proposed neural network pipeline. The input contains information about the crystalline orientation and the morphology of the grains in the polycrystalline microstructure, i.e. the crystalline orientation field $\mathbf{q}(\mathbf{x})$, containing four unit quaternion real-valued components at each voxel. It is passed to a series of trainable layers denoted 3D ResNet, which returns the field $\hat{\mathbf{u}}^*(\mathbf{x})$, containing voxelwise displacement components. In order to reduce the complexity of the task, non-trainable physics-informed layers operate over both $\mathbf{q}(\mathbf{x})$ and $\hat{\mathbf{u}}^*(\mathbf{x})$, such that certain physical *a priori* constraints are verified by construction.

First, the stiffness tensor field is obtained from $\mathbf{q}(\mathbf{x})$ with $\mathbb{C}(\mathbf{x}) = \mathbf{G}(\mathbf{x}) :: \mathbb{C}^0$, where the symbol “::” represents a quadruple tensor contraction. The rank-8 rotation tensor \mathbf{G} for stiffness tensors is given by $G_{ijklIJKL} = g_{Ii}g_{Jj}g_{Kk}g_{Ll}$ [33], where \mathbf{g} denotes the rotation matrix from the laboratory reference frame to that of the grains. We follow [30] (Section A.15) and express \mathbf{g} in terms of \mathbf{q} as the “passive” rotation matrix:

$$\mathbf{g} = \begin{pmatrix} \bar{q} + 2q_2^2 & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & \bar{q} + 2q_3^2 & 2(q_3q_4 + q_1q_2) \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 - q_1q_2) & \bar{q} + 2q_4^2 \end{pmatrix}, \quad (10)$$

with $\bar{q} = 2q_1^2 - q_iq_i$.

Second, the predicted strain field $\hat{\boldsymbol{\varepsilon}}^*(\mathbf{x})$ is obtained from \mathbf{E} and $\hat{\mathbf{u}}^*(\mathbf{x})$, by means of another non-trainable layer that performs:

$$\hat{\boldsymbol{\varepsilon}}(\mathbf{x}) = \mathbf{E} + \frac{1}{2}(\nabla \hat{\mathbf{u}}^*(\mathbf{x}) + \nabla \hat{\mathbf{u}}^{*\top}(\mathbf{x})), \quad (11)$$

which directly follows from Eqs. (4) and (6). Consistently with the FFT scheme, differentiation in the gradient operator is implemented voxelwise via forward finite differences. Finally, the stress field is computed by double contraction of the tensor fields $\mathbb{C}(\mathbf{x})$ and $\hat{\boldsymbol{\varepsilon}}(\mathbf{x})$, i.e. $\hat{\boldsymbol{\sigma}}(\mathbf{x}) = \mathbb{C}(\mathbf{x}) : \hat{\boldsymbol{\varepsilon}}(\mathbf{x})$.

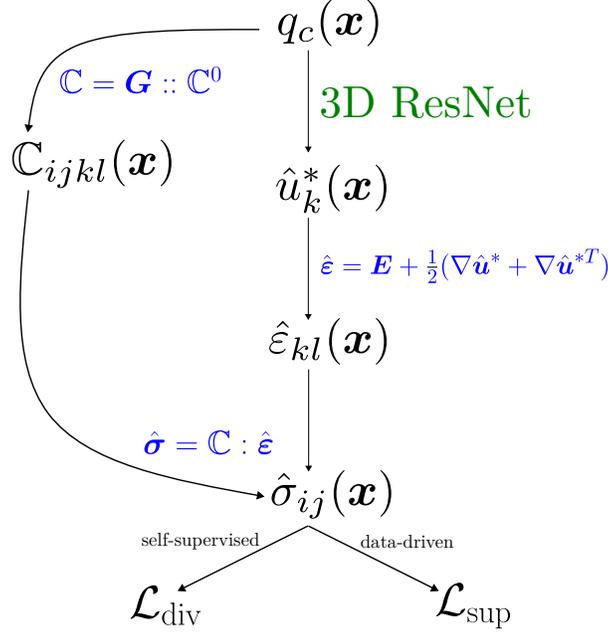


Figure 2: Diagram of the proposed physics-informed surrogate model for solving linear elasticity in polycrystals. Layers with learnable parameters are represented in green, while non-trainable operations are represented in blue. $c \in \{1, 2, 3, 4\}$, $i, j, k, l \in \{1, 2, 3\}$, $x, y, z \in \{1, \dots, 32\}$. The self-supervised loss function is denoted by $\mathcal{L}_{\text{div}}(\hat{\boldsymbol{\sigma}}) = \lambda_1 \langle |\text{div}(\hat{\boldsymbol{\sigma}}(\mathbf{x}))| \rangle$ while the data-driven loss function is denoted by $\mathcal{L}_{\text{sup}}(\hat{\boldsymbol{\sigma}}, \boldsymbol{\sigma}) = \lambda_2 [\langle (\hat{\sigma}_{xx}(\mathbf{x}) - \sigma_{xx}(\mathbf{x}))^2 \rangle + \langle (\hat{\sigma}_{xy}(\mathbf{x}) - \sigma_{xy}(\mathbf{x}))^2 \rangle + \langle (\hat{\sigma}_{xz}(\mathbf{x}) - \sigma_{xz}(\mathbf{x}))^2 \rangle + \langle (\hat{\sigma}_{yy}(\mathbf{x}) - \sigma_{yy}(\mathbf{x}))^2 \rangle + \langle (\hat{\sigma}_{yz}(\mathbf{x}) - \sigma_{yz}(\mathbf{x}))^2 \rangle + \langle (\hat{\sigma}_{zz}(\mathbf{x}) - \sigma_{zz}(\mathbf{x}))^2 \rangle]$, $\lambda_1 = 1 \times 10^{-7}$, $\lambda_2 = 1 \times 10^{-10}$.

We emphasize that this operation produces, by construction, a compatible strain field, i.e. $\nabla \times (\nabla \times \hat{\boldsymbol{\varepsilon}})^\top = \nabla \times (\nabla \times (\mathbf{E} + \frac{1}{2}(\nabla \hat{\mathbf{u}}^* + \nabla \hat{\mathbf{u}}^{*\top}))^\top = \mathbf{0}$. However, mechanical equilibrium (Eq. (2)) is not verified in general. This is similar to the FFT direct scheme, where the strain compatibility condition and constitutive law is enforced at each iteration, but equilibrium only upon convergence. In the neural network framework, this property can only be verified by training with respect to a loss function. We consider two of them, where the first one, denoted by \mathcal{L}_{div} , consists of a self-supervised method that minimizes

$$\mathcal{L}_{\text{div}}(\hat{\boldsymbol{\sigma}}) = \lambda_1 \langle |\text{div}(\hat{\boldsymbol{\sigma}}(\mathbf{x}))| \rangle. \quad (12)$$

The parameter λ_1 in Eq. (12) was empirically chosen such that the loss function $\mathcal{L}_{\text{div}}(\hat{\boldsymbol{\sigma}})$ lies in the range (0, 1) at the first epoch. Moreover, we employ backward finite differences as in [34] to estimate the divergence operator, as used in the FFT “direct” scheme. For the sake of limiting computational time, finite differences in Eqs. (11) and (12) are implemented as convolutions with non-trainable (3, 3, 3) kernels containing weights with values equal to ± 1 or 0. Representations of the kernels that compute backward and forward differences along the x axis are shown on Figure 3. For comparison purposes, we also investigate a data-driven training strategy and use the following supervised loss function:

$$\mathcal{L}_{\text{sup}}(\hat{\boldsymbol{\sigma}}, \boldsymbol{\sigma}) = \lambda_2 \sum_{i,j} \left(1 - \frac{\delta_{ij}}{2}\right) \langle (\hat{\sigma}_{ij}(\mathbf{x}) - \sigma_{ij}(\mathbf{x}))^2 \rangle \quad (13)$$

where δ_{ij} is the Kronecker symbol and we choose λ_2 analogously to the self-supervised case.

4.2. Trainable layers

Given that both the input and output of the proposed surrogate neural network are 3D multichannel images, its architecture is inspired by state-of-the-art neural networks designed for computer vision tasks.

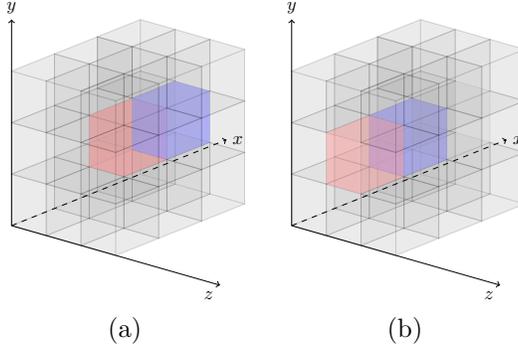


Figure 3: Schematic representation of the non-trainable 3D convolution kernels of size 3 used for performing forward (a) and backward (b) finite differences along the x -axis. Gray, red and blue represent 0, -1 and +1 values, respectively.

More precisely, architectures involving convolutions were considered, as these operations are translation equivariant, i.e. a translational shift in the input causes an equal translational shift in the output. It is important to note that the FFT computations for which we want to build a surrogate are also equivariant with respect to translations. Inspired by [35, 36] we make use of a convolutional network with residual connections (ResNet) to solve a 3D voxelwise regression problem. In the proposed neural network architecture, we discard downsampling operations such as pooling and convolutional layers with stride¹ larger than one, as these operations break translation equivariance [37].

The architecture of the proposed 3D ResNet is detailed on Figure 4. First, all convolutions are performed with cubic kernels of size $k = 3$, which is the smallest odd-sized kernel, allowing for local feature extraction. Further, the kernel stride is set to one voxel. Padding is performed such that the output of each convolution has the same spatial dimensions as the input. Note that the microstructure, as well as the physical problem exhibit periodic boundary conditions, which motivates periodic padding, also called *circular padding* in the adopted PyTorch framework [38].

The proposed 3D ResNet is a composition of a series of layers, namely convolution, instance normalization, activation and residual block. More exactly, the first layer is a 3D convolutional layer, whose output number of channels is set to $N_f = 256$. Next, instance normalization is done on the output of each convolution, allowing for the use of higher learning rates. This choice yields better performance than batch normalization on small batches, as normalization parameters are calculated independently for each instance or sample. For detailed information on that operation, we refer to [39]. Subsequently, the Leaky ReLU function [40] is used as an activation function, given by

$$\text{Leaky ReLU}(x) = \text{ReLU}(x) - \beta \text{ReLU}(-x), \quad \beta \in (0, 1) \quad (14)$$

where $\text{ReLU}(x) = (x + |x|)/2$. Recall that activation functions serve as a way to introduce mathematical non-linearities in the fitted function. Contrarily to the ReLU function, the Leaky ReLU function’s derivative is non-zero for inputs $x \in (-\infty, 0]$, thus preventing the well-known dead neuron phenomenon [41]. Then, the resulting tensors are processed through a sequence of layers called “residual block” (see Figure 5). This involves processing an input feature map \mathbf{F} through a composition of layers (convolution, instance normalization, activation, convolution and instance normalization) to obtain a partial output $\mathcal{F}(\mathbf{F})$. Finally, $\mathbf{F} + \mathcal{F}(\mathbf{F})$ is passed to a Leaky ReLU layer. Note that the residual connection implies that all the dimensions of the feature maps are unchanged.

Since the initial convolutional layers are supposed to learn rather local geometric features present in the input data, later layers process less significant long-range correlations. This intuition motivates the reduction of the channel dimension as the information is propagated through the neural network. Thus,

¹The term “stride” corresponds to the translational step performed by a convolutional kernel along the spatial dimensions.

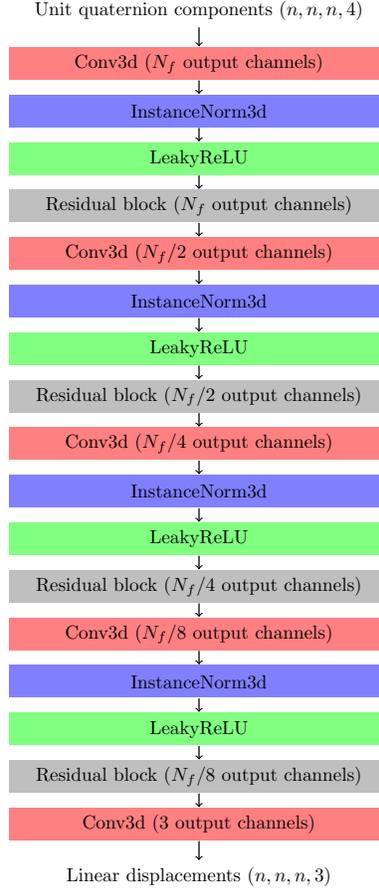


Figure 4: Schematic representation of the proposed 3D ResNet. In the first 3D convolution layer, the number of channels is taken from 4 to N_f . Cubic kernels of size k are employed, with constant-resolution periodic padding and unit stride. Next, an instance normalization layer and a Leaky ReLU activation function act on the resulting tensors. Then, a residual block (see Figure 5) is applied on resulting the feature maps, with the number of channels left unchanged. The channel number is reduced by a factor of 2 by the following convolution, whose output passes through an instance normalization and an activation function. This series of layers is repeated until a final convolution reduces the number of channels to 3.

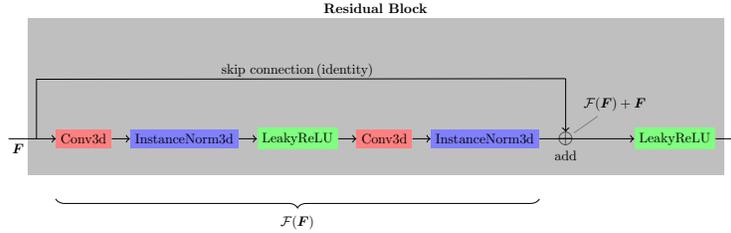


Figure 5: Schematic representation of a residual block, used in the 3D ResNet (Figure 4). It consists of processing an input feature map \mathbf{F} through a series of layers, namely a convolution, an instance normalization, an activation function, another convolution and another instance normalization, whose output is added to the input of the residual block and passed to the final activation function.

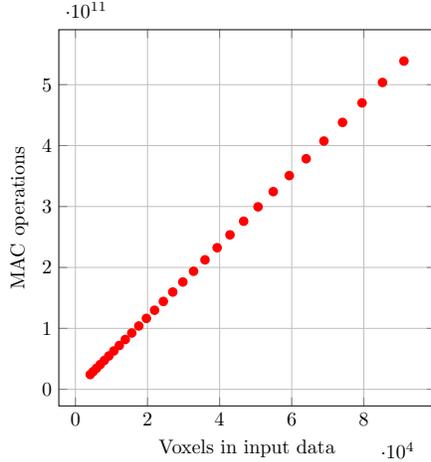


Figure 6: Number of multiply-accumulate operations (MAC), computed using the *ptflops* library [43] versus number of voxels in input data for inferences of the proposed surrogate model on SVEs of various sizes.

the aforementioned series of layers is repeated with a halving of the output number of channels in each convolutional layer (except for convolutions inside residual blocks) until the number of channels reaches $N_f/8 = 32$. Finally, a convolutional layer takes the channel number to 3, which is the number of components of the periodic part of displacement field.

The selection of the amount of halvings in the number of channels was conditioned by the relation between the size n of the training microstructures and the size of the receptive field r_0 . In the present architecture, r_0 is determined by $r_0 = L(k - 1) + 1$, where L denotes the number of convolutional layers and k the kernel size. We choose L such that $r_0 \leq n$. Indeed, in the case of periodic padding, a receptive field larger than the input will induce a learning bias characterized by spurious long range correlations, as the receptive field will cover simultaneously some voxels and their periodic replications. Thus, generalizability for inputs larger than n is impaired, as the network “expects” in the input geometry a periodicity that is no longer present. In the present work, we chose $r_0 = 27$ voxels, which is consistent with that restriction, as $n = 32$ for the training data. The number of trainable parameters in the neural network is 5,893,059.

Moreover the numerical effort of the proposed pipeline was analyzed. The numerical effort of a 3D convolutional layer scales as $\sim n^3 c_{in} c_{out} k^3$, where n^3 is the number of voxels of the input microstructure; c_{in} and c_{out} the number of input and output channels, respectively; and k the size of the cubic kernel. As a consequence of a receptive field with fixed size, the numerical effort of an inference by the neural network is therefore in $O(n^3)$, as confirmed on Figure (6). This is in contrast to the $O(n^3 \log(n^3))$ complexity for FFT computations [11], although alternative FFT schemes may have the same complexity as ours [42]. Regarding costs for inference with CPU only, a 32^3 microstructure takes 4.6 s and 2.2 GB of random-access memory (RAM) and 36.5 s, 17.5 GB, respectively, for a 64^3 microstructure. Despite that, modern implementations of FFT simulations are still much cheaper in terms of time and memory. For instance, a 64^3 -voxel computation takes about 0.4 s and 59.2 MB of memory. We do not intend to propose an even cheaper alternative to these, but rather a building block for solving non-trivial problems, such as inverse problems, some of which are briefly presented in Section 5.4.

5. Results and discussion

In this section, we compare the performance of each learning strategy and their associated computational costs. Additionally, we study how both approaches perform on microstructures exhibiting statistically different properties compared to the ones utilized for training. More precisely, microstructures on larger domains as well as microstructures with different expected numbers and sizes of grains are considered.

5.1. Experiments

During training, we enforce a time-independent uniaxial extension as macroscopic strain together with periodic boundary conditions, consistently with FFT computations, which can be mathematically expressed as

$$\mathbf{E} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ & \varepsilon_{yy} & \varepsilon_{yz} \\ \text{Sym.} & & \varepsilon_{zz} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ & 0 & 0 \\ \text{Sym.} & & 0 \end{bmatrix} \quad (15)$$

The reference stiffness tensor \mathbb{C}^0 models purely γ -phased TiAl polycrystals with tetragonal symmetry [44], given by

$$\mathbb{C}^0 = \begin{bmatrix} C^0_{1111} & C^0_{1122} & C^0_{1133} & C^0_{1123} & C^0_{1113} & C^0_{1112} \\ & C^0_{2222} & C^0_{2233} & C^0_{2223} & C^0_{2213} & C^0_{2212} \\ & & C^0_{3333} & C^0_{3323} & C^0_{3313} & C^0_{3312} \\ & & & C^0_{2323} & C^0_{2313} & C^0_{2312} \\ & \text{Sym.} & & & C^0_{1313} & C^0_{1312} \\ & & & & & C^0_{1212} \end{bmatrix} = \begin{bmatrix} 183 & 74 & 74 & 0 & 0 & 0 \\ & 183 & 74 & 0 & 0 & 0 \\ & & 178 & 0 & 0 & 0 \\ & & & 105 & 0 & 0 \\ & \text{Sym.} & & & 105 & 0 \\ & & & & & 78 \end{bmatrix} \times 10^3 \text{ MPa}, \quad (16)$$

i.e. the Young modulus along the x and y axis and the in-plane shear modulus in (x, y) axis is slightly stiffer compared to the Young modulus along the z axis or the out-of-plane shear modulus.

The dataset used in this study consists of 7000 virtual random polycrystalline microstructures of size 32^3 drawn from the stochastic microstructure model described in Section 2, along with the corresponding stress fields obtained with a standard FFT solver. Recall that these microstructures are multi-channel images, where each voxel contains the real-valued components of a unit quaternion, which represents its crystallographic orientation. The data is split into training, validation and test sets, with proportions of 5/7 for training, 1/7 for validation and 1/7 for testing.

The optimization of the network’s trainable weights was done using the Adam algorithm [45], with batch size equal to 1, and initial learning rate set to 1×10^{-3} . Unitary batch size is used for the sake of minimizing RAM consumption and reducing implementation complexity. In order to prevent overfitting, a criterion was defined to stop the training before reaching the maximum number of epochs. Let e and $e + 1$ denote two consecutive epochs and \mathcal{L}_e and \mathcal{L}_{e+1} the corresponding evaluations of the loss function. Further, let $\delta_{min} \geq 0$ denote the minimum amount of improvement. An improvement with respect to the loss function is characterized by $\mathcal{L}_{e+1} < \mathcal{L}_e - \delta_{min}$. Moreover, the so-called “patience” parameter defines the maximum number of epochs without improvement in the validation loss before stopping the training. In this study, the training procedure is stopped if the best validation loss does not improve within 15 epochs, with $\delta_{min} = 0$.

5.2. Learning strategies: self-supervised vs. data-driven

In this section, we discuss the results of each learning approach for the previously defined set of hyperparameters and boundary conditions. Because the values of both loss functions cannot be directly compared, in order to fairly evaluate and compare the quality of the predictions, we use the mean absolute error (MAE) function as a metric which is defined by

$$\text{MAE}(\hat{\boldsymbol{\sigma}}, \boldsymbol{\sigma}) = \frac{1}{6} \sum_{i \leq j} \langle |\hat{\sigma}_{ij}(\mathbf{x}) - \sigma_{ij}(\mathbf{x})| \rangle. \quad (17)$$

Figure 7 shows the evolution of the loss functions \mathcal{L}_{sup} and \mathcal{L}_{div} as well as the MAE metric for both training and validation datasets. The data-driven pipeline performs worse on the validation dataset than the self-supervised one with respect to the lowest validation MAE: 12,193 MPa for the self-supervised training versus 12,234 MPa for the data-driven one. Also interestingly, whereas the validation loss and MAE of the data-driven model start increasing after around 50 epochs, indicating a degraded generalization ability (overfitting), the validation loss and MAE kept decreasing for the self-supervised training for considerably longer, until the early stopping criterion was met. That behavior suggests an inherent regularization property of the self-supervised loss function \mathcal{L}_{div} .

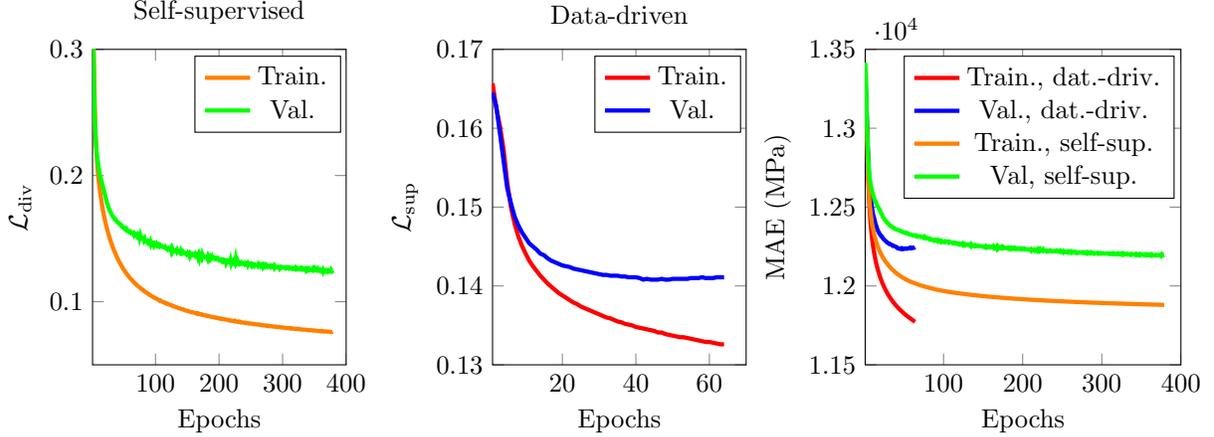


Figure 7: Evolution of the loss functions and the MAE during the training of the self-supervised and data-driven models.

In addition to the MAE, the mean relative error MRE was also computed for the predicted stress field component $\hat{\sigma}_{xx}$ and predicted Von Mises equivalent stress field $\hat{\sigma}_{\text{eq}}$. This metric is given by

$$\text{MRE}(\hat{\sigma}_{xx}, \sigma_{xx}) = \left\langle \frac{\hat{\sigma}_{xx}(\mathbf{x}) - \sigma_{xx}(\mathbf{x})}{\sigma_{xx}(\mathbf{x})} \right\rangle, \quad (18a)$$

$$\text{MRE}(\hat{\sigma}_{\text{eq}}, \sigma_{\text{eq}}) = \left\langle \frac{\hat{\sigma}_{\text{eq}}(\mathbf{x}) - \sigma_{\text{eq}}(\mathbf{x})}{\sigma_{\text{eq}}(\mathbf{x})} \right\rangle. \quad (18b)$$

Regarding the performance over the 32^3 test dataset, the MRE committed by the self-supervised model is 2.16% for $\hat{\sigma}_{xx}$ and 3.99% for $\hat{\sigma}_{\text{eq}}$, whereas for the data-driven model, these values are respectively 2.86% and 4.82%. Figure 8 shows the cumulative distribution functions of the relative error fields on $\hat{\sigma}_{xx}$ and $\hat{\sigma}_{\text{eq}}$. We observe that this error is lower than 5% for 93% of the voxels in the self-supervised $\hat{\sigma}_{xx}$ fields and only 84% for the data-driven counterpart. Regarding self-supervised $\hat{\sigma}_{\text{eq}}$ fields, 69% of the voxels are found in a 5% or lower relative error threshold, whereas for the data-driven predictions, 60% of voxels are present in the aforementioned threshold. In summary, both training strategies have a better accuracy for the σ_{xx} component than for σ_{eq} . Furthermore, the predictions of the self-supervised neural network are more accurate than those of the data-driven pipeline, for both $\hat{\sigma}_{xx}$ and $\hat{\sigma}_{\text{eq}}$.

Moreover, we investigated the accuracy of both training approaches when applied to larger input microstructures, with same expected grain size. More specifically, inference was performed on 125 microstructures of size 64^3 at the same voxel resolution. In order to visually assess the spatial distribution of the errors, Figures 9 and 10 show 3D plots of 32^3 and 64^3 test inputs, their corresponding ground truths, predictions for both trained models and MAE fields. In the present work, the error field (expressed in MPa) is evaluated in terms of L_1 -norm, which is less sensitive to outliers, compared to the L_2 -norm. It was found that the self-supervised predictions have an MRE of 2.39% for $\hat{\sigma}_{xx}$, and 4.57% for $\hat{\sigma}_{\text{eq}}$, while for the data-driven model, these metrics are 2.85% and 4.90% respectively. For a relative error threshold of 5% or less on $\hat{\sigma}_{xx}$, 91% of the voxels in the self-supervised predictions are included, whereas only 84% of the data-driven predictions are included. For $\hat{\sigma}_{\text{eq}}$, these values are 62% and 59%, respectively. Accordingly, the performances of the two approaches are slightly worse on the larger domain, which is related to the fact that 32^3 microstructures were used for training. Indeed, correlations at distances longer than 27 voxels, the size of the receptive field, are not taken into account. The domain outside the receptive field is much smaller in the 32^3 microstructure, than in the 64^3 polycrystal.

The effective stiffness tensor \mathbb{C}^* can be estimated from the test dataset, following either an energetic or mechanical definition, i.e.

$$\langle \boldsymbol{\sigma} : \boldsymbol{\varepsilon} \rangle = \mathbf{E} : \mathbb{C}_e^* : \mathbf{E}, \quad \langle \boldsymbol{\sigma} \rangle = \mathbb{C}_m^* : \mathbf{E}. \quad (19)$$

As a result of the Hill theorem [46] the two definitions above are equivalent and $\mathbb{C}^* = \mathbb{C}_e^* = \mathbb{C}_m^*$ when the

stress is divergence-free. This is not true in the present case and therefore the two estimates provided by the surrogate model are different, i.e. $\mathbb{C}_e^* \neq \mathbb{C}_m^*$. Nevertheless, the minimization of the potential elastic energy requires that [47]

$$\langle \boldsymbol{\sigma} : \boldsymbol{\varepsilon} \rangle \geq \mathbf{E} : \mathbb{C}^* : \mathbf{E}. \quad (20)$$

Therefore, the energetic definition provides an upper-bound of the exact effective stiffness tensor, i.e.,

$$\mathbb{C}_e^* \geq \mathbb{C}^*, \quad (21)$$

in the sense of positive definite tensors. This property also holds in more general situations, see [48] Figure (11) shows correlations between the estimate $C_{m,1111}^*$ and the ground truth C_{1111}^* , computed from FFT. As for the energetic definition, the elastic modulus $C_{m,1111}^*$ is systematically overestimated compared to C_{1111}^* . This bias is an effect of the constraint imposed on the strain field, which in the present case, is rigorously admissible. Despite this, both the self-supervised and data-driven neural networks are capable of accurately predicting the correlation between the crystalline orientation fields and their corresponding mechanical properties. Furthermore, the bias is slightly lower for the self-supervised pipeline than for the data-driven one.

Finally, histograms of the σ_{xx} , $\hat{\sigma}_{xx}$, σ_{eq} and $\hat{\sigma}_{\text{eq}}$ fields were computed (Figure 12). The presence of high stress values are slightly overestimated, consistently with the predictions for the effective elastic moduli, which are stiffer than FFT estimates.

5.3. Accuracy for varied parameters of the stochastic microstructure model

The accuracy of both training methods was also evaluated for microstructures exhibiting different statistical properties than the ones used for training. More precisely, the parameters of the stochastic microstructure model, introduced in Section 2, were systematically varied to derive a different number of seed points, which are more or less homogeneously distributed in space. It is worth recalling that the training dataset is composed of microstructures of size 32^3 , based on a Matérn hardcore point process with intensity $\lambda = 15/32^3$ and hardcore radius $r = 5$. The chosen intensity corresponds to an expected number of 15 grains. Experiments were done for several configurations of expected number of grains (determined by the intensity of the Matérn hardcore process) and hardcore radii. More precisely, both surrogate models were applied on microstructures generated by all combinations of intensity $\lambda \in \{12.5/32^3, 15/32^3, 17.5/32^3\}$ (corresponding to 12.5, 15 and 17.5 expected grains) and hardcore radius $r \in \{2.5, 5, 7.5\}$. Tables 1 and 2 indicate the MRE for both models and for both $\hat{\sigma}_{xx}$ and $\hat{\sigma}_{\text{eq}}$.

Interestingly, both models performed better on the microstructures exhibiting different statistical descriptors than those used for neural network training. Indeed, slightly lower errors are observed for structures generated with either a higher hardcore radius or a higher number of grains. The first case corresponds to a higher repulsion distance between seed points, and lead to more regular cells, whereas in the second case the grains have lower volume.

5.4. Potential applications

In this section, we propose two applications of the physics-informed surrogate model which are not further investigated and left to future work. We first describe an inverse problem consisting of finding uniform-per-grain crystallographic orientations given a known target strain field $\boldsymbol{\varepsilon}^{\text{DIC}}$, such as that obtained by digital image correlation (DIC). It can be expressed by:

$$\min_{\omega_{gc} \in \mathbb{S}^3} \mathcal{L}_{\text{sup}}(\mathbb{C} : \hat{\boldsymbol{\varepsilon}}, \mathbb{C} : \boldsymbol{\varepsilon}^{\text{DIC}}), \quad (22)$$

subject to the constraints:

$$\hat{\boldsymbol{\varepsilon}}(\mathbf{x}) = \mathbf{E} + \frac{\nabla \hat{\mathbf{u}}^*(\mathbf{x}) + \nabla \hat{\mathbf{u}}^{*\top}(\mathbf{x})}{2}, \quad \hat{\mathbf{u}}^*(\mathbf{x}) = \mathcal{N}(\Lambda; \mathbf{q}(\mathbf{x}), \mathbf{E}), \quad q_c(\mathbf{x}) = L_g(\mathbf{x})\omega_{gc}, \quad L_g(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in G_g, \\ 0, & \text{otherwise.} \end{cases} \quad (23)$$

Expected number of grains	Matérn hardcore radius r		
	2.5	5	7.5
12.5	2.34%, 3.03%	2.31%, 2.97%	2.24%, 2.92%
15	2.12%, 2.83%	2.16%, 2.86%	2.07%, 2.77%
17.5	2.08%, 2.78%	2.04%, 2.78%	2.00%, 2.70%

Table 1: $\text{MRE}_{\text{div}}(\sigma_{xx}, \hat{\sigma}_{xx})$ and $\text{MRE}_{\text{sup}}(\sigma_{xx}, \hat{\sigma}_{xx})$ values calculated for test datasets containing 1000×32^3 microstructures for varying combinations of parameters. Recall that, for training, $r = 5$ and 15 is the number of expected grains.

Expected number of grains	Matérn hardcore radius r		
	2.5	5	7.5
12.5	4.27%, 5.16%	4.21%, 5.04%	4.14%, 4.98%
15	3.92%, 4.78%	3.99%, 4.82%	3.86%, 4.68%
17.5	3.86%, 4.68%	3.82%, 4.68%	3.74%, 4.53%

Table 2: $\text{MRE}_{\text{div}}(\sigma_{\text{eq}}, \hat{\sigma}_{\text{eq}})$ and $\text{MRE}_{\text{sup}}(\sigma_{\text{eq}}, \hat{\sigma}_{\text{eq}})$ values calculated for test datasets containing 1000×32^3 microstructures for varying combinations of parameters. Recall that, for training, $r = 5$ and 15 is the number of expected grains.

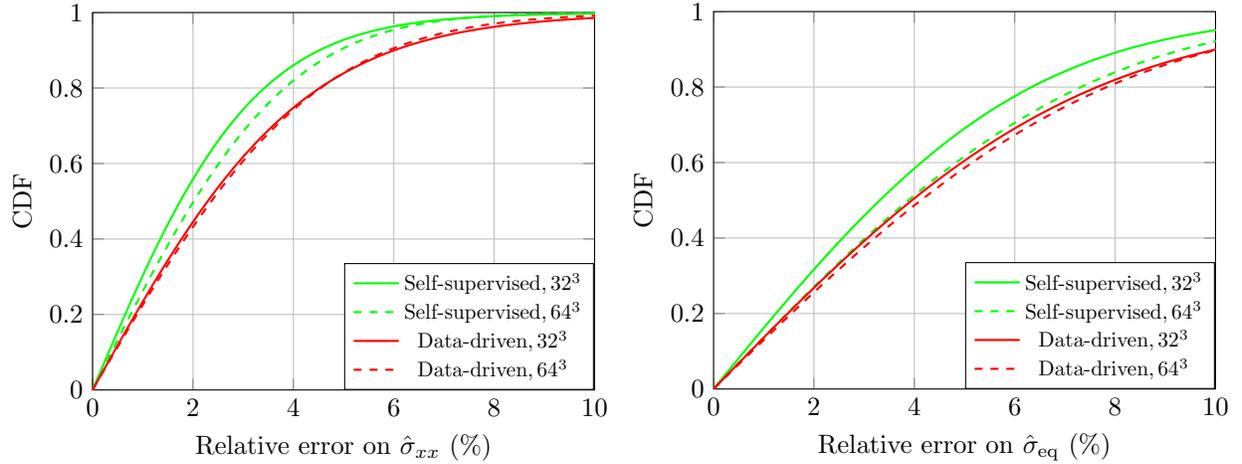


Figure 8: Cumulative distribution functions for the relative error fields on $\hat{\sigma}_{xx}$ (left) and $\hat{\sigma}_{\text{eq}}$ (right) over the test dataset for both self-supervised and data-driven learning strategies and both input sizes (32^3 and 64^3 microstructures).

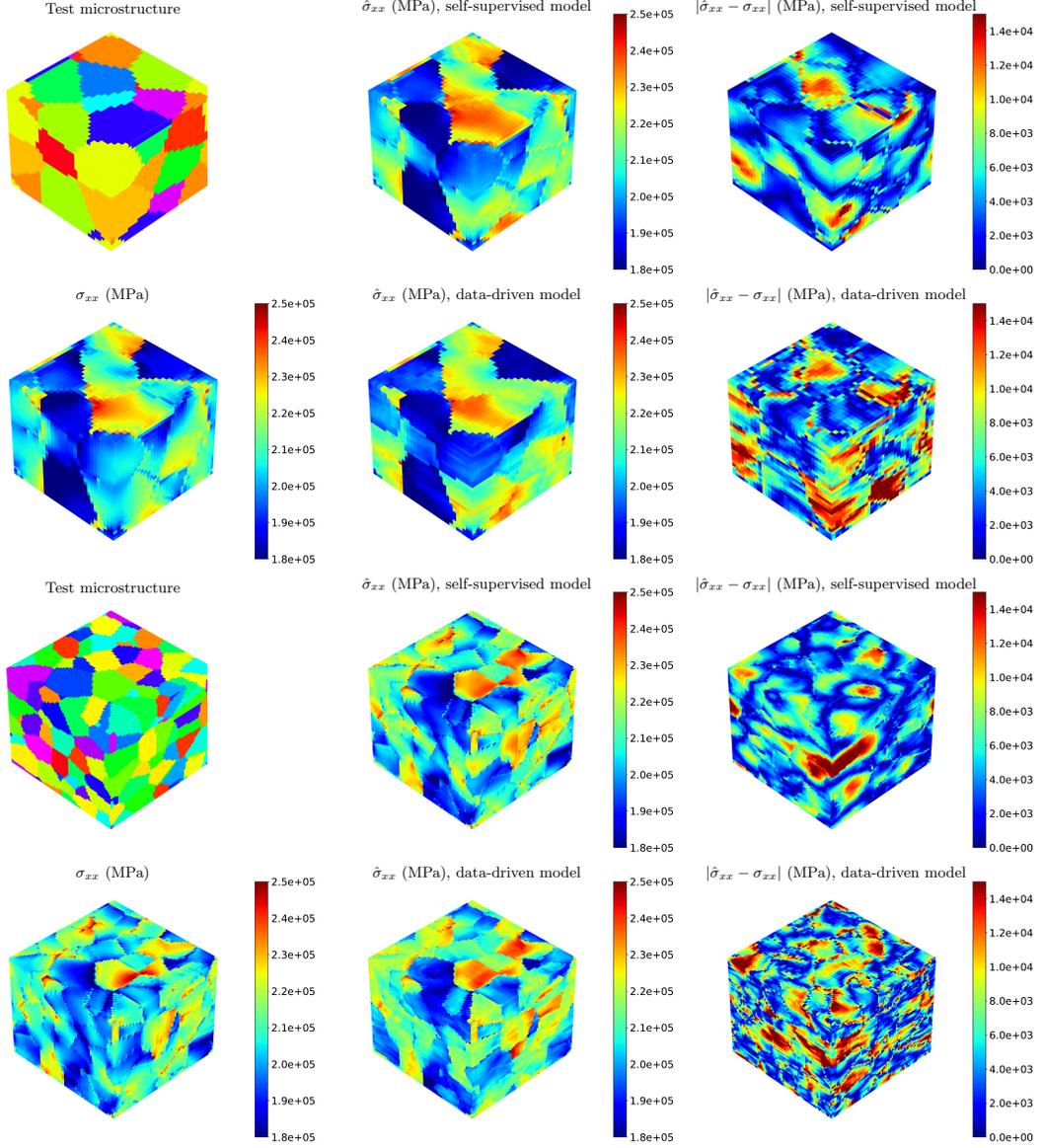


Figure 9: Comparison between 32^3 and 64^3 input samples (coloring based on grain labels) in the test dataset, corresponding stress field component σ_{xx} ground truths from FFT computations and the predictions made by the trained neural networks, as well as MAE fields. Values in MPa.

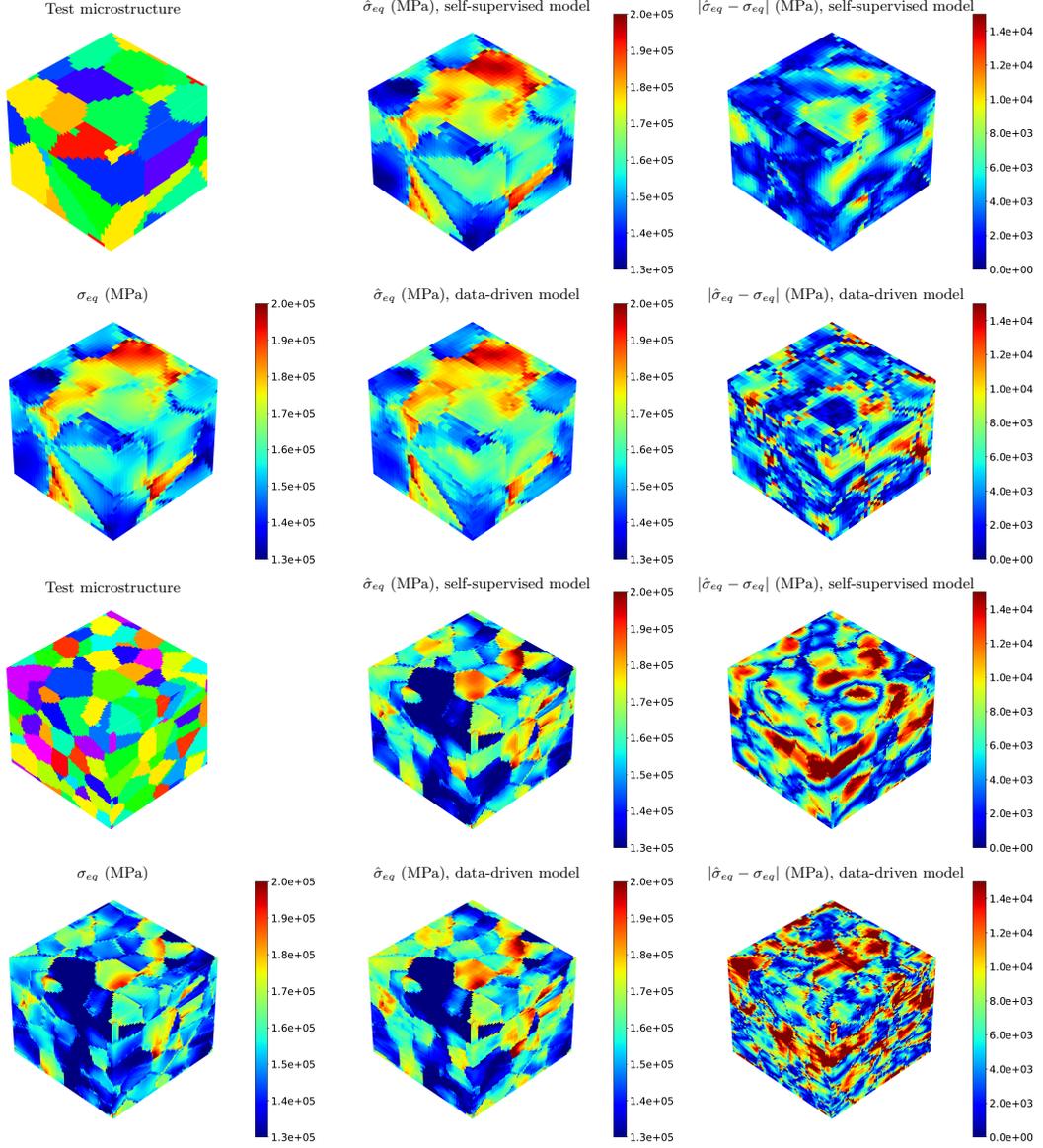


Figure 10: Comparison between 32^3 and 64^3 input samples (coloring based on grain labels) in the test dataset, corresponding Von Mises equivalent stress σ_{eq} ground truths from FFT computations and the predictions made by the trained neural networks, as well as MAE fields. Values in MPa.

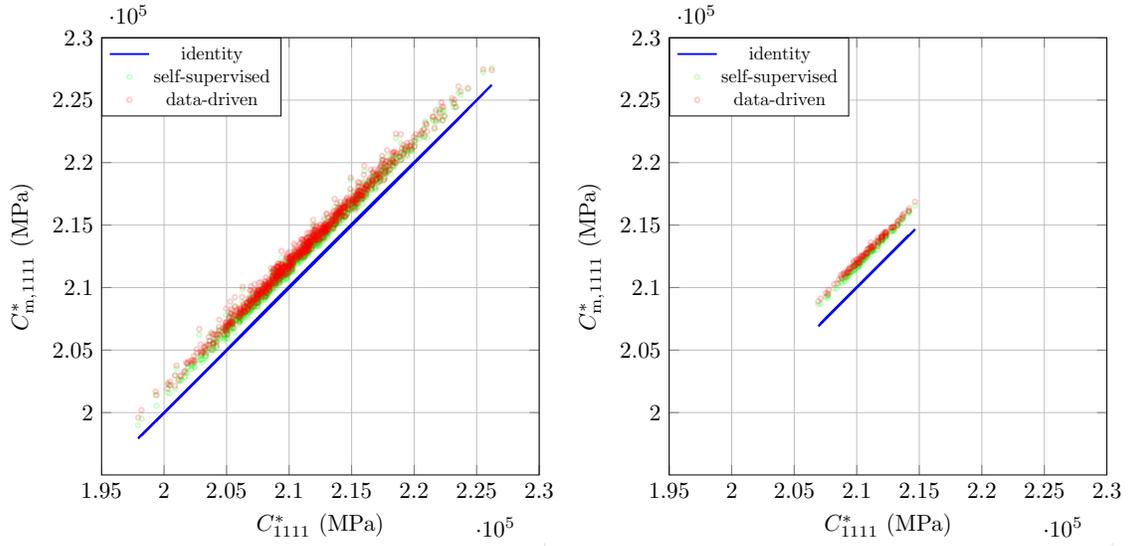


Figure 11: Predictions $C_{m,1111}^*$ versus ground truth values C_{1111}^* of the effective elastic modulus for 32^3 (left) and 64^3 (right) test datasets.

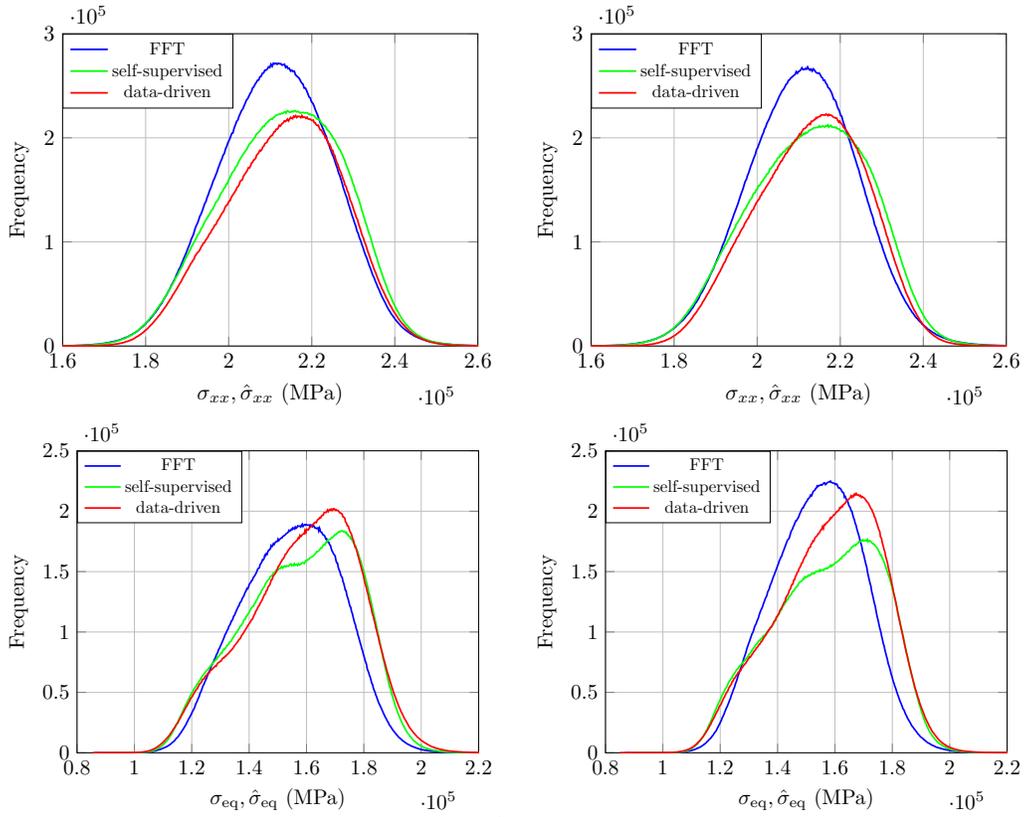


Figure 12: Histograms for ground truths and predictions on 32^3 (left column) and 64^3 (right column) test datasets.

In the above, ω is the set of grainwise unit quaternions to be updated after each step of a gradient descent optimization scheme. The location tensor $L_g(\mathbf{x})$ indicates whether the grain of label g is present on voxel \mathbf{x} .

A second application concerns optimization problems in the general framework of homogenization. One may for instance, seek to minimize the spatial maximum of the strain energy field in the space of isotropic polycrystalline microstructures. Such a problem can be approximated as:

$$\min_{\omega_{gc} \in \mathbb{S}^3} \beta_1 \max \left(\frac{1}{2} \hat{\boldsymbol{\varepsilon}}(\mathbf{x}) : \mathbb{C}(\mathbf{x}) : \hat{\boldsymbol{\varepsilon}}(\mathbf{x}) \right) + \beta_2 |\mathbb{C}_m^* - P(\mathbb{C}_m^*)| \quad (24)$$

subject to the same constraints (23). Here $\beta_{1,2}$ are numerical coefficients and $P(\mathbb{C}_m^*)$ is the Euclidean projection of \mathbb{C}_m^* onto the space of isotropic stiffness tensors [49] and $|\cdot|$ is the Frobenius norm. This term is used to penalize macroscopically anisotropic responses. This problem can be interesting in a context of damage mechanics, when local maxima in the strain energy field are preferential locations for the onset of cracks [50].

6. Conclusion

In this work, we proposed a physics-informed framework for solving linear elasticity on polycrystals by using a fully translation equivariant 3D ResNet as the trainable part of the surrogate model. Non-trainable layers inspired by the operations performed by conventional physical PDE solvers inject beneficial inductive biases into the learning process, improving the interpretability of the neural network and rendering supervision with ground truth data unnecessary. Heuristics for hyperparameter selection based on the network’s receptive field and available training data were proposed. A highlight of the developed model is that it constitutes a fully differentiable expression that links microstructure orientation fields to stress fields, therefore enabling the resolution of inverse problems.

Both self-supervised and data-driven learning settings were analyzed and compared with respect to the average relative error committed voxelwise. While the self-supervised approach is more computationally expensive, it yields slightly better accuracy than its data-driven counterpart while allowing to train a surrogate model in the absence of ground truth data. Furthermore, no overfitting was characterized for the self-supervised method. Apart from better accuracy, the self-supervised approach allows for greatly reduced RAM memory usage during training compared to the data-driven counterpart, as ground truth stress fields are ultimately needless. We also showed that the proposed surrogate models can be accurately applied to inputs with sampling windows larger than that used during training, as well as microstructures with statistically slightly different geometric properties. An important feature of these models is that they produce strictly compatible strain fields. As a consequence, the predictions of the effective response are slightly biased towards stiffer values.

Our pipeline is restricted to the linear elastostatic behavior of polycrystals. Thus, it is not necessary to consider the loading history, but only its final state, which is linearly proportional to the imposed macroscopic strain \boldsymbol{E} . Accounting for the evolution of the response across different time steps, with nonlinear or irreversible constitutive laws, is beyond the scope of this study. Another point worth commenting is the self-supervision of the stress divergence field over the entire domain. More computationally frugal ways for positioning collocation points could be used. The chosen method assumes that every possible collocation point is equally important *a priori*, which is not intuitively true, considering that mechanical fields are rather homogeneous inside grains and highly discontinuous at boundaries. Assigning random positions for collocation points is a possibility to address that, although grain boundaries might be underrepresented, which would reduce the overall accuracy. One could argue in favor of a pre-processing step for identifying stress concentration zones, but that could turn out to be less efficient than the present approach. Moreover, the neural networks’ robustness can be further increased by using a more diverse training dataset i.e. microstructures generated with different stochastic algorithms, expected grain sizes and spatial resolutions.

Finally, two potential applications of the presented surrogate model were proposed: inverse problems involving the optimization of the set of crystallographic orientations for different objectives and subject to certain constraints, among which fixed grain shapes.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge the financial support of the French Agence Nationale de la Recherche (ANR, ANR-21-FAI1-0003) and the Bundesministerium für Bildung und Forschung (BMBF, 01IS21091) within the French-German research project SMILE. FW and LMF are grateful to Vladislav Yastrebov for helpful suggestions.

Code and datasets availability

The code and datasets created to achieve these findings are available to download from <https://github.com/lmonteirofernandes/PI3DCRYSTAL>.

References

- [1] D. François, A. Pineau, A. Zaoui, Mechanical Behaviour of Materials: Volume I: Micro-and Macroscopic Constitutive Behaviour, volume 180, Springer Science & Business Media, 2012. doi:[10.1007/978-94-007-2546-1](https://doi.org/10.1007/978-94-007-2546-1).
- [2] D. François, A. Pineau, A. Zaoui, Mechanical Behaviour of Materials: Volume II: Fracture Mechanics and Damage, volume 191, Springer Science & Business Media, 2012. doi:[10.1007/978-94-007-4930-6](https://doi.org/10.1007/978-94-007-4930-6).
- [3] O. Cazacu, B. Revil-Baudard, N. Chandola, Plasticity-Damage Couplings: From Single Crystal to Polycrystalline Materials, Solid Mechanics and Its Applications, Springer International Publishing, 2019. doi:[10.1007/978-3-319-92922-4](https://doi.org/10.1007/978-3-319-92922-4).
- [4] G. W. Milton, The Theory of Composites, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2002. doi:[10.1017/CB09780511613357](https://doi.org/10.1017/CB09780511613357).
- [5] K. Jöchen, Homogenization of the Linear and Non-linear Mechanical Behavior of Polycrystals, KIT Scientific Publishing, 2014. doi:[10.5445/KSP/1000032289](https://doi.org/10.5445/KSP/1000032289).
- [6] J. Besson, G. Cailletaud, J.-L. Chaboche, S. Forest, Non-Linear Mechanics of Materials, volume 167, Springer Science & Business Media, 2009. doi:[10.1007/978-90-481-3356-7](https://doi.org/10.1007/978-90-481-3356-7).
- [7] R. A. Lebensohn, A. D. Rollett, Spectral methods for full-field micromechanical modelling of polycrystalline materials, Computational Materials Science 173 (2020) 109336. doi:[10.1016/j.commatsci.2019.109336](https://doi.org/10.1016/j.commatsci.2019.109336).
- [8] M. Gille, H. Proudhon, J. Oddershede, R. Quey, T. F. Morgeneyer, 3D strain heterogeneity and fracture studied by X-ray tomography and crystal plasticity in an aluminium alloy, International Journal of Plasticity 183 (2024) 104146. doi:[10.1016/j.ijplas.2024.104146](https://doi.org/10.1016/j.ijplas.2024.104146).
- [9] H. Moulinec, P. Suquet, A fast numerical method for computing the linear and nonlinear mechanical properties of composites, Comptes Rendus de l'Académie des sciences. Série II 318 (1994) 1417–1423.
- [10] M. Koishi, N. Kowatari, B. Figliuzzi, M. Faessel, F. Willot, D. Jeulin, Computational material design of filled rubbers using multi-objective design exploration, in: Constitutive Models for Rubber X, CRC Press, 2017, pp. 467–473. doi:[10.1201/9781315223278-73](https://doi.org/10.1201/9781315223278-73).

- [11] H. Moulinec, P. Suquet, A numerical method for computing the overall response of nonlinear composites with complex microstructure, *Computer Methods in Applied Mechanics and Engineering* 157 (1998) 69–94.
- [12] K. Guo, Z. Yang, C.-H. Yu, M. J. Buehler, Artificial intelligence and machine learning in design of mechanical materials, *Materials Horizons* 8 (2021) 1153–1172. doi:[10.1039/D0MH01451F](https://doi.org/10.1039/D0MH01451F).
- [13] R. Ahmad, M. Liu, M. Ortiz, T. Mukerji, W. Cai, Homogenizing elastic properties of large digital rock images by combining CNN with hierarchical homogenization method (2023). doi:[10.48550/arXiv.2305.06519](https://doi.org/10.48550/arXiv.2305.06519), preprint.
- [14] J. M. Hestroffer, M.-A. Charpagne, M. I. Latypov, I. J. Beyerlein, Graph neural networks for efficient learning of mechanical properties of polycrystals, *Computational Materials Science* 217 (2023) 111894. doi:[10.1016/j.commatsci.2022.111894](https://doi.org/10.1016/j.commatsci.2022.111894).
- [15] R. Liu, A. Kumar, Z. Chen, A. Agrawal, V. Sundararaghavan, A. Choudhary, A predictive machine learning approach for microstructure optimization and materials design, *Scientific Reports* 5 (2015) 11551. doi:[10.1038/srep11551](https://doi.org/10.1038/srep11551).
- [16] J. R. Mianroodi, N. H. Siboni, D. Raabe, Teaching solid mechanics to artificial intelligence - a fast solver for heterogeneous materials, *npj Computational Materials* 7 (2021) 1–10. doi:[10.1038/s41524-021-00571-z](https://doi.org/10.1038/s41524-021-00571-z).
- [17] M. S. Khorrami, J. R. Mianroodi, N. H. Siboni, P. Goyal, B. Svendsen, P. Benner, D. Raabe, An artificial neural network for surrogate modeling of stress fields in viscoplastic polycrystalline materials, *npj Computational Materials* 9 (2023) 1–10. doi:[10.1038/s41524-023-00991-z](https://doi.org/10.1038/s41524-023-00991-z).
- [18] I. E. Lagaris, A. Likas, D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Transactions on Neural Networks* 9 (1998) 987–1000. doi:[10.1109/72.712178](https://doi.org/10.1109/72.712178).
- [19] M. Raissi, Deep hidden physics models: Deep learning of nonlinear partial differential equations, *Journal of Machine Learning Research* 19 (2018) 1–24. doi:[10.5555/3291125.3291150](https://doi.org/10.5555/3291125.3291150).
- [20] S. Kollmannsberger, D. D’Angella, M. Jokeit, L. Herrmann, *Deep Learning in Computational Mechanics: An Introductory Course*, Springer, 2021. doi:[10.1007/978-3-030-76587-3](https://doi.org/10.1007/978-3-030-76587-3).
- [21] A. Henkes, H. Wessels, R. Mahnken, Physics informed neural networks for continuum micromechanics, *Computer Methods in Applied Mechanics and Engineering* 393 (2022) 114790. doi:[10.1016/j.cma.2022.114790](https://doi.org/10.1016/j.cma.2022.114790).
- [22] G. Basso Della Mea, C. Ovalle, L. Laiarinandrasana, E. Decencière, P. Dokládál, Physics informed self-supervised segmentation of elastic composite materials, *Computer Methods in Applied Mechanics and Engineering* 432 (2024) 117355. doi:[10.1016/j.cma.2024.117355](https://doi.org/10.1016/j.cma.2024.117355).
- [23] S. Wang, T.-R. Liu, S. Sankaran, P. Perdikaris, Micrometer: Micromechanics Transformer for Predicting Mechanical Responses of Heterogeneous Materials (2024). doi:[10.48550/arXiv.2410.05281](https://doi.org/10.48550/arXiv.2410.05281), preprint.
- [24] Y.-C. Hsu, C.-H. Yu, M. J. Buehler, Tuning mechanical properties in polycrystalline solids using a deep generative framework, *Advanced Engineering Materials* 23 (2021) 2001339. doi:[10.1002/adem.202001339](https://doi.org/10.1002/adem.202001339).
- [25] T. Kapoor, H. Wang, A. Núñez, R. Dollevoet, Physics-Informed Neural Networks for Solving Forward and Inverse Problems in Complex Beam Systems, *IEEE Transactions on Neural Networks and Learning Systems* 35 (2024) 5981–5995. doi:[10.1109/TNNLS.2023.3310585](https://doi.org/10.1109/TNNLS.2023.3310585).

- [26] H. Jeong, C. Batuwatta-Gamage, J. Bai, C. Rathnayaka, Y. Zhou, Y. Gu, An advanced physics-informed neural network-based framework for nonlinear and complex topology optimization, *Engineering Structures* 322 (2025) 119194. doi:[10.1016/j.engstruct.2024.119194](https://doi.org/10.1016/j.engstruct.2024.119194).
- [27] P. Rieder, M. Neumann, L. Monteiro Fernandes, A. Mulard, H. Proudhon, F. Willot, V. Schmidt, Stochastic 3D microstructure modeling of twinned polycrystals for investigating the mechanical behavior of γ -TiAl intermetallics, *Computational Materials Science* 238 (2024) 112922. doi:[10.1016/j.commatsci.2024.112922](https://doi.org/10.1016/j.commatsci.2024.112922).
- [28] S. N. Chiu, D. Stoyan, W. S. Kendall, J. Mecke, *Stochastic Geometry and its Applications*, 3rd ed., J. Wiley & Sons, 2013. doi:[10.1002/9781118658222](https://doi.org/10.1002/9781118658222).
- [29] B. Stanfill, H. Hofmann, U. Genschel, rotations: An R package for SO(3) data, *The R Journal* 6 (2014) 68–78. doi:[10.32614/RJ-2014-007](https://doi.org/10.32614/RJ-2014-007).
- [30] D. Rowenhorst, A. D. Rollett, G. S. Rohrer, M. Groeber, M. Jackson, P. J. Konijnenberg, M. D. Graef, Consistent representations of and conversions between 3D rotations, *Modelling and Simulation in Materials Science and Engineering* 23 (2015) 083501. doi:[10.1088/0965-0393/23/8/083501](https://doi.org/10.1088/0965-0393/23/8/083501).
- [31] S. Forest, *Mécanique des Milieux Continus: Volume 1: Théorie*, Les Presses des Mines, 2022.
- [32] F. Willot, Fourier-based schemes for computing the mechanical response of composites with accurate local fields, *Comptes Rendus Mécanique* 343 (2015) 232–245. doi:[10.1016/j.crme.2014.12.005](https://doi.org/10.1016/j.crme.2014.12.005).
- [33] D. Depriester, Mean elastic properties of isotropic polycrystals: Comparison between averaging approximations with numerical computations (2019). doi:[10.13140/RG.2.2.36348.59521/3](https://doi.org/10.13140/RG.2.2.36348.59521/3), preprint.
- [34] F. Willot, B. Abdallah, Y.-P. Pellegrini, Fourier-based schemes with modified green operator for computing the electrical response of heterogeneous media with accurate local fields, *International Journal for Numerical Methods in Engineering* 98 (2014) 518–533. doi:[10.1002/nme.4641](https://doi.org/10.1002/nme.4641).
- [35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. doi:[10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [36] V. Santhanam, V. I. Morariu, L. S. Davis, Generalized deep image to image regression, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2017, pp. 5395–5405. doi:[10.1109/CVPR.2017.573](https://doi.org/10.1109/CVPR.2017.573).
- [37] S. Bulusu, M. Favoni, A. Ipp, D. I. Müller, D. Schuh, Generalization capabilities of translationally equivariant neural networks, *Physical Review D* 104 (2021) 074504. doi:[10.1103/PhysRevD.104.074504](https://doi.org/10.1103/PhysRevD.104.074504).
- [38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: *Proceeding of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019, pp. 8024–8035. doi:[10.5555/3454287.3455008](https://doi.org/10.5555/3454287.3455008).
- [39] D. Ulyanov, A. Vedaldi, V. Lempitsky, Instance normalization: The missing ingredient for fast stylization, 2017. doi:[10.48550/arXiv.1607.08022](https://doi.org/10.48550/arXiv.1607.08022), preprint.
- [40] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: *Proceedings of the 30th International Conference on Machine Learning*, 2013, p. 3.
- [41] L. Lu, Y. Shin, Y. Su, G. E. Karniadakis, Dying ReLU and initialization: Theory and numerical examples, *Communications in Computational Physics* 28 (2020) 1671–1706. doi:[10.4208/cicp.OA-2020-0165](https://doi.org/10.4208/cicp.OA-2020-0165).

- [42] J. Yvonnet, A fast method for solving microstructural problems defined by digital images: a space Lippmann–Schwinger scheme, *International Journal for Numerical Methods in Engineering* 92 (2012) 178–205. doi:[10.1002/nme.4334](https://doi.org/10.1002/nme.4334), publisher: John Wiley & Sons, Ltd.
- [43] V. Sovrasov, ptflops: a flops counting tool for neural networks in pytorch framework, 2018-2024. URL: <https://github.com/sovrasov/flops-counter.pytorch>.
- [44] K. Tanaka, T. Ichitsubo, H. Inui, M. Yamaguchi, M. Koiwa, Single-crystal elastic constants of γ -TiAl, *Philosophical Magazine Letters* 73 (1996) 71–78. doi:[10.1080/095008396181019](https://doi.org/10.1080/095008396181019).
- [45] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015, 2015. doi:[10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980).
- [46] R. Hill, Elastic properties of reinforced solids: Some theoretical principles, *Journal of the Mechanics and Physics of Solids* 11 (1963) 357–372. doi:[10.1016/0022-5096\(63\)90036-X](https://doi.org/10.1016/0022-5096(63)90036-X).
- [47] E. Sanchez-Palencia, A. Zaoui, Homogenization Techniques for Composite Media, volume 272 of *Lecture Notes in Physics*, Springer, Berlin, Heidelberg, 1987. doi:[10.1007/3-540-17616-0](https://doi.org/10.1007/3-540-17616-0).
- [48] C. Huet, Application of variational concepts to size effects in elastic heterogeneous bodies, *Journal of the Mechanics and Physics of Solids* 38 (1990) 813–841. doi:[10.1016/0022-5096\(90\)90041-2](https://doi.org/10.1016/0022-5096(90)90041-2).
- [49] M. Moakher, A. N. Norris, The closest elastic tensor of arbitrary symmetry to an elasticity tensor of lower symmetry, *Journal of Elasticity* 85 (2006) 215–263. doi:[10.1007/s10659-006-9082-0](https://doi.org/10.1007/s10659-006-9082-0).
- [50] P. Tiraviriyaporn, S. Aimmanee, Energy-based universal failure criterion and strength-Poisson’s ratio relationship for isotropic materials, *International Journal of Mechanical Sciences* 230 (2022) 107534. URL: <https://www.sciencedirect.com/science/article/pii/S0020740322004313>. doi:[10.1016/j.ijmecsci.2022.107534](https://doi.org/10.1016/j.ijmecsci.2022.107534).