

A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials

Zeliang Liu^{a,*}, C.T. Wu^a, M. Koishi^b

^a*Livermore Software Technology Corporation (LSTC), Livermore, CA 94551, USA*

^b*Koishi Laboratory, The Yokohama Rubber Co., LTD., Kanagawa 254-8601, Japan*

Abstract

In this paper, a new data-driven multiscale material modeling method, which we refer to as deep material network, is developed based on mechanistic homogenization theory of representative volume element (RVE) and advanced machine learning techniques. We propose to use a collection of connected mechanistic building blocks with analytical homogenization solutions which avoids the loss of essential physics in generic neural networks, and this concept is demonstrated for 2-dimensional RVE problems and network depth up to 7. Based on linear elastic RVE data from offline direct numerical simulations, the material network can be effectively trained using stochastic gradient descent with backpropagation algorithm, further enhanced by model compression methods. Importantly, the trained network is valid for any local material laws without the need for additional calibration or micromechanics assumption. Its extrapolations to unknown material and loading spaces for a wide range of problems are validated through numerical experiments, including linear elasticity with high contrast of phase properties, nonlinear history-dependent plasticity and finite-strain hyperelasticity under large deformations.

By discovering a proper topological representation of RVE with fewer degrees of freedom, this intelligent material model is believed to open new possibilities of high-fidelity efficient concurrent simulations for a large-scale heterogeneous structure. It also provides a mechanistic understanding of structure-property relations across material length scales and enables the development of parameterized microstructural database for material design and manufacturing.

Keywords: Material network, building blocks, machine learning, nonlinear plasticity, large deformations

1. Introduction

In the past decade, multi-scale simulation methods have demonstrated significant advantages for computational mechanics due to their ability to consider microscopic heterogeneities inside a material. The macroscopic material properties are strongly affected by the morphology and evolution of the microstructures, especially under extreme events with large material deformations and geometric nonlinearities. While direct simulations for large-scale heterogeneous structures are extremely expensive and uncommon in industrial applications, the representative volume element (RVE) techniques based on homogenization theory [1, 2, 3] are a type of hierarchical multi-scale simulation methods offering the numerical constitutive closure relationship at the macroscopic point.

In terms of RVE analysis and homogenization, direct numerical simulations (DNS), such as finite element method (FEM) [2, 4], meshfree methods [5, 6], and fast Fourier transform (FFT)-based methods [7, 8], offer high accuracy at the expense of high computational costs. A myriad of model reduction techniques have been introduced for predicting the effective mechanical properties in a manner that balances computational

*Corresponding author.

Email address: zlliu@lstc.com (Zeliang Liu)

cost and accuracy. Analytical micromechanics methods [9, 10, 1, 11, 12, 13] can be regarded as one type of reduced-order models with high efficiency. However, due to a loss of detailed physics in the microscale, they normally lose accuracy or require extensive model calibrations when irregular complex morphologies, nonlinear history-dependent properties or large deformations are presented. For heterogeneous hyperelastic materials, manifold-learning methods like isomap are used for nonlinear dimensionality reduction of microscopic strain fields [14]. The model reduction of history-dependent plastic materials can be more complex and challenging. Two examples are non-uniform transformation field analysis (NTFA) [15, 16] and variants of the principle component analysis [17] or proper orthogonal decomposition (POD) [18, 19, 20]. However, they usually require extensive *a priori* simulations for interpolating nonlinear responses, and their extrapolation capability for new material inputs is usually limited. Recently, the self-consistent clustering analysis (SCA) [21, 22] has demonstrated a powerful trade-off between accuracy and efficiency in predicting small-strain elasto-plastic behavior through clustering techniques, and it only requires linear elastic simulations in the offline stage.

Meanwhile, current advanced machine learning models (e.g. artificial neural networks and deep learning) have achieved great successes in broad areas of computer engineering, such as computer vision, gaming, and natural language processing [23, 24, 25, 26]. Although these techniques are able to construct models for complex input-output relations, their applications to mechanics of materials are still limited. Methods like tensor product approximation and neural networks have been employed to directly construct the overall strain energy density surface of the RVE [27, 28, 29, 30]. Artificial neural networks have also been used to approximate constitutive behavior by fitting the stress-strain relation directly [31, 32, 33]. However, these techniques are usually problem-dependent and may suffer from "the danger of extrapolation" beyond the original sampling space, e.g. different material laws and loading paths. Issues like material history dependency, physical invariance and conservation laws are not naturally resolved, mainly due to the loss of physics in the current machine learning models.

The paper introduces a novel multiscale material modeling method called deep material network, which represents the DNS model of RVE by a hierarchical topological structure with mechanistic building blocks, and is able to predict nonlinear material behaviors both accurately and efficiently. In the proposed material network, the above-mentioned limitations of various reduced order methods are addressed simultaneously by meeting three fundamental goals : 1) avoiding an extensive offline sampling stage (e.g. POD, NTFA and generic neural network) and only requiring linear elastic RVE analysis; 2) eliminating the need for extra calibration of the constitutive laws (e.g. NTFA, methods based on isomap) or micro-mechanical assumption of homogenization (e.g. micromechanics methods, self-consistent scheme in SCA); 3) discovering an efficient reduced representation of RVE without the loss of physics and danger of extrapolation (e.g. generic neural network). Due to its intrinsic hierarchy structure, its computational time is proportional to the number of degrees of freedom in the system. After one time offline training, the optimized material network creates a microstructural database of the RVE by virtue of its unique capability of extrapolation to unknown material and loading spaces, which is useful for multiscale concurrent simulation and material design.

In Section 2, the theory of material network, including the building block and network architecture, are explained for RVEs under 2-dimensional (2D) plane strain condition. In Section 3, machine learning approaches, such as stochastic gradient descent (SGD) with backpropagation algorithm and model compression algorithms, are developed for training the material network. Extrapolations to general RVE problems with material and geometric nonlinearities are discussed in Section 4. Finally, applications to several challenging problems are addressed in Section 5, including linear elasticity with high contrast of phase properties, nonlinear history-dependent plasticity and finite-strain hyperelasticity under large deformations. Concluding remarks are given in Section 6.

2. Theory of material network

The basic concept of material network is to use a collection of connected simple building blocks to describe complex RVE responses, similar to the one of artificial neural network where neurons are connected to define an arbitrary function. The building block is chosen to be a simple structure with analytical homogenization solutions, and the architecture of the material network represents the path of the homogenization process

from each individual phase to the overall macroscopic material. With advanced mechanistic machine learning approaches introduced in Section 3, the trained material network provides a possibility to obtain a simplified representation of the RVE with heterogeneous microstructures.

2.1. The physically based building block

The theory of material network is first developed for a two-phase linearly elastic building block in 2-dimensional (2D) plane strain condition. Under small-strain assumption, the strain and stress measures are the infinitesimal strain $\underline{\varepsilon}$ and the Cauchy stress σ , which are related by the fourth-order compliance tensor \mathbf{D} . The overall stress-strain relation of the building block can be expressed as

$$\bar{\underline{\varepsilon}} = \bar{\mathbf{D}} : \bar{\sigma}. \quad (1)$$

For materials 1 and 2, we have

$$\underline{\varepsilon}^1 = \mathbf{D}^1 : \sigma^1, \quad \underline{\varepsilon}^2 = \mathbf{D}^2 : \sigma^2 \quad \text{and} \quad f_1 + f_2 = 1, \quad (2)$$

where f_1 and f_2 are the volume fractions of materials 1 and 2, respectively. In Mandel notation, the stress and strain can be written as

$$\sigma = \{\underline{\sigma}_{11}, \underline{\sigma}_{22}, \sqrt{2}\underline{\sigma}_{12}\}^T = \{\sigma_1, \sigma_2, \sigma_3\}^T \quad (3)$$

and

$$\varepsilon = \{\underline{\varepsilon}_{11}, \underline{\varepsilon}_{22}, \sqrt{2}\underline{\varepsilon}_{12}\}^T = \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}^T,$$

where the subscript 3 denotes the shear direction. The compliance matrix can be written as

$$\varepsilon = \mathbf{D}\sigma, \quad \mathbf{D} = \begin{Bmatrix} D_{11} & D_{12} & D_{13} \\ \text{sym} & D_{22} & D_{23} \\ & & D_{33} \end{Bmatrix}. \quad (4)$$

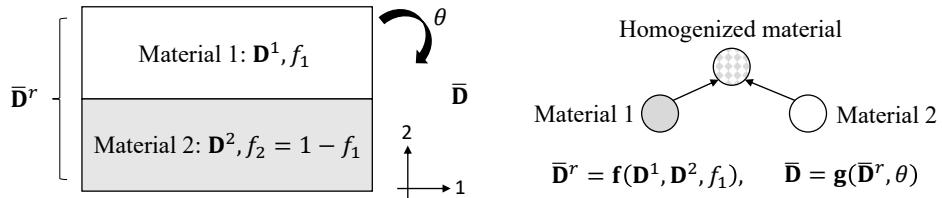


Figure 1: Illustration of the two-layer building block. The compliance matrix after the homogenization operation is $\bar{\mathbf{D}}^r$, and the one after the rotation operation is denoted by $\bar{\mathbf{D}}$.

Similar to artificial neural networks, we want the building block to be simple and easy for the analysis. Here a two-layer structure is proposed. As shown in Figure 1. there are two operations in the building block: 1) homogenization which gives $\bar{\mathbf{D}}^r$ and 2) rotation of the two-layer structure which gives $\bar{\mathbf{D}}$.

The homogenized compliance matrix before the rotation $\bar{\mathbf{D}}^r$ can be expressed as a function of the microscopic compliance matrix of each constituent and its morphological descriptors, in this case, the volume fractions f_1 , with $f_2 = 1 - f_1$. Mathematically, it can be written as

$$\bar{\mathbf{D}}^r = \mathbf{f}(\mathbf{D}^1, \mathbf{D}^2, f_1). \quad (5)$$

Analytical homogenized results are available for this simple two-layer structure, derived based on the equilibrium condition

$$\sigma_2^1 = \sigma_2^2, \quad \sigma_3^1 = \sigma_3^2, \quad (6)$$

and kinematic constraint

$$\varepsilon_1^1 = \varepsilon_1^2. \quad (7)$$

Analytical expressions of the components in the homogenized compliance matrix $\bar{\mathbf{D}}^r$ are

$$\begin{aligned}\bar{D}_{11}^r &= \frac{1}{\Gamma}(D_{11}^1 D_{11}^2), \\ \bar{D}_{12}^r &= \frac{1}{\Gamma}(f_1 D_{12}^1 D_{11}^2 + f_2 D_{12}^2 D_{11}^1), \\ \bar{D}_{13}^r &= \frac{1}{\Gamma}(f_1 D_{13}^1 D_{11}^2 + f_2 D_{13}^2 D_{11}^1), \\ \bar{D}_{22}^r &= f_1 D_{22}^1 + f_2 D_{22}^2 - \frac{1}{\Gamma} f_1 f_2 (D_{12}^1 - D_{12}^2)^2, \\ \bar{D}_{23}^r &= f_1 D_{23}^1 + f_2 D_{23}^2 - \frac{1}{\Gamma} f_1 f_2 (D_{13}^1 - D_{13}^2)(D_{12}^1 - D_{12}^2), \\ \bar{D}_{33}^r &= f_1 D_{33}^1 + f_2 D_{33}^2 - \frac{1}{\Gamma} f_1 f_2 (D_{13}^1 - D_{13}^2)^2,\end{aligned}\tag{8}$$

where

$$\Gamma = f_1 D_{11}^2 + f_2 D_{11}^1 \quad \text{and} \quad f_2 = 1 - f_1.$$

After the homogenization operation, the two-layer structure is rotated and the new compliance matrix $\bar{\mathbf{D}}$ is passed to a child node of the next material building block in the upper level. As shown in Fig 1, the rotation angle is denoted as θ . The matrix \mathbf{R} defines the rotation of a second-order tensor through the angle θ under Mandel notation,

$$\mathbf{R}(\theta) = \begin{Bmatrix} \cos^2 \theta & \sin^2 \theta & \sqrt{2} \sin \theta \cos \theta \\ \sin^2 \theta & \cos^2 \theta & -\sqrt{2} \sin \theta \cos \theta \\ -\sqrt{2} \sin \theta \cos \theta & \sqrt{2} \sin \theta \cos \theta & \cos^2 \theta - \sin^2 \theta \end{Bmatrix},\tag{9}$$

and it satisfies

$$\mathbf{R}^{-1}(\theta) = \mathbf{R}(-\theta).\tag{10}$$

After rotation, the new compliance matrix can be expressed as

$$\bar{\mathbf{D}} = \mathbf{g}(\bar{\mathbf{D}}^r, \theta) = \mathbf{R}(-\theta) \bar{\mathbf{D}}^r \mathbf{R}(\theta).\tag{11}$$

Combining the homogenization and rotation operations in Eq. (5) and (11) yields the completed homogenization function of the two-layer building block,

$$\bar{\mathbf{D}} = \mathbf{g}[\mathbf{f}(\mathbf{D}^1, \mathbf{D}^2, f_1), \theta].\tag{12}$$

Remark 1. It is equivalent to deriving all the analytical homogenization and rotation functions based on the stiffness tensor \mathbf{C} , as we will show in Section 4 and Appendix A. However, the reason why we use the compliance tensor \mathbf{D} here is that the expressions of the analytical solution based on \mathbf{D} are neater for this small-strain 2D building block in plane strain condition.

Note that analytical forms for building blocks with multiple layers could also be derived for multi-phase material, however, the material network would become more complex and more partial derivatives would be evolved in the learning process. For simplicity, this paper will focus on two-phase RVEs, and the two-layer structure, as well as the corresponding binary-tree architectures, will be mainly discussed. In practice, the compliance matrix are stored in a vectorized form with 6 independent variables. The data flow for one building block after the vectorization is shown in Fig. 2.

Derivatives of $\bar{\mathbf{D}}$ and $\bar{\mathbf{D}}^r$ will be used in the stochastic gradient descent with backpropagation algorithm introduced in Section 3.2. Specifically, derivatives of $\bar{\mathbf{D}}$ with respect to the rotation angle θ are

$$\frac{\partial \bar{\mathbf{D}}}{\partial \theta} = -\mathbf{R}'(-\theta) \bar{\mathbf{D}}^r \mathbf{R}(\theta) + \mathbf{R}(-\theta) \bar{\mathbf{D}}^r \mathbf{R}'(\theta)\tag{13}$$

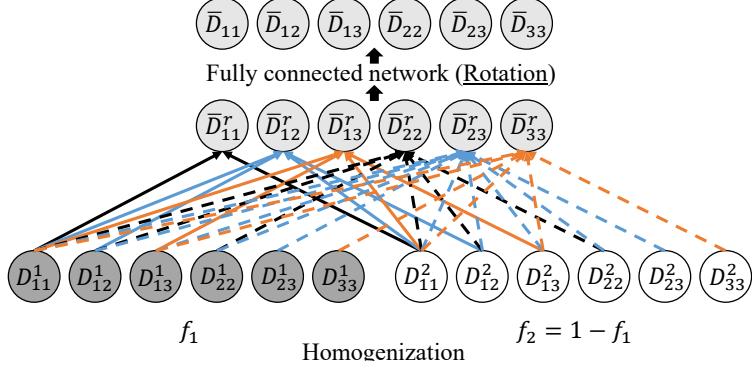


Figure 2: Detailed data flows of the components in the compliance matrices inside a building block.

with

$$\mathbf{R}'(\theta) = \begin{Bmatrix} -\sin 2\theta & \sin 2\theta & \sqrt{2}\cos 2\theta \\ \sin 2\theta & -\sin 2\theta & -\sqrt{2}\cos 2\theta \\ -\sqrt{2}\cos 2\theta & \sqrt{2}\cos 2\theta & -2\sin 2\theta \end{Bmatrix}. \quad (14)$$

In terms of the derivatives of $\bar{\mathbf{D}}$ with respect to the components in $\bar{\mathbf{D}}^r$, we have

$$\frac{\partial \bar{D}_{ij}}{\partial \bar{D}_{kl}^r} = R_{ik}(-\theta)R_{lj}(\theta). \quad (15)$$

Derivatives of $\bar{\mathbf{D}}^r$ with respect to the volume fraction f_1 ($f_2 = 1 - f_1$) are

$$\begin{aligned} \frac{\partial \bar{D}_{11}^r}{\partial f_1} &= \frac{1}{\Gamma} (D_{11}^1 - D_{11}^2) \bar{D}_{11}^r, \\ \frac{\partial \bar{D}_{12}^r}{\partial f_1} &= \frac{1}{\Gamma} [(D_{11}^1 - D_{11}^2) \bar{D}_{12}^r + D_{12}^1 D_{11}^2 - D_{12}^2 D_{11}^1], \\ \frac{\partial \bar{D}_{13}^r}{\partial f_1} &= \frac{1}{\Gamma} [(D_{11}^1 - D_{11}^2) \bar{D}_{13}^r + D_{13}^1 D_{11}^2 - D_{13}^2 D_{11}^1], \\ \frac{\partial \bar{D}_{22}^r}{\partial f_1} &= D_{22}^1 - D_{22}^2 + \frac{1}{\Gamma^2} [(f_1)^2 D_{11}^2 - (f_2)^2 D_{11}^1] (D_{12}^1 - D_{12}^2)^2, \\ \frac{\partial \bar{D}_{23}^r}{\partial f_1} &= D_{23}^1 - D_{23}^2 + \frac{1}{\Gamma^2} [(f_1)^2 D_{11}^2 - (f_2)^2 D_{11}^1] (D_{13}^1 - D_{13}^2)(D_{12}^1 - D_{12}^2), \\ \frac{\partial \bar{D}_{33}^r}{\partial f_1} &= D_{33}^1 - D_{33}^2 + \frac{1}{\Gamma^2} [(f_1)^2 D_{11}^2 - (f_2)^2 D_{11}^1] (D_{13}^1 - D_{13}^2)^2. \end{aligned} \quad (16)$$

Moreover, derivatives of $\bar{\mathbf{D}}^r$ with respect to the components in \mathbf{D}^1 are

$$\begin{aligned} \frac{\partial \bar{\mathbf{D}}^r}{\partial D_{11}^1} &= \frac{1}{\Gamma} \begin{Bmatrix} f_1 D_{11}^2 D_{11}^2 / \Gamma & f_2 (-\bar{D}_{12}^r + D_{12}^2) / \Gamma & f_2 (-\bar{D}_{13}^r + D_{13}^2) / \Gamma \\ \text{sym} & f_1 (f_2)^2 (D_{12}^1 - D_{12}^2)^2 / \Gamma & f_1 (f_2)^2 (D_{13}^1 - D_{13}^2)(D_{12}^1 - D_{12}^2) / \Gamma \\ & f_1 (f_2)^2 (D_{13}^1 - D_{13}^2)^2 / \Gamma \end{Bmatrix}, \\ \frac{\partial \bar{\mathbf{D}}^r}{\partial D_{12}^1} &= \begin{Bmatrix} 0 & f_1 D_{11}^2 / \Gamma & 0 \\ \text{sym} & -2f_1 f_2 (D_{12}^1 - D_{12}^2) / \Gamma & -f_1 f_2 (D_{13}^1 - D_{13}^2) / \Gamma \\ & 0 & 0 \end{Bmatrix}, \end{aligned} \quad (17)$$

$$\frac{\partial \bar{\mathbf{D}}^r}{\partial D_{13}^1} = \begin{cases} 0 & 0 & f_1 D_{11}^2 / \Gamma \\ \text{sym} & 0 & -f_1 f_2 (D_{12}^1 - D_{12}^2) / \Gamma \\ & & -2f_1 f_2 (D_{13}^1 - D_{13}^2) / \Gamma \end{cases},$$

$$\frac{\partial \bar{\mathbf{D}}^r}{\partial D_{22}^1} = \begin{cases} 0 & 0 & 0 \\ \text{sym} & f_1 & 0 \\ & 0 & 0 \end{cases}, \quad \frac{\partial \bar{\mathbf{D}}^r}{\partial D_{23}^1} = \begin{cases} 0 & 0 & 0 \\ \text{sym} & 0 & f_1 \\ & & 0 \end{cases}, \quad \frac{\partial \bar{\mathbf{D}}^r}{\partial D_{33}^1} = \begin{cases} 0 & 0 & 0 \\ \text{sym} & 0 & 0 \\ & & f_1 \end{cases}.$$

Similarly, forms of $\partial \bar{\mathbf{D}}^r / \partial \mathbf{D}^2$ can be obtained by switching the phase index in Eq. (17).

In Section 5.2.3, we will extend the theory of material network to general finite-strain problems, such as heterogeneous hyperelastic RVEs. Furthermore, the concept of two-layer building block can be applied to 3-dimensional (3D) problems, which will be explored in our future work. It should be noted that the existence of an analytical homogenization function is crucial for optimizing the fitting parameters in the material network using gradient-based methods. More details on the training of material network can be found in Section 3.

2.2. Architectures of material network

As shown in Fig. 3, a material network with a binary-tree structure is proposed. The depth of the layer in the tree structure is denoted by N , and i is the layer index. Layer 0 represents the output layer equivalent to the macroscopic homogenized material, and layer $(N+1)$ contains the input from each individual microscopic constituent. Layer N is regarded as the bottom layer. Since a two-layer structure is considered as the basic homogenization building block, each node has 2 child nodes, and there are 2^i nodes at layer i for $i \in [0, N]$.

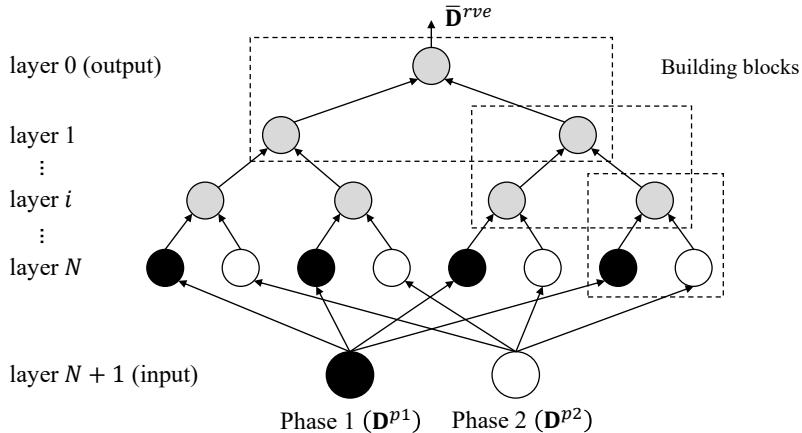


Figure 3: Illustration of material network with depth $N=3$. The nodes in each dashed box form a building block.

To explain the data flows inside the material network, a homogenization building block for the k -th node at layer i is provided in Fig. 4. Correspondingly, the indices of its two child nodes at layer $i+1$ are $2k-1$ and $2k$.

Other than the compliance matrix at the each node, a so-called weighting function w is added into the data flow, which is used for tracking the volume fractions in each building block. When feed forward, the weights of the two child nodes are summed up and passed to their parent node,

$$w_i^k = w_{i+1}^{2k-1} + w_{i+1}^{2k}. \quad (18)$$

By performing the summation recursively, the weight w_i^k can be expressed as a summation of weights of its descendant nodes at the bottom layer N ,

$$w_i^k = \sum_{j=2^{N-i}(k-1)+1}^{2^{N-i}k} w_N^j. \quad (19)$$

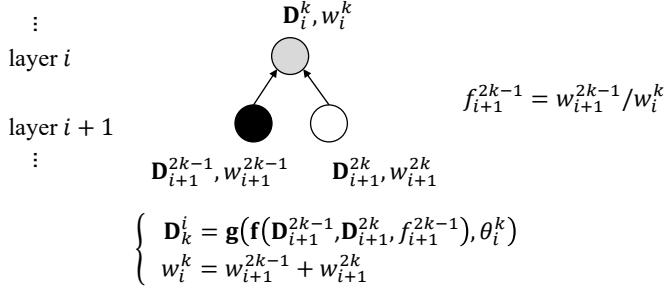


Figure 4: Data flow in the building block for the k -th node at layer i . The volume fraction f_{i+1}^{2k-1} is defined upon the weights.

Therefore, only the weights at bottom layer N are independent parameters. The partial derivative of w_i^k with respect to w_N^j can be written as

$$\frac{\partial w_i^k}{\partial w_N^j} = \begin{cases} 1 & \text{if } k = \lceil j/2^{N-i} \rceil \\ 0 & \text{otherwise} \end{cases} . \quad (20)$$

The weights at layer N are activated through the rectified linear unit (ReLU) [34], which have been widely used in deep learning. For $j \in [1, 2^N]$, we have

$$w_N^j(z_N^j) = Re(z_N^j) = \max(z_N^j, 0). \quad (21)$$

The activations z_N^j are defined at the bottom layer and determine all the weighting functions w in the network. Therefore, z_N^j are treated as the fitting parameters. The derivative of the ReLU is

$$Re'(z_N^j) = \begin{cases} 1 & \text{if } z_N^j > 0 \\ 0 & \text{otherwise} \end{cases} . \quad (22)$$

Once a unit is deactivated ($z_N^j < 0$ and $Re(z_N^j) = 0$), its gradient vanishes and the unit will never be activated again during the training process. This is good for automatically simplifying the material network and increasing the training speed, as long as the learning rate is set appropriately.

Back to the building block in Fig. 4, the volume fraction of the first child nodes in the corresponding two-layer structure can be written as

$$f_{i+1}^{2k-1} = w_{i+1}^{2k-1} / w_i^k. \quad (23)$$

The derivative of f_{i+1}^{2k-1} with respect to w_N^j at the bottom layer is

$$\frac{\partial f_{i+1}^{2k-1}}{\partial w_N^j} = \frac{1}{w_i^k} \left(\frac{\partial w_{i+1}^{2k-1}}{\partial w_N^j} - f_{i+1}^{2k-1} \frac{\partial w_i^k}{\partial w_N^j} \right). \quad (24)$$

The compliance matrices \mathbf{D}^r and \mathbf{D} at the k -th node in layer i become

$$\mathbf{D}_i^{rk} = \mathbf{f}(\mathbf{D}_{i+1}^{2k-1}, \mathbf{D}_{i+1}^{2k}, f_{i+1}^{2k-1}) \quad (25)$$

and

$$\mathbf{D}_i^k = \mathbf{g}[\mathbf{f}(\mathbf{D}_{i+1}^{2k-1}, \mathbf{D}_{i+1}^{2k}, f_{i+1}^{2k-1}), \theta_i^k]. \quad (26)$$

By combining Eq. (19), (21), (23) and (26) the homogenized compliance matrix of the RVE $\bar{\mathbf{D}}^{rve}$ can be written as a function of the compliance matrix from each material phase (\mathbf{D}^{p1} and \mathbf{D}^{p2}) and the fitting

parameters (z and θ),

$$\underbrace{\bar{\mathbf{D}}^{rve}}_{\text{Outputs}} = \mathbf{D}_0^1 = \mathbf{h}(\underbrace{\mathbf{D}^{p1}, \mathbf{D}^{p2}}_{\text{Inputs}}, \overbrace{z_N^{j=1,2,\dots,2^N}, \theta_{i=0,1,\dots,N}^{k=1,2,\dots,2^i}}^{\text{Fitting parameters}}). \quad (27)$$

The number of layers N can be regarded as a hyper-parameter of the material network, and there are totally $(3 \times 2^N - 1)$ fitting parameters in a material network with depth N for a 2D problem. The optimum choice of N needed to be studied through numerical experiments. A shallow material network with a small N may not be sufficient for capturing the RVE behavior, while a deep network with a large N will increase the training time, as well as the computational cost for online extrapolation.

Remark 2. Other than z_N^j in the bottom layer, it is also possible to directly use the volume fraction of the nodes at each layer as the fitting parameters. However, according to our tests, this would cause the vanishing gradient problem of volume fractions in the deep layers and might also result in unwanted early deactivation of sub-tree structures, whereas, the training process based on z_N^j is more continuous and stable.

3. Machine learning of deep material network

3.1. Cost function and dataset for training

The procedure for training the material network is illustrated in Fig. 5. The dataset for training is generated through high-fidelity DNS of an RVE, or experiments (this can be challenging due to limited experimental tests). The inputs, outputs and fitting parameters for this training/optimization problem are listed below, with the number of variables given in the parentheses,

Inputs (12):

$$\mathbf{D}^{p1} \text{ and } \mathbf{D}^{p2}.$$

Outputs (6):

$$\bar{\mathbf{D}}^{dns}.$$

Fitting parameters $(3 \times 2^N - 1)$:

$$z_N^{j=1,2,\dots,2^N}, \theta_{i=0,1,\dots,N}^{k=1,2,\dots,2^i}.$$

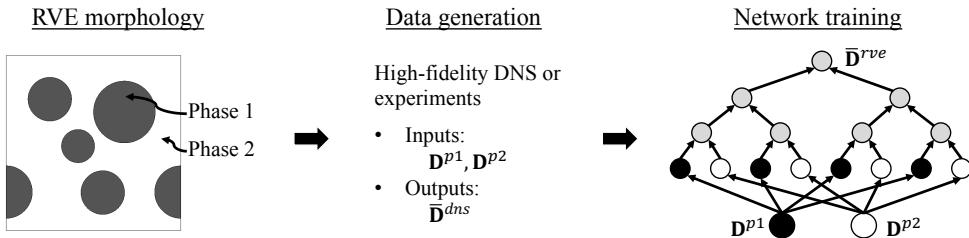


Figure 5: Training procedure of material network. The training data may be generated from high-fidelity DNS and experiments.

When generating the input data, the components in \mathbf{D}^{p1} and \mathbf{D}^{p2} cannot be assigned arbitrarily. For a physical material, situations resulting in negative strain energy density should be avoided. For the ease of data generation, the two material phases are considered to be orthotropic elastic,

$$\mathbf{D}^{p1} = \begin{Bmatrix} 1/E_{11}^{p1} & -\nu_{12}^{p1}/E_{22}^{p1} \\ & 1/E_{22}^{p1} \\ & 1/(2G_{12}^{p1}) \end{Bmatrix} \quad \text{and} \quad \mathbf{D}^{p2} = \begin{Bmatrix} 1/E_{11}^{p2} & -\nu_{12}^{p2}/E_{22}^{p2} \\ & 1/E_{22}^{p2} \\ & 1/(2G_{12}^{p2}) \end{Bmatrix}. \quad (28)$$

To remove the redundancy due to the scaling effect, we have

$$E_{11}^{p1} E_{22}^{p1} = 1, \quad \log_{10}(E_{11}^{p2} E_{22}^{p2}) \in U[-4, 4]. \quad (29)$$

The other inputs are selected randomly as

$$\begin{aligned} \log_{10}(E_{22}^{p1}/E_{11}^{p1}) &\in U[-1, 1], \quad \log_{10}(E_{22}^{p2}/E_{11}^{p2}) \in U[-1, 1], \\ \frac{G_{12}^{p1}}{\sqrt{E_{22}^{p1} E_{11}^{p1}}} &\in U[0.25, 0.5], \quad \frac{G_{12}^{p2}}{\sqrt{E_{22}^{p2} E_{11}^{p2}}} \in U[0.25, 0.5], \end{aligned}$$

where U stands for uniform distribution. Poisson ratios are selected to guarantee the compliance matrices are positive definite for the strain energy density to be positive,

$$\frac{\nu_{12}^{p1}}{\sqrt{E_{22}^{p1}/E_{11}^{p1}}} \in U[0.3, 0.7], \quad \frac{\nu_{12}^{p2}}{\sqrt{E_{22}^{p2}/E_{11}^{p2}}} \in U[0.3, 0.7].$$

Design of experiments based on the Latin hypercube is performed to generate the input space.

To quantify how well we are training the network, a cost function based on the mean square errors (MSE) is proposed:

$$C_0(z, \theta) = \frac{1}{2N_s} \sum_s C_s(z, \theta) = \frac{1}{2N_s} \sum_s \|\bar{\mathbf{D}}_s^{dns} - \mathbf{h}(\mathbf{D}_s^{p1}, \mathbf{D}_s^{p2}, z, \theta)\|^2 / \|\bar{\mathbf{D}}_s^{dns}\|^2. \quad (30)$$

Here, z and θ are the parameters to be fitted, s is the index of the sample (or data point), and N_s is the total number of training samples. Note that the cost function is normalized by the squared norm of $\bar{\mathbf{D}}_s^{dns}$ to remove the scaling effect. The operator $\|\dots\|$ denotes the matrix norm. For a general second-order matrix \mathbf{B} of dimension $m \times m$, the matrix norm is defined as

$$\|\mathbf{B}\| = \sqrt{\sum_{i=1}^m \sum_{j=1}^m b_{ij}^2} = \sqrt{\text{trace}(\mathbf{B}^T \mathbf{B})}, \quad (31)$$

and it is also called the Frobenius norm of matrix \mathbf{B} . Since the trace of a matrix product is invariant under cyclic permutation, it can be proved that the norm of a compliance matrix under Mandel notation is invariant under arbitrary rotation. By using Eq. (11) and the symmetry of the compliance matrix, we have

$$\|\mathbf{D}\| = \sqrt{\text{trace}(\mathbf{DD}^T)} = \sqrt{\text{trace}(\mathbf{R}^{-1} \mathbf{D}^r \mathbf{R} \mathbf{R}^{-1} \mathbf{D}^r \mathbf{R}^T)} = \sqrt{\text{trace}(\mathbf{R} \mathbf{R}^{-1} \mathbf{D}^r \mathbf{D}^r)} = \|\mathbf{D}^r\|. \quad (32)$$

This guarantees that the cost function defined in Eq. (30), as well as the optimized fitting parameters, is independent of the choice of the coordinate system.

To constrain the magnitude of activations z_N^j and make the optimization problem well-posed, an additional term is appended to the cost function C_0 ,

$$C(z, \theta, \lambda) = C_0(z, \theta) + \lambda L(z), \quad (33)$$

where $L(z)$ is defined as

$$L(z) = \left(\sum_j \text{Re}(z_N^j) - \xi \right)^2. \quad (34)$$

The hyper-parameter ξ determines the magnitude of z_N^j . Although the hyper-parameter λ will not alter the optimum fitting parameters for a given training dataset, in practice, it should be set appropriately to expedite the gradient descent algorithm in the training process.

3.2. Stochastic gradient descent and backpropagation algorithm

To minimize the cost function, the stochastic gradient descent (SGD) with backpropagation algorithm will be adopted. The gradient vector of the cost function can be written as

$$\nabla C = \left(\frac{\partial C}{\partial z_N^j}, \frac{\partial C}{\partial \theta_i^k} \right) = \left(\frac{\partial C_0}{\partial z_N^j} + \lambda \frac{\partial L}{\partial z_N^j}, \frac{\partial C_0}{\partial \theta_i^k} \right) \quad (35)$$

with

$$j = 1, 2, \dots, 2^N; \quad i = 0, 1, \dots, N; \quad k = 1, 2, \dots, 2^i.$$

Due to the simplicity of L , we will only focus on deriving the gradients of the cost function C_0 in this section.

In the context of training the neural networks, backpropagation is commonly used to adjust the weight of neurons by calculating the gradient of the cost function. Sometime, this technique is also called backward propagation of errors, because the error is calculated at the output and distributed backward through the network layers. Similar technique can be utilized to train the material network. The heart of backpropagation is the chain rule in calculus, by which the time for computing those partial differentiates can be greatly reduced and less memory is needed. The derivation and implementations of backpropagation for training the material network will be discussed below.

As shown in Fig. 2, each independent component of compliance matrix is regarded as a neuron in the network. After the vectorization, the components in the compliance matrices \mathbf{D} for all the nodes at layer i are denoted by d_i^n , with the index n ranging from 1 to 6×2^i . Meanwhile, we use a_i^n to represent the components in \mathbf{D}^r for all the nodes at layer i . In the algorithm, we define the error δ_i^n of neuron n in layer i by

$$\delta_i^n = \frac{\partial C_0}{\partial d_i^n}, \quad (36)$$

and the error α_i^n by

$$\alpha_i^n = \frac{\partial C_0}{\partial a_i^n}. \quad (37)$$

In the output layer, the components of δ_0 are given by

$$\delta_0^n = \frac{\partial C_0}{\partial d_0^n}. \quad (38)$$

Since C_0 takes a quadratic form in Eq. (30), δ_0^n can be easily computed. For $i \geq 1$, the error α_{i-1}^k and δ_i^n can be computed by

$$\alpha_{i-1}^k = \frac{\partial C_0}{\partial d_{i-1}^j} \frac{\partial d_{i-1}^j}{\partial a_{i-1}^k} = \delta_{i-1}^j \frac{\partial d_{i-1}^j}{\partial a_{i-1}^k}, \quad (39)$$

$$\delta_i^n = \frac{\partial C_0}{\partial a_{i-1}^k} \frac{\partial a_{i-1}^k}{\partial d_i^n} = \alpha_{i-1}^k \frac{\partial a_{i-1}^k}{\partial d_i^n}. \quad (40)$$

All the derivatives have been previously defined. The expressions of $\partial d_{i-1}^j / \partial a_{i-1}^k$ and $\partial a_{i-1}^k / \partial d_i^n$ can be found in Eq. (15) and (17), respectively.

By combining Eq. (38), (39) and (40), we can compute the error δ_i and α_i at any layer in the network. Within one step, the error of cost function propagates backward from the output layer $i = 0$ to the bottom layer $i = N$. After the backpropagation of error for compliance matrices, we can compute the rate of change of the cost function with respect to a rotation angle θ_i^k at layer i by

$$\frac{\partial C_0}{\partial \theta_i^k} = \delta_i^j \frac{\partial d_i^j}{\partial \theta_i^k}. \quad (41)$$

where $\partial d_i^j / \partial \theta_i^k$ can be found in Eq. (13). Meanwhile, the rate of change of the cost function with respect to an activation z_N^j in the bottom layer can be computed by

$$\frac{\partial C_0}{\partial z_N^j} = \left(\alpha_{i-1}^l \frac{\partial a_{i-1}^l}{\partial f_i^m} \frac{\partial f_i^m}{\partial w_N^j} \right) \circ Re'(z_N^j), \quad (42)$$

where \circ denotes the Hadamard (element-wise) product. Analytical forms of $\partial a_{i-1}^l / \partial f_i^m$ are provided in Eq. (16), and $\partial f_i^m / \partial w_N^j$ is defined in Eq. (24).

Finally, we conclude the backpropagation algorithm for finding gradients of the cost function C_0 with respect to the fitting parameters in the material network.

1. **Input:** Set the initial values for z_N and θ_i ($i = 0, 1, \dots, N$)
2. **Feedforward:** For each layer i , compute a_i and d_i
3. **Output error** δ_0 : Compute the vector δ_0 using Eq. (38)
4. **Backpropagate the error:** For each $i = 1, 2, \dots, N$, compute α_{i-1} and δ_i iteratively using Eq. (39) and (40)
5. **Output:** Compute the gradients of the cost function using Eq. (41) and (42)

In a standard gradient descent method, the gradients of each sample ∇C_s are computed and averaged to get ∇C based on Eq. (30),

$$\nabla C = \frac{1}{2N_s} \sum_s \nabla C_s. \quad (43)$$

To accelerate the training speed, the stochastic gradient descent (SGD) is used to train the material network after the gradient vector is obtained. Instead of computing and averaging the gradients over all the samples at each step, a small number M of samples are randomly picked to estimate the gradient ∇C . In this way, the original dataset is divided into several mini-batches which will be used in a sequence of learning steps. If we label the samples in a mini-batch by s_1, s_2, \dots, s_M , the gradient of cost function for the corresponding learning step can be approximated by

$$\nabla C \approx \frac{1}{2M} \sum_{i=1}^M \nabla C_{s_i}. \quad (44)$$

Writing out the gradient decent updating rule in terms of the components, we have

$$z_N^{j'} = z_N^j - \eta \frac{\partial C}{\partial z_N^j} \quad (45)$$

and

$$\theta_i^{k'} = \theta_i^k - \eta \frac{\partial C}{\partial \theta_i^k}, \quad (46)$$

where η is a positive parameter known as the learning rate. Here, an epoch is defined as each time the algorithm has processed all the mini-batches and seen all the samples in the original dataset. In practice, after an epoch is completed, the dataset will be randomly shuffled and prepared for the next epoch to minimize the sample bias.

All fitting parameters are initialized randomly following a uniform distribution at the beginning of the SGD algorithm,

$$z_N^{j(0)} \sim U(0.2, 0.8) \quad \text{and} \quad \theta_i^{k(0)} \sim U(-\pi/2, \pi/2). \quad (47)$$

In order to reduce the influence of the constraint term $L(z)$ at early training stage, the hyper-parameter ξ is chosen to be

$$\xi = E(z_N^{j(0)}) 2^N = 2^{N-1}. \quad (48)$$

This will help to avoid unwanted early deactivation of the material network. Additionally, the learning rate η is an important hyper-parameter that determines the performance of SGD. In the paper, we have utilized a Bold driver algorithm to dynamically adapting the learning rate, and it has effectively improved the training speed in our numerical study. In future, more sophisticated algorithms, like annealing, can be used to tune the learning rate.

3.3. Model compression and parameter reduction

The speed and convergence rate of the network training process can be improved by removing the redundancy in the network, which is called model compression. On the other hand, a less complex network is always beneficial for the extrapolation process in the online stage. As discussed in Section 2.2, the rectified linear unit enables automatic deactivation of nodes during the training; hence it already has a function of network compression. Moreover, two additional approaches are also introduced for further model compression: 1)deletion of the parent node with only one child node; 2) subtree merging based on the similarity search. Illustrations of these two model compression operations are shown in Fig. 6.

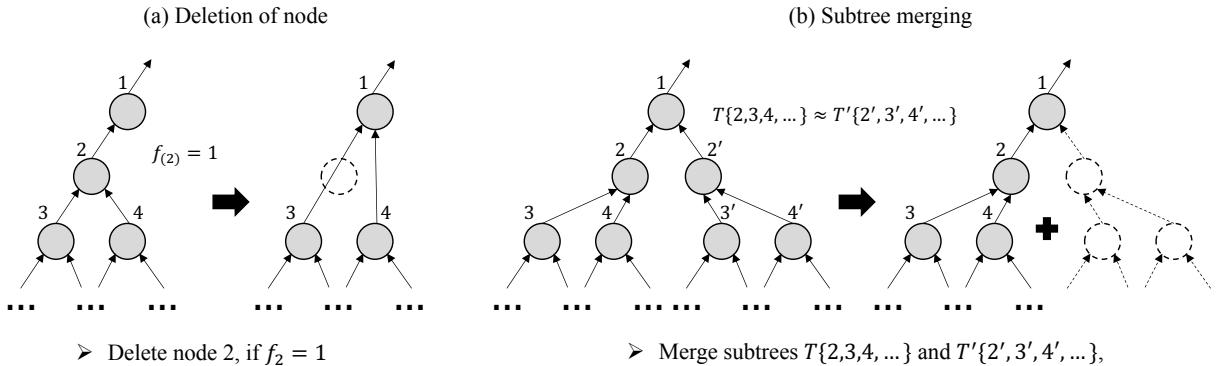


Figure 6: Illustrations of (a) deletion of node and (b) subtree merging for network compression. The merging operation is performed after reordering of the network.

In Fig. 6 (a), node 2 is deleted as its volume fraction is equal to 1. Meanwhile, the rotation angle of node 1 needs to be updated to avoid a sudden jump in the cost function,

$$\theta_{(1)}^{new} = \theta_{(1)}^{old} + \theta_{(2)}^{old}. \quad (49)$$

As shown in Fig. 6 (b), merging of the subtree structures is based on the similarity search. The comparison of the two subtrees, $T\{2, 3, 4, \dots\}$ and $T'\{2', 3', 4', \dots\}$, are performed between all descendant layers of their root nodes 2 and 2'. The differences are evaluated by

$$\delta_{T-T'}^f = \max_{m=3} (|f_{(m)} - f_{(m')}|) \quad \text{and} \quad \delta_{T-T'}^\theta = \max_{m=2} \left(\frac{1}{\pi} |\text{mod}(\theta_{(m)} - \theta_{(m')}, \pi)| \right), \quad (50)$$

where the index (m) denotes the order of nodes in the subtree. If the differences of $\delta_{T-T'}^f$ and $\delta_{T-T'}^\theta$ are both below the tolerances, it is said that

$$T\{2, 3, 4, \dots\} \approx T'\{2', 3', 4', \dots\},$$

and $T'\{2', 3', 4', \dots\}$ will be merged left to compress the network and reduce the number of fitting parameters. Afterwards, the parameters in the new subtree $T\{2, 3, 4, \dots\}$ become

$$w_{(m)}^{new} = w_{(m)}^{old} + w_{(m')}^{old}, \quad \theta_{(m)}^{new} = \frac{1}{2} (\theta_{(m)}^{old} + \theta_{(m')}^{old}) \quad \text{for } m \geq 2. \quad (51)$$

Note that a deletion operation will follow right after the merging operation since $f_{(2)}$ becomes 1 upon the merging of the two subtrees in Fig. 6 (b).

The merging operation should be performed on an ordered material network. At any layer $i \in [1, N - 1]$ of an order material network, the following condition should be satisfied,

$$w_i^{2k-1} \geq w_i^{2k} \quad \forall k \in [1, 2^{i-1}].$$

Each time before the similarity search and subtree merging, the whole material network will be reordered based on the weighting functions. To save the training time, the network compression operations are performed every 10 epochs in our study.

4. Prediction and extrapolation

4.1. Nonlinear small-strain plasticity

Since the material network can capture the essential topological structure of an RVE, it can be used for online prediction of nonlinear plasticity, more than just linear elasticity. Each node at the bottom layer N can be regarded as an individual material node with independent degrees of freedom (DOFs), precisely, the infinitesimal strain ε_N^j . As a result, the total number of DOFs N_{dof} in the material network at the beginning of training is proportional to 2^N . On the other hand, the ReLU activation function and several compression algorithms introduced in Section 3.3 are used to decrease the model complexity during the training. For the final trained network, the number of DOFs becomes proportional to the number of active/remaining nodes at the bottom layer, denoted by N_a ,

$$N_{dof} \propto N_a. \quad (52)$$

In the bottom layer, the Cauchy stress at the j -th node is denoted by σ_N^j . At each loading step, its stress-strain relation can be written as

$$\Delta\varepsilon_N^j = \mathbf{D}_N^j \Delta\sigma_N^j + \delta\varepsilon_N^j. \quad (53)$$

The compliance matrix and incremental stress are given by the local constitutive law,

$$\Delta\sigma_N^j = \Delta\sigma_N^j(\Delta\varepsilon_N^j, \varepsilon_N^j, \sigma_N^j, \beta_N^j, \dots) \quad (54)$$

and

$$\mathbf{D}_N^j = \mathbf{D}_N^j(\Delta\varepsilon_N^j, \varepsilon_N^j, \sigma_N^j, \beta_N^j, \dots), \quad (55)$$

where β_N^j is the vector of history-dependent internal variables. Once the compliance matrix and incremental stress are obtained, we can compute the residual strain in the bottom layer by

$$\delta\varepsilon_N^j = \Delta\varepsilon_N^j - \mathbf{D}_N^j \Delta\sigma_N^j. \quad (56)$$

Note that the residual strain comes from the material non-linearity. In other words, if all the materials are linear elastic, there will be no residual strain and the homogenization procedure will be same as the one described for the training process.

In a network with material non-linearity, both of the compliance matrix \mathbf{D} and residual strain $\delta\varepsilon$ are feed forward from the bottom layer to the output layer. In Fig. 4, the stress-strain relations for the $2k$ -th and $(2k - 1)$ -th nodes on layer $i + 1$ at each loading step can be written as

$$\Delta\varepsilon_{i+1}^{2k-1} = \mathbf{D}_{i+1}^{2k-1} \Delta\sigma_{i+1}^{2k-1} + \delta\varepsilon_{i+1}^{2k-1} \quad \text{and} \quad \Delta\varepsilon_{i+1}^{2k} = \mathbf{D}_{i+1}^{2k} \Delta\sigma_{i+1}^{2k} + \delta\varepsilon_{i+1}^{2k}, \quad (57)$$

and the stress-strain relation of their parent node at layer i is

$$\Delta\varepsilon_i^k = \mathbf{D}_i^k \Delta\sigma_i^k + \delta\varepsilon_i^k. \quad (58)$$

In the forward homogenization process, the compliance matrix and residual strain at the parent node are calculated from the ones in its child nodes,

$$\mathbf{D}_i^k = \mathbf{p}(\mathbf{D}_{i+1}^{2k-1}, \mathbf{D}_{i+1}^{2k}) \quad (59)$$

and

$$\delta\boldsymbol{\varepsilon}_i^k = \mathbf{q}(\mathbf{D}_{i+1}^{2k-1}, \mathbf{D}_{i+1}^{2k}, \delta\boldsymbol{\varepsilon}_{i+1}^{2k-1}, \delta\boldsymbol{\varepsilon}_{i+1}^{2k}). \quad (60)$$

Again, due to the simplicity of the two-layer building block, analytical forms of function \mathbf{p} and \mathbf{q} can be derived. Since the function \mathbf{p} for compliance matrices takes the same form as Eq. (12) in Section 2.1, we will only provide the form of function \mathbf{q} for residual strain here. For the generic two-layer structure illustrated in Fig. 1, we have

$$\delta\bar{\boldsymbol{\varepsilon}} = \mathbf{q}(\mathbf{D}^1, \mathbf{D}^2, \delta\boldsymbol{\varepsilon}^1, \delta\boldsymbol{\varepsilon}^2), \quad (61)$$

where $\delta\bar{\boldsymbol{\varepsilon}}$ is the overall residual strain at the parent node and $\delta\boldsymbol{\varepsilon}^1, \delta\boldsymbol{\varepsilon}^2$ are the residual strains at the two child nodes. Similarly, the function \mathbf{q} contains a homogenization operation and a rotation operation. The homogenization operation gives the homogenized residual strain $\delta\bar{\boldsymbol{\varepsilon}}^r$ of the two-layer structure,

$$\delta\bar{\boldsymbol{\varepsilon}}^r = \delta\bar{\boldsymbol{\varepsilon}}^r(\mathbf{D}^1, \mathbf{D}^2, \boldsymbol{\varepsilon}^1, \boldsymbol{\varepsilon}^2, f_1), \quad (62)$$

and the rotation operation outputs the overall residual strain $\delta\bar{\boldsymbol{\varepsilon}}$ by

$$\delta\bar{\boldsymbol{\varepsilon}} = \mathbf{R}(-\theta)\delta\bar{\boldsymbol{\varepsilon}}^r, \quad (63)$$

where $\mathbf{R}(-\theta)$ is the rotation matrix defined in Eq. (9). Based on the equilibrium condition and kinematic constraints of the two-layer structure, the analytical forms of $\delta\bar{\boldsymbol{\varepsilon}}^r$ in Eq. (62) can be derived as

$$\begin{aligned} \delta\bar{\varepsilon}_{11}^r &= \frac{1}{\Gamma}(f_1 D_{11}^2 \delta\varepsilon_{11}^1 + f_2 D_{11}^1 \delta\varepsilon_{11}^2), \\ \delta\bar{\varepsilon}_{22}^r &= f_1 \delta\varepsilon_{22}^1 + f_2 \delta\varepsilon_{22}^2 - \frac{1}{\Gamma} f_1 f_2 (D_{12}^1 - D_{12}^2) (\delta\varepsilon_{11}^1 - \delta\varepsilon_{11}^2), \\ \delta\bar{\varepsilon}_{12}^r &= f_1 \delta\varepsilon_{12}^1 + f_2 \delta\varepsilon_{12}^2 - \frac{1}{\Gamma} f_1 f_2 (D_{13}^1 - D_{13}^2) (\delta\varepsilon_{11}^1 - \delta\varepsilon_{11}^2), \end{aligned} \quad (64)$$

with

$$\Gamma = f_1 D_{11}^2 + f_2 D_{11}^1 \quad \text{and} \quad f_2 = 1 - f_1.$$

Finally, after the forward homogenization process, the stress-strain relation at the output layer $i = 0$ is obtained,

$$\Delta\boldsymbol{\varepsilon}_0^1 = \mathbf{D}_0^1 \Delta\boldsymbol{\sigma}_0^1 + \delta\boldsymbol{\varepsilon}_0^1. \quad (65)$$

Macroscopic boundary conditions are applied on the output layer, and $\Delta\boldsymbol{\varepsilon}_0^1$ and $\Delta\boldsymbol{\sigma}_0^1$ represent the macroscopic strain and stress, respectively. In the de-homogenization process, the incremental strain and stress data is feed backward from the output layer to the bottom layer, using the homogenized constitutive law Eq. (58) at each node.

Remark 3. For nonlinear history-dependent materials, the internal variables are stored at each individual active node in the bottom layer. Once each material law is evaluated at the bottom layer (e.g. loading-unloading conditions), the information of compliance (or stiffness) tensor and residual strain (or stress) are propagated forward through the network to give the homogenized properties at the output layer.

Newton's method is used for solving the nonlinear system. A Newton's iteration consists of one forward homogenization process and one backward de-homogenization process,

$$\text{layer } N \xrightarrow[\text{homogenization}]{\text{forward}} \text{layer } 0 \text{ (macroscale)} \xrightarrow[\text{de-homogenization}]{\text{backward}} \text{layer } N.$$

After each iteration, the new strain increment $\Delta\boldsymbol{\varepsilon}_N^{j(\text{new})}$ at the bottom layer can be computed. The relative difference between the strain increments at the current and previous iterations will be computed and used for the convergence check. If convergence is not yet achieved, the compliance matrices and residual strains at the bottom layer will be recomputed based upon the new $\Delta\boldsymbol{\varepsilon}_N^{j(\text{new})}$, and used for the next iteration. Upon convergence, the internal variables at each active node in the bottom layer will be updated, and the analysis moves on to the next loading step. Note that the residual strain $\delta\boldsymbol{\varepsilon}_N^j$ is not necessarily equal to zero for nonlinear materials. The same iterative procedure can be applied to finite-strain problem as will be discussed in Section 4.2.

If we define the homogenization and de-homogenization processes in Fig. 4 as one unit operation, the total number of operations of the whole material network in one Newton's iteration is proportional to the number of DOFs,

$$N_{op} \propto N_{dof} \quad \text{or} \quad N_{op} \propto N_a. \quad (66)$$

This is advantageous over most existing homogenization methods, such as FEM ($N_{op} \propto (N_{dof})^{2.x}$), integral equation-based methods ($N_{op} \propto (N_{dof})^3$) [21]. The computational cost of material network will be evaluated numerically in Section 5.2.2.

4.2. Finite-strain problem

The online extrapolation procedure of the material network based on the two-layer building block can also be extended to finite-strain problems with large deformation. In finite-strain problems, we choose the deformation gradient \mathbf{F} and the first Piola-Kirchhoff stress \mathbf{P} as the strain and stress measures, respectively. Here, we defined the tangent elasticity tensor \mathbf{A} in the rate form as

$$\dot{\mathbf{P}} = \mathbf{A} : \dot{\mathbf{F}} \quad \text{or} \quad \dot{P}_{ij} = A_{ijkl} \dot{F}_{kl} \quad (67)$$

and

$$A_{ijkl} = \mathbf{Q}_{jnpl}^{SE} F_{in} F_{kp} + \mathbf{S}_{jl} \delta_{ki}, \quad (68)$$

where \mathbf{Q}^{SE} is the tangent stiffness tensor defined upon the second Piola-Kirchhoff stress \mathbf{S} and Green strain \mathbf{E} . The first elasticity tensor has major symmetry, $A_{ijkl} = A_{klji}$, but does not have minor symmetries. As a result, Voigt notation cannot be applied in this case, instead, we represent the deformation gradient and first Piola-Kirchhoff stress in 2D as

$$\mathbf{F} = \{F_{11}, F_{22}, F_{12}, F_{21}\}^T = \{F_1, F_2, F_3, F_4\}^T, \quad (69)$$

$$\mathbf{P} = \{P_{11}, P_{22}, P_{12}, P_{21}\}^T = \{P_1, P_2, P_3, P_4\}^T.$$

Furthermore, the stiffness matrix \mathbf{A} can be written as

$$\dot{\mathbf{P}} = \mathbf{A} \dot{\mathbf{F}}, \quad \mathbf{A} = \begin{Bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ & A_{22} & A_{23} & A_{24} \\ & & A_{33} & A_{34} \\ \text{sym} & & & A_{44} \end{Bmatrix}. \quad (70)$$

Due to major symmetry, the matrix \mathbf{A} is symmetric. It can be seen from Eq. (70) that the stiffness matrix for finite strain now has 10 independent components.

For each loading step, the stress-strain relation at each node in the material network takes the following form,

$$\Delta\mathbf{P} = \mathbf{A} \Delta\mathbf{F} + \delta\mathbf{P}, \quad (71)$$

where $\Delta\mathbf{P}$ and $\Delta\mathbf{F}$ are the incremental stress and strain, and $\delta\mathbf{P}$ is the residual first Piola-Kirchhoff stress. In the forward homogenization process, \mathbf{A}_N and $\delta\mathbf{P}_N$ at the bottom layer $i = N$ are given by the material constitutive laws. Infos of stiffness tensor and residual stress are feed forward all the way to the output layer $i = 0$ through the homogenization and rotation operations at each building block.

In the generic two-layer structure shown in Fig. 1, the homogenized stiffness tensor $\bar{\mathbf{A}}^r$ before the rotation can be written as a function of the stiffness tensors of its two constituents (\mathbf{A}^1 , \mathbf{A}^2) and their volume fractions (f_1 and $f_2 = 1 - f_1$),

$$\bar{\mathbf{A}}^r = \bar{\mathbf{A}}^r(\mathbf{A}^1, \mathbf{A}^2, f_1). \quad (72)$$

The homogenization function of $\delta\mathbf{P}$ takes the following form,

$$\delta\bar{\mathbf{P}}^r = \delta\bar{\mathbf{P}}^r(\mathbf{A}^1, \mathbf{A}^2, \delta\mathbf{P}^1, \delta\mathbf{P}^2, f_1). \quad (73)$$

Analytical forms of Eq. (72) and (73) can be derived based on the equilibrium condition,

$$P_2^1 = P_2^2, \quad P_3^1 = P_3^2, \quad (74)$$

and the kinematic constraints,

$$F_1^1 = F_1^2, \quad F_4^1 = F_4^2. \quad (75)$$

Explicit forms of $\bar{\mathbf{A}}^r$ and $\delta\bar{\mathbf{P}}^r$ can be found in Eq. (A.6) and (A.7) in Appendix A with derivations.

Remark 4. In finite strain problems, we use the \mathbf{P} and \mathbf{F} as the stress and strain measures to keep the rotational information, instead of the symmetric ones such as second Piola-Kirchhoff stress and Green strain. To satisfy the equilibrium and kinematic conditions at the interface of the 2D building block, we need to constrain at least 4 DOFs as shown in Eq. (74) and (75).

Meanwhile, the finite-strain rotation matrix \mathbf{R}^f for a given angle θ becomes

$$\mathbf{R}^f(\theta) = \begin{Bmatrix} \cos^2 \theta & \sin^2 \theta & \sin \theta \cos \theta & \sin \theta \cos \theta \\ \sin^2 \theta & \cos^2 \theta & -\sin \theta \cos \theta & -\sin \theta \cos \theta \\ -\sin \theta \cos \theta & \sin \theta \cos \theta & \cos^2 \theta & -\sin^2 \theta \\ -\sin \theta \cos \theta & \sin \theta \cos \theta & -\sin^2 \theta & \cos^2 \theta \end{Bmatrix}. \quad (76)$$

In the rotation operation, the new stiffness matrix and residual stress can be computed by

$$\bar{\mathbf{A}} = \mathbf{R}^f(-\theta)\bar{\mathbf{A}}^r\mathbf{R}^f(\theta) \quad \text{and} \quad \delta\bar{\mathbf{P}} = \mathbf{R}^f(-\theta)\delta\bar{\mathbf{P}}^r. \quad (77)$$

After solving the macroscopic boundary value problem at the output layer, data of incremental deformation gradient $\Delta\mathbf{F}$ and first Piola-Kirchhoff stress $\Delta\mathbf{P}$ is feed backward from the output layer to the bottom layer. Similar to the procedure for nonlinear plasticity, Newton's method is used for finding the converged solution at each loading step.

5. Evaluations

5.1. Network training for various RVE morphologies

Material networks are trained for four different types of RVE morphologies under 2D plane strain conditions: a) uniform material with a single phase; b) matrix-inclusion material with circular inclusions embedded in the matrix; c) amorphous material from solving the Cahn-Hilliard equation [35] for spinodal decomposition; d) anisotropic material with phases penetrated in one direction.

The geometries and the corresponding finite element meshes of these four RVEs are provided in Fig. 7. Volume fraction of phase 1 material $v f_1$ in these four RVEs are 1.00, 0.29, 0.50 and 0.64. Number of elements in their FE meshes N_{ele} are 400, 3186, 9868 and 4355. Direct numerical simulations (DNS) of RVEs are performed using the RVE package based on implicit finite element method in LS-DYNA® with periodic boundary conditions.

For each RVE, 200 training samples and 100 validation samples are generated, following the procedure introduced in Section 3.1. In the SGD training process, the mini-batch size is chosen to be 20, so that there are 10 learning steps in each epoch. All the material networks are trained for 10000 epochs. The fitting

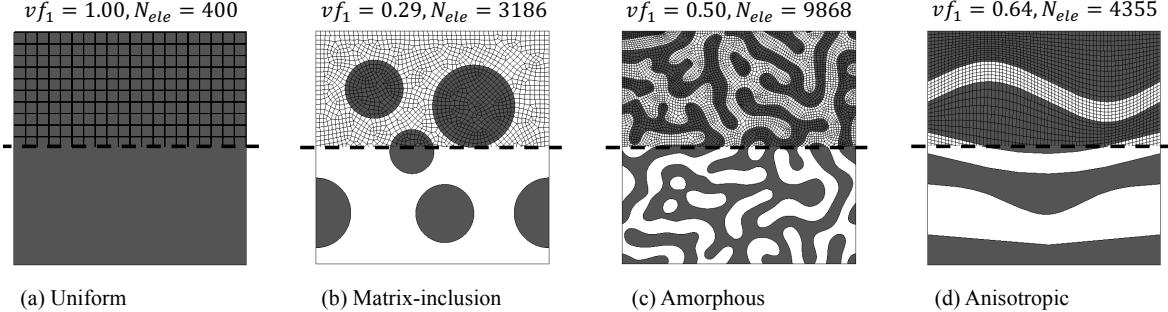


Figure 7: Geometry and finite element meshes (shown in the top half) of various two-phase RVEs evaluated in the work. Phase 1 is denoted by darker faces. The volume fraction of phase 1 material vf_1 and the number of elements in the FE mesh N_{ele} are shown at the top of each RVE plot.

parameters z_N^j and θ_i^k are initialized randomly according to Eq. (47). Note that the choice of initial fitting parameters may influence the topological structures of the trained network, and this initialization effect will be investigated in our future work.

Python libraries have been created for both network training and online extrapolation, and all the numerical examples including DNS are tested on one Intel® Xeon® CPU E5-2640 v4 2.40GHz processor. The computational time of each epoch in the SGD algorithm is approximately proportional to the number of active nodes in the bottom layer N_a . In our current study, the typical training times for $N = 3, 5$ and 7 on one processor are $0.4, 2.1$ and 9.1 hours, respectively. Several approaches can be potentially applied to further increase the training speed, such as code vectorization, hyper-parameter optimization, parallel programming and GPU computing.

5.1.1. Training history of material network

Performance of the training algorithm is evaluated by tracking the training and validation errors at each epoch. The (relative) error of a sample s at the t -th epoch is defined as

$$e_s^{(t)} = \frac{\|\bar{\mathbf{D}}_s^{dns} - \bar{\mathbf{D}}_s^{mn(t)}\|}{\|\bar{\mathbf{D}}_s^{dns}\|}, \quad (78)$$

where $\bar{\mathbf{D}}_s^{dns}$ and $\bar{\mathbf{D}}_s^{mn(t)}$ are the compliance matrices predicted by DNS and the material network, respectively. The matrix norm $\|\dots\|$ has been defined in Eq. (31). The average error of a dataset with S samples at the t -th epoch is

$$\bar{e}^{(t)} = \frac{1}{S} \sum_s e_s^{(t)}. \quad (79)$$

Table 1: Average errors of training datasets for various RVEs after 10000 epochs.

	uniform	matrix-inclusion	amorphous	anisotropic
$N = 3$	0.02%	3.02%	7.86%	8.33%
$N = 5$	0.02%	0.55%	1.08%	0.37%
$N = 7$	0.02%	0.51%	0.70%	0.48%

Histories of average training and validation errors for material networks with different depths are provided in the Fig. 8. The average errors of the training dataset for material networks with depth $N = 3, 5$ and 7

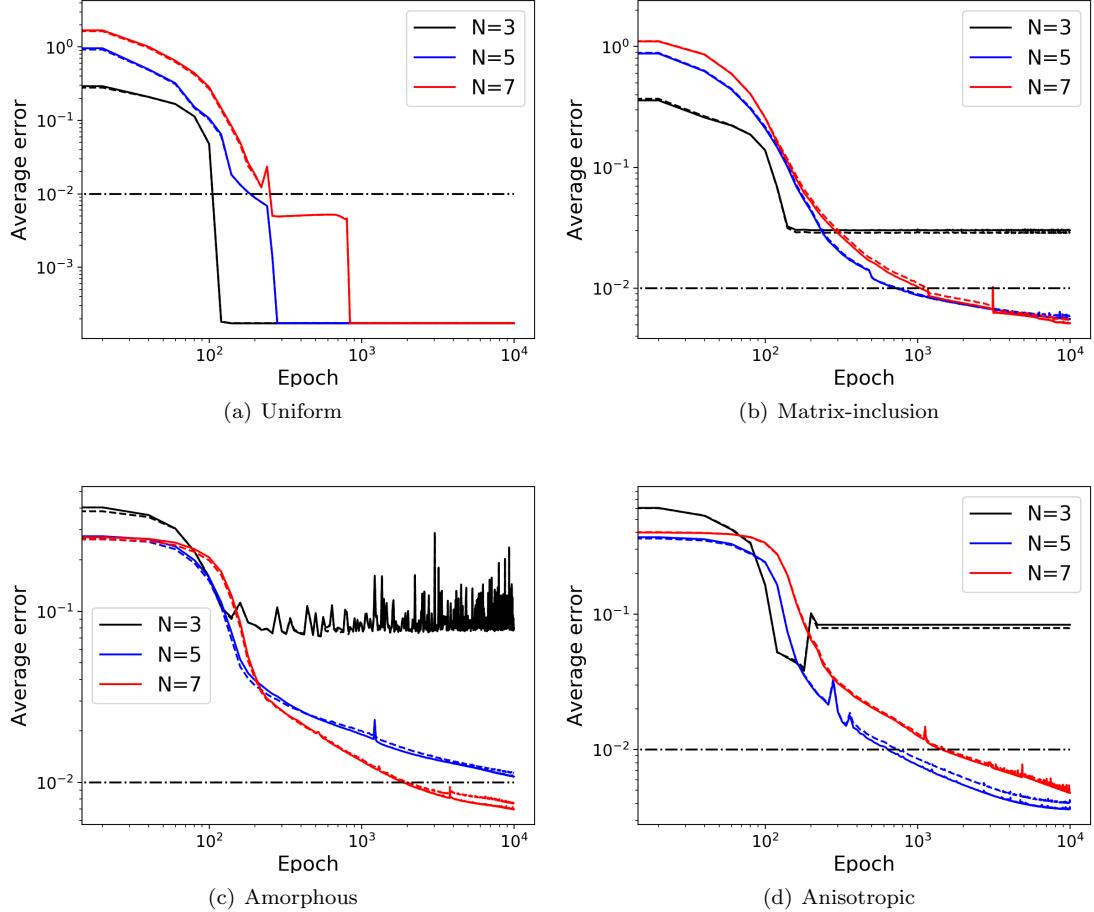


Figure 8: Histories of average training errors (solid lines) and validation errors (dashed lines) for various RVEs. The dash-dot lines denote the average error equal to 1%.

are listed in Table 1. It is observed that the average errors of all the RVEs were below 0.7% after 10000 epochs of training when $N = 7$. Actually, a good accuracy of 2% was reached after 1000 epochs for $N = 5$ and 7 in all the cases. Other than the uniform RVE, a network with $N = 3$ is in general not deep enough to capture the RVE responses, herein, its average training error stopped decreasing at a relatively large value. Additionally, the difference between the training and validation errors was almost negligible, indicating that the hyper-parameters in the model are well tuned and no over-fitting has appeared.

Remark 5. *The training results depend on the initial parameters and the SGD algorithm converges to a band of critical points which are local minimal close to the "global optimum", as studied by Choromanska et al. [36] for multilayer neural networks. In addition, multiple realizations of the network with different sets of initial parameters can be trained separately, and the one with the minimum training or testing errors will be selected as the database for the RVE.*

Fig. 9 provides the distributions of the training error after 10000 epochs for network depth $N = 7$. The figure shows that the maximum relative errors in the training datasets are below 2.5% for all the RVEs. It can be concluded that a material network with sufficient depth ($N \geq 5$) can represent the DNS model accurately, while with much less DOFs. Meanwhile, the proposed material network based on a simple two-layer building block can be effectively trained by SGD with backpropagation algorithms and network

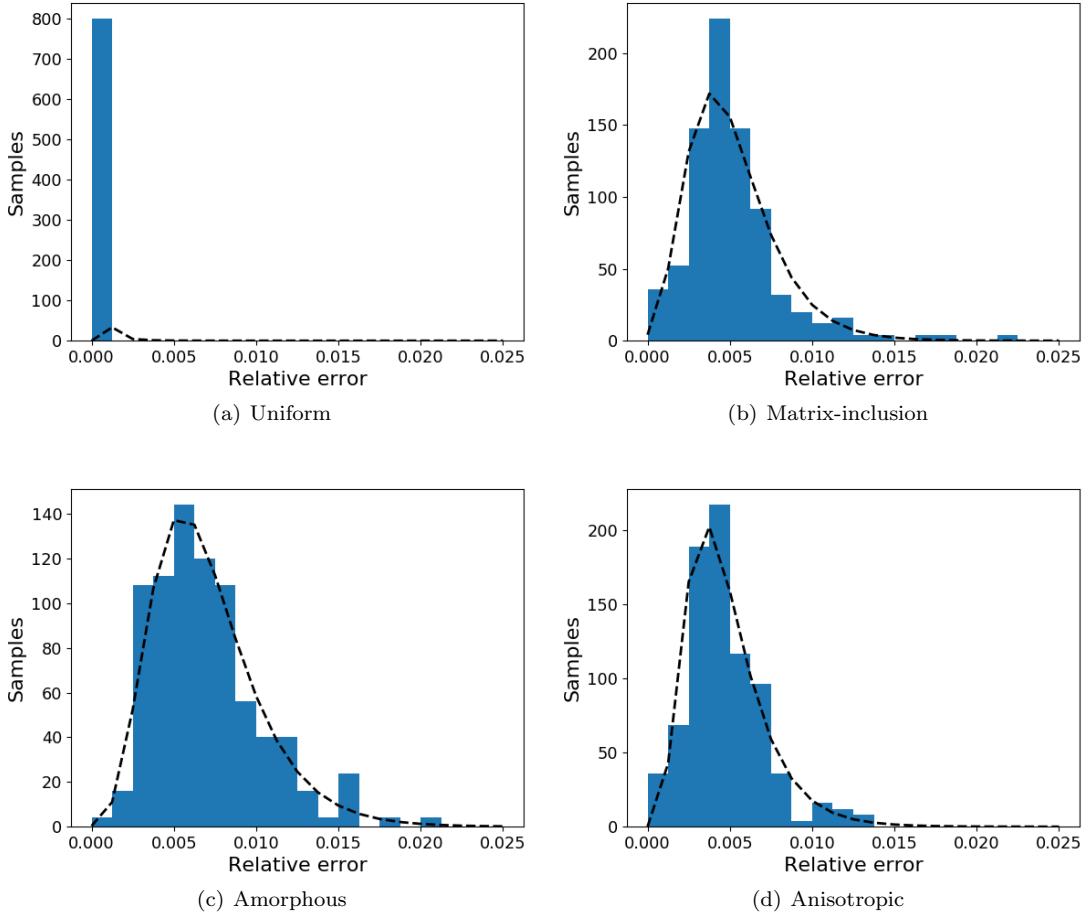


Figure 9: Distributions of training errors for material networks with depth $N = 7$ after 10000 epochs. The histograms are normalized.

compression methods.

Table 2: Number of active nodes in the bottom layer (N_a) after 10000 epochs of training.

	uniform	matrix-inclusion	amorphous	anisotropic	<i>initial N_a</i>
$N = 3$	1	4	6	2	8
$N = 5$	1	16	28	16	32
$N = 7$	1	44	86	62	128

Numbers of active nodes in the bottom layer (N_a) of the trained network after 10000 epochs are listed in Table 2. With the ReLU activation function and merging operation for network compression, N_a gradually decreases during the training. The final N_a of a trained network with a given depth N depends on the RVE morphology. In our case, the amorphous RVE needs the most nodes (or DOFs) to minimize the training error. For uniform RVE, material networks with different depths all degenerated to the same structure with only one active node $N_a = 1$. In fact, this structure is equivalent to the DNS model. Therefore, their relative errors are only about 0.02%, which should come from the numerical error in the DNS calculation.

5.1.2. Topologies of trained material network

During the training, the topology of the network was kept varying to minimize the cost function. Here we use the so-called treemap to track and display the hierarchical structure of the material network using a set of nested rectangles. In a treemap, each rectangle represents an active node in the bottom layer of the material network, and its area is determined by the weighting function w_N at the corresponding node.

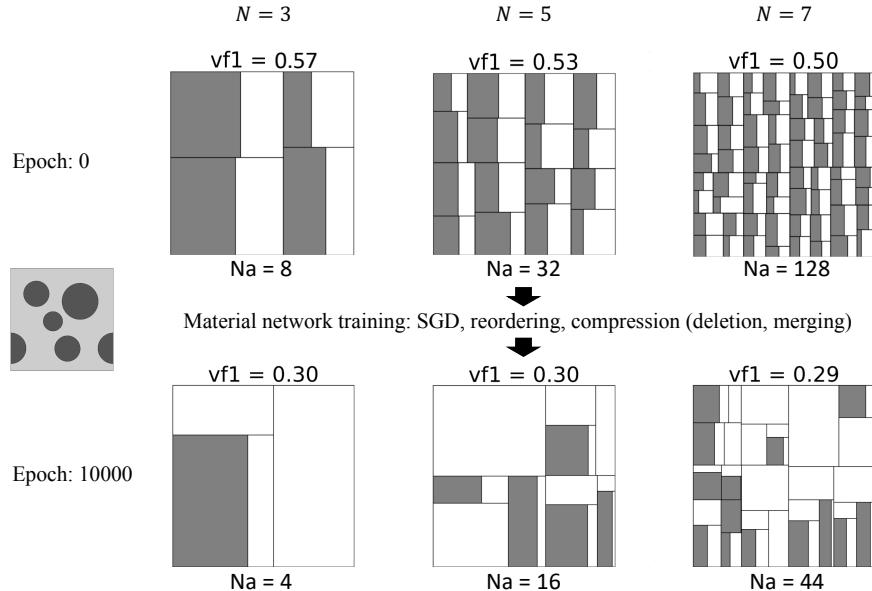


Figure 10: Treemaps of material network for the matrix-inclusion RVE at the beginning and the end of the training. Three network depths $N = 3, 5, 7$ are considered. Volume fraction of phase 1 is provided at the top of each plot, while the number of active nodes at the bottom layer are shown at the bottom of each plot.

Fig. 10 presents the treemaps of material network for the matrix-inclusion RVE at the beginning and the end of the training, where the dark rectangles represent the phase 1 material. Since the weights of nodes at the bottom layer are assigned randomly for the initial network, the volume fraction of phase 1 is around 0.5 at the beginning of training. Note that the treemap only reflects the weight of each node, so that the rotation information at each node is not included in the plots.

Treemaps of trained material networks for all the RVEs after 10000 epochs are shown in Fig. 11. Three network depths $N = 3, 5$ and 7 are considered. A material network with $N \geq 5$ is capable of accurately learning the phase volume fraction vf_1 for all the RVEs. This indicates that the training dataset, which only contains the homogenized mechanical properties under different combinations of phases, can be used for extracting the geometric RVE information. It also suggests that the material network based on the two-layer building block can well represent the topological structure of the RVE.

An intriguing feature of the material network is that its topological structure is intrinsically parameterized. The rate of change or gradient of the overall material properties with respect to the geometric descriptors (e.g. volume fraction) and mechanical properties (e.g. modulus) of any node in the network can always be derived analytically. Thus, various gradient-based optimization methods could be easily employed to design the material both geometrically and mechanically. Potential applications to the area of material design will be investigated in our future work.

5.2. Online extrapolation of trained network

Based on the concept of material network and its machine learning algorithms, a reduced topological representation of the DNS model can be effectively mined from high-fidelity RVE training data. In this section, we will further investigate the material network's capability of online extrapolation to three different

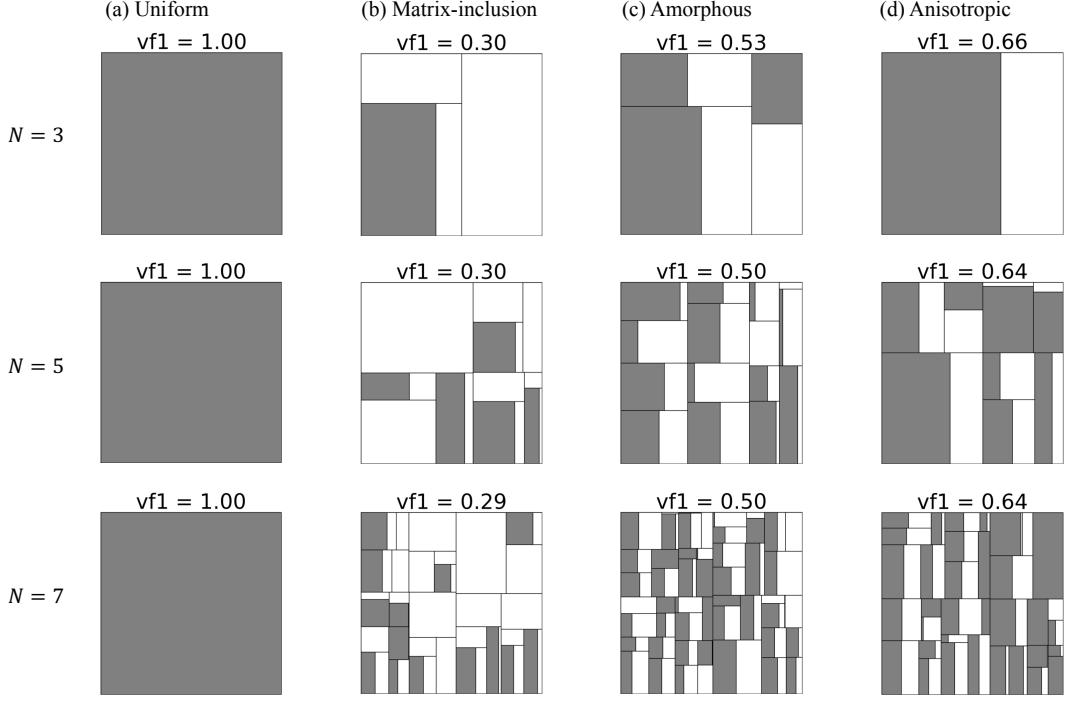


Figure 11: Treemaps of trained material network at the 10000-th epoch. Three network depths $N = 3, 5$, and 7 are shown for each RVE. Volume fraction of phase 1 is provided at the top of each plot.

cases: 1) linear elasticity with high contrast of phase properties; 2) small-strain nonlinear plasticity; 3) finite-strain hyperelasticity under large deformations.

5.2.1. Linear elasticity with high contrast of phase properties

The proposed material network is first extrapolated to linear elasticity problems with high contrast of phase properties. Comparing to the training dataset, wider ranges of material constants were used here. Both phases are assumed to be isotropic linear elastic.

In this new testing dataset, ranges of elastic constants of phases 1 and 2 are

$$E^{p1} = 1, \quad \nu^{p1} \in U[0.005, 0.495] \quad (80)$$

and

$$\log_{10}(E^{p2}) \in U[-3, 3], \quad \nu^{p2} \in U[0.005, 0.495], \quad (81)$$

where U denotes the uniform distribution. The largest ratio of elastic moduli between the two phases is 1000. For each RVE, 100 testing samples are generated using Latin hypercube. The relative error of linear elastic compliance matrices is defined in Eq. (78).

Table 3: Average relative errors of the testing dataset on trained material networks.

	uniform	matrix-inclusion	amorphous	anisotropic
$N = 3$	0.02%	2.32%	12.60%	9.93%
$N = 5$	0.02%	0.92%	2.34%	0.61%
$N = 7$	0.02%	0.88%	1.70%	0.67%

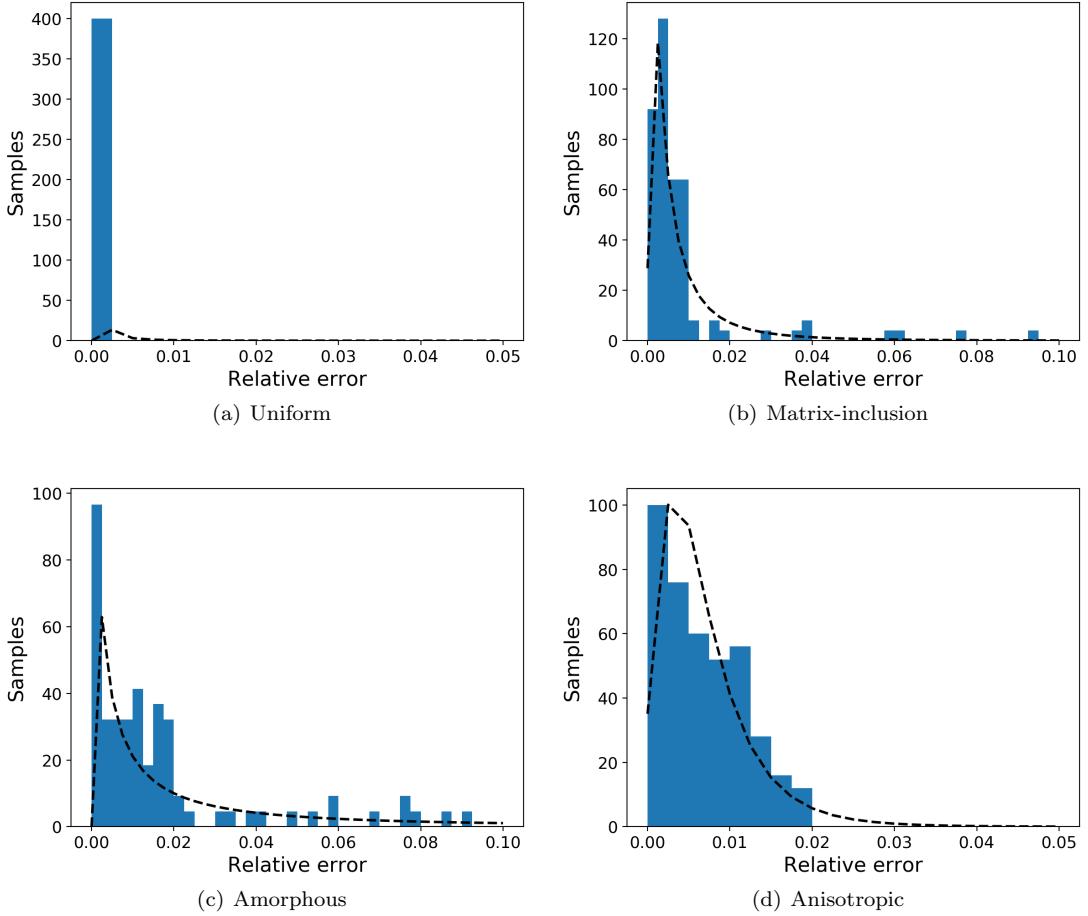


Figure 12: Distributions of testing errors for trained networks with depth $N = 7$. The histograms are normalized.

Average relative errors of the material networks with $N = 3$, $N = 5$ and $N = 7$ are listed in Table 3 for all the RVEs. Distributions of the relative testing errors of material networks with $N = 7$ are shown in Fig. 12. A log-normal density function is fitted to the histogram for each case.

For the uniform RVE, the relative errors are negligible since all the networks have degenerated to the same one, that is equivalent to the DNS model. As expected, a material network with depth $N = 3$ is not sufficient to maintain accurate predictions. For $N = 7$, the average relative errors of the matrix-inclusion and anisotropic RVEs are below 1%, while the one of the amorphous RVE reaches 1.7%. As shown in Fig. 12, the maximum relative error among the 100 testing samples of the anisotropic RVE is still below 2%. However, maximum errors of the matrix-inclusion and amorphous RVEs go up to 9.5%, while the errors of their training and validation datasets are below 2.5% as shown in Fig. 9. To reduce their prediction errors for models with high contrast of phase properties, it will be helpful to increase the sampling ranges of the training datasets.

5.2.2. Small-strain nonlinear plasticity

Nonlinear elasto-plastic RVEs under small-strain assumption are studied in this section. The phase 2 material is considered to be elasto-plastic with an isotropic von Mises yield surface and piece-wise linear hardening law. Its elastic constants are

$$E^{p2} = 100 \text{ GPa}, \nu^{p2} = 0.3. \quad (82)$$

The yield stress σ_Y is determined by the hardening law as a function of the effective plastic strain $\bar{\varepsilon}^{pl}$,

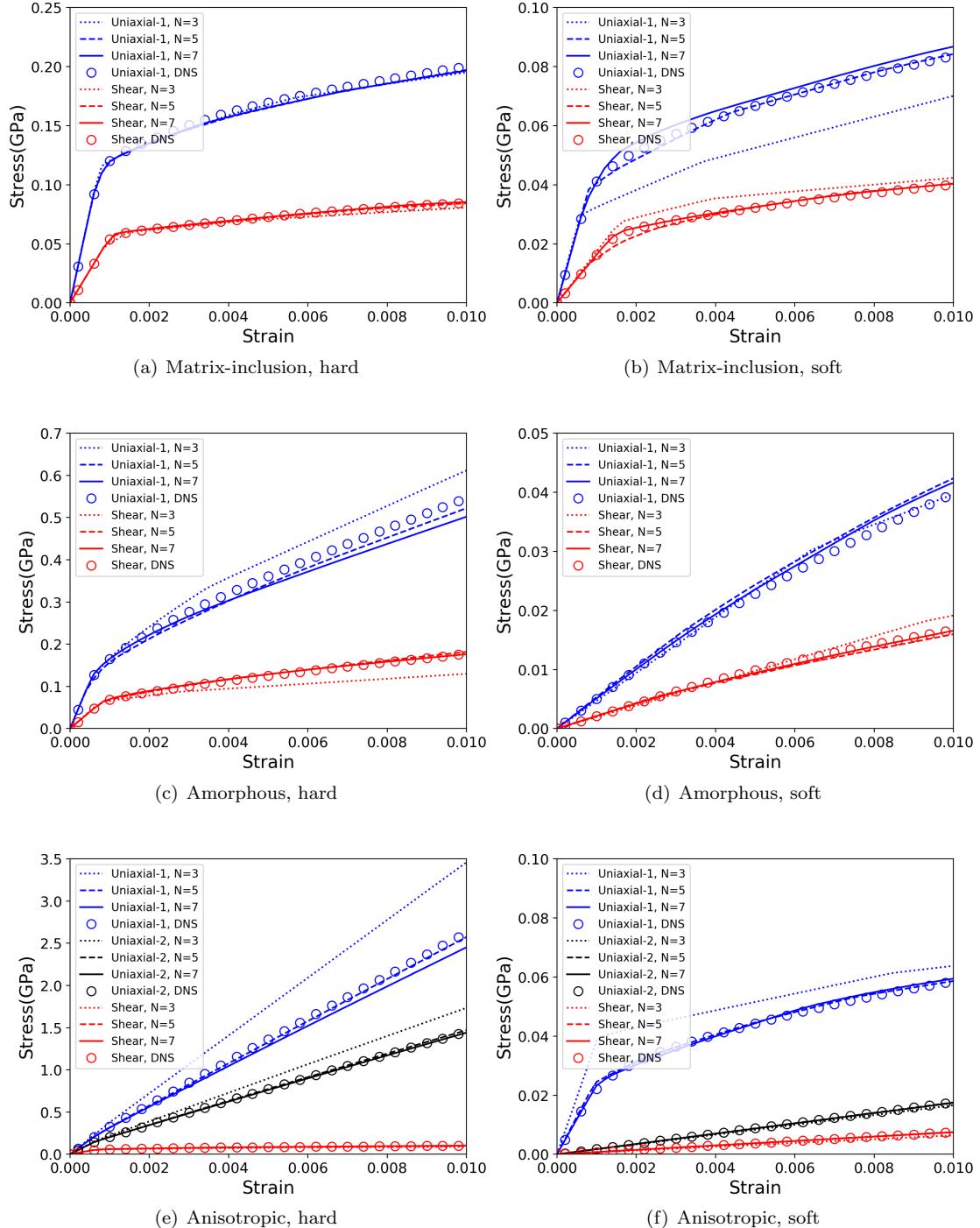


Figure 13: Comparisons between material network and DNS for nonlinear small-strain plasticity for various RVEs under uniaxial tension and shear loading conditions. For the anisotropic RVE (e, f), uniaxial tension loadings are applied in two orthogonal directions. Both the hard (a, c, e) and soft (b, d, f) cases are considered. The network depths are $N = 3$ (dotted), 5 (dashed) and 7 (solid). DNS results are marked by the circles (○).

which is a monotonically increasing internal state variable of the plastic material during the deformation. The yielding stress σ_Y^0 is equal to 0.1 GPa. The hardening law is considered to be piecewise linear and isotropic,

$$\sigma_{p2}^Y(\bar{\varepsilon}_{p2}^{pl}) = \begin{cases} 0.1 + 5\bar{\varepsilon}_{p2}^{pl} & \bar{\varepsilon}_{p2}^{pl} \in [0, 0.008] \\ 0.124 + 2\bar{\varepsilon}_{p2}^{pl} & \bar{\varepsilon}_{p2}^{pl} \in [0.008, \infty) \end{cases} \text{ GPa.} \quad (83)$$

Phase 1 remains as a linear elastic material, but either hard or soft phases are considered by making the Young's modulus of phase 1 either harder or softer than phase 2. For these two cases, properties of phase 1 material are

$$E^{p1} = 500 \text{ GPa}, \nu^{p1} = 0.19 \text{ (hard)} \quad \text{and} \quad E^{p1} = 1 \text{ GPa}, \nu^{p1} = 0.19 \text{ (soft)}. \quad (84)$$

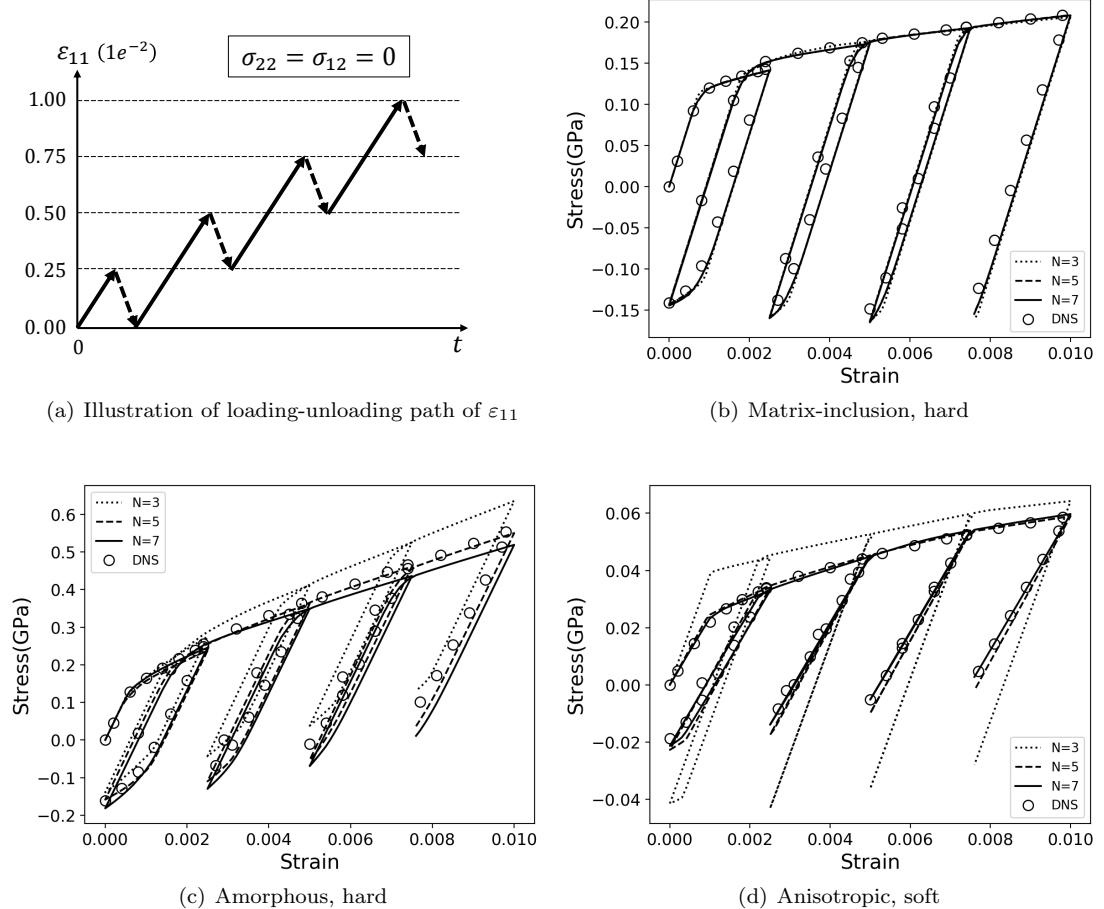


Figure 14: Comparisons between material network and DNS under uniaxial loading-unloading path shown in (a). RVEs with apparent hardening regime are considered: matrix-inclusion and amorphous RVEs with hard phase 1, anisotropic RVE with soft phase 1. The network depths are $N = 3$ (dotted), 5 (dashed) and 7 (solid). DNS results are marked by the circles (\circ).

Fig. 13 includes the stress-strain curves considering two different loading cases, uniaxial tension and pure shear, predicted by DNS and the material network. In the case of anisotropic RVE in Fig. 13 (e) and (f), the uniaxial tension tests are performed in two orthogonal directions. By looking at the plots, we can conclude that the proposed method is capable of capturing the nonlinear plastic behavior for both hard

and soft cases under different loading conditions, meanwhile, with significantly fewer degrees of freedom. Note that it can also accurately capture anisotropic RVE behaviors, which can be very challenging for most micromechanics-based methods [11, 21]. Furthermore, the network should be deep enough to capture the nonlinear RVE behavior accurately, and in our case, $N = 5$ is a good depth to start from.

Fig. 14 (a) shows a uniaxial loading-unloading path which contains 4 loading and 4 unloading steps at different stress/strain levels. Three RVEs with apparent hardening regime are considered: matrix-inclusion RVE with hard phase 1, amorphous RVE with hard phase 1 and anisotropic RVE with soft phase 1. The stress-strain curves are provided in Fig. 14 (b,c,d). For all the cases, the loading-unloading behavior can be well captured by the material networks for $N \geq 5$.

The material networks were also validated against a complex loading path for different RVEs. Macro-strain constraints are applied on ε_{11} and ε_{12} , while $\sigma_{22} = 0$. As shown in Fig. 15 (a), there are three steps within the loading path and the RVE eventually returns to the initial state ($\varepsilon_{11} = 0$; $\varepsilon_{12} = 0$). Due to plasticity, σ_{22} does not necessarily vanish at the end of loading. The stress-strain curves σ_{11} -vs- ε_{11} and σ_{12} -vs- ε_{12} given by the material networks and the DNS results are shown in Fig. 15 (b,c,d). Once again good agreement is observed for all three RVEs.

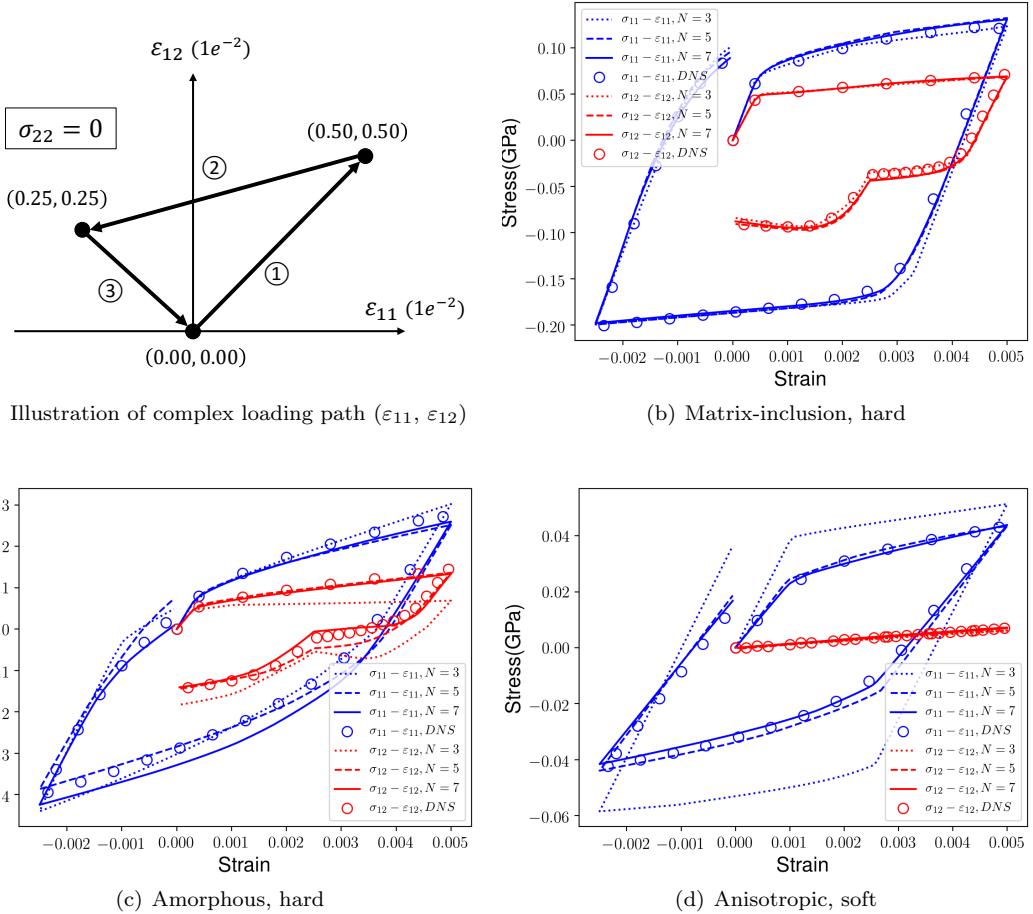


Figure 15: Comparisons between material network and DNS under a complex loading path shown in (a). Constraints are applied on ε_{11} and ε_{12} , and $\sigma_{22} = 0$. RVEs with apparent hardening regime are considered: matrix-inclusion and amorphous RVEs with hard phase 1, anisotropic RVE with soft phase 1. The network depths are $N = 3$ (dotted), 5 (dashed) and 7 (solid). DNS results are marked by the circles (\circ).

In theory, a deeper network can always be more accurate, since its solution space includes the one of a shallower network. However, with more fitting parameters, the surface of the cost function of the deeper network has more local minimal, so that the training process will require more epochs to find the optimum representation of the RVE. For example, given 10000 epochs of training, a network with $N = 7$ does not always provide a better prediction than the one with $N = 5$ as we can see from Fig. 13, 14 and 15. Other than the method discussed in *Remark 5* (multiple realizations), it is also possible to use different types of training datasets to improve the convergence of training algorithms. In practice, the optimum choice of network depth depends on one's devoted training time and desired accuracy.

The computational times of material networks for the matrix-inclusion and amorphous RVEs are presented in Fig. 16 for different numbers of active nodes in the bottom layer. Uniaxial tension loading was considered, and there were 25 loading steps in each simulation. Typical FE simulations of the amorphous RVEs took about 79 s, while the trained material network took 0.12 s for $N_a = 6$ ($N = 3$), 0.49 s for $N_a = 28$ ($N = 5$), and 1.58 s for $N_a = 86$ ($N = 7$). With the same N_a , the network for the matrix-inclusion has more active nodes belonging to the nonlinear plastic material, therefore, it costs more time in evaluating the local constitutive laws than the one of the amorphous RVE. For the 2D amorphous RVE, the trained network with $N = 5$ is 150 times faster than the corresponding DNS, and we expect the improvement of efficiency from the proposed material network will be more significant for 3D RVEs.

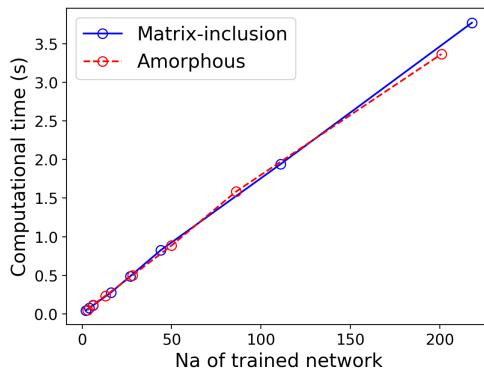
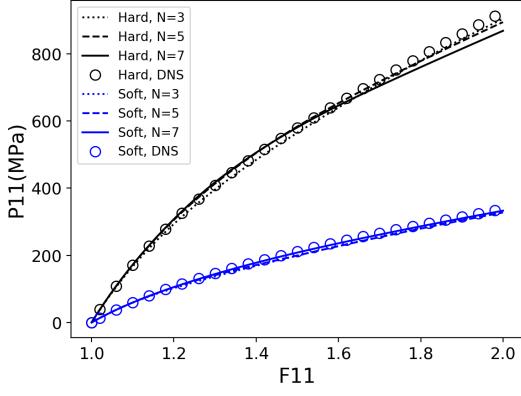


Figure 16: Computational time vs. number of active nodes in the bottom layer N_a for nonlinear small-strain plasticity. Trained material networks for matrix-inclusion and amorphous RVEs are considered. 25 uniaxial loading steps are simulated with e_{11} up to 0.01.

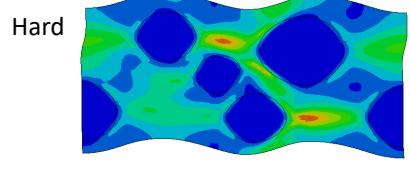
It can be seen from the figure that the computational time is proportional to N_a in the network, consistent of our statement in Section 4.1. This is an important feature of the proposed material network based on hierarchical homogenization structure. As in most RVE homogenization techniques, the computational costs are proportional to $(N_{dof})^{2.x}$ (FEM), $N_{dof} \log(N_{dof})$ (FFT-based methods [7]), or $(N_{dof})^3$ (SCA [21], TFA [37]). Moreover, the material network is solved directly without the need for an iterative solver (e.g. the one used in the FFT-based method [7]), herein, its robustness and reliability are well preserved. All these features make the proposed material network a promising tool for concurrent multiscale simulations and material design.

5.2.3. Finite-strain hyperelasticity under large deformations

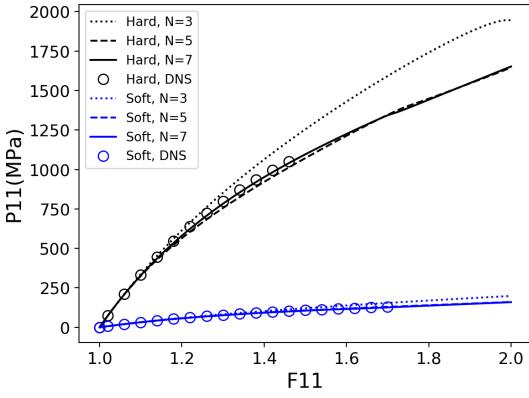
In this section, the trained material networks are extrapolated to predict the responses of finite-strain hyperelastic RVEs under large deformation. To the authors' knowledge, finite-strain problems with large deformations are challenging for most existing model reduction methods. Analytical and semi-analytical micromechanics methods are no longer valid due to the failure of analytical solutions. Methods based on reduced basis also lose their accuracy when large geometric non-linearity is involved. Even DNS models may encounter some numerical difficulties due to severe material distortions. Nonetheless, we will demonstrate



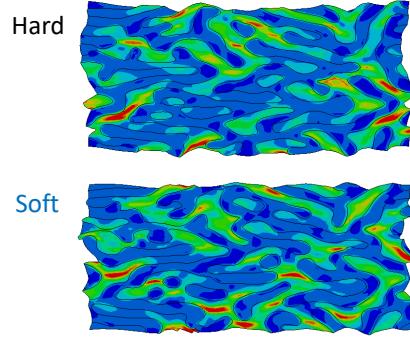
(a) Matrix-inclusion: P_{11} – F_{11} plots



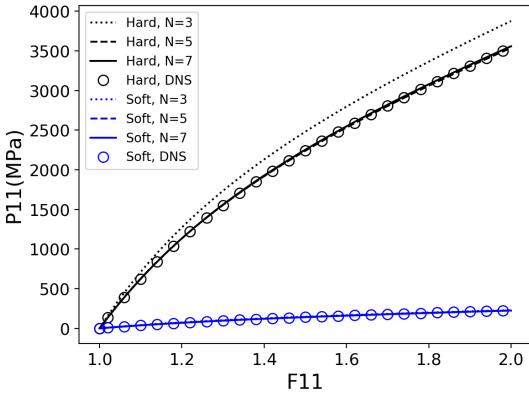
(b) Matrix-inclusion: deformed RVEs at $F_{11} = 1.5$



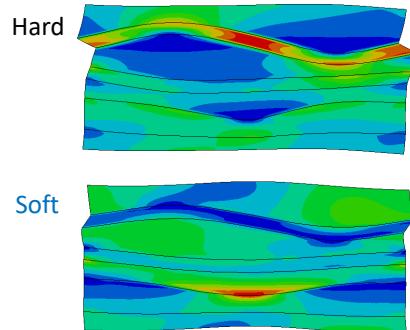
(c) Armophous: P_{11} – F_{11} plots



(d) Armophous: deformed RVEs at $F_{11} = 1.5$



(e) Anisotropic: P_{11} – F_{11} plots



(f) Anisotropic: deformed RVEs at $F_{11} = 1.5$

Figure 17: Comparisons between material network and DNS for finite-strain hyperelasticity under uniaxial tension loading for various RVEs. Both hard and soft cases are considered. The P_{11} vs. F_{11} plots are shown on the left (a, c, e), and Green strain fields in the deformed RVEs at $F_{11} = 1.5$ from DNS are shown on the right (b, d, f). The network depths are $N = 3$ (dotted), 5 (dashed) and 7 (solid). DNS results are marked by the circles (o).

that the proposed material network with reduced DOFs is able to predict both global and local RVE responses accurately under large deformations.

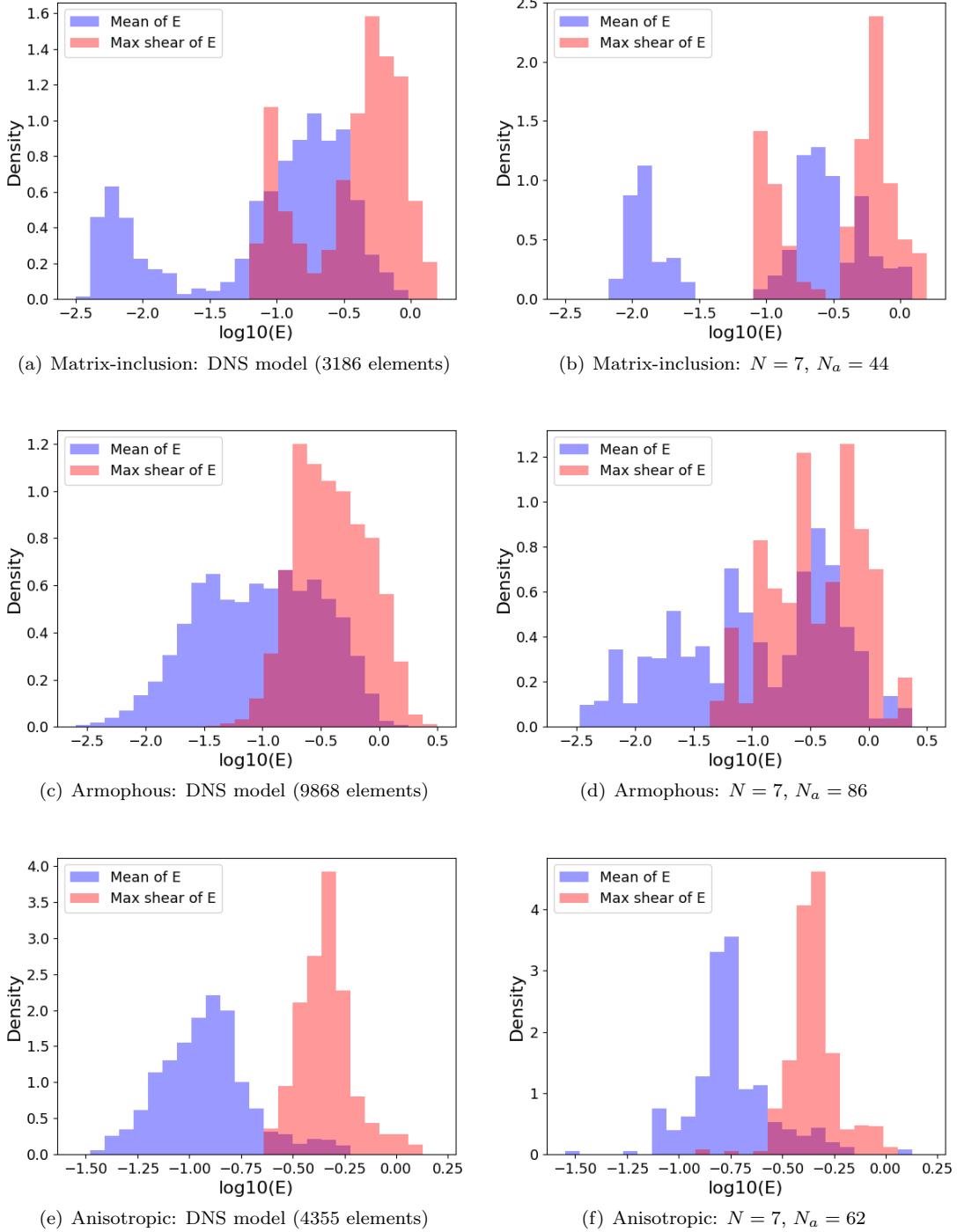


Figure 18: Distributions of the mean Green strain and the max shear Green strain inside the RVE (hard) from DNS (left) and material networks with $N = 7$ under uniaxial tension loading at $F_{11} = 1.5$. The histograms are normalized.

Both phase 1 and phase 2 are considered to be Mooney-Rivlin hyperelastic materials. The strain energy density function of the Mooney-Rivlin material is defined as

$$W = A(I - 3) + B(II - 3) + C(III^{-2} - 1) + D(III - 1)^2. \quad (85)$$

The independent material constants are A , B and the Poisson ratio ν , and $2(A+B)$ gives the shear modulus of linear elasticity. The other parameters in strain energy density function can be calculated by

$$C = 0.5A + B, \quad D = \frac{A(5\nu - 2) + B(11\nu - 5)}{2(1 - 2\nu)}, \quad (86)$$

and I, II, III are the invariants of right Cauchy-Green tensor $C = F^T F$.

Both phases are assumed to be nearly incompressible ($\nu \approx 0.5$). The material constants of phase 2 are

$$A^{p2} = 100 \text{ MPa}, B^{p2} = 50 \text{ MPa}, \nu^{p2} = 0.49. \quad (87)$$

Phase 1 is considered to be either harder or softer than phase 2. For these two cases, the properties of phase 1 are

$$A^{p1} = 1000 \text{ MPa}, B^{p1} = 500 \text{ MPa}, \nu^{p1} = 0.49 \text{ (hard)} \quad (88)$$

and

$$A^{p1} = 10 \text{ MPa}, B^{p1} = 5 \text{ MPa}, \nu^{p1} = 0.49 \text{ (soft).}$$

Fig. 17 provides comparisons of the $P_{11} - F_{11}$ curves obtained from DNS and from the material network under uniaxial tension loading, as well as plots of the deformed RVE at $F_{11} = 1.5$. By default, each RVE is pulled to $F_{11} = 2$, however, DNS for the amorphous RVE were terminated early due to severe distortion of elements during the analysis. Again, good agreements between the results from material networks and DNS are observed for all the cases.

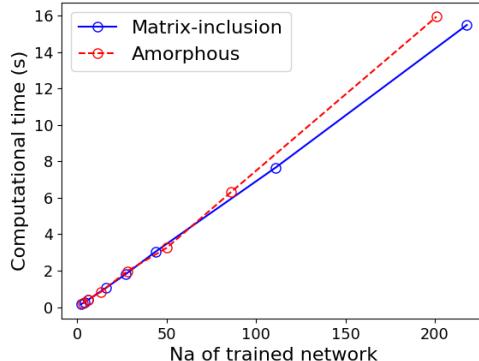


Figure 19: Computational time vs. number of active nodes in the bottom layer N_a for nonlinear finite-strain hyperelasticity. Trained material networks for the matrix-inclusion and amorphous RVEs are considered. 50 uniaxial loading steps are simulated with F_{11} up to 2.0.

The distributions of local Green strains inside the RVEs predicted by DNS and material networks with $N = 7$ under uniaxial tension loading at $F_{11} = 1.5$ are shown in Fig. 18. RVEs with hard phase 1 material were considered. For each case, the statistical characteristics of the local green strain in the material network agree well with the DNS results. The mean and max-shear of a Green strain $\mathbf{E} = \{E_{11}, E_{22}, E_{12}\}$ are defined as,

$$E_{mean} = \frac{1}{2}(E_{11} + E_{22}) \quad (89)$$

and

$$E_{max-shear} = \sqrt{\frac{1}{4}(E_{11} - E_{22})^2 + (E_{12})^2}. \quad (90)$$

The computational time of material networks for the matrix-inclusion and amorphous RVEs are presented in Fig. 19 for different N_a . There were 50 loading steps in each simulation. A typical FE simulation of the matrix-inclusion RVE (3186 elements) took about 30 s, while the material network of took 1.05 s for $N_a = 16(N = 5)$, and 3.06 s for $N_a = 44(N = 7)$. Again, the computational time is still approximately proportional to N_a in the network. However, a deeper network with more active nodes in its bottom layers usually has larger local deformations, herein, requires more newton's iterations to converge at each loading step. The material network is a simplified topological structure that reduces the number of DOFs and computational complexity of the DNS model, and the speed-up ratio will be more significant for DNS with a larger mesh and 3D problems.

6. Conclusion

A new data-driven multiscale material modeling method called deep material network is developed for machine learning of RVE topologies and accelerated online predictions. Based on a simple two-layer building block with analytical homogenization solutions, the material network constructs a hierarchical structure that is capable of capturing complex RVE behavior. Optimization and training of the deep material network are enabled by the synergistic choices of various machine learning approaches, such as SGD with backpropagation algorithm, nodal deletion and subtree merging for model compression. Importantly, limitations encountered in previous reduced order methods (e.g. extensive offline sampling, extra model assumption and calibration, danger of extrapolation, loss of physics) have been addressed by the proposed material network, with the following key features:

1. A novel RVE model reduction approach based on mechanistic building block and hierarchical topological structure;
2. A complete machine learning procedure for training the deep material network based on offline numerical DNS data or experimental testing data;
3. Efficient and accurate online predictions for challenging RVE homogenization problems, including nonlinear history-dependent plasticity and finite-strain hyperelasticity under large deformations.

There are a vast number of research opportunities for improvement and extension of the deep material network, in terms of both theory development and applications. A few future directions are listed as below:

1. Choices of proper data-sampling strategy and advanced machine learning techniques can help to optimize the material network, while a deeper understanding of the whole training process still requires more study.
2. Extension of the present 2D building block to 3D is of fundamental importance, and the possibility of including multiphysics effect can also be explored.
3. Applications of the deep material network to industrial RVE problems, generally with more complex morphology and material laws, can be of significant interests, such as visco-hyperelastic polymer composite, carbon fiber reinforced polymer (CFRP), and polycrystalline alloys modeled by crystal plasticity.

Finally, this intelligent model reduction method provides a possibility to obtain accurate constitutive behaviors for accelerating the multi-scale concurrent computation in a large-scale heterogeneous structure. Additionally, it provides a way to avoid the numerical difficulty associated with large material distortion in conventional RVE approaches. Due to its efficiency and intrinsically parameterized structure, the proposed material network also offers a high application potential for multiscale material design.

Acknowledgement

The authors would like to warmly thank Dr. John O. Hallquist of LSTC for his support to this research. The support from the Yokohama Rubber Co., LTD under the Yosemite project is also gratefully acknowledged.

Appendix A. Analytical solutions of 2D building block in finite strain

The analytical solutions in finite strain are derived based on the equilibrium condition and kinematic constraints at the interface between the two layers, which are listed below,

$$P_2^1 = P_2^2, \quad P_3^1 = P_3^2, \quad F_1^1 = F_1^2, \quad F_4^1 = F_4^2. \quad (\text{A.1})$$

First, assume an arbitrary overall deformation gradient after homogenization as

$$\bar{\mathbf{F}}^r = \{\bar{F}_1^r, \bar{F}_2^r, \bar{F}_3^r, \bar{F}_4^r\}^T,$$

Based on Eq. (A.1) and definition of homogenization, we have

$$F_1^1 = F_1^2 = \bar{F}_1^r, \quad F_2^2 = \frac{1}{f_2} \bar{F}_2^r - \frac{f_1}{f_2} F_2^1, \quad F_3^2 = \frac{1}{f_2} \bar{F}_3^r - \frac{f_1}{f_2} F_3^1, \quad F_4^1 = F_4^2 = \bar{F}_4^r. \quad (\text{A.2})$$

with

$$f_2 = 1 - f_1.$$

Applying the constitutive laws (without residual stress) for both phases and substituting the above equation into the equilibrium condition yields

$$\begin{Bmatrix} F_2^1 \\ F_3^1 \end{Bmatrix} = \begin{Bmatrix} \hat{A}_{22} & \hat{A}_{23} \\ \hat{A}_{32} & \hat{A}_{33} \end{Bmatrix}^{-1} \begin{Bmatrix} f_2 \Delta A_{12} & A_{22}^2 & A_{23}^2 & f_2 \Delta A_{24} \\ f_2 \Delta A_{13} & A_{32}^2 & A_{33}^2 & f_2 \Delta A_{34} \end{Bmatrix} \bar{\mathbf{F}}^r = \mathbf{s}_{2 \times 4}^1 \bar{\mathbf{F}}^r, \quad (\text{A.3})$$

where

$$\hat{\mathbf{A}} = f_2 \mathbf{A}^1 + f_1 \mathbf{A}^2 \quad \text{and} \quad \Delta \mathbf{A} = \mathbf{A}^2 - \mathbf{A}^1. \quad (\text{A.4})$$

We can then combine Eq. (A.2) and (A.3) to define the concentration tensor for layer/material 1,

$$\mathbf{F}^1 = \mathbf{S}^1 \bar{\mathbf{F}}^r, \quad \text{with } S_{11}^1 = S_{44}^1 = 1, \quad \mathbf{S}_{([2,3],:)}^1 = \mathbf{s}_{2 \times 4}^1. \quad (\text{A.5})$$

The homogenized stiffness tensor $\bar{\mathbf{A}}^r$ in Eq. (72) can be derived as

$$\bar{\mathbf{A}}^r = \mathbf{A}^2 - f_1 \Delta \mathbf{A} \mathbf{S}^1. \quad (\text{A.6})$$

When residual stress is considered, we set the overall deformation gradient $\bar{\mathbf{F}}^r = \mathbf{0}$ and the homogenized residual stress $\delta \bar{\mathbf{P}}^r$ in Eq. (73) is equal to the homogenized stress in the RVE, which takes the following form,

$$\delta \bar{\mathbf{P}}^r = f_1 \delta \mathbf{P}^1 + f_2 \delta \mathbf{P}^2 - f_1 f_2 \Delta \mathbf{A}_{(:,[2,3])} \begin{Bmatrix} \hat{A}_{22} & \hat{A}_{23} \\ \hat{A}_{32} & \hat{A}_{33} \end{Bmatrix}^{-1} \begin{Bmatrix} \delta P_2^1 \\ \delta P_3^1 \end{Bmatrix}. \quad (\text{A.7})$$

Note that the same procedure can also be used to derive the analytical solutions of 3D two-layer building block in both small-strain and finite-strain formulation.

References

- [1] R. Hill, A self-consistent mechanics of composite materials, *Journal of the Mechanics and Physics of Solids* 13 (4) (1965) 213–222.
- [2] F. Feyel, J.-L. Chaboche, Fe 2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials, *Computer methods in applied mechanics and engineering* 183 (3) (2000) 309–330.
- [3] M. G. Geers, V. G. Kouznetsova, W. Brekelmans, Multi-scale computational homogenization: Trends and challenges, *Journal of computational and applied mathematics* 234 (7) (2010) 2175–2182.
- [4] T. Belytschko, S. Loehnert, J.-H. Song, Multiscale aggregating discontinuities: A method for circumventing loss of material stability, *International Journal for Numerical Methods in Engineering* 73 (6) (2008) 869–894.
- [5] C. T. Wu, M. Koishi, Three-dimensional meshfree-enriched finite element formulation for micromechanical hyperelastic modeling of particulate rubber composites, *International Journal for Numerical Methods in Engineering* 91 (11) (2012) 1137–1157.
- [6] C. T. Wu, D. Wang, Y. Guo, An immersed particle modeling technique for the three-dimensional large strain simulation of particulate-reinforced metal-matrix composites, *Applied Mathematical Modelling* 40 (4) (2016) 2500–2513.
- [7] 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 (1–2) (1998) 69 – 94.
- [8] T. De Geus, J. Vondřejc, J. Zeman, R. Peerlings, M. Geers, Finite strain fft-based non-linear solvers made simple, *Computer Methods in Applied Mechanics and Engineering* 318 (2017) 412–430.
- [9] J. D. Eshelby, The determination of the elastic field of an ellipsoidal inclusion, and related problems, *Proc. R. Soc. Lond. A* 241 (1226) (1957) 376–396.
- [10] Z. Hashin, S. Shtrikman, A variational approach to the theory of the elastic behaviour of multiphase materials, *Journal of the Mechanics and Physics of Solids* 11 (2) (1963) 127–140.
- [11] T. Mura, *Micromechanics of defects in solids*, Vol. 3, Springer Science & Business Media, 1987.
- [12] Z. Liu, J. A. Moore, S. M. Aldousari, H. S. Hedia, S. A. Asiri, W. K. Liu, A statistical descriptor based volume-integral micromechanics model of heterogeneous material with arbitrary inclusion shape, *Computational Mechanics* (2015) 1–19.
- [13] Z. Liu, J. A. Moore, W. K. Liu, An extended micromechanics method for probing interphase properties in polymer nanocomposites, *Journal of the Mechanics and Physics of Solids* 95 (2016) 663–680.
- [14] S. Bhattacharjee, K. Matouš, A nonlinear manifold-based reduced order model for multiscale analysis of heterogeneous hyperelastic materials, *Journal of Computational Physics* 313 (2016) 635–653.
- [15] J. Michel, P. Suquet, Nonuniform transformation field analysis, *International Journal of Solids and Structures* 40 (25) (2003) 6937 – 6955, special issue in Honor of George J. Dvorak.
- [16] J.-C. Michel, P. Suquet, A model-reduction approach in micromechanics of materials preserving the variational structure of constitutive relations, *Journal of the Mechanics and Physics of Solids* 90 (2016) 254–285.
- [17] I. Jolliffe, *Principal component analysis*, Wiley Online Library, 2002.
- [18] J. Yvonnet, Q.-C. He, The reduced model multiscale method (r3m) for the non-linear homogenization of hyperelastic media at finite strains, *Journal of Computational Physics* 223 (1) (2007) 341–368.
- [19] P. Kerfriden, O. Goury, T. Rabczuk, S. P.-A. Bordas, A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics, *Computer Methods in Applied Mechanics and Engineering* 256 (2013) 169–188.
- [20] J. Oliver, M. Caicedo, A. Huespe, J. Hernández, E. Roubin, Reduced order modeling strategies for computational multiscale fracture, *Computer Methods in Applied Mechanics and Engineering* 313 (2017) 560–595.
- [21] Z. Liu, M. Bessa, W. K. Liu, Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials, *Computer Methods in Applied Mechanics and Engineering* 306 (2016) 319–341.
- [22] Z. Liu, M. Fleming, W. K. Liu, Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials, *Computer Methods in Applied Mechanics and Engineering* 330 (2018) 547–577.
- [23] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, *IEEE Signal Processing Magazine* 29 (6) (2012) 82–97.
- [24] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *nature* 521 (7553) (2015) 436.
- [25] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016.
- [26] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., Mastering the game of go without human knowledge, *Nature* 550 (7676) (2017) 354.
- [27] B. Le, J. Yvonnet, Q.-C. He, Computational homogenization of nonlinear elastic materials using neural networks, *International Journal for Numerical Methods in Engineering* 104 (12) (2015) 1061–1084.
- [28] J. Yvonnet, E. Monteiro, Q.-C. He, Computational homogenization method and reduced database model for hyperelastic heterogeneous structures, *International Journal for Multiscale Computational Engineering* 11 (3) (2013) 201–225.
- [29] M. Bessa, R. Bostanabad, L. Liu, A. Hu, D. Apley, C. Brinson, W. Chen, W. Liu, A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality, *Computer Methods in Applied Mechanics and Engineering* 320 (2017) 633–667.
- [30] R. Ibañez, D. Borzacchiello, J. V. Aguado, E. Abisset-Chavanne, E. Cueto, P. Ladevèze, F. Chinesta, Data-driven nonlinear elasticity: constitutive manifold construction and problem discretization, *Computational Mechanics* 60 (5) (2017) 813–826.
- [31] J. Ghaboussi, J. Garrett Jr, X. Wu, Knowledge-based modeling of material behavior with neural networks, *Journal of engineering mechanics* 117 (1) (1991) 132–153.

- [32] J. F. Unger, C. Könke, Coupling of scales in a multiscale simulation using neural networks, *Computers & Structures* 86 (21-22) (2008) 1994–2003.
- [33] K. Wang, W. Sun, A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning, *Computer Methods in Applied Mechanics and Engineering* 334 (2018) 337–380.
- [34] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.
- [35] J. W. Cahn, J. E. Hilliard, Free energy of a nonuniform system. i. interfacial free energy, *The Journal of chemical physics* 28 (2) (1958) 258–267.
- [36] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, Y. LeCun, The loss surfaces of multilayer networks, in: *Artificial Intelligence and Statistics*, 2015, pp. 192–204.
- [37] G. J. Dvorak, Transformation field analysis of inelastic composite materials, *Proc. R. Soc. Lond. A* 437 (1992) 311–327.