# Micromechanics-based material networks revisited from the interaction viewpoint; robust and efficient implementation for multi-phase composites

Van Dung Nguyen[a,1], Ludovic Noels[a,*]

[a]*Computational & Multiscale Mechanical of Materials (CM3),*
*Department of Aerospace and Mechanical Engineering,*
*University of Liège,*
*Quartier Polytech 1, Allée de la Découverte 9, B-4000 Liège, Belgium*

## Abstract

A material network consists of discrete material nodes, which, when interacting, can represent complex microstructure responses. In this work, we investigate this concept of material networks under the viewpoint of the hierarchical network interactions. Within this viewpoint, the response of the material network is governed by a well-defined system of equations and an arbitrary number of phases can be considered, independently of the network architecture. The predictive capability is achieved by, on the one hand, sufficiently deep and rich network interactions to tie the discrete material nodes together, and, on the other hand, an efficient offline training procedure. For this purpose, a unified and efficient framework for an arbitrary network architecture is developed, not only for the offline training, but also for the online evaluation. The efficiency and prediction accuracy of the material network as a surrogate of a homogenization-based multiscale model in predicting the stress-strain response in both contexts of a virtual test and of FE$^2$ multiscale simulations are demonstrated through numerical examples with two-phase and three-phase fiber-reinforced composites.

*Keywords:* Deep material network, Finite strain, Data driven, Closed form implementation

## 1. Introduction

When performing numerical simulations in mechanics, the definition of a constitutive model governing the stress-strain response is a key role to solve the underlying boundary value problem. On the one hand, single-scale models can be derived using phenomenological or physical arguments to capture the physics of the problems. The significant problem with these constitutive models consists in the prediction errors, which could arise by the fact that the real material behavior is so complex that it cannot be accurately captured by a simple model or by the difficulty in identifying the parameters. On the other hand, when considering the simulations of large-scale heterogeneous structures, multiscale models are widely developed

[*] Corresponding author, Phone: +32 4 366 48 26, Fax: +32 4 366 95 05

*Email addresses:* `vandung.nguyen@ulg.ac.be` (Van Dung Nguyen), `L.Noels@ulg.ac.be` (Ludovic Noels)

[1] Postdoctoral Researcher at the Belgian National Fund for Scientific Research (FNRS)

to incorporate the information from the material microstructure and constitutive behaviors at the lower scales through a homogenization process, see reviews by [1, 2]. These so-called homogenization models provide closed forms of the constitutive relationships at the structural scale while remaining of reduced computational cost as compared to the direct numerical simulations that embed the microstructural details.

Among the existing different multiscale methods, computational homogenization (so-called FE$^2$) is the most versatile homogenization method since it can be applied on a large variety of heterogeneous material systems using a full-field finite element discretization of the micro-structure [3, 4]. In this framework, at each material point of the macro-scale Boundary Value Problem (BVP), a microscopic BVP is defined on a representative volume element (RVE) extracted from the material microstructure at that point. With a suitable boundary condition defined from the kinematic variables (*e.g.* strains), this lower-scale BVP is solved yielding the homogenized quantities (*e.g.* stresses). Even though a relative coarse mesh is used at the macro-scale, extreme events involving large deformations and nonlinearities require solving a larger number of microscopic BVPs and lead to intractable computation time and memory that make this approach infeasible for simulations of large-scale heterogeneous structures.

A popular remedy to alleviate the computation cost in the FE$^2$ scheme is to substitute the microscopic BVPs by Reduced Order Models (ROM), which balance the computational cost and accuracy [5, 6, 7, 8, 9, 10, 11, 12]. The governing equations can be solved with a reduced number of degrees of freedom in a reduced-order space based on full-field analyzes by means of proper orthogonal decomposition of the displacement field; this step is possibly followed by a so-called hyper-reduction in order to reduce the evaluation of the internal variables, see the review in [13] for a complete discussion. Within the context of multiscale simulations, the use of ROMs allows obtaining a considerably reduced computational cost [6, 9, 11]. Although ROMs show significant computational benefits, they can become inefficient for the cases involving high nonlinearities. Indeed, the computational efficiency is not really improved by several orders of magnitude because of the computational cost related to the assembly of the reduced residuals unless a hyper-reduction stage follows, but whose accuracy can be limited for highly non-linear models.

Meanwhile, Machine Learning (ML) models, *e.g.* Artificial Neural Networks (ANN), have received an increasing attention in the field of computational mechanics. Thanks to the capability of the ML models to act as universal approximators, the complex input-output relationships representing the constitutive behavior can be approximated in disregard of any physical arguments through a training step using data obtained from offline simulations. The ANNs can either be used to replace different parts of single-scale constitutive models, *e.g.* evolution laws of internal variables [14, 15] or they can serve as complete constitutive laws [16, 17, 18]. In the context of the FE$^2$ scheme, ANN-based surrogate models have been extensively used. In elasticity and nonlinear elasticity, the so-called feed-forward ANNs can be used to construct the homogenized strain energy surface from which the stress-strain relationship can be derived [19, 20] or to directly approximate this stress-strain relationship [21, 22, 23]. Although the feed-forward ANNs have been shown to be reliable surrogate models in these history-independent cases, the difficulty arises with irreversible behaviors, *e.g.* plasticity, in both the ANN architecture and offline training since the information moves only along one direction. As a result, the material response under complex loading conditions (*e.g.* unloading, reloading) cannot be captured. Nevertheless, the inputs of the feed-forward ANNs can be enriched by the ones related to the strain paths, *e.g.* length of strain paths, to introduce the history-dependency [24]. Recently, Recurrent Neural Networks (RNN) have been used in [25, 23, 26, 27, 28] to approximate the history-dependent homogenized stress-strain relationships with accuracy, thanks to their ability to account for the information about what has been computed so far. Although these ANN-based surrogate models provide a promising speedup, the prediction could be inaccurate when extrapolating beyond the offline sampling space, which is not easily built large enough when dealing with irreversible material responses since this requires a large amount of offline simulations.

Recently, the concept of Deep Material Network (DMN) was pioneered in [29] as a novel data-driven modeling technique for two-phase heterogeneous materials. Based on two-phase laminates of given orientations serving as mechanistic building blocks, a binary hierarchical topological structure, so-called DMN, is created. During the training process for a fixed microstructure, this DMN is able to “learn”, not only the contribution of its constituents, but also its constituents interactions. Once trained, the DMN handles well extrapolation, since it is able to predict nonlinear material behaviors both accurately and efficiently although only linear elastic data were used during the offline training. The online evaluation of the DMN is obtained through a forward homogenization process and a backward de-homogenization process of the linearization of the stress-strain behavior. This DMN was applied to 3-dimensional microstructures [30] and interface failure [31] showing that the predicted stress-strain curves are close to direct numerical simulation results. The theoretical basis of the DNM from the viewpoint of the classical small-strain micromechanics was investigated in [32], from which the rotation-free DMNs and a new online evaluation technique based on the energy minimization statement were proposed. Only one direction characterizing the laminate direction was considered in the rotation-free DMNs, leading to a smaller number of degrees of freedom compared to the micro-orientation defined by three angles in the original version [29]. More complex micro-mechanical models such as mean-field homogenization in the mechanistic building blocks was also recently considered for woven composite materials [33], allowing reducing the needed number of blocks to a few couples.

The aim of this work is to derive the DMN theory under the viewpoint of network interactions with the following highlights:

- A material network is formed from discrete material nodes as a surrogate of a computational micro-structural models with an arbitrary kinematic assumption. Consequently, a well-defined system of DMN governing equations can be derived and allows easily estimating the response of the material network during the online stage.

- Not only the stress measure but also the tangent operator are computed in a closed form way as the output of the DMNs. As a result, the DMNs can be considered as an efficient surrogate model for estimating the stress-strain relationships in FE$^2$ multiscale simulations.

- A complete and efficient offline training procedure is proposed for material networks, in which an arbitrary architecture and an arbitrary number of physical phases of the micro-structure can be considered.

The paper is organized as follows. In Section 2, the theory of homogenization-based multiscale model is briefly recalled. In Section 3, the micromechanics-based material network structure is introduced in the context of the network interactions and the unified framework is detailed. A complete and efficient offline training procedure for the material network is proposed in Section 4. The accuracy and computational efficiency of the proposed framework are demonstrated through various numerical examples with two-phase and three-phase fiber-reinforced composites in Section 5.

## 2. Homogenization-based multiscale model

The goal of a homogenization-based multiscale model is to derive constitutive relations from the resolution of a microscopic boundary value problem (BVP). This microscopic BVP is defined by a representative volume element (RVE) $V_0$ whose boundary is $\partial V_0$, by an appropriate boundary condition, and by the constitutive models of the different phases. The microscopic BVP is usually formulated using classical continuum under the assumption of the separation of scales, in which the characteristic length of the microscopic BVP is much smaller than the one of the macro-scale [3]. In this section, the theory of the homogenization-based multiscale model in the context of finite strains is briefly summarized.

At a material point $\mathbf{X} \in V_0$, the deformation gradient $\mathbf{F}$ is given by

$$\mathbf{F} = \mathbf{x} \otimes \boldsymbol{\nabla}_0 \,, \tag{1}$$

where $\boldsymbol{\nabla}_0$ is the gradient operator with respect to the microscopic reference configuration and $\mathbf{x}$ is the position of the material point $\mathbf{X} \in V_0$ in the current configuration. In the absence of dynamic effects, the equilibrium state of the microscopic BVP is governed by the following equations

$$\mathbf{P} \cdot \boldsymbol{\nabla}_0 = \mathbf{0} \quad \forall \mathbf{x} \in V_0 \text{ and} \tag{2}$$

$$\mathbf{P} \cdot \mathbf{N} = \mathbf{T} \quad \forall \mathbf{x} \in \partial V_0 \,, \tag{3}$$

where $\mathbf{P}$ is the microscopic first Piola-Kirchhoff stress tensor and $\mathbf{T}$ is the surface traction per unit reference surface on the boundary $\partial V_0$ whose outward unit normal $\mathbf{N}$ is expressed in the reference configuration. The RVE consists of $P$ constituents, also called physical phases in this paper, whose mechanical behaviors are explicitly provided by

$$\mathbf{P}\left(t\right) = \mathfrak{P}^p\left(\mathbf{F}\left(t\right), \mathbf{Z}\left(t\right)\right) \text{ and evolution laws for } \mathbf{Z} \,, \tag{4}$$

for $p = 0 \,, \ldots \,, P-1$, and where $\mathbf{Z}$ is a set of internal variables defined to follow the history-dependent processes.

The macroscopic deformation gradient $\bar{\mathbf{F}}$ and first Piola-Kirchhoff stress $\bar{\mathbf{P}}$ are related to their microscopic counterparts by the following equations

$$\left\langle \mathbf{F} \right\rangle_{V_0} = \bar{\mathbf{F}} \,, \tag{5}$$

$$\left\langle \mathbf{P} \right\rangle_{V_0} = \bar{\mathbf{P}} \,, \text{ and} \tag{6}$$

$$\left\langle \mathbf{P} : \delta\mathbf{F} \right\rangle_{V_0} = \bar{\mathbf{P}} : \delta\bar{\mathbf{F}} \,, \tag{7}$$

where $\langle \bullet \rangle_{V_0} = \frac{1}{V_0}\int_{V_0} \bullet \, dV$ denotes the averaging operator over $V_0$. Equations (5, 6) are respectively the averaging strain and stress identities. Equation (7) is the Hill-Mandel condition, which corresponds to the requirement of energy consistency between the different scales.

In the context of strain-driven homogenization, $\bar{\mathbf{F}}$ is known and $\bar{\mathbf{P}}$ is sought. From the value of $\bar{\mathbf{F}}$, the boundary condition of the microscopic BVP is defined in such a way that Eqs. (5, 7) are satisfied *a priori*. The microscopic BVP is solved and the homogenized stress $\bar{\mathbf{P}}$ is computed following Eq. (6). In order to solve the microscopic BVP, a perturbation field $\boldsymbol{\xi}$ is introduced over $V_0$ such that

$$\boldsymbol{\xi} = \mathbf{x} - \bar{\mathbf{F}} \cdot \mathbf{X} \,. \tag{8}$$

Consequently, Eq. (1) can be rewritten as

$$\mathbf{F} = \bar{\mathbf{F}} + \boldsymbol{\xi} \otimes \boldsymbol{\nabla}_0 \,, \tag{9}$$

where $\boldsymbol{\xi} \otimes \boldsymbol{\nabla}_0$ represents the fluctuation deformation gradient. Equation (5) therefore implies

$$\int_{\partial V_0} \boldsymbol{\xi} \otimes \mathbf{N} \, dA = \mathbf{0} \,. \tag{10}$$

By using the strong form stated by Eqs. (2, 3) and performing an integration by parts, Eq. (7) is rewritten as

$$\int_{V_0} \mathbf{P} : \left(\delta\boldsymbol{\xi} \otimes \boldsymbol{\nabla}_0\right) \, dV = \int_{\partial V_0} \mathbf{T} \cdot \delta\boldsymbol{\xi} \, dA = 0 \,. \tag{11}$$

In order to resolve the strong form stated by Eqs. (2, 3) under the constraints described by Eqs. (10, 11), a minimal kinematic vector field is defined as

$$\mathcal{U}^{\min}\left(V_0\right) = \left\{ \boldsymbol{\xi} \middle| \int_{\partial V_0} \boldsymbol{\xi} \otimes \mathbf{N} \, dA = \mathbf{0} \right\} \,. \tag{12}$$

Using the Hill-Mandel condition (11), the weak form of the microscopic BVP is stated as finding $\boldsymbol{\xi} \in \mathcal{U}\left(V_0\right) \subseteq \mathcal{U}^{\min}\left(V_0\right)$ such that

$$\int_{V_0} \mathbf{P} : \left(\delta\boldsymbol{\xi} \otimes \boldsymbol{\nabla}_0\right) \, dV = 0 \;\forall \delta\boldsymbol{\xi} \in \mathcal{U}\left(V_0\right) \,, \tag{13}$$

where $\mathcal{U}\left(V_0\right)$ is either a subset of or the set $\mathcal{U}^{\min}\left(V_0\right)$. It is noted that the resolution of this weak form (13) automatically fulfills the Hill-Mandel condition stated by Eq. (11). The space $\mathcal{U}\left(V_0\right)$ is obtained by defining a specific boundary condition kinematically stronger than $\mathcal{U}^{\min}\left(V_0\right)$, *e.g.* linear displacement boundary condition, minimal kinematic boundary condition, periodic boundary condition [34]. In this work, the periodic boundary condition is employed because of its good convergence rate in terms of the microstructure RVE size [35, 36]. The kinematic space for the periodic boundary condition is given by

$$\mathcal{U}\left(V_0\right) = \left\{ \boldsymbol{\xi} | \boldsymbol{\xi}\left(\mathbf{X}^+\right) = \boldsymbol{\xi}\left(\mathbf{X}^-\right) \forall \mathbf{X}^+ \in \partial V_0^+ \text{ and corresponding } \mathbf{X}^- \in \partial V_0^- \right\} \,, \tag{14}$$

where $\partial V_0^+$ and $\partial V_0^-$ are two non-overlapping and separate parts of the boundary $\partial V_0$. The weak form (13) coupled with the constitutive laws specified in Eq. (4) can be solved using the finite element method.

To summarize, in the context of the homogenization-based multiscale approach, the homogenized stress $\bar{\mathbf{P}}$ is estimated from $\bar{\mathbf{F}}$ following:

- An RVE $V_0$ is chosen with its discretization;
- The kinematic space $\mathcal{U}(V_0)$ is chosen. The application of the periodic boundary condition on non periodic micro-structures follows the interpolation method proposed in [37].
- For a given $\bar{\mathbf{F}}$ history, the constrained microscopic BVP (13) is iteratively solved using at the micro-scale the constitutive laws (4) as detailed in [38].
- The extraction of $\bar{\mathbf{P}}$ follows Eq. (6). When this model is used within a macroscopic finite element simulation as a constitutive law, the homogenized tangent operator $\bar{\mathbf{L}} = \frac{\partial \bar{\mathbf{P}}}{\partial \bar{\mathbf{F}}}$ needs to be estimated in order to perform the macro-scale Newton-Raphson iterations. The estimation of $\bar{\mathbf{L}}$ can be found in [38].

From the computational point of view, the homogenization-based multiscale model acts as the usual constitutive laws. Its explicit form generally cannot be achieved, but the stress-strain relationship is always available through the resolutions of the microscopic BVP. This technique provides an accurate prediction as it enables the incorporation of both geometrical and material nonlinearities. However, the finite element simulations with embedded computational homogenization-based models, so-called FE$^2$ simulations, need to perform a large number of the iterative resolutions of the microscopic BVP (at all macro-scale Gauss points and for all macroscopic Newton-Raphson iterations), for which an intractable computation time is required. In order to speed up the multiscale analysis, a surrogate model needs to be considered to replace the costly homogenization-based multiscale model.

## 3. Micromechanics-based material network

In this section, a micromechanical viewpoint of the concept of material networks is provided. A material network is built from a list of discrete nodes and the definition of their interactions. These interactions are expressed in terms of fitting parameters, which are identified through an offline training stage. Such interactions can be obtained with the hierarchical tree architectures considered in [29, 30, 32], from which the predictive capability of these models can be explained. Furthermore, the interaction viewpoint in this work allows, not only easily evaluating the response of material networks, thus seen as a surrogate, through the resolution of the governing equations, but also considering an arbitrary number of physical phases independently of the tree architecture. In this section, the theory of the micromechanics-based material network is introduced.

### *3.1. Governing equations of material networks*

Let us consider $\mathcal{K}$, a list of indexes, representing $N$ discrete material nodes indexed from 0 to $N-1$ as

$$\mathcal{K} = [0\,,1\,,\dots\,,N-1]\ . \tag{15}$$

An arbitrary material node $i \in \mathcal{K}$ possesses

- a positive weight $W^i$ in order to quantify its contribution into the network; and
- a physical state, consisting of the deformation gradient $\mathbf{F}^i$, the first Piola-Kirchhoff stress $\mathbf{P}^i$, and internal state $\mathbf{Z}^i$, which is governed by a constitutive law.

In the remaining of this paper, the superscript "$i$" is used to refer to the node $i \in \mathcal{K}$.

Assuming that there exists a homogenization-based multiscale model defined on an RVE $V_0$ is decomposed into $N$ parts $V_0^i$, where $V_0^i$ is associated to the node $i$, with $i = 0\,,\dots\,,N-1$. One thus has

$$\frac{V_0^i}{V_0} = \frac{W^i}{\mathcal{S}\left(\mathcal{K}\right)} \text{ with } i = 0\,,\dots\,,N-1\,, \tag{16}$$

where $\mathcal{S}\left(\bullet\right)$ is the weighted sum operator defined on the list of indexes $\bullet$ of an arbitrary material network as

$$\mathcal{S}\left(\bullet\right) = \sum_{i\in\bullet} W^i\,. \tag{17}$$

Each part $V_0^i \in V_0$ corresponds to the material node $i$ in a homogenization sense in which Eqs. (5, 6, 7) are postulated, yielding

$$\mathbf{F}^i = \left\langle\mathbf{F}\right\rangle_{V_0^i}\,, \mathbf{P}^i = \left\langle\mathbf{P}\right\rangle_{V_0^i}\ ,\ \text{and } \mathbf{P}^i : \delta\mathbf{F}^i = \left\langle\mathbf{P}:\delta\mathbf{F}\right\rangle_{V_0^i}\ , \tag{18}$$

where $\left\langle\bullet\right\rangle_{V_0^i} = \frac{1}{V_0^i}\int_{V_0^i} \bullet\, dV$ denotes the averaging operator over $V_0^i$. As a result, Eqs. (5, 6, 7) can be rewritten under discrete forms

$$\begin{aligned}
\frac{1}{\mathcal{S}\left(\mathcal{K}\right)}\sum_{i=0}^{N-1} W^i\mathbf{F}^i &= \bar{\mathbf{F}}\,, && (19)\\
\frac{1}{\mathcal{S}\left(\mathcal{K}\right)}\sum_{i=0}^{N-1} W^i\mathbf{P}^i &= \bar{\mathbf{P}}\,,\ \text{and} && (20)\\
\frac{1}{\mathcal{S}\left(\mathcal{K}\right)}\sum_{i=0}^{N-1} W^i\mathbf{P}^i : \delta\mathbf{F}^i &= \bar{\mathbf{P}} : \delta\bar{\mathbf{F}}\,. && (21)
\end{aligned}$$

The behavior of each material node is governed by a material law specified in Eq. (4). For the material node $i$ with $i = 0, \dots, N-1$, the local stress $\mathbf{P}^i$ and the internal variables $\mathbf{Z}^i$ are estimated from the current deformation gradient $\mathbf{F}^i$ (and previous values of the internal variables) as

$$\mathbf{P}^i\left(t\right) = \mathfrak{P}^{p_i}\left(\mathbf{F}^i\left(t\right), \mathbf{Z}^i\left(t\right)\right) \text{ and evolution laws for } \mathbf{Z}^i\,, \tag{22}$$

where $p_i$ denotes the phase index of the material node $i$.

Equations (19, 20, 21) govern the response of the material network with $\mathbf{F}^0, \ldots, \mathbf{F}^{N-1}$ being unknowns. The interactions between these material nodes must be introduced to tie them together and to close the problem statement. Inspired from the work [32], these interactions are obtained through a so-called interaction mapping $\mathcal{I}$, which is defined by

$$\begin{aligned} \mathcal{I}:\left(\mathbf{a}^0, \ldots, \mathbf{a}^{M-1}\right) &\rightarrow \left(\mathbf{F}^0, \ldots, \mathbf{F}^{N-1}\right) \\ &: \quad \mathbf{F}^i = \bar{\mathbf{F}} + \sum_{j=0}^{M-1}\alpha^{i,j}\mathbf{a}^j\otimes\mathbf{G}^j \text{ with } i = 0, \ldots, N-1\,, \end{aligned} \tag{23}$$

where $M$ is the number of interaction mechanisms defined by the DMN structure, where $\alpha^{i,j}$ and the $M$ unit vectors $\mathbf{G}^j$ with $i = 0, \ldots, N-1$ and $j = 0, \ldots, M-1$ are the fitting parameters supposed to be known, *e.g.* following an offline training stage, and where the $M$ vectors $\mathbf{a}^j$ with $j = 0, \ldots, M-1$ are the new unknowns substituting to the local deformation gradients $\mathbf{F}^i$ with $i = 0, \ldots, N-1$. For a mechanism $j$ at node $i$, the unit normal vector $\mathbf{G}^j$ characterizes the interaction direction while $\mathbf{a}^j$ characterizes the interaction incompatibility. Hereafter, $\mathbf{G}^j$ and $\mathbf{a}^j$ are called as the interaction direction and interaction incompatibility vector, respectively. The interaction mapping (23) is equivalent to Eq. (9) since the local deformation gradient $\mathbf{F}^i$ at a material node is also decomposed into a homogeneous part $\bar{\mathbf{F}}$ and a fluctuation part $\sum_{j=0}^{M-1}\alpha^{i,j}\mathbf{a}^j\otimes\mathbf{G}^j$ from which the interaction with other material nodes is defined.

Using the interaction mapping described in Eq. (23), Eqs. (19, 21) can be respectively rewritten as

$$\sum_{j=0}^{M-1}\left(\sum_{i=0}^{N-1}W^i\alpha^{i,j}\right)\mathbf{a}^j\otimes\mathbf{G}^j = \mathbf{0}\,, \text{ and} \tag{24}$$

$$\sum_{j=0}^{M-1}\left[\left(\sum_{i=0}^{N-1}W^i\mathbf{P}^i\alpha^{i,j}\right)\cdot\mathbf{G}^j\right]\cdot\delta\mathbf{a}^j = \mathbf{0}\,. \tag{25}$$

Equations (24, 25) represent respectively the constrained kinematic space and the weak form from which the solution of this material network can be found. An obvious choice to fulfill Eq. (24) *a priori* is

$$\sum_{i=0}^{N-1}W^i\alpha^{i,j} = 0 \text{ with } j = 0, \ldots, M-1\,. \tag{26}$$

In this last equation, a mechanism $j$ is characterized by a list of values $\left[\alpha^{0,j}, \ldots, \alpha^{N-1,j}\right]$, which can be chosen as functions of the values of $W^i$ with $i = 0, \ldots, N-1$ and satisfies Eq. (26). A particular choice for this mechanism will be detailed in the next section. Because of

Eq. (26), Eq. (24) is satisfied independently to the values of $\mathbf{a}^j$. As a result, one can choose $\mathbf{a}^0, \mathbf{a}^1, \dots, \mathbf{a}^{M-1}$ being independent. Equation (25) therefore results in the following system of $M$ equations

$$\left(\sum_{i=0}^{N-1} W^i \mathbf{P}^i \alpha^{i,j}\right) \cdot \mathbf{G}^j = \mathbf{0} \text{ with } j = 0\,, \dots\,, M-1\,. \tag{27}$$

It is noted that the resolution of Eq. (27) automatically fulfills the Hill-Mandel condition stated by Eq. (21). The homogenized stress $\bar{\mathbf{P}}$ is estimated following Eq. (20) as an observable quantity.

Equation (26) is an inevitable condition of the underlying framework to fulfill *a priori* Eq. (19) which is equivalently stated by the strain averaging theorem often employed in the homogenization theories. It is noted that the governing equations (19, 20, 21, 22) of the DMN do not rely on any assumption about the number of material phases, the nature of their constitutive behaviors, or the kinematic formalism (*i.e.* small strain or large strain). Although the framework is written in the large strain formalism, the small strain formalism is naturally obtained by considering the stress measure as the Cauchy stress. In the next section, the condition (26) is satisfied through the definition of the network interactions, *i.e.* for each $j = 0\,, \dots\,, M-1$, $\alpha^{0,j}, \dots, \alpha^{N-1,j}$ are defined as functions of $W^0\,, \dots\,, W^{N-1}$ satisfying Eq. (26).

Eventualy, a material network $\mathcal{M}$ is defined as a network of material nodes indexed in $\mathcal{K}$ with their interactions defined through the interaction mapping $\mathcal{I}$. The homogenized stress $\bar{\mathbf{P}}$ is estimated from $\bar{\mathbf{F}}$ in the context of the material network following

- The set of $N$ material nodes, with their parameters $W^i$ and $M$ interaction mechanisms characterized by $\alpha^{i,j}$ and $\mathbf{G}^j$ with $i = 0\,, \dots\,, N-1$ and $j = 0\,, \dots\,, M-1$, are determined from an offline stage.

- The governing equation (27) is iteratively solved in terms of the unknowns $\mathbf{a}^k$, with $k = 0\,, \dots\,, M-1$, as detailed in Section 3.4 of this work.

- The extraction of $\bar{\mathbf{P}}$ follows Eq. (20). When this model is used within a macro-scale finite element simulation as a constitutive law, the homogenized tangent operator $\bar{\mathbf{L}} = \frac{\partial \bar{\mathbf{P}}}{\partial \bar{\mathbf{F}}}$ needs to be estimated in order to perform the Newton-Raphson iterations. The estimation of $\bar{\mathbf{L}}$ is detailed in Section 3.4 of this work.

### *3.2. Network interactions*

The network interactions are defined to satisfy *a priori* Eq. (26). From Eq. (23), one can define an *interaction mechanism* $j$, with $j = 0, \dots, M-1$, as the pair $(\boldsymbol{\alpha}^j, \mathbf{G}^j)$ in which $\boldsymbol{\alpha}^j$ is given as

$$\boldsymbol{\alpha}^j = \left[\alpha^{0,j}, \dots, \alpha^{N-1,j}\right]\,, \tag{28}$$

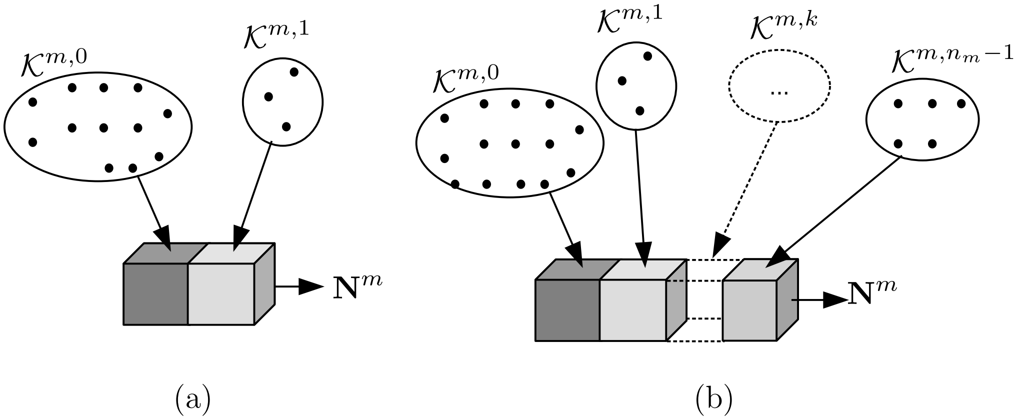
and satisfies *a priori* Eq. (26). The list of these $M$ interaction mechanisms is denoted by $\boldsymbol{\mathcal{H}}$ with

$$\boldsymbol{\mathcal{H}} = \left[\left(\boldsymbol{\alpha}^0, \mathbf{G}^0\right)\,, \dots\,, \left(\boldsymbol{\alpha}^{M-1}, \mathbf{G}^{M-1}\right)\right]\,, \tag{29}$$

which gathers all the fitting parameters of the material network and is assumed to be determined from an offline training stage.

A *network interaction* corresponds to the creation of one or more interaction mechanisms. For a material network $\mathcal{M}$ with a nodes list indexed in $\mathcal{K}$, each network interaction is defined by making a connection either between two sub-lists of material nodes extracted from $\mathcal{K}$, leading to a *two-list interaction*, or, more generally, between $n$ (with $n \geq 2$) sub-lists of material nodes extracted from $\mathcal{K}$, leading to an *n-list interaction* (also-called *multiple-list interaction* if the value of $n > 2$ is not mentioned). A two-list interaction results into a single interaction mechanism while an $n$-list interaction results into $n-1$ interaction mechanisms. These network interaction types are sketched in Fig. 1 and detailed subsequently.

To build all interaction mechanisms for the material network $\mathcal{M}$, $\boldsymbol{\mathcal{H}}$ starts as an empty list. Once a network interaction is created, its resulting interaction mechanisms are appended to $\boldsymbol{\mathcal{H}}$. Without loss of generality, in this section, one assumes that a network interaction indexed by $m$ induces $n_m - 1$ mechanisms indexed by $q_m, \ldots, q_m + n_m - 2$, where $q_m$ denotes the starting index of the mechanisms associated to the network interaction $m$, *i.e.* $q_m = \sum_{k=0}^{m-1}(n_k - 1)$. Assuming the existence of $L$ network interactions, one has $\sum_{m=0}^{L-1}(n_m - 1) = M$.



(a)

(b)

Figure 1: Sketch of a network interaction indexed by $m$ of $n_m$ sub-lists inducing $n_m - 1$ mechanisms indexed by $q_m$, ..., $q_m + n_m - 2$: (a) a two-list interaction and (b) a multiple-list interaction as an extension of the two-list interaction. $\mathbf{N}^m$ denotes the interaction direction.

Let us first consider a list of material nodes $\mathcal{K}$ with two of its arbitrary nonempty and non-overlapping sub-lists $\mathcal{K}^{m,0}$ and $\mathcal{K}^{m,1}$ of material nodes, *i.e.* $\mathcal{K}^{m,0} \neq \emptyset$, $\mathcal{K}^{m,1} \neq \emptyset$, and $\mathcal{K}^{m,0} \cap \mathcal{K}^{m,1} = \emptyset$. The network interaction between $\mathcal{K}^{m,0}$ and $\mathcal{K}^{m,1}$, the so-called two-list interaction, following the direction $\mathbf{N}^m$, the so-called interaction direction, as sketched in Fig. 1(a), results into an interaction mechanism characterized by a pair $(\boldsymbol{\alpha}^{q_m}, \mathbf{G}^{q_m})$, in which $\mathbf{G}^{q_m} = \mathbf{N}^m$ and $\boldsymbol{\alpha}^{q_m}$ can be obtained by choosing the values of $\alpha^{i,q_m}$ with $i \in \mathcal{K}$ satisfying Eq. (26) following

$$\alpha^{i,q_m} = \begin{cases} \dfrac{1}{\mathcal{S}\left(\mathcal{K}^{m,0}\right)} & \text{if } i \in \mathcal{K}^{m,0}\,, \\ -\dfrac{1}{\mathcal{S}\left(\mathcal{K}^{m,1}\right)} & \text{if } i \in \mathcal{K}^{m,1}\,, \text{ and} \\ 0 & \text{if } i \in \mathcal{K}\backslash\left(\mathcal{K}^{m,0} \cup \mathcal{K}^{m,1}\right)\,, \end{cases} \tag{30}$$

since $\mathcal{S}\left(\mathcal{K}^{m,0}\right)=\sum_{i\in\mathcal{K}^{m,0}} W^i$ and $\mathcal{S}\left(\mathcal{K}^{m,1}\right)=\sum_{i\in\mathcal{K}^{m,1}} W^i$. Following Eq. (23), this interaction contributes to the deformation gradient of each material node in the network by

$$\begin{cases} \text{In } \mathcal{K}^{m,0} & : \dfrac{1}{\mathcal{S}\left(\mathcal{K}^{m,0}\right)}\mathbf{a}^{q_m}\otimes\mathbf{N}^m\,, \\ \text{In } \mathcal{K}^{m,1} & : -\dfrac{1}{\mathcal{S}\left(\mathcal{K}^{m,1}\right)}\mathbf{a}^{q_m}\otimes\mathbf{N}^m\,,\ \text{and} \\ \text{In } \mathcal{K}\backslash\left(\mathcal{K}^{m,0}\cup\mathcal{K}^{m,1}\right) & : \mathbf{0}\,. \end{cases} \tag{31}$$

As a result, Eq. (27) can be rewritten as

$$\left[\frac{1}{\mathcal{S}\left(\mathcal{K}^{m,0}\right)}\sum_{i\in\mathcal{K}^{m,0}} W^i\mathbf{P}^i-\frac{1}{\mathcal{S}\left(\mathcal{K}^{m,1}\right)}\sum_{i\in\mathcal{K}^{m,1}} W^i\mathbf{P}^i\right]\cdot\mathbf{N}^m=\mathbf{0}\,, \tag{32}$$

in which $\frac{1}{\mathcal{S}\left(\mathcal{K}^{m,0}\right)}\sum_{i\in\mathcal{K}^{m,0}} W^i\mathbf{P}^i$ and $\frac{1}{\mathcal{S}\left(\mathcal{K}^{m,1}\right)}\sum_{i\in\mathcal{K}^{m,1}} W^i\mathbf{P}^i$ represent respectively the average stresses in $\mathcal{K}^{m,0}$ and $\mathcal{K}^{m,1}$. The last equation implies the balance between these two average stresses following the direction $\mathbf{N}^m$.

The two-list interaction can be directly extended into a general case of $n_m$ (with $n_m\geq 2$) non-overlapping and nonempty sub-lists $\mathcal{K}^{m,0}$, $\mathcal{K}^{m,1}$, ..., $\mathcal{K}^{m,n_m-1}$ of $\mathcal{K}$, *i.e.* $\mathcal{K}^{m,k}\neq\emptyset\,,\mathcal{K}^{m,k}\cap\mathcal{K}^{m,l}=\emptyset$ with $k=0,\ldots,n_m-1$, $l=0,\ldots,n_m-1$, and $l\neq k$. By assuming a unique vector $\mathbf{G}^{q_m}$ for a given multiple-list interaction or $n_m$-list interaction, the latter can be obtained by applying $n_m-1$ two-list interactions of $n_m-1$ pairs of two consecutive sub-lists $\mathcal{K}^{m,k}$ and $\mathcal{K}^{m,k+1}$ with $k=0,\ldots,n_m-2$ following the direction $\mathbf{N}^m$ as sketched in Fig. 1(b). This network interaction results into $n_m-1$ different interaction mechanisms as

$$\left[\left(\boldsymbol{\alpha}^{q_m},\mathbf{G}^{q_m}\right),\ldots,\left(\boldsymbol{\alpha}^{q_m+n_m-2},\mathbf{G}^{q_m+n_m-2}\right)\right]\,, \tag{33}$$

in which $\mathbf{G}^{q_m}=\ldots=\mathbf{G}^{q_m+n_m-2}=\mathbf{N}^m$ and $\boldsymbol{\alpha}^{q_m+k}$ with $k=0,\ldots,n_m-2$ can be obtained by choosing the values of $\alpha^{i,q_m+k}$ with $i\in\mathcal{K}$ satisfying Eq. (26) following

$$\alpha^{i,q_m+k}=\begin{cases} \dfrac{1}{\mathcal{S}\left(\mathcal{K}^{m,k}\right)} & \text{if } i\in\mathcal{K}^{m,k}\,, \\ -\dfrac{1}{\mathcal{S}\left(\mathcal{K}^{m,k+1}\right)} & \text{if } i\in\mathcal{K}^{m,k+1}\,,\ \text{and} \\ 0 & \text{if } i\in\mathcal{K}\backslash\left(\mathcal{K}^{m,k}\cup\mathcal{K}^{m,k+1}\right)\,, \end{cases} \qquad \text{with } k=0,\ldots,n_m-2\,. \tag{34}$$

Following the interaction mapping (23), the contribution of this interaction to the deformation gradient in each material node of $\mathcal{K}^{m,k}$ with $k = 0, \ldots, n_m - 1$ reads

$$\begin{cases} \text{In } \mathcal{K}^{m,0} & : \dfrac{1}{\mathcal{S}\left(\mathcal{K}^{m,0}\right)} \mathbf{a}^{q_m} \otimes \mathbf{N}^m \,, \\ \text{In } \mathcal{K}^{m,1} & : \dfrac{1}{\mathcal{S}\left(\mathcal{K}^{m,1}\right)} \left(\mathbf{a}^{q_m+1} - \mathbf{a}^{q_m}\right) \otimes \mathbf{N}^m \,, \\ \vdots & \\ \text{In } \mathcal{K}^{m,n_m-1} & : \dfrac{1}{\mathcal{S}\left(\mathcal{K}^{m,n_m-1}\right)} \left(-\mathbf{a}^{q_m+n_m-2}\right) \otimes \mathbf{N}^m \,, \text{ and} \\ \text{In } \mathcal{K} \backslash \left(\mathcal{K}^{m,0} \cup \ldots \cup \mathcal{K}^{m,n_m-1}\right) & : \mathbf{0} \,. \end{cases} \tag{35}$$

As a result, Eq. (27) yields

$$\left[\frac{1}{\mathcal{S}\left(\mathcal{K}^{m,k}\right)} \sum_{i \in \mathcal{K}^{m,k}} W^i \mathbf{P}^i - \frac{1}{\mathcal{S}\left(\mathcal{K}^{m,k+1}\right)} \sum_{i \in \mathcal{K}^{m,k+1}} W^i \mathbf{P}^i\right] \cdot \mathbf{N}^m = \mathbf{0} \text{ with } k = 0, \ldots, n_m - 2 \,, \tag{36}$$

where $\frac{1}{\mathcal{S}\left(\mathcal{K}^{m,k}\right)} \sum_{i \in \mathcal{K}^{j,k}} W^i \mathbf{P}^i$ represents the average stress in $\mathcal{K}^{m,k}$. The last equation implies the balance between these $n_m$ averaging stresses following the direction $\mathbf{N}^m$.

In the following sections, one assumes that the material network $\mathcal{M}$ of a list of material nodes $\mathcal{K}$ is built with $L$ network interactions leading to $M$ interaction mechanisms. All the interaction directions are stored in a list of vectors as

$$\boldsymbol{\mathcal{N}} = \left[\mathbf{N}^0, \ldots, \mathbf{N}^{L-1}\right] \,. \tag{37}$$

Once all the network interactions are defined, the interaction mechanisms given in Eq. (29) are known as functions of the weights $W^i$ for $i \in \mathcal{K}$ and the interaction directions $\boldsymbol{\mathcal{N}}$.

### *3.3. Hierarchical interaction-based deep material network*

The basic concept of the deep material network (DMN) is to use a collection of connected material nodes to describe the complex response of microstructured materials [29]. In this work, the DMN is built from an arbitrary list of material nodes by specifying multiple hierarchical network interactions. This strategy can be described as follows:

- At the lowest level (so-called level 0), each material node is separately found in a separate sub-list.

- Once a network interaction is formed, a merged sub-list is defined as the union of all the sub-lists participating to this network interaction. This merged sub-list is then used as a sub-list in a network interaction at the upper level.

- At current level (level $c$ with $c \geq 1$), multiple network interactions are created by forming multiple $n$-list interactions between $n$ sub-lists, with the former $n$-list interactions corresponding to the merged sub-lists of the lower level (level $c - 1$) for $c > 1$. These network interactions are non-overlapping, *i.e.* a sub-list at the level $c - 1$ cannot take part in two different network interactions at the level $c$. This recursive procedure stops when all the material nodes of the material network under consideration are found in one unique merged sub-list.

An example of this strategy for 10 material nodes is illustrated in Fig. 2(a). First, at level 0, 10 material nodes are separately present. Then, at level 1, 5 non-overlapping two-list interactions are created. Next, at level 2, a two-list interaction and a three-list interaction are created from the merged sub-lists of the interactions at the level 1. Finally, at the level 3 (the highest level), the merged sub-lists of the two interactions existing at the level 2 are used to define a unique two-list interaction. These hierarchical network interactions can be arranged under a tree architecture, as shown in Fig. 2(b) for the data reported in Fig. 2(a). It is noted that the leaves of the tree correspond to material nodes while regular nodes, *i.e.* nodes having children nodes, of the tree correspond to the network interactions. The notations "depth", "interaction level", and "position" are also reported for each interaction. Clearly, each node of this tree corresponds to a material network, subsequently called sub-material network in order to distinguish with the material network under consideration, with a corresponding number of material nodes.

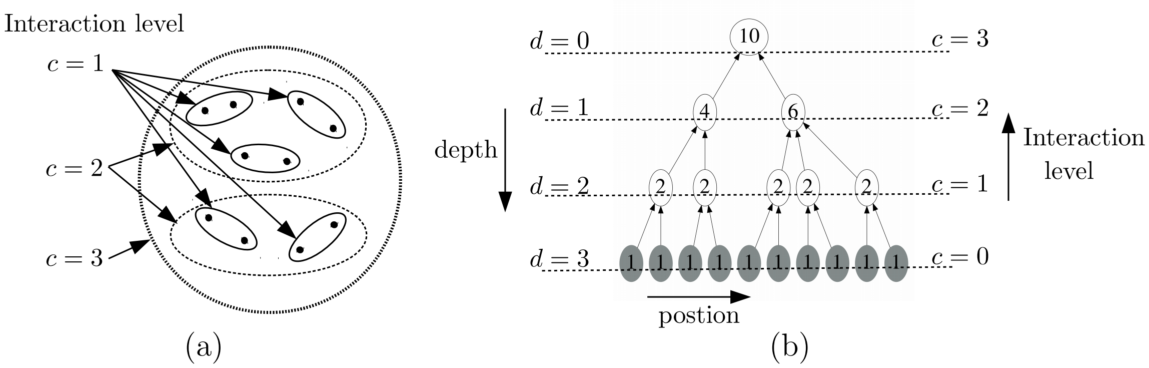


Figure 2: Example of an interaction-based deep material network of 10 material nodes: (a) interaction level definition and (b) equivalent tree architecture in which the blank nodes represent network interactions while filled nodes correspond to material nodes; the number inside each node corresponds to the number of nodes in the respective interaction; the notations "depth", "interaction level", and "position" of an interaction are also reported.

In a hierarchical network structure as shown in Fig. 2, a depth list, denoted by $\mathcal{D}_d$ at the depth $d$, can be defined by including the lists of indexes of all sub material networks at the same depth $d$, with

$$\mathcal{D}_d : \left[\mathcal{K}_d^0, \dots, \mathcal{K}_d^{T_d-1}\right] \text{ for } d = 0, \dots, d_{\max}, \tag{38}$$

where $T_d$ is the number of sub material networks at the depth $d$ and $d_{\max}$ denotes the deepest depth of the material network. Each sub-material network at the deepest depth, *i.e.* $\mathcal{D}_{d_{\max}}$, contains only one material node. When the number of sub-lists in each network interaction is fixed to a unique value of $n \geq 2$, with the total number of material nodes being $N = n^{d_{\max}}$, a perfect $n$-tree of the deepest depth $d_{\max}$ is obtained.

An arising question is how to link the $N$ material nodes of the material network $\mathcal{M}$ with $P$ physical phases whose behaviors are governed by $P$ constitutive laws provided by Eq. (4).

For a multiple-list interaction at the level 1 consisting of $n$ material nodes, the condition $n \leq P$ must be respected when designing a network interaction and two possibilities are distinguished:

- if $n = P$, the $P$ constitutive laws are then randomly distributed into these $P$ material nodes; and
- if $n < P$, $n$ constitutive laws are randomly taken from the total $P$ constitutive laws and these $n$ constitutive laws are then randomly distributed into these $n$ material nodes.

We can find in Figs. 3(a) and (b) respective illustrations of the distribution for 2 and 3 physical phases, *i.e.* 2 and 3 constitutive laws, into the tree structure sketched in Fig. 2(b).

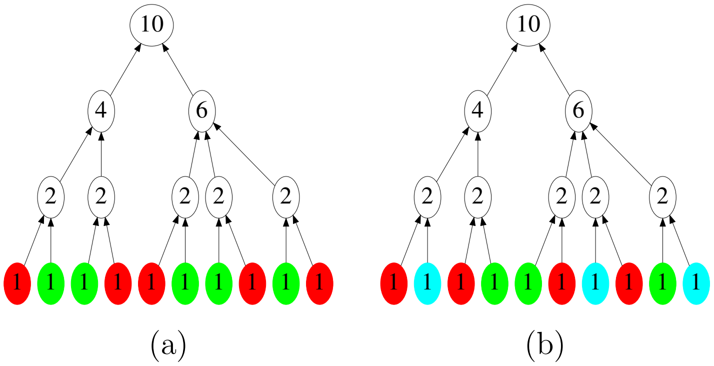


Figure 3: Random distribution of $P$ physical phases for the material network sketched in Fig. 2(b): (a) $P = 2$ (colored in red and green, respectively) and (b) $P = 3$ (colored in red, green, and cyan, respectively).

A small number of material nodes in a DMN allows reducing the computational cost during both the offline training and the online predictions, but this decreases the accuracy. To find a necessary number of material nodes for a given microstructure, different DMN architectures should be tested by increasing the number of material nodes until reaching a required accuracy. Once the architecture of the DMN is initialized, its fitting parameters are randomly initialized as well. As a result, the volume fractions of the physical phases in the DMN do not necessarily correspond to the values in the microstructure under consideration. However, after training, the original volume fractions are recovered for a sufficiently large number of material nodes [29]. It is noted that the model compression, *e.g.* nodal deletion and sub-tree merging, can be used to reduce the number of material nodes, from which the training process can be accelerated [29, 30, 33]. Such an optimization procedure however is not considered in the current framework.

The accurate predictive capability of the DMN presented in the works [29, 30, 32] was explained by the suitable presence of network interactions. In contrast to these works, in the herein developed interaction-based viewpoint, the response of the material network is governed by a micromechanics-based system of equations and an arbitrary number of phases can be considered independently of the network architecture, which can also be an arbitrary hierarchical tree. Moreover, this interaction-based strategy allows facilitating the network evaluation as it will be detailed in the next section.

In summary, the hierarchical interaction-based DMN is viewed as a trainable system involving fitting parameters:

- The weights of the material nodes $W^i$ satisfying $W^i > 0$, $\forall i \in \mathcal{K}$; and
- The interaction directions $\boldsymbol{\mathcal{N}}$ consisting of $L$ unit normal vectors: $\mathbf{N}^m$ with $m = 0, \ldots, L-1$ as described in Section 3.2.

It is noted that the values of $\alpha^{i,j}$ in Eq. (23) are known from the values of $W^i$ when specifying the network interactions. In order to make this DMN become a surrogate of a full-field model, it must be trained to identify these fitting parameters by minimizing a loss function as detailed in Section 4.

### *3.4. Material network evaluation*

Let us consider the material network $\boldsymbol{\mathcal{M}}$ of $N$ material nodes given by a list of indexes $\mathcal{K}$. The governing equations of this DMN are provided in Section 3.1. Assuming that the weights $W^i$ for $i \in \mathcal{K}$ and the interaction directions $\boldsymbol{\mathcal{N}}$ are known following a training stage, see Section 4 for details, from which all the interaction mechanisms $\boldsymbol{\mathcal{H}}$ as described in Eq. (29) are available, the unknowns of the DMN involve only the $M$ vectors $\mathbf{a}^j$ considered in Eq. (23). These unknown vectors are collected in a column vector denoted by $\mathbf{A}$ as

$$\mathbf{A} = \left[ \left(\mathbf{a}^j\right)^T \text{ for } j = 0\,, \ldots\,, M-1 \right]^T . \tag{39}$$

In a kinematically driven and finite strain framework, the input of the DMN is the homogenized deformation gradient $\bar{\mathbf{F}}$ while the outputs consist of the homogenized first Piola-Kirchhoff stress $\bar{\mathbf{P}}$ and of the homogenized tangent operator

$$\bar{\mathbf{L}} = \frac{\partial \bar{\mathbf{P}}}{\partial \bar{\mathbf{F}}} . \tag{40}$$

The estimation of the tangent operator $\bar{\mathbf{L}}$ is mandatory when integrating the DMN in a multiscale finite element setting as a surrogate model predicting the local stress-strain behavior. The evaluation of a DMN follows an iterative resolution with the following steps:

(i) *Downscaling*: the homogenized deformation gradient tensor $\bar{\mathbf{F}}$ is propagated downward to all the material nodes through the network interactions specified by Eq. (23), leading to the deformation gradient at each material node, which depends not only on $\bar{\mathbf{F}}$ but also on $\mathbf{A}$. The local constitutive law of each material node, Eq. (4), allows estimating the local stresses and tangent operators.

(ii) *Nonlinear system resolution*: the balance equation (27) is evaluated. If a convergence criterion is not yet satisfied, the unknown vector $\mathbf{A}$ is updated by solving the linearization of the system of equations (27) and the resolution continues with step (i). Otherwise, the step (iii) follows.

(iii) *Upscaling*: the homogenized stress $\bar{\mathbf{P}}$ and the homogenized tangent operator $\bar{\mathbf{L}}$ (if required) are computed.

These three steps are detailed in the following sections. For the ease of implementation, the bijective operators $\operatorname{vec}(\bullet)$ and $\operatorname{mat}(\bullet)$ are introduced in order to convert second-order tensors (*e.g.* $\mathbf{F}$ and $\mathbf{P}$) and fourth order tensors (*e.g.* $\mathbf{L} = \frac{\partial \mathbf{P}}{\partial \mathbf{F}}$) respectively into vectors and matrices while preserving the tensor operations through matrix-vector multiplications, see Appendix A for details.

### *3.4.1. Downscaling*

Using the unknown vector $\mathbf{A}$ defined by Eq. (39), the interaction mapping described by Eq. (23) can be rewritten for a material node $i \in \mathcal{K}$ under the vector form

$$\text{vec}\left(\mathbf{F}^i\right) = \text{vec}\left(\bar{\mathbf{F}}\right) + \mathbf{D}^i\mathbf{A} \ \forall i \in \mathcal{K}\,, \tag{41}$$

where vec $(\bullet)$ is the vector representation operator of an arbitrary second-order tensor, see Appendix A for details, and $\mathbf{D}^i$ is a row-block matrix depending only on the weights and interaction directions,

$$\mathbf{D}^i = \left[\alpha^{i,j}\mathbf{R}^j \text{ for } j = 0, \ldots, M-1\right]\,. \tag{42}$$

In the last equation, $\mathbf{R}^j$ is a $9 \times 3$ matrix constructed from $\mathbf{G}^j$ when converting a dyadic tensor product into a matrix-vector operation, *i.e.*

$$\text{vec}\left(\mathbf{a}^j \otimes \mathbf{G}^j\right) = \mathbf{R}^j\mathbf{a}^j\,, \tag{43}$$

where

$$\mathbf{R}^j = \begin{bmatrix} G_0^j & 0 & 0 & G_1^j & 0 & 0 & G_2^j & 0 & 0 \\ 0 & G_0^j & 0 & 0 & G_1^j & 0 & 0 & G_2^j & 0 \\ 0 & 0 & G_0^j & 0 & 0 & G_1^j & 0 & 0 & G_2^j \end{bmatrix}^T\,. \tag{44}$$

From the local deformation gradient estimated by Eq. (41) at each material node, the material constitutive law associated to this node as defined by Eq. (4) is used to estimate the local stress $\mathbf{P}^i$ and the internal variables $\mathbf{Z}^i$. This is achieved with an appropriate integration scheme in the time interval $[t - \Delta t\,, t]$ from the current deformation gradient $\mathbf{F}^i$ and the internal variables of the previous converged solution $\mathbf{Z}^i\left(t - \Delta t\right)$, or in other words

$$\begin{cases} \mathbf{P}^i\left(t\right) & = \hat{\mathfrak{P}}^i\left(\mathbf{F}^i\left(t\right), \mathbf{Z}^i\left(t - \Delta t\right)\right) \\ \mathbf{Z}^i\left(t\right) & = \hat{\mathfrak{Q}}^i\left(\mathbf{F}^i\left(t\right), \mathbf{Z}^i\left(t - \Delta t\right)\right) \end{cases} \forall i \in \mathcal{K}\,, \tag{45}$$

where $t$ and $\Delta t$ denote respectively the current time and time step. We assume that the local tangent operator $\mathbf{L}^i = \frac{\partial \mathbf{P}^i}{\partial \mathbf{F}^i}$ is available under the form of a closed form expression, which can be obtained by linearization of any material model.

### *3.4.2. Nonlinear system resolution*

Using the matrix $\mathbf{D}^i$ expressed in Eq. (42), the residual vector of the system of $M$ equations described by the system (27) can be rewritten as

$$\mathbf{r} = \sum_{i \in \mathcal{K}} W^i \left(\mathbf{D}^i\right)^T \text{vec}\left(\mathbf{P}^i\right)\,. \tag{46}$$

The convergence is achieved if the following condition is satisfied:

$$||\mathbf{r}||_\infty < \tau^{\text{abs}} \text{ or } ||\mathbf{r}||_\infty < \tau^{\text{rel}}||\mathbf{r}_0||_\infty \,, \tag{47}$$

where $||\bullet||_\infty$ represents the infinity norm operator, $\tau^{\text{abs}}$ and $\tau^{\text{rel}}$ are respectively the absolute and relative tolerances, and $\mathbf{r}_0$ is the initial residual. In this work, $\tau^{\text{abs}} = 10^{-12}$ and $\tau^{\text{rel}} = 10^{-6}$ are considered.

If the condition (47) is not satisfied, the unknown vector $\mathbf{A}$ is corrected with

$$\delta\mathbf{A} = -\mathbf{K}^{-1}\mathbf{r} \text{ and } \mathbf{A} \leftarrow \mathbf{A} + \delta\mathbf{A} \,, \tag{48}$$

where $\mathbf{K}$ is the Jacobian matrix

$$\mathbf{K} = \frac{\partial \mathbf{r}}{\partial \mathbf{A}} \,. \tag{49}$$

Using Eqs. (41, 45, 46), Eq. (49) can be rewritten as

$$\mathbf{K} = \sum_{i\in\mathcal{K}} W^i \left(\mathbf{D}^i\right)^T \text{mat}\left(\mathbf{L}^i\right) \mathbf{D}^i \,, \tag{50}$$

where $\text{mat}\left(\bullet\right)$ is the matrix representation operator of a fourth order tensor, see Appendix A for details.

### *3.4.3. Upscaling*

The homogenized stress tensor $\bar{\mathbf{P}}$ is computed by Eq. (20). If the DMN is considered in an iterative multiscale simulation as a constitutive law, the homogenized tangent operator $\bar{\mathbf{L}}$ also needs to be estimated following Eq. (40). Using Eqs. (20, 41), Eq. (40) yields

$$\text{mat}\left(\bar{\mathbf{L}}\right) \;=\; \frac{1}{\mathcal{S}\left(\mathcal{K}\right)} \sum_{i\in\mathcal{K}} W^i \text{mat}\left(\mathbf{L}^i\right) + \boldsymbol{\mathcal{Y}} \frac{\partial \mathbf{A}}{\partial \text{vec}\left(\bar{\mathbf{F}}\right)} \,, \tag{51}$$

where

$$\boldsymbol{\mathcal{Y}} = \frac{1}{\mathcal{S}\left(\mathcal{K}\right)} \sum_{i\in\mathcal{K}} W^i \text{mat}\left(\mathbf{L}^i\right) \mathbf{D}^i \,. \tag{52}$$

In order to estimate $\bar{\mathbf{L}}$ following Eq. (51), the matrix $\frac{\partial \mathbf{A}}{\partial \text{vec}\left(\bar{\mathbf{F}}\right)}$ needs to be computed. To this end, the consistency of Eq. (46) is expressed as

$$\delta\mathbf{r} = \sum_{i\in\mathcal{K}} W^i \left(\mathbf{D}^i\right)^T \text{mat}\left(\mathbf{L}^i\right) \left(\text{vec}\left(\delta\bar{\mathbf{F}}\right) + \mathbf{D}^i \delta\mathbf{A}\right) = \mathbf{0} \,. \tag{53}$$

This last equation allows computing

$$\frac{\partial \mathbf{A}}{\partial \text{vec}\left(\bar{\mathbf{F}}\right)} = -\mathbf{K}^{-1}\boldsymbol{\mathcal{X}} \,, \tag{54}$$

where $\mathbf{K}$ is given by Eq. (50) and

$$\boldsymbol{\mathcal{X}} = \sum_{i\in\mathcal{K}} W^i \left(\mathbf{D}^i\right)^T \operatorname{mat}\left(\mathbf{L}^i\right) \, . \tag{55}$$

Equation (54) is a linear system of multiple-right hand side and the matrix $\mathbf{K}$ was previously used to solve the system of Eq. (48). Since the solution of Eq. (48) often follows a direct factorization procedure for $\mathbf{K}$, the resolution of the multiple-right hand side system (54) is then performed using this factorized matrix at a reduced computational time [38].

Equation (51) can be rewritten under the form of a Schur complement as

$$\operatorname{mat}\left(\bar{\mathbf{L}}\right) = \boldsymbol{\mathcal{J}} - \boldsymbol{\mathcal{Y}}\mathbf{K}^{-1}\boldsymbol{\mathcal{X}} \, , \tag{56}$$

where $\boldsymbol{\mathcal{J}} = \frac{1}{\mathcal{S}(\mathcal{K})}\sum_{i\in\mathcal{K}} W^i \operatorname{mat}\left(\mathbf{L}^i\right)$. If the tangent operators of all material nodes are symmetric, *i.e.* $\left(\operatorname{mat}\left(\mathbf{L}^i\right)\right)^T = \operatorname{mat}\left(\mathbf{L}^i\right) \ \forall i \in \mathcal{K}$, which is the case for an elastoplastic model relying on an elastic potential and an associative plastic flow rule, one has

$$\boldsymbol{\mathcal{Y}} = \frac{1}{\mathcal{S}\left(\mathcal{K}\right)}\boldsymbol{\mathcal{X}}^T \text{ and } \operatorname{mat}\left(\bar{\mathbf{L}}\right) = \boldsymbol{\mathcal{J}} - \frac{1}{\mathcal{S}\left(\mathcal{K}\right)}\boldsymbol{\mathcal{X}}^T\mathbf{K}^{-1}\boldsymbol{\mathcal{X}} \, , \tag{57}$$

implying that the tangent operator $\operatorname{mat}\left(\bar{\mathbf{L}}\right)$ is also symmetric. In the general case, the tangent operator $\operatorname{mat}\left(\mathbf{L}^i\right)$ is not necessarily symmetric and the symmetry of $\operatorname{mat}\left(\bar{\mathbf{L}}\right)$ is thus not guaranteed.

### *3.4.4. Summary*

For a material network $\mathcal{M}$ indexed in $\mathcal{K}$, and assuming its weights and its network interactions $\boldsymbol{\mathcal{H}}$ have been determined, the resolution framework for estimating its response is summarized as follows:

(I) Initialization:

(i) assemble $\mathbf{A}$ following Eq. (39);
(ii) initialize $\mathbf{A} = \mathbf{0}$;
(iii) assemble $\mathbf{D}^i$ $\forall i \in \mathcal{K}$ following Eq. (42).

(II) Evaluation at time $t$ with a time step $\Delta t$ for the current value of $\bar{\mathbf{F}}\left(t\right)$, and knowing the values of $\mathbf{A}\left(t-\Delta t\right)$ and the history data at material nodes $\mathbf{Z}^i\left(t-\Delta t\right)$, $\forall i \in \mathcal{K}$, at the previous converged solution (at time $t-\Delta t$):

(i) initialize $\mathbf{A}(t) = \mathbf{A}\left(t-\Delta t\right)$ and $\mathbf{Z}^i\left(t\right) = \mathbf{Z}^i\left(t-\Delta t\right)$, $\forall i \in \mathcal{K}$;
(ii) downscale $\bar{\mathbf{F}}(t)$ and $\mathbf{A}(t)$ following Eq. (41);
(iii) evaluate the local constitutive law at all material nodes following Eq. (45), yielding $\mathbf{P}^i\left(t\right)$, $\mathbf{Z}^i\left(t\right)$, and $\mathbf{L}^i\left(t\right)$, $\forall i \in \mathcal{K}$;
(iv) evaluate the residual $\mathbf{r}$ following Eq. (46);
(v) **if** the convergence criterion following Eq. (47) is achieved **go to** (viii); **else go to** (vi);
(vi) correct the value of $\mathbf{A}(t)$ following Eq. (48);

(vii) go to (ii);

(viii) compute the homogenized stress $\bar{\mathbf{P}}(t)$ following Eq. (20);

(ix) compute the homogenized tangent operator $\bar{\mathbf{L}}(t)$ following Eq. (51) if required;

(x) store $\mathbf{A}(t)$ and history data $\mathbf{Z}^i(t)$ at all material nodes for the next step;

(xi) exit.

## 4. Machine learning algorithm for tuning the network parameters

In Section 3, we have provided the theory of the DMN from the network interaction viewpoint, assuming that the fitting parameters consisting of the weights and network interaction parameters were known. In order to make a DMN become a surrogate of the full-field resolution of the homogenization problem described in Section 2, the DMN must be trained to identify these fitting parameters.

The original homogenization model described in Section 2 is defined by i) a microstructure topology, ii) the microscopic boundary condition on the RVE, and iii) the constitutive laws of the underlying constituents. Since the theory of DMNs focuses on the topological learning, the first two aspects, i) and ii), of the homogenization problem can be replaced by the network of discrete material nodes with their interactions. In other words, the microstructure and microscopic boundary condition in the full-field model are represented by the fitting parameters of the DMN, and as a result the latter can be identified by an offline training. This work follows the training strategy proposed in [29], in which it was shown that it is enough to consider the homogenized material tensor, at zero-strain, in order to train the DMN. To be complete, the point iii) of the homogenization problem, *i.e.* the constitutive laws of the underlying constituents, is explicitly accounted for by the DMN during the online simulations by using the same constitutive models in the corresponding DMN material nodes.

As a result, the fitting parameters result from an optimization process conducted using machine learning algorithms with a training dataset consisting of the homogenized material tensors at zero-strain obtained by full-field simulations. This training dataset is obtained from offline simulations with the full-field model, in which the different constituents are characterized by their own material tensor. These constituent material tensors are not physical but artificially and randomly generated in order to form a large dataset and are considered to be orthotropic in order to reduce the dimension of the sampling space and to ease the random generation process [29, 30]. Another possibility based on a rank-one perturbation was considered in [39], in which the structure of the stiffness matrices encompasses the possible algorithmic tangents of $J_2$-elastoplasticity.

Although the training framework of two-phase composites can be found in [29, 32], the training procedure is detailed in this section in the case of an arbitrary number of physical phases and of an arbitrary DMN architecture. A complete training procedure consists of the followings steps:

- First, the fitting parameters of the DMN are explicated.
- Then, the procedure to build a training dataset is summarized.

- Next, a fast algorithm is proposed to estimate, not only the homogenized material tensor of the DMN, but also its derivatives with respect to the fitting parameters. The latter allows considering an arbitrary gradient-based optimizer for the training process.
- Finally, the training procedure based on a simple adaptive scheme of the stochastic gradient descent is provided to minimize a cost function defined on the training dataset.

These steps are detailed as follows.

### *4.1. Fitting parameters*

Let us consider the DMN $\mathcal{M}$ of $N$ material nodes $\mathcal{K}$ as described in Section 3. Its fitting parameters consist of the weights of the material nodes $W^i$, $\forall i \in \mathcal{K}$, and of the interaction directions $\boldsymbol{\mathcal{N}}$ defined by $L$ unit normal vectors $\mathbf{N}^m$ with $m = 0, \dots, L-1$. However, the positiveness of the weights, *i.e.* $W^i > 0$, $\forall i \in \mathcal{K}$, and the unit length nature of the interactions directions, *i.e.* $\mathbf{N}^m \cdot \mathbf{N}^m = 1$ with $m = 0, \dots, L-1$, have to be enforced during the training process, implying that a constrained minimization problem needs to be considered. In order to avoid this constrained minimization, and thus to ease the training process that relies on a gradient descent algorithm, *ad hoc* changes of variables are considered. In particular, each weight of a material node is considered as a function whose image is in the positive range, and each interactions direction is expressed through an angle parametrization.

First, the condition $W^i > 0$, $\forall i \in \mathcal{K}$, must be satisfied in order to obtain a valid contribution of that material node into the network. To enforce this condition, the rectified linear unit (abbreviated by "relu") was considered in [29]. However, the use of relu leads to the "death" of nodes with a gradient-based training algorithm, *i.e.* when a weight becomes negative, its gradient vanishes and this node is never reactivated again. To avoid dead nodes, this work considers the smoothed version of relu, abbreviated by "$\mathrm{relu}^{\mathrm{s}}(\bullet)$", as

$$y = \frac{1}{s} \ln\left(1 + \mathrm{e}^{sx}\right) = \mathrm{relu}^{\mathrm{s}}(x) \ , \tag{58}$$

where $s$ is the sharpness, see Fig. 4 for the influence of $s$ on $\mathrm{relu}^{\mathrm{s}}(\bullet)$. When $s \to \infty$, the relu activation function is recovered. In the remaining of the paper, the value $s = 10$ is used. As a result, a weight $W^i$ is obtained through a function of a fitting parameter $z^i$ as

$$W^i = \mathrm{relu}^{\mathrm{s}}\left(z^i\right) \ , \ \forall i \in \mathcal{K} \, . \tag{59}$$

A positive value of $W^i$ is always guaranteed since the output of $\mathrm{relu}^{\mathrm{s}}(\bullet)$ is always positive.

Then, since any unit vector can be expressed in terms of either the polar angle in a bidimensional state or the spherical angles in a general 3-dimensional state, the vector $\mathbf{N}^m$ is rewritten by

$$\mathbf{N}^m = \begin{cases} \begin{bmatrix} \cos\left(2\pi\varphi^m\right) & \sin\left(2\pi\varphi^m\right) & 0 \end{bmatrix}^T & \text{in plane strain}\, , \\ \begin{bmatrix} \cos\left(2\pi\varphi^m\right)\sin\left(\pi\theta^m\right) & \sin\left(2\pi\varphi^m\right)\sin\left(\pi\theta^m\right) & \cos\left(\pi\theta^m\right) \end{bmatrix}^T & \text{otherwise}\, , \end{cases} \tag{60}$$

with $[\varphi^m \ \theta^m] \in [0 \ 1) \times [0 \ 1]$ being normalized angles. The relationship above allows defining

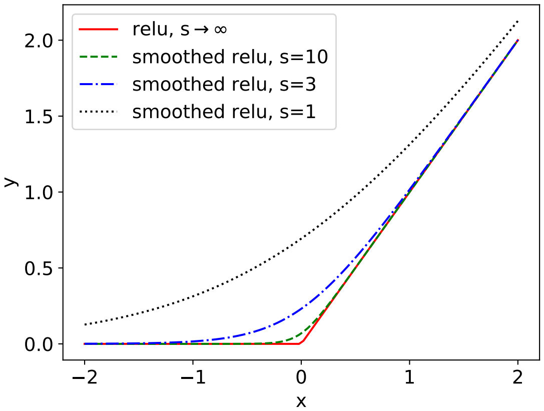


Figure 4: Activation function.

the operator angle $(\bullet)$ by

$$\text{angle}\,(\mathbf{N}) = \begin{cases} [\varphi] & \text{in plane strain}\,, \\ [\varphi\ \theta]^T & \text{otherwise}\,, \end{cases} \tag{61}$$

for any arbitrary unit vector $\mathbf{N}$. As a result, these normalized angles are considered as the tuning parameters instead of directly using the interaction directions.

Finally, all the fitting parameters of the DMN are then assembled in a vector $\boldsymbol{\mathcal{F}}$ as

$$\boldsymbol{\mathcal{F}} = \left[z^i\ \forall i \in \mathcal{K} \quad (\text{angle}\,(\mathbf{N}))^T\ \ \forall \mathbf{N} \in \boldsymbol{\mathcal{N}}\right]^T . \tag{62}$$

### *4.2. Dataset for training*

Considering a composite material of $P$ physical phases ($P \geq 2$), the constitutive law of each material node corresponds to one these $P$ phases. Assuming that at the training stage, the response of each material node is characterized by a material tensor $\mathbb{C}^{(p)}$ with $p$ chosen among $0, \ldots, P-1$, the response of the whole network is characterized by a homogenized material tensor, denoted by $\bar{\mathbb{C}}$, whose explicit form is given by Eq. (51). As a result, one has the following relation

$$\underbrace{\bar{\mathbb{C}}}_{\text{output}} = \bar{\mathbb{C}}\left(\underbrace{\mathbb{C}^{(0)}, \ldots, \mathbb{C}^{(P-1)}}_{\text{input}};\quad \underbrace{\boldsymbol{\mathcal{F}}}_{\text{fitting parameters}}\right) , \tag{63}$$

where the fitting parameters vector $\boldsymbol{\mathcal{F}}$ is described by Eq. (62). In order to identify $\boldsymbol{\mathcal{F}}$ for a given microstructure $V_0$, a training dataset of input-output pairs must be created. Following the expression (63), one has

- input: $\mathbb{X} = \left[\mathbb{C}^{(0)}, \ldots, \mathbb{C}^{(P-1)}\right]$; and
- output: $\mathbb{Y} = \bar{\mathbb{C}}$.

Since Eq. (63) is viewed as a universal input-output relation, the values $\mathbb{C}^{(0)}, \ldots, \mathbb{C}^{(P-1)}$ do not need to be the physical ones of the physical phases following Eq. (4). For this purpose, following [29], $N_s$ samples of the input, *i.e.* $\mathbb{X}_s$ with $s = 0, \ldots, N_s - 1$, are first randomly generated. The number of elements $N_s$ required in the training dataset to train the DMN depends on the number of fitting parameters, and consequently, depends on the complexity of the microstructure. A more complex microstructure requires a higher number of samples $N_s$. In numerical applications considered in Section 5, the value of $N_s$ is specified for each microstructure. Then the corresponding outputs, *i.e.* $\mathbb{Y}_s^{\text{DNS}}$ with $s = 0, \ldots, N_s - 1$, are obtained as the homogenized tangent operators of a microscopic boundary value problem defined over $V_0$ using the periodic boundary condition, see [38] for implementation details.

In order to take advantage of the symmetric property of the material tensor $\mathbb{C}$, a bijective operator $[\bullet]$ is introduced to convert a fourth-order material tensor into a symmetric matrix following Voigt's notations, we refer to Eq. (B.4) in Appendix B.1 for details. Using this operator, a fourth order tensor $\mathbb{C}$ becomes either a $3 \times 3$ symmetric matrix in plane strain state or a $6 \times 6$ symmetric matrix in a general three-dimensional state. The generation of the training dataset is summarized as follows:

(I) initialize the full-field computational homogenization model:

  (i) create the finite element mesh of the microstructure with $P$ phases;

  (ii) define the periodic boundary condition;

(II) **for** $s$ **from** $0$ **to** $N_s - 1$:

  (i) generate randomly $P$ material tensors $\mathbb{C}^{(0)}, \ldots, \mathbb{C}^{(P-1)}$ leading to an input sample $\mathbb{X}_s$ of the $P$ phase material tensors of the microstructure. The generation procedure for general 3-dimensional problems can be found in [30], in which $P$ material tensors are generated through randomly generating the Young's moduli and Poison ratios of an arbitrary orthotropic material tensor. The particular plane strain state is reported in Appendix B.2;

  (ii) assign these $P$ random material tensors to the $P$ phases in the microstructure;

  (iii) compute the corresponding homogenization material tensor $\mathbb{Y}_s^{\text{DNS}} = \bar{\mathbb{C}}^{\text{DNS}}$ using the full-field computational homogenization model. The computation details can be found in [38];

  (iv) append the pair $\left(\mathbb{X}_s, \mathbb{Y}_s^{\text{DNS}}\right)$ in the training dataset.

### *4.3. Efficient evaluation of the homogenized material tensor for DMN training*

In order to accelerate the training process, an efficient evaluation of Eq. (63) is necessary since this function must be evaluated at each sample of the training dataset. Theoretically, the DMN homogenized material tensor $\bar{\mathbb{C}}$ can be directly computed by Eq. (51). However, the computation of the derivative of $\bar{\mathbb{C}}$ with respect to the fitting parameters $\boldsymbol{\mathcal{F}}$ required for a gradient-based training algorithm needs a considerable computation time because of the large number of operations. For a fast evaluation, a recursive strategy was considered in [29] for a perfect binary tree structure, in which the homogenized behavior is obtained by recursively applying depth-by-depth the analytical homogenization solution of a two-phase composite from its deepest depth. This work follows a similar strategy with the analytic solution of each network interaction, but generalizes it in order to estimate the material tensor of an arbitrary tree architecture.

Let us consider a DMN $\mathcal{M}$ of a list of material nodes indexed in $\mathcal{K}$ and with the fitting parameters $\boldsymbol{\mathcal{F}}$ and let us consider an arbitrary sub-list $\mathcal{K}_d^p$ located at a depth $d$ and position $p$ in the DMN tree structure as illustrated in Fig. 2. Since $\mathcal{K}_d^p$ is also a list of indexes of a material network, one can define the operator $\bar{\mathbb{C}}\left(\mathcal{K}_d^p;\boldsymbol{\mathcal{F}}\right)$ to indicate the homogenized material tensor of $\mathcal{K}_d^p$ for a given $\boldsymbol{\mathcal{F}}$. In order to estimate $\bar{\mathbb{C}}\left(\mathcal{K}_d^p;\boldsymbol{\mathcal{F}}\right)$, one can distinguish two possibilities:

- If $\mathcal{K}_d^p$ contains only one material node $i$, one directly has

  $$\bar{\mathbb{C}}\left(\mathcal{K}_d^p;\boldsymbol{\mathcal{F}}\right) = \mathbb{C}^i\,, \tag{64}$$

  where $\mathbb{C}^i$ is the material tensor at that material node $i$.

- If $\mathcal{K}_d^p$ contains more than one material node, there exists a network interaction $m$ whose interaction direction is $\mathbf{N}^m$ and which is created from $n_d^p$ ($n_d^p \geq 2$) non-overlapping sub-lists $\mathcal{K}_{d+1}^{r_m+k}$ with $k = 0\,,\ldots\,,n_d^p - 1$ and $r_m = \sum_{k=0}^{p-1} n_d^k$ being the location of the first sub-list of this interaction at the depth $d+1$. One has thus

  $$\mathcal{K}_d^p = \mathcal{K}_{d+1}^{r_m} \cup \ldots \cup \mathcal{K}_{d+1}^{r_m+n_d^p-1}\,. \tag{65}$$

  This network interaction leads to the following relation

  $$\left[\bar{\mathbb{C}}_d^p\right] = \mathbb{H}_{n_d^p}\left(\left[\bar{\mathbb{C}}_{d+1}^{r_m}\right], \ldots, \left[\bar{\mathbb{C}}_{d+1}^{r_m+n_d^p-1}\right], f_d^{p,0}, \ldots, f_d^{p,n_d^p-1}, \mathbf{N}^m\right)\,. \tag{66}$$

  In this last equation, the notation $\bar{\mathbb{C}}\left(\mathcal{K}_r^s;\boldsymbol{\mathcal{F}}\right)$ is replaced by $\bar{\mathbb{C}}_r^s$ for an arbitrary material network $\mathcal{K}_r^s$ for simplicity; the bijective operator $[\bullet]$ is introduced to convert this symmetric fourth-order tensor into a symmetric matrix following Voigt's notations, see Appendix B.1 for details; $\mathbb{H}_{n_d^p}$ is a functional operator used to refer the case of $n_d^p$ sub-lists; and

  $$f_d^{p,k} = \frac{\mathcal{S}\left(\mathcal{K}_{d+1}^{r_m+k}\right)}{\mathcal{S}\left(\mathcal{K}_d^p\right)} \text{ for } k = 0\,,\ldots\,,n_d^p - 1\,. \tag{67}$$

  The analytical evaluation of Eq. (66) is provided in Appendix B.3.

The homogenized material tensor can be obtained as

$$\bar{\mathbb{C}} = \bar{\mathbb{C}}\left(\mathcal{K};\boldsymbol{\mathcal{F}}\right) \equiv \bar{\mathbb{C}}_0^0\,, \tag{68}$$

by using recursively depth-by-depth Eq. (66) until all the sub-lists contain only one material node, for which the material tensor is known. With this process, not only the value of $\bar{\mathbb{C}}$ but also its derivative $\partial\bar{\mathbb{C}}/\partial\boldsymbol{\mathcal{F}}$ can be estimated, see the details reported in Appendix B.4.

### *4.4. Offline training procedure*

Considering $N_s$ input-output pairs serving as the training dataset described in Section 4.2, the fitting parameters identified in Section 4.1 are obtained though an optimization process. A gradient descent-based method can be found in [29] in which a large number of training steps are needed in order to reach the required accuracy. In this work, a simpler and faster training algorithm using an adaptive scheme of the stochastic gradient descent (SGD) is investigated.

*4.4.1. Cost function and its gradient with respect to the fitting parameters*

For each data point $(\mathbb{X}_s, \mathbb{Y}_s^{\text{DNS}})$, with $\mathbb{X}_s = \left[\left[\mathbb{C}_s^{(0)}\right], \ \ldots \ \left[\mathbb{C}_s^{(P-1)}\right]\right]$ and $\mathbb{Y}_s^{\text{DNS}} = \left[\bar{\mathbb{C}}_s^{\text{DNS}}\right]$, in the training dataset, a relative error between the results predicted by the DNS and by the DMN is measured by

$$C_s\left(\boldsymbol{\mathcal{F}}\right) = \frac{\|\left[\bar{\mathbb{C}}_s^{\text{DNS}}\right] - \left[\bar{\mathbb{C}}_s^{\text{DMN}}\right]\|_{\text{F}}}{\|\left[\bar{\mathbb{C}}_s^{\text{DNS}}\right]\|_{\text{F}}}\,, \tag{69}$$

where $\left[\bar{\mathbb{C}}_s^{\text{DMN}}\right] = \left[\bar{\mathbb{C}}_s^{\text{DMN}}\right]\left(\mathbb{X}_s; \boldsymbol{\mathcal{F}}\right)$ is obtained using Eq. (63) and $\| \bullet \|_{\text{F}}$ represents the Frobenius norm. The cost function is defined over the training dataset as

$$C^g\left(\boldsymbol{\mathcal{F}}\right) = \frac{1}{N_s}\sum_{s=0}^{N_s-1} C_s\left(\boldsymbol{\mathcal{F}}\right)\,, \tag{70}$$

where $N_s$ is the size of the training dataset. To investigate the overfitting, a validation set consisting of $N_v$ samples is also created. The prediction error in this validation set is also estimated using Eq. (70).

The gradient of the error (69) with respect to the tuning parameters must be estimated in the context of the gradient descent algorithm. One has

$$\mathbf{g}_s = \frac{\partial C_s}{\partial \boldsymbol{\mathcal{F}}} = \left(\frac{1}{\|\left[\bar{\mathbb{C}}_s^{\text{DNS}}\right]\|_{\text{F}}}\,\frac{\left[\bar{\mathbb{C}}_s^{\text{DNS}}\right] - \left[\bar{\mathbb{C}}_s^{\text{DMN}}\right]}{\|\left[\bar{\mathbb{C}}_s^{\text{DNS}}\right] - \left[\bar{\mathbb{C}}_s^{\text{DMN}}\right]\|_{\text{F}}}\right) \circ \frac{\partial\left[\bar{\mathbb{C}}_s^{\text{DMN}}\right]}{\partial \boldsymbol{\mathcal{F}}}\,, \tag{71}$$

where $\frac{\partial\left[\bar{\mathbb{C}}_s^{\text{DMN}}\right]}{\partial \boldsymbol{\mathcal{F}}}$ is a block vector whose components are matrices with the dimension of $\left[\bar{\mathbb{C}}_s^{\text{DMN}}\right]$ and $\circ$ represents the element-wise product of the previous matrix with each component of the block vector. Using the procedure described in Section 4.3, the term $\frac{\partial\left[\bar{\mathbb{C}}_s^{\text{DMN}}\right]}{\partial \boldsymbol{\mathcal{F}}}$ required in Eq. (71) is easily computed, see Appendix B.4.

*4.4.2. Fast training algorithm with stochastic gradient descent*

In this training algorithm, the gradient of the cost function (70) is approximated by the gradient error on a single sample and the learning rate is adapted in order to reduce the loss function. The input parameters of the training algorithm includes

- the maximum number of training steps $N_{\max}$;
- the maximum and minimum learning rates $\eta_{\max}$ and $\eta_{\min}$; and
- the learning rate adaptive factor $\kappa$.

The training algorithm is detailed as follows:

(I) initialization:

- (i) initialize the values of $\boldsymbol{\mathcal{F}}_0$ following uniform distributions, yielding

  $$\boldsymbol{\mathcal{F}}_0 = \left[z^i \in U\left(0.2, 0.8\right)\ \forall i \in \mathcal{K} \quad \left(\text{angle}\left(\mathbf{N}\right)\right)^T \in \left[U\left(0,1\right)\right]^d\ \forall \mathbf{N} \in \boldsymbol{\mathcal{N}}\right]^T , \tag{72}$$

  where $U\left(a, b\right)$ stands for a uniform distribution in the range $[a, b]$, and $d$ is the number of components in angle $\left(\mathbf{N}\right)$; the values of $z^i$, $\forall i \in \mathcal{K}$, are normalized with the total weights $\mathcal{S}\left(\mathcal{K}\right)$ as

  $$z^i \leftarrow \frac{z^i}{\mathcal{S}\left(\mathcal{K}\right)} , \ \forall i \in \mathcal{K} ; \tag{73}$$

- (ii) compute the initial cost function $C^g_0$ using Eq. (70);
- (iii) initialize the learning rate $\eta = \eta_{\text{max}}$;

(II) **for** $l$ **from** 0 **to** $N_{\text{max}} - 1$:

- (i) initialize $\boldsymbol{\mathcal{F}}_{l+1} = \boldsymbol{\mathcal{F}}_l$;
- (ii) apply the stochastic gradient descent algorithm on the $N_s$ training samples:
  **for** $s$ **from** 0 **to** $N_s - 1$:
  - (a) compute the gradient $\mathbf{g}_s$ of the sample $s$ with respect to $\boldsymbol{\mathcal{F}}_{l+1}$ using Eq. (71);
  - (b) update $\boldsymbol{\mathcal{F}}_{l+1}$ using the gradient $\mathbf{g}_s$ and the learning rate $\eta$ following

    $$\boldsymbol{\mathcal{F}}_{l+1} \leftarrow \boldsymbol{\mathcal{F}}_{l+1} - \eta \mathbf{g}_s \, . \tag{74}$$

- (iii) compute the cost function $C^g_{l+1}$ using Eq. (70);
- (iv) compute the validation loss using Eq. (70) with the validation dataset;
- (v) check convergence: **if** an early stopping criterion exists and **if** it is satisfied: set $N_{\text{max}} = l + 1$ and **go to** (III);
- (vi) check $C^g_{l+1} \leq C^g_l$; **if** true **go to** (viii);
- (vii) modify the learning rate $\eta = \max(\kappa\eta, \eta_{\text{min}})$ and **go to** (i);
- (viii) go to next training step: $l \leftarrow l + 1$;

(III) exit.

In this work, we use $\eta_{\text{min}} = 10^{-4}$, $\eta_{\text{max}} = 1$ and $\kappa = 0.8$. The value $N_{\text{max}}$ can vary depending on the required training error and can be adapted on the fly by defining an early stopping criterion. Since the gradient of the loss function is easily computed, it is always possible to consider an arbitrary gradient-based optimizer, see *e.g.* [40] for different possibilities. Anyway, the stochastic gradient descent optimizer works well in this current framework as demonstrated through different examples in the next section. Moreover, the learning rate adaptation allows avoiding oscillations although a large initial value of the learning rate, *i.e.* $\eta_{\text{max}} = 1$, is used. The learning rate decay factor $\kappa$ can be arbitrarily chosen smaller than 1, but should not be too small in order to avoid a large drop in the learning rate.

## 5. Numerical applications

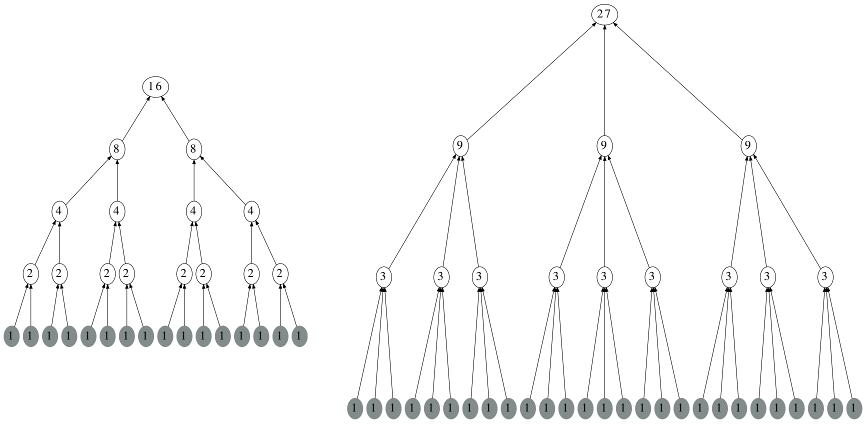
### *5.1. Material network architecture*

Although this work can employ an arbitrary tree structure, only the following two types will be investigated.

Firstly, we consider a *perfect tree* denoted by

$$T^{\text{perfect}}\left(n, d_{\max}\right) \,, \tag{75}$$

in which $n \geq 2$ denotes the number of sub-lists in all the network interactions and $d_{\max} \geq 2$ is the deepest depth of the tree. The value of $n^{d_{\max}}$ represents the total number of material nodes. With $P$ being the number of physical phases, the condition $P \geq n \geq 2$ must be satisfied in order to avoid the interactions of the material nodes with the same material behavior at the first level of interactions. This condition implies that an arbitrary perfect $n$-tree can be used for any $P \geq n$. Figure 5 sketches the architectures of $T^{\text{perfect}}\left(2, 4\right)$ and $T^{\text{perfect}}\left(3, 3\right)$ as examples of perfect trees.



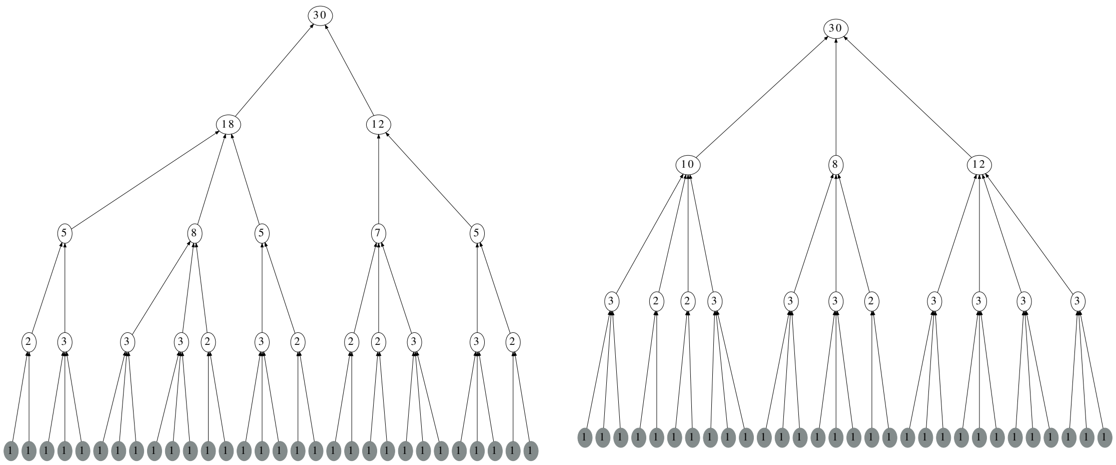
(a) $T^{\text{perfect}}\left(2, 4\right)$ (b) $T^{\text{perfect}}\left(3, 3\right)$

Figure 5: Examples of a perfect tree: (a) $T^{\text{perfect}}\left(2, 4\right)$ and (b) $T^{\text{perfect}}\left(3, 3\right)$.

Secondly, we consider a *random tree* denoted by

$$T^{\text{rand}}\left(N, n^{\max,1}, n^{\max}\right) \,, \tag{76}$$

in which $N$ is the number of material nodes, $n^{\max,\,1}$ is the maximal number of the sub-lists at the first level of the network interaction, and $n^{\max}$ is the corresponding value at the other levels. The minimum number of sub-lists in a network interaction is equal to 2. Consequently, the number of the sub material networks in each network interaction is randomly taken: (i) between 2 and its maximum value $n^{\max,\,1}$ for the first level of interaction and (ii) between 2 and $n^{\max}$ for other levels. The two parameters $n^{\max,\,1}$ and $n^{\max}$ are considered in order to discriminate the interaction conditions at the first level and at the other levels. Indeed, at the first level, the condition $2 \leq n^{\max,\,1} \leq P$, with $P$ being the number of physical phases, must be satisfied in order to avoid defining an interaction between two material nodes obeying to the same material model. At a higher level, the value of $n^{\max}$ is only constrained by the condition $n^{\max} \geq 2$. Since the tree architecture is totally random, different architectures can be obtained with the same parameters $N$, $n^{\max,1}$, and $n^{\max}$. Figure 6 shows two different realizations of $T^{\text{rand}}(30, 3, 4)$. There exist many other possibilities to create a random tree but this is beyond the scope of the current work.



(a) $T^{\text{rand}}(30, 3, 4)$, $1^{st}$ realization

(b) $T^{\text{rand}}(30, 3, 4)$, $2^{nd}$ realization

Figure 6: Examples of random trees: (a) $T^{\text{rand}}(30, 3, 4)$, $1^{st}$ realization and (b) $T^{\text{rand}}(30, 3, 4)$, $2^{nd}$ realization.

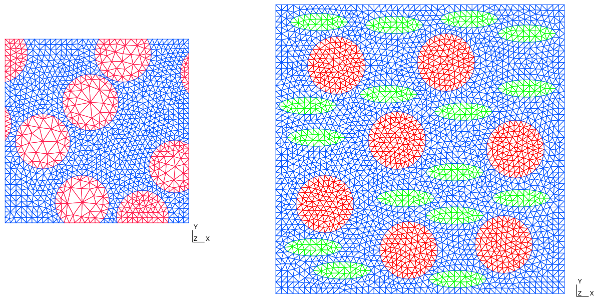
Once the tree structure is available, the constitutive laws of the different constituents can be assigned to the material nodes as previously discussed in Section 3.3. Both perfect and random material networks are considered in this section to represent different fiber-reinforced composites. To identify the fitting parameters of these material networks, the offline training requires only the homogenized material tensor at zero-strain of the network, as explained in Section 4. The offline training dataset is artificially and randomly generated by considering random material tensors for the material phases. During the online predictions, the complete and physical material models of the microstructure constituents are considered. The matrix behavior is governed by a finite strain $J_2$ elastoplastic model, see Appendix C for details, while the fibers obey a finite strain bi-logarithmic elastic law, which corresponds to the finite strain $J_2$ elastoplastic model without plasticity[2]. To obtain the offline training data and the reference solutions of the online predictions for comparison purpose, direct numerical simulations (DNS) are performed using the resolution strategy of the constrained micro-scale finite element problem as detailed in [38]. The periodic boundary condition is considered for a general mesh setting following the interpolation method [37]. All the processes, including both the offline training and the online predictions, are performed on a single cpu.

[2] A very high value of the initial yield stress governing the onset of the plastic stage is considered.

### *5.2. Offline training for various microstructures*

The material networks are trained for the unidirectional fiber-reinforced composite materials (abbreviated by UDC material) in the cases of two-phase and three-phase materials under a plane strain state and for short fiber-reinforced composite materials (abbreviated by SFC material) in the case of a two-phase material under a general three dimensional state. The number of epochs for training is first set to a value of 50 and then extended by a value of 50 each time. The training stops when the loss is not much improved.

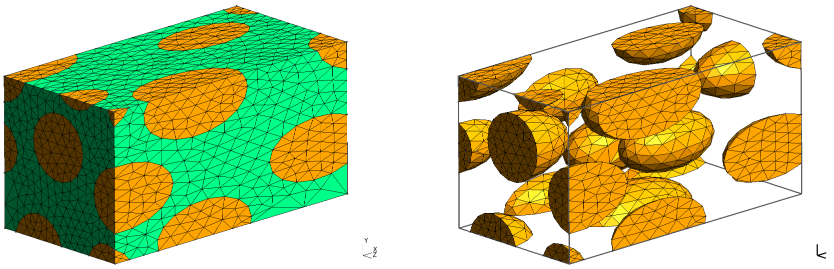
The finite element meshes of the two-phase and three-phase UDC microstructures are shown in Figs. 7(a) and (b), respectively. For the two-phase microstructure, see Fig. 7(a), the volume fraction of the fibers is equal to 39.9% and the finite element mesh consists of 2237 six-node triangular elements. For the three-phase microstructure in which round and elliptic fibers are present as two different fiber types, see Fig. 7(b), the volume fractions of the round and elliptic fibers are equal to 22% and 15.1%, respectively and the finite mesh consists of 5502 six-node triangular elements.



(a) (b)

Figure 7: UDC material - investigated microstructures: (a) two phases (39.9% fibers) and (b) three phases (20% round fibers and 15.1% elliptic fibers).

The finite element mesh of the SFC microstructure is shown in Fig. 8, in which the ellipsoid fibers are embedded in a matrix. The volume fraction of the fiber is equal to 20.66% and the finite element mesh consists of 12724 uadratic tetrahedral elements.



(a) (b)

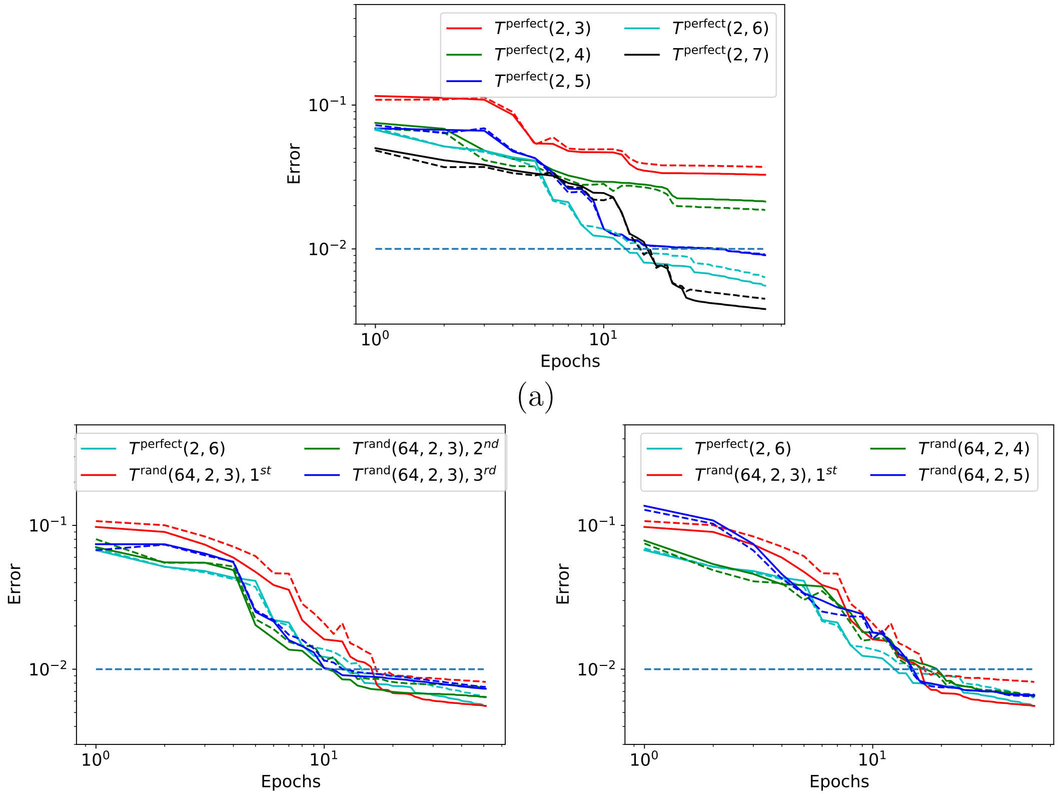
Figure 8: SFC material - investigated microstructure: (a) overall view and (b) fibers only.

The use of the non-periodic microstructures in Figs. 7(a) and Fig. 8 in this study is motivated by the fact that the spatial arrangement of fibers in a real composite microstructure is normally not periodic. This corresponds to the case encountered when the microstructure is directly extracted from real materials, *e.g.* via the Scanning Electron Microscope (SEM) [41].

### 5.2.1. Two-phase UDC microstructure

For the two-phase UDC microstructure, see Fig. 7(a), a dataset of 300 samples is generated for the offline training. This data is divided into a training dataset consisting of 200 samples and a validation dataset consisting of 100 samples.

Histories of the average training and validation errors are reported in Fig. 9 using 50 epochs with different network architectures. Figure 9(a) represents the results with the perfect binary trees. When the number of material nodes reaches $2^5 = 32$, an average error smaller than 1% is obtained. Figure 9(b) represents the results with three different realizations of the random tree $T^{\text{rand}}(64, 2, 3)$. The average error after 50 epochs is also smaller than 1% for all these three cases showing that a number of 64 material nodes is high enough for a good prediction. Average errors smaller than 1% are also obtained with different random architectures of 64 material nodes as shown in Fig. 9(c). When considering material networks with a limited number of material nodes, *e.g.* $T^{\text{perfect}}(2, 3)$ and $T^{\text{perfect}}(2, 4)$ in Fig. 9(a), the average training and validation errors stabilize with relatively high values.



(a)

(b)

(c)

Figure 9: Two-phase UDC microstructure - training histories under plane strain state after 50 epochs: (a) perfect trees and (b, c) random trees as compared to $T^{\text{perfect}}(2, 6)$. Continuous and dashed lines represent respectively the average training and validation errors. The 1%-error line is also reported for comparison purpose.

The training histories are also summarized in Tab. 1. The reached fiber volume faction is very close to the value of 39.9% of the microstructure used for training. Moreover, the training time is relatively small. The most time consuming case corresponds to the $\mathcal{T}^{\text{perfect}}(2,7)$ tree and is only 74 seconds.

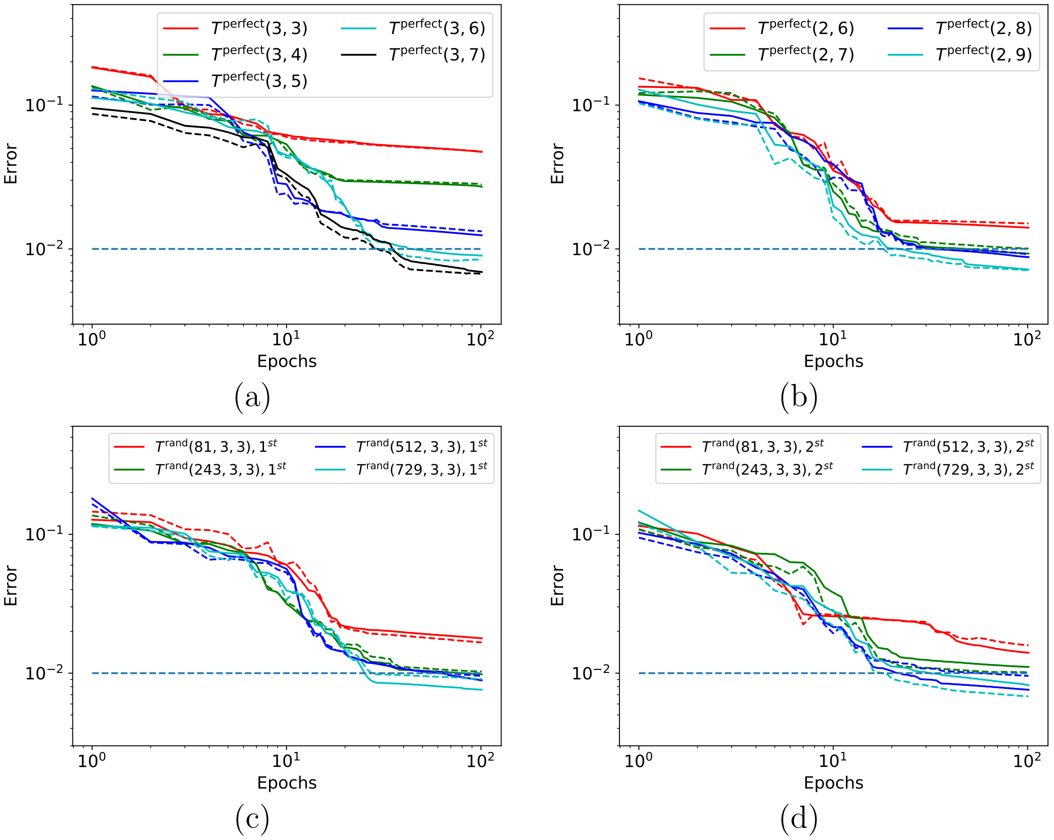
Table 1: Two-phase UDC microstructure - training results for different DMN architectures. The number of epochs $N_{\max}$, number of fitting parameters $N_f$, training time $T$, averaging training error $C^{g,\,\text{train}}$, averaging validation error $C^{g,\,\text{test}}$, and the predicted fiber volume fraction $V_f$ of the DMN are reported.

| DMN | $N_{\max}$ | $N_f$ | $T$ | $C^{g,\,\text{train}}$ | $C^{g,\,\text{test}}$ | $V_f$ |
|---|---|---|---|---|---|---|
| $\mathcal{T}^{\text{perfect}}(2,3)$ | 50 | 15 | 5s | 3.28% | 3.72% | 39.84% |
| $\mathcal{T}^{\text{perfect}}(2,4)$ | 50 | 31 | 10s | 2.13% | 1.86% | 40.30% |
| $\mathcal{T}^{\text{perfect}}(2,5)$ | 50 | 63 | 19s | 0.90% | 0.92% | 40.02% |
| $\mathcal{T}^{\text{perfect}}(2,6)$ | 50 | 127 | 37s | 0.55% | 0.64% | 39.97% |
| $\mathcal{T}^{\text{perfect}}(2,7)$ | 50 | 255 | 74s | 0.38% | 0.45% | 39.97% |
| | | | | | | |
| $\mathcal{T}^{\text{rand}}(64,2,3), 1^{st}$ | 50 | 116 | 41s | 0.56% | 0.82% | 39.98% |
| $\mathcal{T}^{\text{rand}}(64,2,3), 2^{nd}$ | 50 | 116 | 38s | 0.66% | 0.75% | 40.05% |
| $\mathcal{T}^{\text{rand}}(64,2,3), 3^{rd}$ | 50 | 116 | 41s | 0.54% | 0.57% | 39.98% |
| $\mathcal{T}^{\text{rand}}(64,2,4)$ | 50 | 112 | 31s | 0.65% | 0.67% | 40.00% |
| $\mathcal{T}^{\text{rand}}(64,2,5)$ | 50 | 109 | 32s | 0.69% | 0.87% | 40.01% |

### *5.2.2. Three-phase UDC microstructure*

For the three-phase UDC microstructure, see Fig. 7(b), a dataset of 500 samples is generated for the offline training. This data is divided into a training dataset consisting of 400 samples and a validation dataset consisting of 100 samples. Since three phases are present in this microstructure, this work considers not only the perfect ternary trees, *i.e.* $\mathcal{T}^{\text{perfect}}(3,\bullet)$ with $\bullet$ being a number ranging from 3 to 7, but also perfect binary trees, *i.e.* $\mathcal{T}^{\text{perfect}}(2,\bullet)$ with $\bullet$ being a number ranging from 6 to 9, and random trees. The random trees in this section are limited to the cases $\mathcal{T}^{\text{rand}}(\bullet,3,3)$ with $\bullet$ being a number ranging from 81 to 729. This section aims at demonstrating the statement of Section 3 in which the material network is formed from the viewpoint of the network interaction so that an arbitrary number of phases can be considered for a given architecture.

Histories of the average training and validation errors of the different network architectures are reported in Fig. 10 using 100 epochs for the different network architectures. Figure 10(a) represents the results with the perfect ternary trees. When the number of material nodes reaches $3^6 = 729$, the average error becomes smaller than 1%. As demonstrated in Fig. 10(b) for the prefect binary trees, an average error smaller than 1% is also obtained for a number of material nodes larger than $2^8 = 256$. The results with two different realizations of the random tree $\mathcal{T}^{\text{rand}}(\bullet,3,3)$ with $\bullet$ being successively 81, 243, 512, and 729 are shown in Figs. 10(c) and (d), in which an average error smaller than 1% is also obtained for a number of material nodes larger than 512. Generally, a network with limited number of material nodes cannot correctly capture the full-field response. The training process therefore stabilizes with a higher error when the number of material nodes decreases.



(a) (b) (c) (d)

Figure 10: Three-phase UDC microstructure - training histories under plane strain state after 100 epochs: (a) perfect ternary trees, (b) perfect binary trees, and (c, d) random trees. Continuous and dashed lines represent respectively the average training and validation errors. The 1%-error line is also reported for comparison purpose.

The training histories for this three-phase UDC microstructure are also summarized in Tab. 2. The reached fiber volume factions of the round and elliptic fibers are very close to the values of 20% and 15.1% of the microstructure used for training, we refer to Fig. 7(b) for this microstructure. Thanks to the efficient training framework, the training time is relatively small with a wide range of the number of material nodes, *e.g.* the $T^{\text{perfect}}(3,7)$ tree with 2187 material nodes requires only 1h 18min to complete the training process. The training time for all other cases is smaller than 35 minutes. Clearly, the error of 1% can be obtained with different network architectures as long as the number of material nodes is sufficiently high to obtain rich enough network interactions.

### *5.2.3. Two-phase SFC microstructure*

For the two-phase SFC microstructure, see Figs. 8(a) and (b), a dataset of 500 samples is generated for the offline training. This data is divided into a training dataset consisting of 400 samples and a validation dataset consisting of 100 samples.

Histories of the average training error and of the average validation error are reported in Fig. 11 using 200 epochs with perfect binary and random trees. From the errors of the binary trees reported in Fig. 11(a), it can be seen that a good accuracy can be achieved with $2^7 = 128$ and $2^8 = 256$ material nodes. For the random trees with the two different

Table 2: Three-phase UDC microstructure - training results for different DMN architectures. The number of epochs $N_{\max}$, number of fitting parameters $N_f$, training time $T$, averaging training error $C^{g,\,\mathrm{train}}$, averaging validation error $C^{g,\,\mathrm{test}}$, and the predicted fiber volume fractions $V_f$ of round and elliptic fibers of the DMN are reported.

| DMN | $N_{\max}$ | $N_f$ | $T$ | $C^{g,\,\mathrm{train}}$ | $C^{g,\,\mathrm{test}}$ | $V_f$ round | $V_f$ elliptic |
|---|---|---|---|---|---|---|---|
| $\mathcal{T}^{\mathrm{perfect}}(3,3)$ | 100 | 40 | 1min 6s | 4.73% | 4.77% | 20.50% | 14.75% |
| $\mathcal{T}^{\mathrm{perfect}}(3,4)$ | 100 | 121 | 2min 6s | 2.71% | 2.8% | 21.43% | 14.90% |
| $\mathcal{T}^{\mathrm{perfect}}(3,5)$ | 100 | 364 | 12min 2s | 1.24% | 1.33% | 21.93% | 15.03% |
| $\mathcal{T}^{\mathrm{perfect}}(3,6)$ | 100 | 1093 | 26min 28s | 0.90% | 0.84% | 22.01% | 15.00% |
| $\mathcal{T}^{\mathrm{perfect}}(3,7)$ | 100 | 3280 | 1h 18min | 0.69% | 0.67% | 21.90% | 15.12% |
| | | | | | | | |
| $\mathcal{T}^{\mathrm{perfect}}(2,6)$ | 100 | 127 | 1min 48s | 1.41% | 1.50% | 21.92% | 15.13% |
| $\mathcal{T}^{\mathrm{perfect}}(2,7)$ | 100 | 255 | 3min 44s | 0.93% | 1.01% | 22.07% | 15.01% |
| $\mathcal{T}^{\mathrm{perfect}}(2,8)$ | 100 | 511 | 11min 2s | 0.88% | 0.92% | 21.97% | 15.09% |
| $\mathcal{T}^{\mathrm{perfect}}(2,9)$ | 100 | 1023 | 20min 21s | 0.72% | 0.71% | 21.98% | 15.08% |
| | | | | | | | |
| $\mathcal{T}^{\mathrm{rand}}(81,3,3),1^{st}$ | 100 | 133 | 3min 45s | 1.78% | 1.66% | 22.00% | 15.00% |
| $\mathcal{T}^{\mathrm{rand}}(81,3,3),2^{nd}$ | 100 | 133 | 3min 43s | 1.40% | 1.59% | 21.95% | 15.10% |
| $\mathcal{T}^{\mathrm{rand}}(243,3,3),1^{st}$ | 100 | 403 | 10min 15s | 0.98% | 1.03% | 22.00% | 14.99% |
| $\mathcal{T}^{\mathrm{rand}}(243,3,3),2^{nd}$ | 100 | 403 | 10min 30s | 1.11% | 1.01% | 21.99% | 15.08% |
| $\mathcal{T}^{\mathrm{rand}}(512,3,3),1^{st}$ | 100 | 859 | 19min 11s | 0.89% | 0.96% | 21.98% | 15.10% |
| $\mathcal{T}^{\mathrm{rand}}(512,3,3),2^{nd}$ | 100 | 859 | 19min 11s | 0.89% | 0.96% | 21.98% | 15.10% |
| $\mathcal{T}^{\mathrm{rand}}(729,3,3),1^{st}$ | 100 | 1221 | 34min 35s | 0.76% | 0.91% | 21.98% | 15.07% |
| $\mathcal{T}^{\mathrm{rand}}(729,3,3),2^{nd}$ | 100 | 1223 | 30min 16s | 0.82% | 0.68% | 21.94% | 15.11% |

realizations reported in Fig. 11(b), a similar accuracy can be obtained but this requires a larger number of material nodes.

The training histories of this SFC microstructure are summarized in Tab. 3. The obtained fiber volume faction is very close to the value of 20.66% of the microstructure used for training. Since the matrix operations with full 3-dimensional material tensors must be employed during the training process, the training time is much higher than the ones of the UDC microstructures reported in Tabs. 1 and 2. A perfect binary tree requires less training time compared to a random tree with the same number of material nodes as a result of the three-list interactions present in the latter.

### 5.3. *Online predictions of the trained material networks*

The online predictions of the trained material network are carried out based on the concept of the network interactions as detailed in Section 3.4. The material parameters of the fibers and matrix are reported in Tab. 4, in which the elastic fibers are embedded in the elastoplastic matrix without decohesion. Hard fibers are considered in the two-phase UDC and SFC microstructures while both hard and soft fibers are considered, respectively for the circular and elliptic inclusions, in the three-phase UDC microstructure. The microstructures are subjected to cyclic uniaxial strain loadings: either a prescribed shear strain by imposing the value of $F_{01}$ or a prescribed tension strain by imposing the value of $F_{00}$.

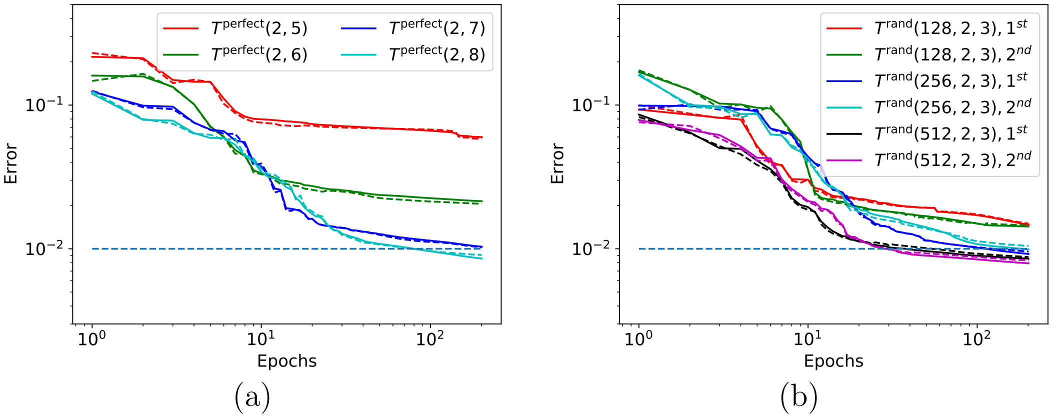


Figure 11: SFC microstructure - training histories after 200 epochs: (a) perfect binary trees, and (b) random trees. Continuous and dashed lines represent average respectively the training error and the average validation error.

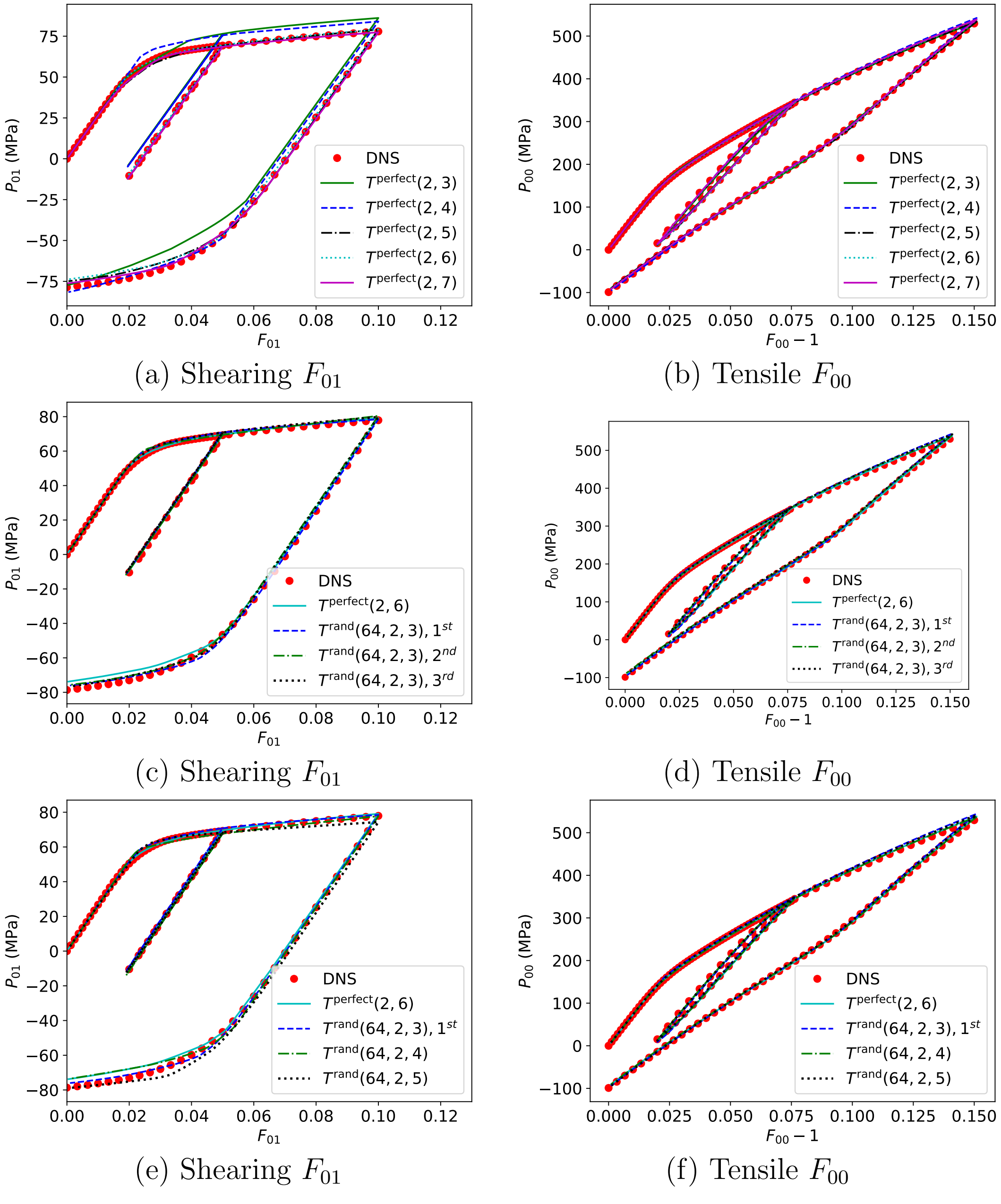
Table 3: SFC microstructure - training results for different DMN architectures. The number of epochs $N_{\max}$, number of fitting parameters $N_f$, training time $T$, averaging training error $C^{g,\,\text{train}}$, averaging validation error $C^{g,\,\text{test}}$, and the predicted fiber volume fraction $V_f$ with the DMN are reported.

| DMN | $N_{\max}$ | $N_f$ | $T$ | $C^{g,\,\text{train}}$ | $C^{g,\,\text{test}}$ | $V_f$ |
|---|---|---|---|---|---|---|
| $T^{\text{perfect}}(2,5)$ | 200 | 94 | 17min | 4.95% | 4.88% | 19.32% |
| $T^{\text{perfect}}(2,6)$ | 200 | 190 | 35min | 2.14% | 2.06% | 20.80% |
| $T^{\text{perfect}}(2,7)$ | 200 | 382 | 1h9min | 1.03% | 1.02% | 20.58% |
| $T^{\text{perfect}}(2,8)$ | 200 | 766 | 2h10min | 0.86% | 0.90% | 20.58% |
| | | | | | | |
| $T^{\text{rand}}(128,2,3),1^{st}$ | 200 | 348 | 1h36min | 1.47% | 1.50% | 20.64% |
| $T^{\text{rand}}(128,2,3),2^{nd}$ | 200 | 348 | 1h38min | 1.43% | 1.45% | 20.49% |
| $T^{\text{rand}}(256,2,3),1^{st}$ | 200 | 684 | 3h13min | 0.92% | 0.95% | 20.61% |
| $T^{\text{rand}}(256,2,3),2^{nd}$ | 200 | 684 | 3h5min | 0.99% | 1.00% | 20.72% |
| $T^{\text{rand}}(512,2,3),1^{st}$ | 200 | 1384 | 6h | 0.85% | 0.88% | 20.68% |
| $T^{\text{rand}}(512,2,3),2^{nd}$ | 200 | 1384 | 5h | 0.79% | 0.83% | 20.58% |

In the case of the two-phase UDC microstructure, the homogenized stress evolution in terms of the prescribed homogenized strain is shown in Fig. 12 for the trained material networks reported in Tab. 1. The corresponding full-field results (denoted by direct numerical simulation -DNS) are also reported for comparison purpose. For all cases, the full-field results are well reproduced by the material networks for a sufficiently high number of material nodes (at least 32 for this microstructure). It is shown that the material network can capture the nonlinear behavior of the composite material with much less degrees of freedom.

In the case of the three-phase UDC microstructure, the homogenized stress responses in terms of the prescribed homogenized strain are shown in Fig. 13 for the trained material networks reported in Tab. 2. It is shown that material networks with deeper and richer interactions provide better predictions disregarding their architecture.

In the case of the two-phase SFC microstructure, the homogenized stress evolution curves in terms of the prescribed homogenized strain are shown in Fig. 14 for the trained material



(a) Shearing $F_{01}$

(b) Tensile $F_{00}$

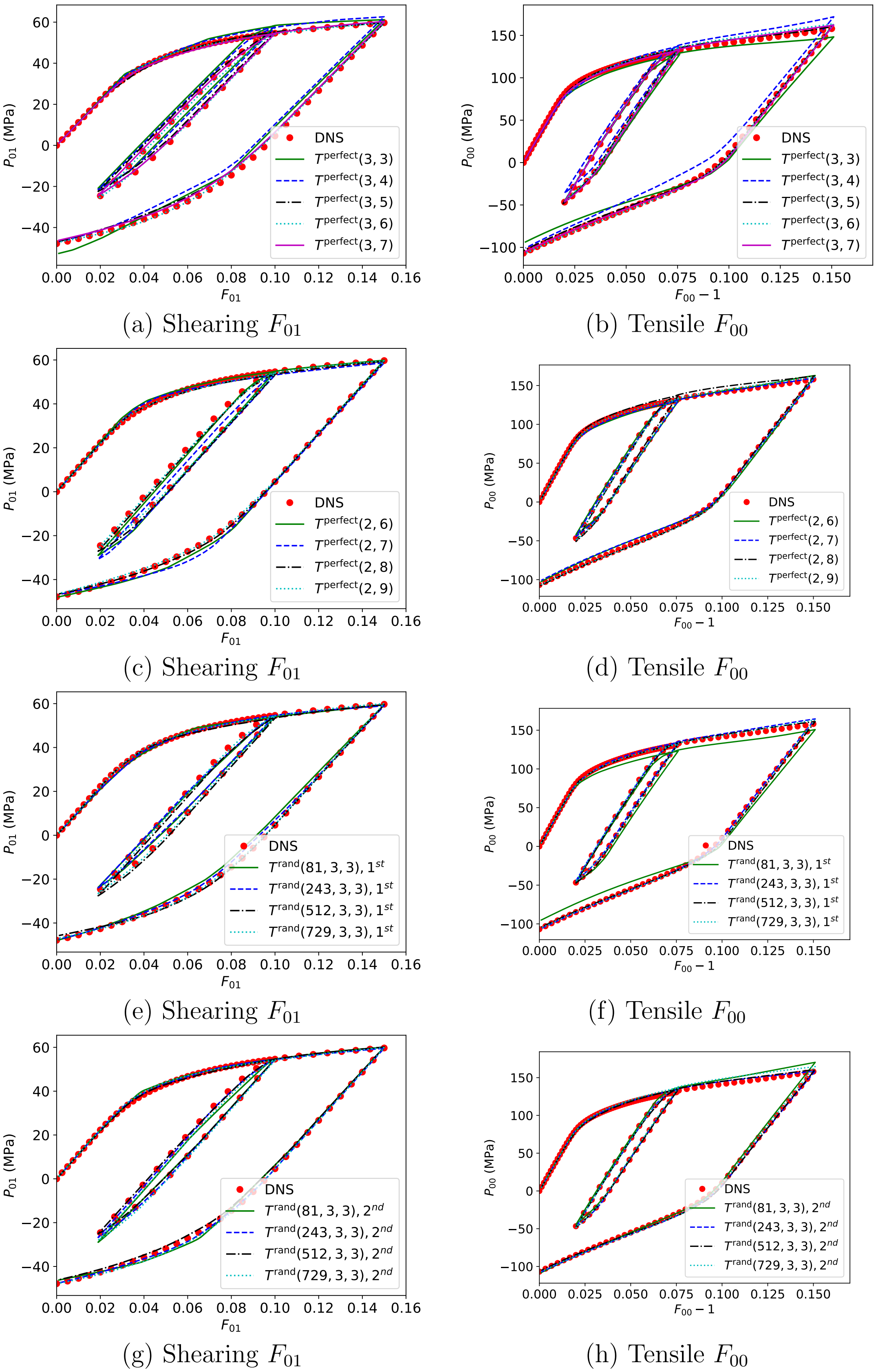
(c) Shearing $F_{01}$

(d) Tensile $F_{00}$

(e) Shearing $F_{01}$

(f) Tensile $F_{00}$

Figure 12: Two-phase UDC microstructure - comparison of the results predicted by the material network with different architectures and by the direct numerical simulation (DNS) under pure shear (a, c, e) and tension (b, d, f).



(a) Shearing $F_{01}$

(b) Tensile $F_{00}$

(c) Shearing $F_{01}$

(d) Tensile $F_{00}$

(e) Shearing $F_{01}$

(f) Tensile $F_{00}$

(g) Shearing $F_{01}$

(h) Tensile $F_{00}$

Figure 13: Three-phase UDC microstructure - comparison of the results predicted by the material network with different architectures and by the direct numerical simulation (DNS) under pure shear (a, c, e, g) and tension (b, d, f, h).

Table 4: Material parameters for fibers and matrix

| | Bulk modulus<br>$K$ [GPa] | Shear modulus<br>$\mu$ [GPa] | Isotropic flow stress<br>$\tau_{\mathrm{y}}(\gamma)$ [MPa]<br>($\gamma$ - equivalent plastic strain) |
|---|---|---|---|
| Hard fibers | 17.78 | 7.68 | $+\infty$ |
| Soft fibers | 0.13 | 0.7 | $+\infty$ |
| Matrix | 2.67 | 1.23 | $100 + 64\gamma$ |

networks reported in Tab. 3. For all cases, the DNS is well captured by the material networks.

### *5.4. Computational cost*

The use of the deep material networks allows accelerating the prediction of the microstructure nonlinear response. The computational efficiency in comparison with the full-field finite element resolution (so-called direct numerical simulation - DNS) is characterized by a speedup factor, which is defined as

$$\text{Speedup} = \frac{T_{\text{DNS}}}{T_{\text{DMN}}}\,, \tag{77}$$

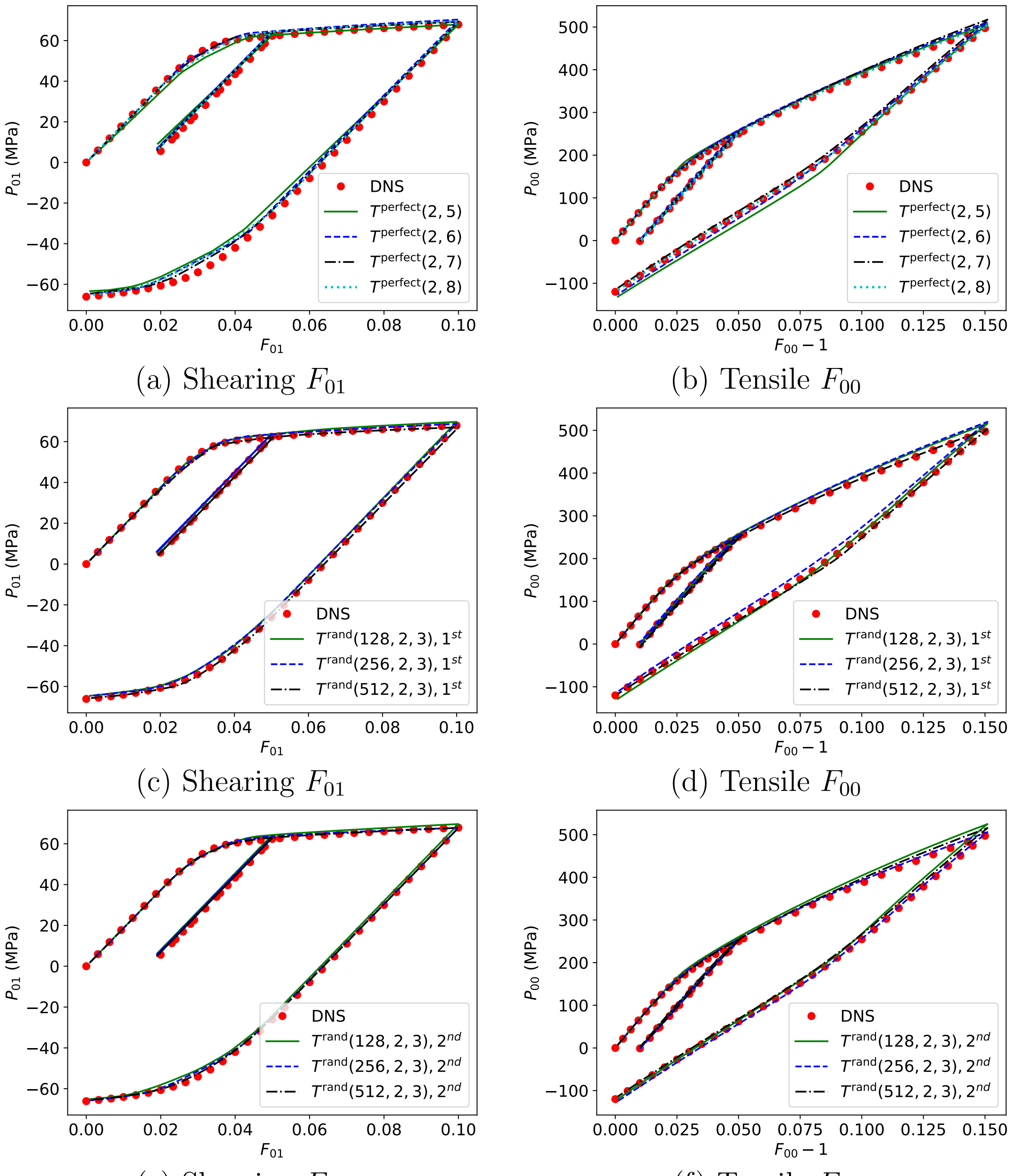
where $T_{\text{DNS}}$ and $T_{\text{DMN}}$ denote the wall-clock times required to complete DNS and DMN predictions, respectively.

The computational time for generating the offline training dataset and for the direct numerical simulations are summarized in Tab. 5. The three-dimensional finite element simulations in the case of the SFC microstructure require much more computational time than the two-dimensional finite element simulations of the UDC microstructures.

Table 5: Computational time for generating the offline training dataset and for the direct numerical simulations (DNS).

| | Training<br>dataset ($N_s$) | Shearing DNS<br>(Average) | Tensile DNS<br>(Average) |
|---|---|---|---|
| Two-phase UDC | 0.15h (300 samples) | 0.25h | 0.5h |
| Three-phase UDC | 0.75h (500 samples) | 0.4h | 0.7h |
| Two-phase SFC | 150h (400 samples) | 12h | 22h |

The speedup obtained by the DMN is shown in Fig. 15 for the online predictions reported in Figs. 12, 13, and 14 respectively for the two-phase UDC, three-phase UDC, and two-phase SFC microstructures in comparison with their corresponding DNS reported in Tab. 5. It can be seen that the DMN predictions are much faster than the DNS, especially for the SFC microstructure with a speedup larger than 1000. A higher speedup is obtained with a smaller number of the material nodes. When the number of the material nodes increases, the speedup decreases as the number of material nodes reflects the complexity of the material network, *i.e.* a larger number of material nodes leads to more degrees of freedom and more evaluations of the local constitutive behavior. Although the DMN requires the creation of



(a) Shearing $F_{01}$

(b) Tensile $F_{00}$

(c) Shearing $F_{01}$

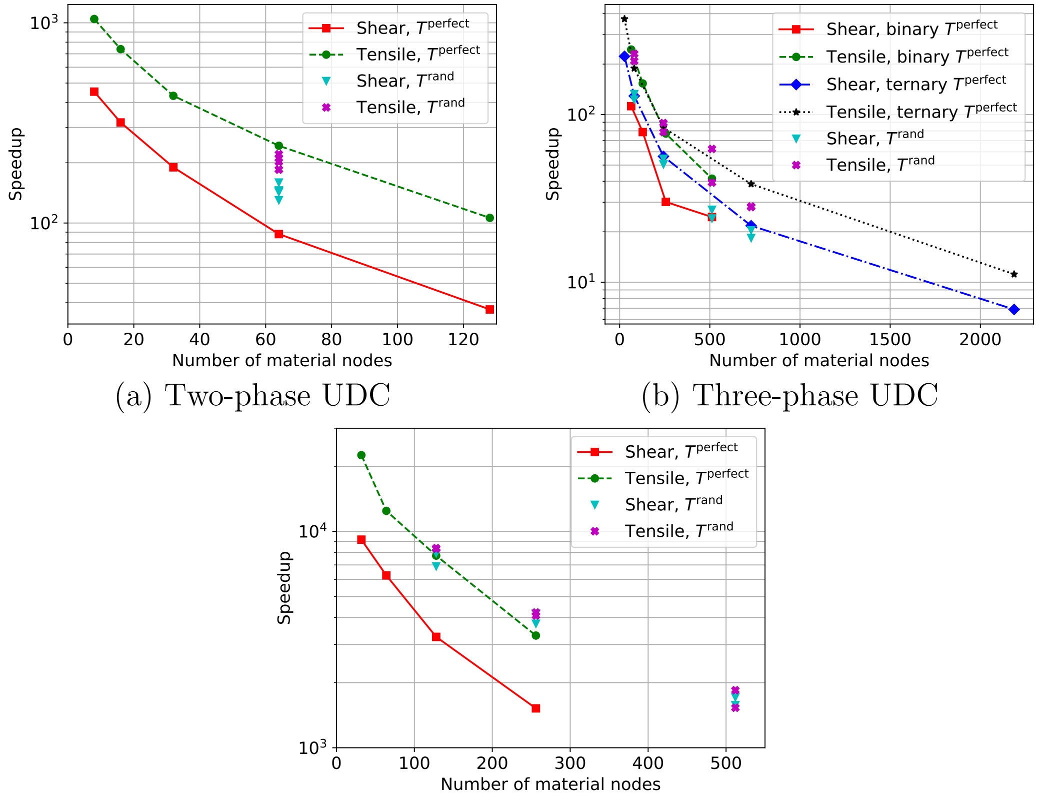
(d) Tensile $F_{00}$

(e) Shearing $F_{01}$

(f) Tensile $F_{00}$

Figure 14: SFC microstructure - comparison of the results predicted by the material network with different network architectures and by the direct numerical simulation (DNS) under pure shear (a, c, e) and tension (b, d, f).

the offline dataset and training, their use provides a promising computational efficiency since the network is trained only once with the homogenised material tensors at zero-strain as data and can subsequently be used as a predictive model for different constitutive behaviors and loading paths.



(a) Two-phase UDC

(b) Three-phase UDC

(c) Two-phase SFC

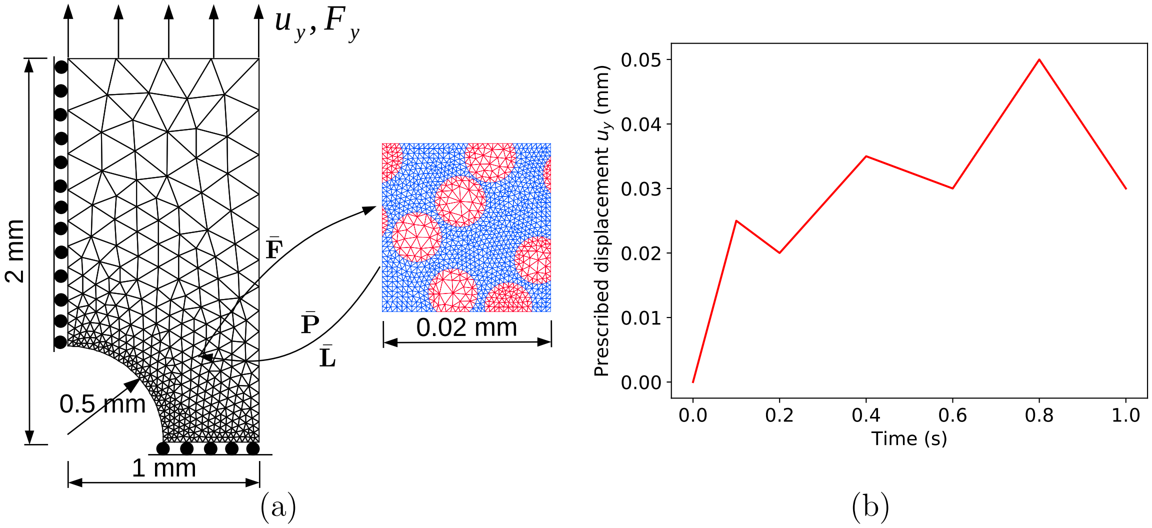
Figure 15: Speedup of DMNs: (a) two-phase UDC microstructure for the predictions reported in Fig. 12, (b) three-phase UDC microstructure for the predictions reported in Fig. 13, and (c) two-phase SFC microstructure for the predictions reported in Fig. 14.

### *5.5. Two-scale simulation of an open-hole sample*

The trained DMNs are now used as a reduced order model of the corresponding homogenization problem in a multiscale simulation. In this section, the simulation of an open hole sample performed in [27] is reconsidered. The result obtained with a fully coupled FE$^2$ analysis is compared to the ones obtained using the DMNs.

Table 6: Open-hole sample - material parameters for fibers and matrix in the multiscale simulation [27].

| | Bulk modulus | Shear modulus | Isotropic flow stress |
|---|---|---|---|
| | $K$ [GPa] | $\mu$ [GPa] | $\tau_{\text{y}}(\gamma)$ [MPa] |
| | | | ($\gamma$ - equivalent plastic strain) |
| Fibers | 16.67 | 12.5 | $+\infty$ |
| Matrix | 2.5 | 1.15 | $100 + 20\,[1 - \exp(-30\gamma)]$ |



(a) (b)

Figure 16: Open-hole sample following [27]: (a) multiscale setting and (b) prescribed displacement at the top boundary of the sample. The microstructure volume element corresponds to the one shown in Fig. 7(a) in Section 5.2.

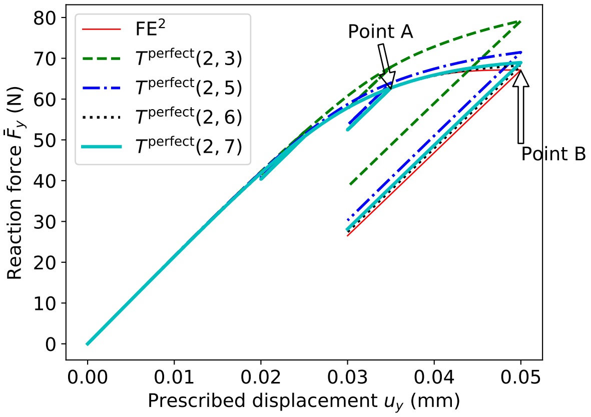


Figure 17: Open-hole sample - comparison between the $\mathrm{FE}^2$ and the finite element simulation using the DMN as a surrogate: reaction force versus prescribed displacement curves.

The multiscale setting of the open-hole sample is illustrated in Fig. 16(a). The sample is loaded on its top edge under the prescribed displacement depicted in Fig. 16(b). Because of the geometrical and loading symmetries, only one quarter of the sample is modeled. The material properties of the matrix and fibers used in the work [27] are reported in Tab. 6. Since the microscopic volume element used in the multiscale analysis was used to train the DMNs in Section 5.2.1, these trained DMNs can be readily used in this section for the multiscale analyzes, without requiring additional training although the material parameters have changed. The reference solution, a concurrent $\mathrm{FE}^2$ analysis was conducted in [27] in which the micromechanics model is solved using the finite element method in a concurrent way with the macro-scale problem. The result in terms of the reaction force versus prescribed displacement is reported in Fig. 17. The simulations obtained with successively $T^{\mathrm{perfect}}(2,3)$, $T^{\mathrm{perfect}}(2,5)$, $T^{\mathrm{perfect}}(2,6)$, and $T^{\mathrm{perfect}}(2,7)$ are also depicted. The models $T^{\mathrm{perfect}}(2,3)$ and $T^{\mathrm{perfect}}(2,5)$ overestimate the reference solution while the other two models $T^{\mathrm{perfect}}(2,6)$ and

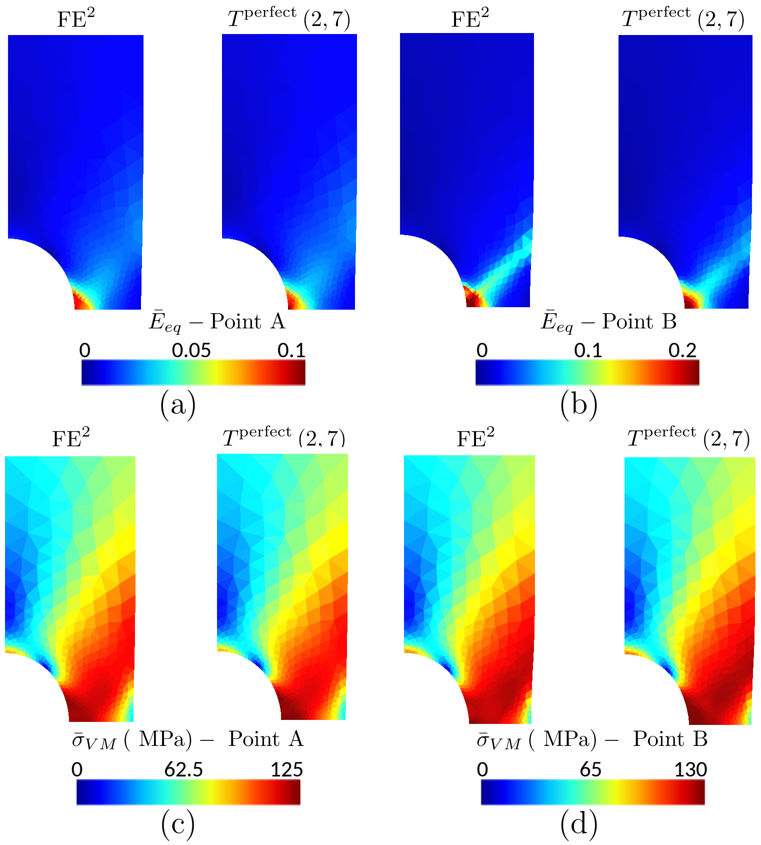


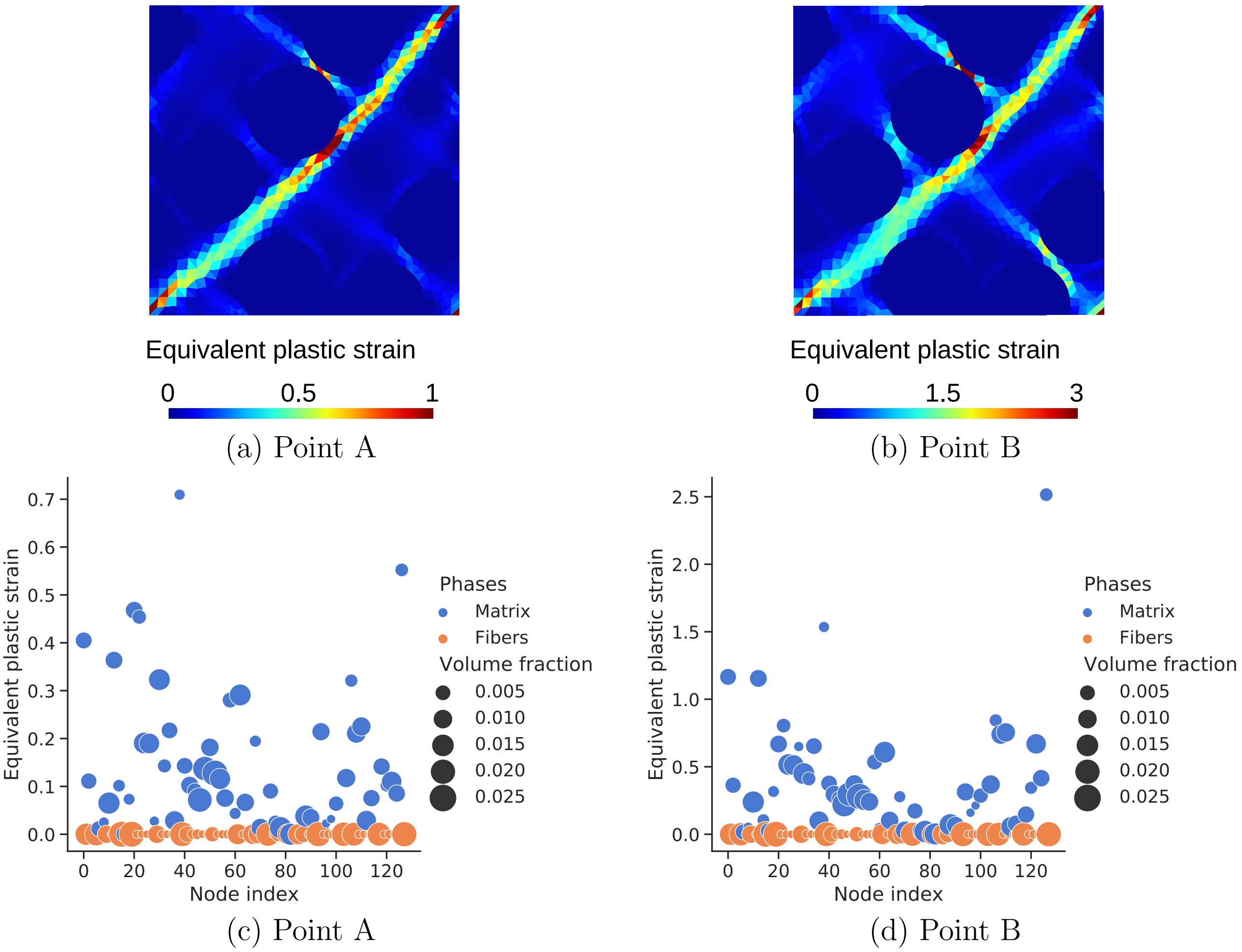
Figure 18: Open-hole sample - comparison between the FE$^2$ and the finite element simulation using $T^{\text{perfect}}(2,7)$ as the surrogate: (a, b) distribution of the macroscopic Green-Lagrange equivalent strain $\bar{E}_{eq}$, and (c, d) distribution of the macroscopic von Mises equivalent stress $\bar{\sigma}_{VM}$. The loading points A and B are reported in Fig. 17.

$T^{\text{perfect}}(2,7)$ provide good predictions.

Figure 18 compares the distributions of the macroscopic strain and stress obtained with the FE$^2$ simulation and with the simulation using $T^{\text{perfect}}(2,7)$ as a surrogate. The macroscopic Green-Lagrange equivalent strain $\bar{E}_{eq}$ and the macroscopic von Mises equivalent stress $\bar{\sigma}_{VM}$ distributions are depicted for two loading levels denoted by “Point A” and “Point B” in Fig. 17 and are computed by

$$\bar{E}_{eq} = \sqrt{\frac{2}{3}\bar{\boldsymbol{E}}_{GL}^{\text{dev}} : \bar{\boldsymbol{E}}_{GL}^{\text{dev}}} \text{ and } \bar{\sigma}_{VM} = \sqrt{\frac{3}{2}\bar{\boldsymbol{\sigma}}^{\text{dev}} : \bar{\boldsymbol{\sigma}}^{\text{dev}}}\,, \tag{78}$$
$$\text{with } \bar{\boldsymbol{E}}_{GL} = \frac{1}{2}\left(\bar{\mathbf{F}}^T \cdot \bar{\mathbf{F}} - \mathbf{I}\right) \text{ and } \bar{\boldsymbol{\sigma}} = \left(\det \bar{\mathbf{F}}\right)^{-1} \bar{\mathbf{P}} \cdot \bar{\mathbf{F}}^T\,,$$

where $\bar{\boldsymbol{E}}_{GL}^{\text{dev}}$ and $\bar{\boldsymbol{\sigma}}^{\text{dev}}$ are respectively the deviatoric parts of the macroscopic Green-Lagrange strain tensor $\bar{\boldsymbol{E}}_{GL}$ and of the macroscopic Cauchy stress tensor $\bar{\boldsymbol{\sigma}}$, and $\bar{\mathbf{F}}$ and $\bar{\mathbf{P}}$ are respectively the macroscopic deformation gradient and the first Piola-Kirchhoff stress tensors. It can be seen that the model using $T^{\text{perfect}}(2,7)$ as a surrogate can predict the strain and stress distributions in good agreement with the ones of the FE$^2$ simulation. At “Point A”, the strain and stress distributions obtained with these two models are almost identical as shown in Figs.



(a) Point A

(b) Point B

(c) Point A

(d) Point B

Figure 19: Open-hole sample - local equivalent plastic strain computed for the macroscopic point close to the hole in the high strain region: (a, b) predicted by the micro-scale BVP resolution of the FE$^2$ analysis, and (c, d) at the material nodes of the DMN surrogate, in which case the circle sizes represent the volume fractions of the material nodes. The loading points A and B are reported in Fig. 17.

18(a) and (c). At “Point B”, while the stress distributions are similar as seen in Fig. 18(d), the model using $T^{\text{perfect}}(2,7)$ as a surrogate slightly underestimates the strain distribution inside the band with high deformations compared to the FE$^2$ model as shown in Fig. 18(b). The extrapolation capability of the DMN is demonstrated since the DMN training considers only the material operators at zero-strain, while elastoplastic behaviors are considered for the online prediction. However, in the case of micro-structure evolution during the deformation, this extrapolation accuracy could be reduced. In this case, the offline training should be enriched with the nonlinear behaviors to better capture the interaction between the material nodes. Considering a macroscopic point close to the hole in the high strain region, the local equivalent plastic strains predicted by the micro-scale BVP resolution in the FE$^2$ analysis are compared in Fig. 19 to the level of plastic strain reached in the material nodes of the DMN. The levels of plastic strain in the micro-scale BVP and in the DMN material nodes are of equivalent magnitude for the two loading levels “Point A” and “Point B”.

Table 7: Open-hole sample - computational cost of the multiscale simulations.

| | Offline sampling | Training | Online |
|---|---|---|---|
| FE$^2$ [27] | - | - | 18000 hour-cpu |
| Recurrent Neural Network [27] | 18000 hour-cpu | 3 day-cpu | 0.5 hour-cpu |
| $T^{\text{perfect}}(2,3)$ | $<$ 10 minute-cpu | $<$ 2 minute-cpu | 0.4 hour-cpu |
| $T^{\text{perfect}}(2,5)$ | $<$ 10 minute-cpu | $<$ 2 minute-cpu | 3 hour-cpu |
| $T^{\text{perfect}}(2,6)$ | $<$ 10 minute-cpu | $<$ 2 minute-cpu | 15 hour-cpu |
| $T^{\text{perfect}}(2,7)$ | $<$ 10 minute-cpu | $<$ 2 minute-cpu | 34 hour-cpu |

The computational cost breakdown for multiscale analyzes with the different methodologies is reported in Tab. 7. The computation time of the FE$^2$ approach took around 30 h using 600 processors on a cluster. For comparison, the simulation using the Recurrent Neural Network (RNN) as surrogate developed in [27] required around one half hour on a single cpu, but required a considerable time to generate the training data (generation of 9000 loading paths with around 2 hour per path) and to train the RNN (around 3 days on a single processor). The computational efficiency of the simulations using DMNs as surrogates is here demonstrated. Although a larger amount of computational time is required for online simulations, *e.g.* it took 15 hours and 34 hours to complete the simulations with $T^{\text{perfect}}(2,6)$ and $T^{\text{perfect}}(2,7)$ respectively, the training time is relatively small. Moreover, if the RNN-based surrogate model [27] provides a high speedup, the prediction becomes inaccurate when extrapolating beyond the offline sampling space. The DMN-based surrogate models can resolve this problem as the creation of the offline training data do not require iterative resolutions, but only the evaluation of the homogenized material tensor at zero-strain.

## 6. Conclusion

In this work, the framework of material network is revisited. The material network is formed from the discrete material nodes under the viewpoint of the hierarchical network interactions. This viewpoint allows not only easily evaluating the response of material networks with an arbitrary constitutive law considered at each material node, through the resolution of well-defined governing equations, but also considering an arbitrary number of phases independently of the network architecture. The predictive capability of the material network is then guaranteed for sufficiently deep and rich network interactions.

We also provide a complete and efficient offline training procedure for the material networks. The training procedure works with an arbitrary number of phases and with an arbitrary DMN architecture. In order to accelerate the training process, a fast algorithm to estimate the DMN homogenized material tensor at zero-strain is based on the analytical resolution strategy of a multiple-phase laminate. Finally, the training procedure based on a simple adaptive scheme of the stochastic gradient descent is proposed to minimize a cost function defined on the training dataset. Through the numerical applications, the capability of the offline training procedure is demonstrated as it can be performed at a reduced computational cost.

The DMNs was then used as surrogates of the computational micromechanics models in multiscale simulations. Once trained, the DMNs can be used to predict the homogenized response with a good accuracy and in much lower computational time as compared to the use of a direct finite element simulation of microscale BVP. In this work, only $J_2$ plasticity model is considered. In the future, the DMN should be extended for more complex material laws, *e.g.* damage and failure. A remaining important challenge is to extend the method to cases in which the microstructure evolves with the deformation, as in the cases of porous materials under large deformation.

## Data availability

The raw/processed data required to reproduce these findings is available on [42] under the Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

## Appendix A. Vector, matrix representations of arbitrary second-order and fourth-order tensors

An arbitrary second-order tensor **H** can be rewritten as a column vector **A** of 9 components, with

$$\mathrm{vec}\left(\mathbf{H}\right) = \mathbf{A} \leftrightarrow H_{ij} = A_{i+3j}\,. \tag{A.1}$$

Similarly, an arbitrary fourth-order tensor **T** can rewritten as a square matrix **B** of 81 components, with

$$\mathrm{mat}\left(\mathbf{T}\right) = \mathbf{B} \leftrightarrow T_{ijkl} = B_{pq} \text{ with } p = i + 3j \text{ and } q = k + 3l\,. \tag{A.2}$$

The reciprocal operators $\mathrm{vec}^{-1}\left(\bullet\right)$ and $\mathrm{mat}^{-1}\left(\bullet\right)$ can be directly deduced. One has the following property

$$\mathrm{vec}\left(\mathbf{T} : \mathbf{H}\right) = \mathrm{mat}\left(\mathbf{T}\right)\mathrm{vec}\left(\mathbf{H}\right)\,. \tag{A.3}$$

## Appendix B. Material tensor derivations at zero-strain for training purpose

In this section we assume infinitesimal symmetric strain and stress tensors in order to evaluate the material tensor at zero-strain state. We use respectively the notations $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ for the infinitesimal symmetric stress and strain tensors.

### *Appendix B.1. Vector and matrix representations of tensors at zero-strain*

The operator $[\bullet]$ is introduced as follows:

- With a vector $\mathbf{a}$:

$$[\mathbf{a}] = \begin{cases} \begin{bmatrix} a_0 & a_1 \end{bmatrix}^T & \text{if plane strain state} \\ \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix}^T & \text{otherwise} \end{cases} . \tag{B.1}$$

- With the true deformation tensor $\boldsymbol{\varepsilon}$:

$$[\boldsymbol{\varepsilon}] = \begin{cases} \begin{bmatrix} \varepsilon_{00} & \varepsilon_{11} & 2\varepsilon_{01} \end{bmatrix}^T & \text{if plane strain state} \\ \begin{bmatrix} \varepsilon_{00} & \varepsilon_{11} & \varepsilon_{22} & 2\varepsilon_{01} & 2\varepsilon_{02} & 2\varepsilon_{12} \end{bmatrix}^T & \text{otherwise} \end{cases} . \tag{B.2}$$

- With the Cauchy stress tensor $\boldsymbol{\sigma}$:

$$[\boldsymbol{\sigma}] = \begin{cases} \begin{bmatrix} \sigma_{00} & \sigma_{11} & \sigma_{01} \end{bmatrix}^T & \text{if plane strain state} \\ \begin{bmatrix} \sigma_{00} & \sigma_{11} & \sigma_{22} & \sigma_{01} & \sigma_{02} & \sigma_{12} \end{bmatrix}^T & \text{otherwise} \end{cases} . \tag{B.3}$$

- With the material tensor $\mathbb{C}$:

$$[\mathbb{C}] = \begin{cases} \begin{bmatrix} \mathbb{C}_{0000} & \mathbb{C}_{0011} & \mathbb{C}_{0001} \\ \mathbb{C}_{1100} & \mathbb{C}_{1111} & \mathbb{C}_{1101} \\ \mathbb{C}_{0100} & \mathbb{C}_{0111} & \mathbb{C}_{0101} \end{bmatrix} & \text{if plane strain state} \\ \begin{bmatrix} \mathbb{C}_{0000} & \mathbb{C}_{0011} & \mathbb{C}_{0022} & \mathbb{C}_{0001} & \mathbb{C}_{0002} & \mathbb{C}_{0012} \\ \mathbb{C}_{1100} & \mathbb{C}_{1111} & \mathbb{C}_{1122} & \mathbb{C}_{1101} & \mathbb{C}_{1102} & \mathbb{C}_{1112} \\ \mathbb{C}_{2200} & \mathbb{C}_{2211} & \mathbb{C}_{2222} & \mathbb{C}_{2201} & \mathbb{C}_{2202} & \mathbb{C}_{2212} \\ \mathbb{C}_{0100} & \mathbb{C}_{0111} & \mathbb{C}_{0122} & \mathbb{C}_{0101} & \mathbb{C}_{0102} & \mathbb{C}_{0112} \\ \mathbb{C}_{0200} & \mathbb{C}_{0211} & \mathbb{C}_{0222} & \mathbb{C}_{0201} & \mathbb{C}_{0202} & \mathbb{C}_{0212} \\ \mathbb{C}_{1200} & \mathbb{C}_{1211} & \mathbb{C}_{1222} & \mathbb{C}_{1201} & \mathbb{C}_{1202} & \mathbb{C}_{1212} \end{bmatrix} & \text{otherwise} \end{cases} . \tag{B.4}$$

The stress-strain relation is preserved with the operator $[\bullet]$, *i.e.*

$$[\boldsymbol{\sigma}] = [\mathbb{C} : \boldsymbol{\varepsilon}] = [\mathbb{C}]\,[\boldsymbol{\varepsilon}] \ . \tag{B.5}$$

Moreover, one has the following properties

$$[\mathbf{a} \otimes^s \mathbf{N}] = \mathbf{H}\,[\mathbf{a}] \;\text{ and }\; [\boldsymbol{\sigma} \cdot \mathbf{N}] = \mathbf{H}^T\,[\boldsymbol{\sigma}] \;, \tag{B.6}$$

where $[\mathbf{a} \otimes^s \mathbf{N}]$ is considered as a strain tensor, $\otimes^s$ denotes the symmetric dyadic tensor product, *i.e.* $\mathbf{c} \otimes^s \mathbf{d} = \frac{1}{2}\left(c_i d_j + c_j d_i\right)$ for two arbitrary vectors $\mathbf{c}$ and $\mathbf{d}$, and where $\mathbf{H} = \mathbf{H}\left(\mathbf{N}\right)$ is detailed as

$$\mathbf{H}\left(\mathbf{N}\right) = \begin{cases} \begin{bmatrix} N_0 & 0 \\ 0 & N_1 \\ N_1 & N_0 \end{bmatrix} & \text{if plane strain state} \\ \begin{bmatrix} N_0 & 0 & 0 \\ 0 & N_1 & 0 \\ 0 & 0 & N_2 \\ N_1 & N_0 & 0 \\ N_2 & 0 & N_0 \\ 0 & N_2 & N_1 \end{bmatrix} & \text{otherwise} \end{cases} . \tag{B.7}$$

*Appendix B.2. Material sampling in the plane strain state*

For plane strain problems, $N_s$ samples $\mathbb{X}_s$ with $s = 0\,, \ldots\,, N_s - 1$ are directly generated. For this purpose, the orthotropic material tensor of each phase is written as

$$\left[\mathbb{C}^{(i)}\right] = \begin{bmatrix} C_{00}^{(i)} & C_{01}^{(i)} & 0 \\ C_{01}^{(i)} & C_{11}^{(i)} & 0 \\ 0 & 0 & C_{22}^{(i)} \end{bmatrix} \text{ with } i = 0, \ldots, P-1\,, \tag{B.8}$$

where the conditions $C_{00}^{(i)} C_{11}^{(i)} - \left(C_{01}^{(i)}\right)^2 > 0$ and $C_{22}^{(i)} > 0$, with $i = 0, \ldots, P-1$, are used to obtain a positive definite matrix. To avoid scaling issues, *i.e.* considering the pair $[\mathbb{C}^{(0)}, \ldots, \mathbb{C}^{(P-1)}]$ versus the pair $[\gamma\mathbb{C}^{(0)}, \ldots, \gamma\mathbb{C}^{(P-1)}]$ for $\gamma \neq 0$, one first generates

$$C_{00}^{(0)} = 1 \text{ and } \ln\left(\frac{C_{00}^{(i)}}{C_{00}^{(i-1)}}\right) \in U\left(-1, 1\right) \text{ with } i = 1, \ldots, P-1\,, \tag{B.9}$$

and the other components are then generated as

$$\ln\left(\frac{C_{11}^{(i)}}{C_{00}^{(i)}}\right) \in U\left(-1,1\right)\,, \frac{C_{01}^{(i)}}{\sqrt{C_{00}^{(i)} C_{11}^{(i)}}} \in U\left(0, 0.9\right)\,, \text{ and } \ln\left(\frac{C_{22}^{(i)}}{\sqrt{C_{00}^{(i)} C_{11}^{(i)}}}\right) \in U\left(-1,1\right)\,, \tag{B.10}$$
$$\text{for } i = 0, \ldots, P-1,$$

where $U\left(a, b\right)$ stands for a uniform distribution in the range $[a, b]$.

### *Appendix B.3. Zero-strain material tensor of a network interaction*

Let us consider the DMN $\mathcal{M}$ defined by the list $\mathcal{K}$ of material nodes and the fitting parameters $\boldsymbol{\mathcal{F}}$. Let us consider an arbitrary sub material network with its list of indexes $\mathcal{K}_d^p$ for a node located at depth $d$ and position $p$ in the DMN tree structure as illustrated in Fig. 2. We denote $\bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_d^p\right)$, $\bar{\boldsymbol{\sigma}}\left(\mathcal{K}_d^p\right)$, and $\bar{\mathbb{C}}\left(\mathcal{K}_d^p;\boldsymbol{\mathcal{F}}\right)$ respectively the homogenized infinitesimal symmetric strain tensor, homogenized Cauchy stress tensor, and the homogenized zero-strain material tensor of the sub-material network $\mathcal{K}_d^p$. One has[3]

$$\bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_d^p\right) = \frac{1}{\mathcal{S}\left(\mathcal{K}_d^p\right)} \sum_{i\in\mathcal{K}_d^p} W^i \boldsymbol{\varepsilon}^i\,, \text{ and } \bar{\boldsymbol{\sigma}}\left(\mathcal{K}_d^p\right) = \frac{1}{\mathcal{S}\left(\mathcal{K}_d^p\right)} \sum_{i\in\mathcal{K}_d^p} W^i \boldsymbol{\sigma}^i\,, \tag{B.11}$$

where $\boldsymbol{\varepsilon}^i$ and $\boldsymbol{\sigma}^i$ are respectively the strain and stress at the material node $i$. The material tensor for the sub material network of nodes indexed in $\mathcal{K}_d^p$ leads to the following relation

$$\bar{\boldsymbol{\sigma}}\left(\mathcal{K}_d^p\right) = \bar{\mathbb{C}}\left(\mathcal{K}_d^p;\boldsymbol{\mathcal{F}}\right) : \bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_d^p\right)\,. \tag{B.12}$$

One assumes that there exists a network interaction $m$ created from $n_d^p$ ($n_d^p \geq 2$) non-overlapping sub-lists $\mathcal{K}_{d+1}^{r_m+k}$ with $k = 0, \ldots, n_d^p - 1$, in which $r_m = \sum_{k=0}^{p-1} n_d^k$ denotes the location of the first sub-list of this interaction at the depth $d+1$. One thus has $\mathcal{K}_d^p = \mathcal{K}_{d+1}^{r_m} \cup \ldots \cup \mathcal{K}_{d+1}^{r_m+n_d^p-1}$. This network interaction is formed following the direction $\mathbf{N}^m$ and induces $n_d^p-1$ interaction mechanisms indexed from $q_m$ to $q_m+n_d^p-2$. This network interaction contributes to the deformation gradient of each material node in $\mathcal{K}_d^p$ following Eq. (35) as[4]

$$\begin{cases}
\bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_{d+1}^{r_m}\right) = & \bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_d^p\right) + \dfrac{1}{\mathcal{S}\left(\mathcal{K}_{d+1}^{r_m}\right)} \mathbf{a}^{q_m} \otimes^s \mathbf{N}^m\,, \\
\bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_{d+1}^{r_m+1}\right) = & \bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_d^p\right) + \dfrac{1}{\mathcal{S}\left(\mathcal{K}_{d+1}^{r_m+1}\right)} \left(\mathbf{a}^{q_m+1} - \mathbf{a}^{q_m}\right) \otimes^s \mathbf{N}^m\,, \\
\vdots & \\
\bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_{d+1}^{r_m+n_d^p-1}\right) = & \bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_d^p\right) + \dfrac{1}{\mathcal{S}\left(\mathcal{K}_{d+1}^{r_m+n_d^p-1}\right)} \left(-\mathbf{a}^{q_m+n_d^p-2}\right) \otimes^s \mathbf{N}^m\,,
\end{cases} \tag{B.13}$$

where $\otimes^s$ denotes the symmetric dyadic tensor product, *i.e.* $\mathbf{c} \otimes^s \mathbf{d} = \frac{1}{2}\left(c_i d_j + c_j d_i\right)$ for two arbitrary vectors $\mathbf{c}$ and $\mathbf{d}$. Equation (36) is rewritten in terms of the Cauchy stress tensors, yielding

$$\left(\bar{\boldsymbol{\sigma}}\left(\mathcal{K}_{d+1}^{r_m+k}\right) - \bar{\boldsymbol{\sigma}}\left(\mathcal{K}_{d+1}^{r_m+k+1}\right)\right) \cdot \mathbf{N}^m = \mathbf{0} \text{ with } k = 0, \ldots, n_d^p - 2\,, \tag{B.14}$$

with, as a result of Eq. (B.12),

$$\bar{\boldsymbol{\sigma}}\left(\mathcal{K}_{d+1}^{r_m+k}\right) = \bar{\mathbb{C}}\left(\mathcal{K}_{d+1}^{r_m+k};\boldsymbol{\mathcal{F}}\right) : \bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_{d+1}^{r_m+k}\right) \text{ with } k = 0, \ldots, n_d^p - 1\,. \tag{B.15}$$

[3] Eqs. (19, 20) are used but with the infinetesimal symmetric strain and Cauchy stress measures.

[4] The symmetric infinitesimal strain tensor $\boldsymbol{\varepsilon}$ is estimated from the deformation gradient tensor $\mathbf{F}$ using the relation $\boldsymbol{\varepsilon} = \frac{1}{2}\left(\mathbf{F}^T + \mathbf{F}\right) - \mathbf{I}$.

Besides, since $\mathcal{K}_{d+1}^{r_m+k}$, with $k = 0, \dots, n_d^p - 1$, are $n_d^p$ non-overlapping sub-lists of $\mathcal{K}_d^p$, one always has

$$\begin{aligned} \bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_d^p\right) &= \sum_{k=0}^{n_d^p-1} f_d^{p,k} \bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_{d+1}^{r_m+k}\right) \text{ , and} && \text{(B.16)} \\ \bar{\boldsymbol{\sigma}}\left(\mathcal{K}_d^p\right) &= \sum_{k=0}^{n_d^p-1} f_d^{p,k} \bar{\boldsymbol{\sigma}}\left(\mathcal{K}_{d+1}^{r_m+k}\right) \text{ ,} && \text{(B.17)} \end{aligned}$$

where

$$f_d^{p,k} = \frac{\mathcal{S}\left(\mathcal{K}_{d+1}^{r_m+k}\right)}{\mathcal{S}\left(\mathcal{K}_d^p\right)} \text{ for } k = 0, \dots, n_d^p - 1 \,. \tag{B.18}$$

Clearly the values of $\bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}_{d+1}^{r_m+k}\right)$ with $k = 0, \dots, n_d^p - 1$ following Eq. (B.13) automatically satisfy Eq. (B.16). Using Eqs. (B.13, B.15), Eq. (B.14) results into a linear system of $n_d^p - 1$ equations whose unknowns are

$$\mathbf{b}^{q_m+k} = \frac{\mathbf{a}^{q_m+k}}{\mathcal{S}\left(\mathcal{K}_d^p\right)} \text{ with } k = 0, \dots, n_d^p - 2 \,. \tag{B.19}$$

From this solution, one can obtain, as shown below, the material tensor (66), which is rewritten here as

$$\left[\bar{\mathbb{C}}_d^p\right] = \mathbb{H}_{n_d^p}\left(\left[\bar{\mathbb{C}}_{d+1}^{r_m}\right], \dots, \left[\bar{\mathbb{C}}_{d+1}^{r_m+n_d^p-1}\right], f_d^{p,0}, \dots, f_d^{p,n_d^p-1}, \mathbf{N}^m\right) \tag{B.20}$$

where the notation $\bar{\mathbb{C}}\left(\mathcal{K}_r^s; \boldsymbol{\mathcal{F}}\right)$ is replaced by $\bar{\mathbb{C}}_r^s$ for an arbitrary material network $\mathcal{K}_r^s$ for simplicity; the bijective operator $[\bullet]$ is introduced to convert this fourth-order tensor into a symmetric matrix following Voigt's notations, see Appendix B.1 for details; and $\mathbb{H}_{n_d^p}$ is a functional operator used to refer to the case of $n_d^p$ sub-lists.

In the following sections, the analytical form of $\mathbb{H}_2$ is first derived in the case of a two-list interaction. Then this solution is extended to the general case $\mathbb{H}_{n_d^p}$ with $n_d^p > 2$ of a multiple-list interaction.

### *Appendix B.3.1. Analytical solution for a two-list interaction*

In this section, only indexes representing the order of sub-lists are kept for ease of readability. In order to find the analytical form of the functional operator $\mathbb{H}_2$ expressed in Eq. (B.20), Eqs. (B.12 - B.17) are rewritten for the case of a two-list interaction, leading to following equations

$$
\begin{aligned}
\bar{\boldsymbol{\sigma}} &= \bar{\mathbb{C}} : \bar{\boldsymbol{\varepsilon}} \, , && \text{(B.21)}\\
\bar{\boldsymbol{\varepsilon}}^0 &= \bar{\boldsymbol{\varepsilon}} + \frac{1}{f^0}\mathbf{b} \otimes^s \mathbf{N} \, , && \text{(B.22)}\\
\bar{\boldsymbol{\varepsilon}}^1 &= \bar{\boldsymbol{\varepsilon}} - \frac{1}{f^1}\mathbf{b} \otimes^s \mathbf{N} \, , && \text{(B.23)}\\
\left(\bar{\boldsymbol{\sigma}}^0 - \bar{\boldsymbol{\sigma}}^1\right) \cdot \mathbf{N} &= \mathbf{0} \, , && \text{(B.24)}\\
\bar{\boldsymbol{\sigma}}^0 &= \bar{\mathbb{C}}^0 : \bar{\boldsymbol{\varepsilon}}^0 \, , && \text{(B.25)}\\
\bar{\boldsymbol{\sigma}}^1 &= \bar{\mathbb{C}}^1 : \bar{\boldsymbol{\varepsilon}}^1 \, , && \text{(B.26)}\\
\bar{\boldsymbol{\varepsilon}} &= f^0 \bar{\boldsymbol{\varepsilon}}^0 + f^1 \bar{\boldsymbol{\varepsilon}}^1 \, , \text{ and} && \text{(B.27)}\\
\bar{\boldsymbol{\sigma}} &= f^0 \bar{\boldsymbol{\sigma}}^0 + f^1 \bar{\boldsymbol{\sigma}}^1 \, , && \text{(B.28)}
\end{aligned}
$$

where $f^1 = 1 - f^0$ and where Eq. (B.19) has been used. The operator $\mathbb{H}_2$ is defined as

$$
\mathbb{H}_2 = \frac{\partial \bar{\boldsymbol{\sigma}}}{\partial \bar{\boldsymbol{\varepsilon}}} \, . \tag{B.29}
$$

Using the bijective operator $[\bullet]$, see Appendix B.1 for details, and the property

$$
\left[\mathbf{b} \otimes^s \mathbf{N}\right] = \mathbf{H}\left[\mathbf{b}\right] \, , \tag{B.30}
$$

where the matrix $\mathbf{H}$ depends only on $\mathbf{N}$, we refer to Eq. (B.7) in Appendix B.1 for its detailed expression, Eqs. (B.22, B.23) become

$$
\begin{cases}
\left[\bar{\boldsymbol{\varepsilon}}^0\right] & = \left[\bar{\boldsymbol{\varepsilon}}\right] + \dfrac{1}{f^0}\mathbf{H}\left[\mathbf{b}\right] \\
\left[\bar{\boldsymbol{\varepsilon}}^1\right] & = \left[\bar{\boldsymbol{\varepsilon}}\right] - \dfrac{1}{f^1}\mathbf{H}\left[\mathbf{b}\right]
\end{cases} . \tag{B.31}
$$

Moreover, Eq. (B.24) is rewritten as

$$
\mathbf{H}^T\left(\left[\bar{\mathbb{C}}^0\right]\left[\bar{\boldsymbol{\varepsilon}}^0\right] - \left[\bar{\mathbb{C}}^1\right] : \left[\bar{\boldsymbol{\varepsilon}}^1\right]\right) = \mathbf{0} \, , \tag{B.32}
$$

in which the property $\left[\boldsymbol{\sigma} \cdot \mathbf{N}\right] = \mathbf{H}^T\left[\boldsymbol{\sigma}\right]$ is used, see Appendix B.1 for details. Using Eqs. (B.31), Eq. (B.32) becomes

$$
\mathbf{H}^T\left(\left[\bar{\mathbb{C}}^0\right] - \left[\bar{\mathbb{C}}^1\right]\right)\left[\bar{\boldsymbol{\varepsilon}}\right] + \mathbf{H}^T\left(\frac{1}{f^0}\bar{\mathbb{C}}^0 + \frac{1}{f^1}\bar{\mathbb{C}}^1\right)\mathbf{H}\left[\mathbf{b}\right] = \mathbf{0} \, . \tag{B.33}
$$

The equation above leads to the solution of $\mathbf{b}$:

$$
\left[\mathbf{b}\right] = -f^0 f^1 \mathbf{S}^{-1}\mathbf{H}^T\left(\left[\bar{\mathbb{C}}^0\right] - \left[\bar{\mathbb{C}}^1\right]\right)\left[\bar{\boldsymbol{\varepsilon}}\right] \, , \tag{B.34}
$$

where

$$\mathbf{S} = \mathbf{H}^T \left( f^1 \left[ \bar{\mathbb{C}}^0 \right] + f^0 \left[ \bar{\mathbb{C}}^1 \right] \right) \mathbf{H} \, . \tag{B.35}$$

As a result, Eq. (B.29) can be detailed as

$$\mathbb{H}_2 \equiv \left[ \bar{\mathbb{C}} \right] = \frac{\partial \left[ \bar{\boldsymbol{\sigma}} \right]}{\partial \left[ \bar{\boldsymbol{\varepsilon}} \right]} = f^0 \left[ \bar{\mathbb{C}}^0 \right] + f^1 \left[ \bar{\mathbb{C}}^1 \right] - f^0 f^1 \left( \left[ \bar{\mathbb{C}}^0 \right] - \left[ \bar{\mathbb{C}}^1 \right] \right) \mathbf{Q} \left( \left[ \bar{\mathbb{C}}^0 \right] - \left[ \bar{\mathbb{C}}^1 \right] \right) \, , \tag{B.36}$$

where

$$\mathbf{Q} = \mathbf{H} \mathbf{S}^{-1} \mathbf{H}^T \, . \tag{B.37}$$

Equation (B.36) can be rewritten under the indexed form as

$$\left[ \bar{\mathbb{C}} \right]_{ij} = f^0 \left[ \bar{\mathbb{C}}^0 \right]_{ij} + f^1 \left[ \bar{\mathbb{C}}^1 \right]_{ij} - f^0 f^1 \left[ \Delta \bar{\mathbb{C}} \right]_{ik} Q_{kl} \left[ \Delta \bar{\mathbb{C}} \right]_{lj} \, , \tag{B.38}$$

where $\Delta \bar{\mathbb{C}} = \bar{\mathbb{C}}^0 - \bar{\mathbb{C}}^1$, and the indexes range from 0 to 5 as these matrices are expressed in the 6-dimension space. This consideration is applied in the remaining part of this section.

The derivatives of $\left[ \bar{\mathbb{C}} \right]$ with respect to the inputs in the relation (B.38) are computed as follows:

(I) With respect to $\mathbb{C}^0$ and $\mathbb{C}^1$:

$$\begin{aligned} \frac{\partial \left[ \bar{\mathbb{C}} \right]_{ij}}{\partial \left[ \bar{\mathbb{C}}^0 \right]_{pq}} = \quad & f^0 \mathbb{I}_{ijpq} - f^0 f^1 \mathbb{I}_{ikpq} Q_{kl} \left[ \Delta \bar{\mathbb{C}} \right]_{lj} - f^0 f^1 \left[ \Delta \bar{\mathbb{C}} \right]_{ik} Q_{kl} \mathbb{I}_{ljpq} \\ & - f^0 f^1 \left[ \Delta \bar{\mathbb{C}} \right]_{ik} \frac{\partial Q_{kl}}{\partial \left[ \bar{\mathbb{C}}^0 \right]_{pq}} \left[ \Delta \bar{\mathbb{C}} \right]_{lj} \, , \text{ and} \end{aligned} \tag{B.39}$$

$$\begin{aligned} \frac{\partial \left[ \bar{\mathbb{C}} \right]_{ij}}{\partial \left[ \bar{\mathbb{C}}^1 \right]_{pq}} = \quad & f^1 \mathbb{I}_{ijpq} + f^0 f^1 \mathbb{I}_{ikpq} Q_{kl} \left[ \Delta \bar{\mathbb{C}} \right]_{lj} + f^0 f^1 \left[ \Delta \bar{\mathbb{C}} \right]_{ik} Q_{kl} \mathbb{I}_{ljpq} \\ & - f^0 f^1 \left[ \Delta \bar{\mathbb{C}} \right]_{ik} \frac{\partial Q_{kl}}{\partial \left[ \bar{\mathbb{C}}^1 \right]_{pq}} \left[ \Delta \bar{\mathbb{C}} \right]_{lj} \, , \end{aligned} \tag{B.40}$$

where $\mathbb{I}$ is the symmetric fourth order tensor in a 6-dimension space, *i.e.*

$$\mathbb{I}_{ijpq} = \frac{1}{2} \left( \delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right) \text{ with } \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \, , \tag{B.41}$$

and the terms $\dfrac{\partial Q_{kl}}{\partial \left[ \bar{\mathbb{C}}^i \right]_{pq}}$ with $i = 0, 1$ are estimated using Eq. (B.37) as

$$\frac{\partial Q_{kl}}{\partial \left[ \bar{\mathbb{C}}^i \right]_{pq}} = H_{kr} \frac{\partial S^{-1}_{rs}}{\partial S_{ab}} \frac{\partial S_{ab}}{\partial \left[ \bar{\mathbb{C}}^i \right]_{pq}} H_{ls} \, . \tag{B.42}$$

In this last equation, one has

$$\frac{\partial S^{-1}_{ij}}{\partial S_{kl}} = -\frac{1}{2}\left(S^{-1}_{ik}S^{-1}_{lj} + S^{-1}_{il}S^{-1}_{kj}\right) , \tag{B.43}$$

and Eq. (B.35) results into

$$\frac{\partial S_{ij}}{\partial \left[\bar{\mathbb{C}}^0\right]_{pq}} = f_1 H_{ki}\mathbb{I}_{klpq}H_{lj} \, , \text{ and} \tag{B.44}$$

$$\frac{\partial S_{ij}}{\partial \left[\bar{\mathbb{C}}^1\right]_{pq}} = f_0 H_{ki}\mathbb{I}_{klpq}H_{lj} \, . \tag{B.45}$$

(II) With respect to $f^0$ and $f^1$:

$$\begin{aligned}\frac{\partial \left[\bar{\mathbb{C}}\right]_{ij}}{\partial f^0} &= \left[\bar{\mathbb{C}}^0\right]_{ij} - f^1\left[\Delta\bar{\mathbb{C}}\right]_{ik} Q_{kl}\left[\Delta\bar{\mathbb{C}}\right]_{lj} \\ &\quad - f^0 f^1\left[\Delta\bar{\mathbb{C}}\right]_{ik}\frac{\partial Q_{kl}}{\partial f^0}\left[\Delta\bar{\mathbb{C}}\right]_{lj} \, , \text{ and}\end{aligned} \tag{B.46}$$

$$\begin{aligned}\frac{\partial \left[\bar{\mathbb{C}}\right]_{ij}}{\partial f^1} &= \left[\bar{\mathbb{C}}^1\right]_{ij} - f^0\left[\Delta\mathbb{C}\right]_{ik} Q_{kl}\left[\Delta\bar{\mathbb{C}}\right]_{lj} \\ &\quad - f^0 f^1\left[\Delta\bar{\mathbb{C}}\right]_{ik}\frac{\partial Q_{kl}}{\partial f^1}\left[\Delta\bar{\mathbb{C}}\right]_{lj} \, ,\end{aligned} \tag{B.47}$$

where

$$\frac{\partial Q_{kl}}{\partial f^i} = H_{kr}\frac{\partial S^{-1}_{rs}}{\partial S_{ab}}\frac{\partial S_{ab}}{\partial f^i}H_{ls} \text{ with } i = 0, 1 \, . \tag{B.48}$$

In the last equation, the term $\dfrac{\partial S^{-1}_{rs}}{\partial S_{ab}}$ is estimated using Eq. (B.43), and one has

$$\frac{\partial S_{ij}}{\partial f^0} = H_{ki}\left[\bar{\mathbb{C}}^1\right]_{kl}H_{lj} \, , \text{ and } \frac{\partial S_{ij}}{\partial f^1} = H_{ki}\left[\bar{\mathbb{C}}^0\right]_{kl}H_{lj} \, , \tag{B.49}$$

as a result of Eq. (B.35).

(III) With respect to $\mathbf{N}$:

$$\frac{\partial \left[\bar{\mathbb{C}}\right]_{ij}}{\partial N_p} = -f^0 f^1\left[\Delta\bar{\mathbb{C}}\right]_{ik}\frac{\partial Q_{kl}}{\partial N_p}\left[\Delta\bar{\mathbb{C}}\right]_{lj} \, , \tag{B.50}$$

where $\dfrac{\partial Q_{ij}}{\partial N_p}$ is estimated using Eq. (B.37) as

$$\frac{\partial Q_{ij}}{\partial N_p} = \frac{\partial H_{ik}}{\partial N_p}S^{-1}_{kl}H_{jl} + H_{ik}S^{-1}_{kl}\frac{\partial H_{jl}}{\partial N_p} + H_{ik}\frac{\partial S^{-1}_{kl}}{\partial S_{rs}}\frac{\partial S_{rs}}{\partial N_p}H_{jl} \, . \tag{B.51}$$

In this last equation, the term $\dfrac{\partial S^{-1}_{kl}}{\partial S_{rs}}$ is given by Eq. (B.43) and one has as a result of Eq. (B.35) the following equation

$$\begin{aligned}\frac{\partial S_{ij}}{\partial N_p} &= \frac{\partial H_{ki}}{\partial N_p}\left(f^1\left[\bar{\mathbb{C}}^0\right]_{kl} + f^0\left[\bar{\mathbb{C}}^1\right]_{kl}\right)H_{lj} \\ &\quad + H_{ki}\left(f^1\left[\bar{\mathbb{C}}^0\right]_{kl} + f^0\left[\bar{\mathbb{C}}^1\right]_{kl}\right)\frac{\partial H_{lj}}{\partial N_p}\,,\end{aligned} \tag{B.52}$$

where the expression of $\mathbf{H}$ is given in Eq. (B.7), leading to

- For the plane strain state:

$$\frac{\partial \mathbf{H}}{\partial N_0} = \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix}, \frac{\partial \mathbf{H}}{\partial N_1} = \begin{bmatrix}0 & 0\\ 0 & 1\\ 1 & 0\end{bmatrix}, \text{ and } \frac{\partial \mathbf{H}}{\partial N_2} = \mathbf{0}\,. \tag{B.53}$$

- For the general state:

$$\frac{\partial \mathbf{H}}{\partial N_0} = \begin{bmatrix}1 & 0 & 0 & \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & \end{bmatrix}, \frac{\partial \mathbf{H}}{\partial N_1} = \begin{bmatrix}0 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 0 & \\ 1 & 0 & 0 & \\ 0 & 0 & 0 & \\ 0 & 0 & 1 & \end{bmatrix}, \text{ and } \frac{\partial \mathbf{H}}{\partial N_2} = \begin{bmatrix}0 & 0 & 0 & \\ 0 & 0 & 0 & \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & \\ 1 & 0 & 0 & \\ 0 & 1 & 0 & \end{bmatrix}. \tag{B.54}$$

*Appendix B.3.2. Analytical solution for a multiple-list interaction*

The operator $\mathbb{H}_{n^p_d}$ with $n^p_d > 2$ of a multiple-list interaction can be obtained as an extension of the solution $\mathbb{H}_2$ of a two-list interaction. Since $\mathcal{K}^{r_m}_{d+1} \cap \mathcal{K}^{r_m+1}_{d+1} = \emptyset$, the two first equations in the set of Eqs. (B.13) lead to the following relations

$$\begin{aligned}\bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}^{r_m}_{d+1} \cup \mathcal{K}^{r_m+1}_{d+1}\right) &= \frac{\mathcal{S}\left(\mathcal{K}^{r_m}_{d+1}\right)}{\mathcal{S}\left(\mathcal{K}^{r_m}_{d+1} \cup \mathcal{K}^{r_m+1}_{d+1}\right)}\bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}^{r_m}_{d+1}\right) + \frac{\mathcal{S}\left(\mathcal{K}^{r_m+1}_{d+1}\right)}{\mathcal{S}\left(\mathcal{K}^{r_m}_{d+1} \cup \mathcal{K}^{r_m+1}_{d+1}\right)}\bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}^{r_m+1}_{d+1}\right) \\ &= \bar{\boldsymbol{\varepsilon}}\left(\mathcal{K}^p_d\right) + \frac{1}{\mathcal{S}\left(\mathcal{K}^{r_m}_{d+1} \cup \mathcal{K}^{r_m+1}_{d+1}\right)}\mathbf{a}^{q_m+1} \otimes \mathbf{N}^m\,,\end{aligned} \tag{B.55}$$

and

$$\begin{aligned}\bar{\boldsymbol{\sigma}}\left(\mathcal{K}^{r_m}_{d+1} \cup \mathcal{K}^{r_m+1}_{d+1}\right) &= \frac{\mathcal{S}\left(\mathcal{K}^{r_m}_{d+1}\right)}{\mathcal{S}\left(\mathcal{K}^{r_m}_{d+1} \cup \mathcal{K}^{r_m+1}_{d+1}\right)}\bar{\boldsymbol{\sigma}}\left(\mathcal{K}^{r_m}_{d+1}\right) + \frac{\mathcal{S}\left(\mathcal{K}^{r_m+1}_{d+1}\right)}{\mathcal{S}\left(\mathcal{K}^{r_m}_{d+1} \cup \mathcal{K}^{r_m+1}_{d+1}\right)}\bar{\boldsymbol{\sigma}}\left(\mathcal{K}^{r_m+1}_{d+1}\right) \\ &= \frac{f^{p,0}_d}{f^{p,0}_d + f^{p,1}_d}\bar{\boldsymbol{\sigma}}\left(\mathcal{K}^{r_m}_{d+1}\right) + \frac{f^{p,1}_d}{f^{p,0}_d + f^{p,1}_d}\bar{\boldsymbol{\sigma}}\left(\mathcal{K}^{r_m+1}_{d+1}\right)\,.\end{aligned} \tag{B.56}$$

Moreover, using this last result, the balance condition expressed in Eq. (B.14) for the two lists $\mathcal{K}_{d+1}^{r_m}$, $\mathcal{K}_{d+1}^{r_m+1}$ reads

$$\bar{\boldsymbol{\sigma}}\left(\mathcal{K}_{d+1}^{r_m}\cup\mathcal{K}_{d+1}^{r_m+1}\right)\cdot\mathbf{N}^m=\bar{\boldsymbol{\sigma}}\left(\mathcal{K}_{d+1}^{r_m}\right)\cdot\mathbf{N}^m=\bar{\boldsymbol{\sigma}}\left(\mathcal{K}_{d+1}^{r_m+1}\right)\cdot\mathbf{N}^m\,. \tag{B.57}$$

Consequently, Eqs. (B.55, B.56, B.57) imply that the interaction between $\mathcal{K}_{d+1}^{r_m}$, $\mathcal{K}_{d+1}^{r_m+1}$, $\mathcal{K}_{d+1}^{r_m+2}$, …, $\mathcal{K}_{d+1}^{r_m+n_d^p-1}$ with $n_d^p>2$ is equivalent to the interaction between $\mathcal{K}_{d+1}^{r_m}\cup\mathcal{K}_{d+1}^{r_m+1}$, $\mathcal{K}_{d+1}^{r_m+2}$, …, $\mathcal{K}_{d+1}^{r_m+n_d^p-1}$, *i.e.* Eq. (B.20) can be rewritten as

$$\begin{aligned}\left[\bar{\mathbb{C}}\left(\mathcal{K}_d^p;\boldsymbol{\mathcal{F}}\right)\right]=\mathbb{H}_{n_d^p-1}\Bigg(&\left[\bar{\mathbb{C}}\left(\mathcal{K}_{d+1}^{r_m}\cup\mathcal{K}_{d+1}^{r_m+1};\boldsymbol{\mathcal{F}}\right)\right],\left[\bar{\mathbb{C}}\left(\mathcal{K}_{d+1}^{r_m+2};\boldsymbol{\mathcal{F}}\right)\right],\dots,\left[\bar{\mathbb{C}}\left(\mathcal{K}_{d+1}^{r_m+n_d^p-1};\boldsymbol{\mathcal{F}}\right)\right]\\&,f_d^{p,0}+f_d^{p,1},f_d^{p,2},\dots,f_d^{p,n_d^p-1},\mathbf{N}^m\Bigg)\,,\end{aligned} \tag{B.58}$$

where $\left[\bar{\mathbb{C}}\left(\mathcal{K}_{d+1}^{r_m}\cup\mathcal{K}_{d+1}^{r_m+1};\boldsymbol{\mathcal{F}}\right)\right]$ corresponds to the solution of the two-list interaction between $\mathcal{K}_{d+1}^{r_m}$ and $\mathcal{K}_{d+1}^{r_m+1}$ as

$$\left[\bar{\mathbb{C}}\left(\mathcal{K}_{d+1}^{r_m}\cup\mathcal{K}_{d+1}^{r_m+1};\boldsymbol{\mathcal{F}}\right)\right]=\mathbb{H}_2\left(\left[\bar{\mathbb{C}}\left(\mathcal{K}_{d+1}^{r_m};\boldsymbol{\mathcal{F}}\right)\right],\left[\bar{\mathbb{C}}\left(\mathcal{K}_{d+1}^{r_m+1};\boldsymbol{\mathcal{F}}\right)\right],\widehat{f_d^{p,0}},\widehat{f_d^{p,1}},\mathbf{N}^m\right)\,, \tag{B.59}$$

with

$$\widehat{f_d^{p,0}}=\frac{f_d^{p,0}}{f_d^{p,0}+f_d^{p,1}}\text{ and }\widehat{f_d^{p,1}}=\frac{f_d^{p,1}}{f_d^{p,0}+f_d^{p,1}}=1-\widehat{f_d^{p,0}}\,. \tag{B.60}$$

Clearly, Eq. (B.58) can be recursively applied until reaching the operator $\mathbb{H}_2$. Therefore it suffices to provide the solution for the case $\mathbb{H}_2$ in order to evaluate Eq. (B.20) in the case $n_d^p>2$ by recursively applying Eq. (B.58). Equation (B.59) corresponds to an explicit form of the functional operator $\mathbb{H}_2$ given in Appendix B.3.1, in which the derivatives $\frac{\partial\mathbb{H}_2}{\partial\left[\bar{\mathbb{C}}_{d+1}^{r_m}\right]}$, $\frac{\partial\mathbb{H}_2}{\partial\left[\bar{\mathbb{C}}_{d+1}^{r_m+1}\right]}$, $\frac{\partial\mathbb{H}_2}{\partial\widehat{f_d^{p,0}}}$, $\frac{\partial\mathbb{H}_2}{\partial\widehat{f_d^{p,1}}}$, $\frac{\partial\mathbb{H}_2}{\partial N_0^m}$, $\frac{\partial\mathbb{H}_2}{\partial N_1^m}$, and $\frac{\partial\mathbb{H}_2}{\partial N_2^m}$ are detailed.

Equation (B.20) is evaluated by recursively applying Eqs. (B.58, B.59). For ease of implementation, in the remaining of this section, only indexes representing the order of the sub-material networks in Eq. (B.20) are kept, *i.e.*

$$\mathbb{H}_n\;=\;\mathbb{H}_n\left(\left[\bar{\mathbb{C}}^0\right],\dots,\left[\bar{\mathbb{C}}^{n-1}\right],f^0,\dots,f^{n-1},\mathbf{N}\right)\,. \tag{B.61}$$

When $n>2$, Eq. (B.58) is rewritten as

$$\mathbb{H}_n\;=\;\mathbb{H}_{n-1}\left(\mathbb{H}_2,\underbrace{\left[\bar{\mathbb{C}}^2\right],\dots,\left[\bar{\mathbb{C}}^{n-1}\right]}_{\text{remaining }n-2\text{ sub-lists}},f^0+f^1,\underbrace{f^2,\dots,f^{n-1}}_{\text{remaining }n-2\text{ sub-lists}},\mathbf{N}\right)\,. \tag{B.62}$$

where $\mathbb{H}_2$ is detailed by Eq. (B.36) and can be rewritten under a functional form

$$\mathbb{H}_2\;=\;\mathbb{H}_2\left(\left[\bar{\mathbb{C}}^0\right],\left[\bar{\mathbb{C}}^1\right],\frac{f^0}{f^0+f^1},\frac{f^1}{f^0+f^1},\mathbf{N}\right)\,. \tag{B.63}$$

In order to evaluate $\mathbb{H}_n$, Eq. (B.63) is applied for the $(k+1)^{th}$ sub-list and the material network made from all sub-lists indexed from 0 to $k$, and starting for $k=0$ until reaching $k=n-2$. This procedure is detailed as follows:

(I) start with $\left[\bar{\mathbb{C}}^{0,0}\right] = \left[\bar{\mathbb{C}}^{0}\right]$;

(II) **for** $k$ **from** $0$ **to** $n-2$:

  (i) compute the function value:

$$\left[\bar{\mathbb{C}}^{0,k+1}\right] = \mathbb{H}_2\left(\left[\bar{\mathbb{C}}^{0,k}\right], \left[\bar{\mathbb{C}}^{k+1}\right], f^{0,k+1}, f^{1,k+1}, \mathbf{N}\right), \tag{B.64}$$

  where the function $\mathbb{H}_2$ is given in Eq. (B.36) and

$$f^{0,k+1} = \frac{\sum_{l=0}^{k} f^l}{\sum_{l=0}^{k+1} f^l} \text{ and } f^{1,k+1} = \frac{f^{k+1}}{\sum_{l=0}^{k+1} f^l} = 1 - f^{0,k+1}; \tag{B.65}$$

  (ii) compute the derivatives of the elastic operator (B.64) using the derivatives of the operator $\mathbb{H}_2$ explicited in Appendix B.3.1:

$$\mathbb{A}^k = \frac{\partial\left[\bar{\mathbb{C}}^{0,k+1}\right]}{\partial\left[\bar{\mathbb{C}}^{0,k}\right]}, \mathbb{B}^k = \frac{\partial\left[\bar{\mathbb{C}}^{0,k+1}\right]}{\partial\left[\bar{\mathbb{C}}^{k+1}\right]}, \mathbb{F}_0^k = \frac{\partial\left[\bar{\mathbb{C}}^{0,k+1}\right]}{\partial f^{0,k+1}}, \mathbb{F}_1^k = \frac{\partial\left[\bar{\mathbb{C}}^{0,k+1}\right]}{\partial f^{1,k+1}},$$
$$\mathbb{N}_0^k = \frac{\partial\left[\bar{\mathbb{C}}^{0,k+1}\right]}{\partial N_0}, \mathbb{N}_1^k = \frac{\partial\left[\bar{\mathbb{C}}^{0,k+1}\right]}{\partial N_1}, \text{ and } \mathbb{N}_2^k = \frac{\partial\left[\bar{\mathbb{C}}^{0,k+1}\right]}{\partial N_2},$$

  in which $\mathbb{A}^k$ and $\mathbb{B}^k$ are viewed as fourth-order symmetric tensors while $\mathbb{F}_0^k$, $\mathbb{F}_1^k$, $\mathbb{N}_0^k$, $\mathbb{N}_1^k$, and $\mathbb{N}_2^k$ are viewed as second-order symmetric tensors in the 6-dimensional space. The tensor operations with these quantities are then simply extended from the ones in the conventional 3-dimensional space by varying the indexes from 0 to 5;

(III) evaluate the homogenized material tensor $\mathbb{H}_n = \left[\bar{\mathbb{C}}^{0,n-1}\right]$;

(IV) compute the derivatives of $\mathbb{H}_n$ with respect to the inputs

(i) with respect to $\bar{\mathbb{C}}^i$ with $i = 0, \ldots, n-1$:

$$\frac{\partial \mathbb{H}_n}{\partial \left[\bar{\mathbb{C}}^{n-1}\right]} = \mathbb{B}^{n-2}\,, \tag{B.66}$$

$$\frac{\partial \mathbb{H}_n}{\partial \left[\bar{\mathbb{C}}^{n-2}\right]} = \mathbb{A}^{n-2} : \mathbb{B}^{n-3}\,, \tag{B.67}$$

$$\frac{\partial \mathbb{H}_n}{\partial \left[\bar{\mathbb{C}}^{n-3}\right]} = \mathbb{A}^{n-2} : \mathbb{A}^{n-3} : \mathbb{B}^{n-4}\,, \tag{B.68}$$

$$\vdots \tag{B.69}$$

$$\frac{\partial \mathbb{H}_n}{\partial \left[\bar{\mathbb{C}}^{1}\right]} = \mathbb{A}^{n-2} : \mathbb{A}^{n-3} : \cdots \mathbb{A}^{1} : \mathbb{B}^{0}\,, \text{ and} \tag{B.70}$$

$$\frac{\partial \mathbb{H}_n}{\partial \left[\bar{\mathbb{C}}^{0}\right]} = \mathbb{A}^{n-2} : \mathbb{A}^{n-3} : \cdots : \mathbb{A}^{0}\,; \tag{B.71}$$

(ii) with respect to $f^k$ with $k = 0, \ldots, n-1$:

$$\begin{aligned} \frac{\partial \mathbb{H}_n}{\partial f^k} = {} & \mathbb{F}_0^{n-2} \frac{\partial f_0^{0,n-1}}{\partial f^k} + \mathbb{F}_1^{n-2} \frac{\partial f^{(1,n-1)}}{\partial f^k} \\ & + \mathbb{A}^{n-2} : \left( \mathbb{F}_0^{n-3} \frac{\partial f^{(0,n-2)}}{\partial f^k} + \mathbb{F}_1^{n-3} \frac{\partial f^{(1,n-2)}}{\partial f^k} \right) \\ & + \mathbb{A}^{n-2} : \mathbb{A}^{n-3} : \left( \mathbb{F}_0^{n-4} \frac{\partial f^{0,n-3}}{\partial f^k} + \mathbb{F}_1^{n-4} \frac{\partial f^{1,n-3}}{\partial f^k} \right) \\ & \vdots \\ & + \mathbb{A}^{n-2} : \mathbb{A}^{n-3} : \cdots : \mathbb{A}^{1} : \left( \mathbb{F}_0^{0} \frac{\partial f^{0,1}}{\partial f^k} + \mathbb{F}_1^{0} \frac{\partial f^{1,1}}{\partial f^k} \right)\,, \end{aligned} \tag{B.72}$$

where the derivatives $\frac{\partial f^{0,l}}{\partial f^k}$ and $\frac{\partial f^{1,l}}{\partial f^k}$ with $l = 1, \ldots, n-1$ and $k = 0, \ldots, n-1$ can be easily estimated using Eq. (B.65);

(iii) with respect to $\mathbf{N}$:

$$\begin{aligned} \frac{\partial \mathbb{H}_n}{\partial N_l} = {} & \mathbb{N}_l^{n-2} + \mathbb{A}^{n-2} : \mathbb{N}_l^{n-3} + \mathbb{A}^{n-2} : \mathbb{A}^{n-3} : \mathbb{N}_l^{n-4} + \\ & \ldots + \mathbb{A}^{n-2} : \mathbb{A}^{n-3} : \cdots : \mathbb{A}^{1} : \mathbb{N}_l^{0} \text{ with } l = 0, 1, 2\,; \end{aligned} \tag{B.73}$$

(IV) exit.

Using the procedure described here above, the function (B.20) and its derivatives required for the DMN training are implicitly available.

### *Appendix B.4. Homogenized tangent operator and its derivative with respect to the fitting parameters*

Since each sub-list in the right hand side of Eq. (66) continues to be sub-divided through network interactions until all the sub-lists contain only one material node, the material tensor

$\bar{\mathbb{C}}\left(\mathcal{K};\boldsymbol{\mathcal{F}}\right) \equiv \bar{\mathbb{C}} \equiv \bar{\mathbb{C}}_0^0$ can be estimated by recursively applying Eq. (66) depth-by-depth from the deepest depth, where the node elastic tangent operators are known. A procedure for estimating both $\bar{\mathbb{C}}$ and $\partial\bar{\mathbb{C}}/\partial\boldsymbol{\mathcal{F}}$ based on the depth lists described by Eq. (38) is proposed. The evaluation procedure starts from the depth list $\mathcal{D}_{d_{\max}-1}$ (since $\mathcal{D}_{d_{\max}}$ does not contain any network interaction) and progresses to the depth list $\mathcal{D}_0$ as follows:

(I) initialize $d = d_{\max} - 1$;

(II) **for** each sub material network $\mathcal{K}_d^p$ in the list $\mathcal{D}_d$:

(i) **if** $\mathcal{K}_d^p$ contains only one material node $i$:

$$\left[\mathbb{C}_d^p\right] = \left[\mathbb{C}^i\right] \text{ and } \frac{\partial\left[\mathbb{C}_d^p\right]}{\partial\boldsymbol{\mathcal{F}}} = \mathbf{0}\,. \tag{B.74}$$

(ii) **if** $\mathcal{K}_d^p$ contains more than one material node:

(a) there exists a network interaction $m$ whose interaction direction is $\mathbf{N}^m$ and which is created from $n_d^p$ ($n_d^p \geq 2$) non-overlapping sub-lists $\mathcal{K}_{d+1}^{r_m+k}$ with $k = 0\,,\ldots\,, n_d^p - 1$ and $r_m = \sum_{k=0}^{p-1} n_d^k$ denoting the location of the first sub-list of this interaction at the depth $d+1$, and satisfying $\mathcal{K}_d^p = \mathcal{K}_{d+1}^{r_m} \cup \ldots \cup \mathcal{K}_{d+1}^{r_m+n_d^p-1}$;

(b) compute $\left[\mathbb{C}_d^p\right]$ following Eq. (66);

(c) compute $\dfrac{\partial\left[\mathbb{C}_d^p\right]}{\partial\boldsymbol{\mathcal{F}}}$ as follows:

$$\frac{\partial\left[\mathbb{C}_d^p\right]}{\partial\boldsymbol{\mathcal{F}}} = \sum_{k=0}^{n_d^p-1} \frac{\partial\left[\mathbb{C}_d^p\right]}{\partial\left[\mathbb{C}_{d+1}^{r_m+k}\right]} : \frac{\partial\left[\mathbb{C}_{d+1}^{r_m+k}\right]}{\partial\boldsymbol{\mathcal{F}}} + \sum_{k=0}^{n_d^p-1} \frac{\partial\left[\mathbb{C}_d^p\right]}{\partial f_d^{p,k}} \frac{\partial f_d^{p,k}}{\partial\boldsymbol{\mathcal{F}}} + \sum_{l=0}^{2} \frac{\partial\left[\mathbb{C}_d^p\right]}{\partial N_l^j} \frac{\partial N_l^m}{\partial\boldsymbol{\mathcal{F}}}\,, \tag{B.75}$$

in which

(a) the values of $\left[\mathbb{C}_{d+1}^{r_m+k}\right]$ and of $\frac{\partial\left[\mathbb{C}_{d+1}^{r_m+k}\right]}{\partial\boldsymbol{\mathcal{F}}}$ with $k = 0\,,\ldots\,, n_d^p - 1$ are known from the recursive process;

(b) the fourth-order tensor notation is an abuse of notation for the term $\frac{\partial\left[\mathbb{C}_d^p\right]}{\partial\left[\mathbb{C}_{d+1}^{r_m+k}\right]}$ and the double-dot product (:) is applied for this fourth-order tensor and each component of the block vector $\frac{\partial\left[\mathbb{C}_{d+1}^{r_m+k}\right]}{\partial\boldsymbol{\mathcal{F}}}$;

(c) the terms $\frac{\partial\left[\mathbb{C}_d^p\right]}{\partial\left[\mathbb{C}_{d+1}^{r_m+k}\right]}$, $\frac{\partial\left[\mathbb{C}_d^p\right]}{\partial f_d^{p,k}}$ with $k = 0, \ldots, n_d^p - 1$ and $\frac{\partial\left[\mathbb{C}_d^p\right]}{\partial N_l^m}$ with $l = 0, 1, 2$ result from the solution of the network interaction given by Eq. (66) and are computed following Appendix B.3;

(d) the terms $\frac{\partial f_d^{p,k}}{\partial\boldsymbol{\mathcal{F}}}$ with $k = 0, \ldots, n_d^p - 1$ can be easily estimated from Eqs. (59, 67) which are rewritten as

$$f_d^{p,k} = \frac{\sum_{i\in\mathcal{K}_{d+1}^{r_m+k}} \mathrm{relu}^{\mathrm{s}}\left(z^i\right)}{\sum_{i\in\mathcal{K}_d^p} \mathrm{relu}^{\mathrm{s}}\left(z^i\right)}\,, \tag{B.76}$$

in which the positive value of $W^i = \text{relu}^{\text{s}}\left(z^i\right)$ is enforced;

(e) the term $\frac{\partial N_l^j}{\partial \boldsymbol{\mathcal{F}}}$ with $l = 0, 1, 2$ is estimated using Eq. (60);

(III) **if** $d = 0$, go to (IV) **else** assign $d = d - 1$ and go to (II);

(IV) exit.

## Appendix C. $J_2$ elastoplastic model at finite strains

The deformation gradient $\mathbf{F}$ is decomposed into its reversible elastic part $\mathbf{F}^{\text{e}}$ and its irreversible plastic part $\mathbf{F}^{\text{p}}$ such that $\mathbf{F} = \mathbf{F}^{\text{e}} \cdot \mathbf{F}^{\text{p}}$. The elastic potential energy is defined as

$$\psi(\mathbf{C}^{\text{e}}) = \frac{K}{2} \ln^2 J + \frac{\mu}{4} \left(\ln \mathbf{C}^{\text{e}}\right)^{\text{dev}} : \left(\ln \mathbf{C}^{\text{e}}\right)^{\text{dev}} , \tag{C.1}$$

where $\mathbf{C}^{\text{e}} = \mathbf{F}^{\text{eT}} \cdot \mathbf{F}^{\text{e}}$, and $K$, and $\mu$ correspond to the bulk and shear modulii of the material. The first Piola-Kirchhoff stress tensor $\mathbf{P}$ derives from the elastic potential (C.1) under an elastic state, *i.e.* at constant $\mathbf{F}^{\text{p}}$, yielding

$$\mathbf{P} = \frac{\partial \psi\left(\mathbf{F}; \mathbf{F}^{\text{p}}\right)}{\partial \mathbf{F}} = K \mathbf{F}^{\text{-T}} \ln J + \mu \mathbf{F}^{\text{e}} \cdot \left[\mathbf{C}^{\text{e}\,-1} \cdot \left(\ln \mathbf{C}^{\text{e}}\right)^{\text{dev}}\right] \cdot \mathbf{F}^{\text{p}\,\text{-T}} . \tag{C.2}$$

The elastic part $\mathbf{F}^{\text{e}}$ and the plastic part $\mathbf{F}^{\text{p}}$ of the deformation gradient are obtained through a $J_2$ plastic flow expressed in terms of the Kirchhoff stress. The Kirchhoff stress $\boldsymbol{\kappa} = \mathbf{P} \cdot \mathbf{F}^{\text{T}}$ is first computed using Eq. (C.2) as

$$\boldsymbol{\kappa} = K \ln J \mathbf{I} + \mu \mathbf{F}^{\text{e}} \cdot \left[\mathbf{C}^{\text{e}-1} \cdot \left(\ln \mathbf{C}^{\text{e}}\right)^{\text{dev}}\right] \cdot \mathbf{F}^{\text{e}\,\text{T}} . \tag{C.3}$$

The equivalent von Mises stress is then calculated through the deviatoric part of $\boldsymbol{\kappa}$, *i.e.* $\tau_{eq} = \sqrt{\frac{3}{2} \boldsymbol{\kappa}^{\text{dev}} : \boldsymbol{\kappa}^{\text{dev}}}$. According to the $J_2$-plasticity theory, the von Mises stress criterion reads

$$f = \tau_{\text{eq}} - \tau_{\text{y}}^0 - R(\gamma) \leq 0 , \tag{C.4}$$

where $f$ is the yield surface, $\tau_{\text{y}}^0$ is the initial yield stress, $\gamma$ is the equivalent plastic strain and $R(\gamma)$ is the isotropic hardening stress. The evolution of $\mathbf{F}^{\text{p}}$ is governed by the associative plastic flow theory following

$$\dot{\mathbf{F}}^{\text{p}} = \dot{\gamma} \mathbf{N} \cdot \mathbf{F}^{\text{p}} , \tag{C.5}$$

where $\mathbf{N}$ is the normal to the yield surface, see [43] for more details.

The input parameters of this $J_2$ plasticity model including the bulk modulus $K$, the shear modulus $\mu$, and the parameters specifying the isotropic flow stress $\tau_{\text{y}}\left(\gamma\right) = \tau_{\text{y}}^0 + R(\gamma)$. The elastic behavior can be degenerated by simply considering $\tau_{\text{y}}^0 = +\infty$.

The numerical integration of the plastic problem follows a predictor-corrector scheme as described in [43], in which the expresion of $\mathbf{L} = \frac{\partial \mathbf{P}}{\partial \mathbf{F}}$ is derived.

## References

[1] P. Kanouté, D. Boso, J. Chaboche, B. Schrefler, Multiscale methods for composites: a review, Archives of Computational Methods in Engineering 16 (1) (2009) 31–75 (2009).

[2] L. Noels, L. Wu, L. Adam, J. Seyfarth, G. Soni, J. Segurado, G. Laschet, G. Chen, M. Lesueur, M. Lobos, T. Bhlke, T. Reiter, S. Oberpeilsteiner, D. Salaberger, D. Weichert, C. Broeckmann, Effective Properties, John Wiley & Sons, Ltd, 2016, Ch. 6, pp. 433–485 (2016). doi:10.1002/9783527693566.ch6.

[3] M. G. Geers, V. G. Kouznetsova, K. Matouš, J. Yvonnet, Homogenization methods and multiscale modeling: nonlinear problems, Encyclopedia of Computational Mechanics Second Edition (2017) 1–34 (2017).

[4] K. Matouš, M. G. Geers, V. G. Kouznetsova, A. Gillman, A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials, Journal of Computational Physics 330 (2017) 192 – 220 (2017). doi:https://doi.org/10.1016/j.jcp.2016.10.070.

[5] 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).

[6] J. Hernández, J. Oliver, A. Huespe, M. Caicedo, J. Cante, High-performance model reduction techniques in computational multiscale homogenization, Computer Methods in Applied Mechanics and Engineering 276 (2014) 149 – 189 (2014). doi:https://doi.org/10.1016/j.cma.2014.03.011.

[7] 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 (2016). doi:https://doi.org/10.1016/j.cma.2016.04.004.

[8] 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 (2016). doi:https://doi.org/10.1016/j.jmps.2016.02.005.

[9] D. Soldner, B. Brands, R. Zabihyan, P. Steinmann, J. Mergheim, A numerical study of different projection-based model reduction techniques applied to computational homogenisation, Computational mechanics 60 (4) (2017) 613–625 (2017).

[10] M. J. Zahr, P. Avery, C. Farhat, A multilevel projection-based model order reduction framework for nonlinear dynamic multiscale problems in structural and solid mechanics, International Journal for Numerical Methods in Engineering 112 (8) (2017) 855–881 (2017). doi:10.1002/nme.5535.

[11] 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 (2017). doi:https://doi.org/10.1016/j.cma.2016.09.039.

[12] S. Wulfinghoff, F. Cavaliere, S. Reese, Model order reduction of nonlinear homogenization problems using a hashinshtrikman type finite element method, Computer Methods in Applied Mechanics and Engineering 330 (2018) 149 – 179 (2018). doi:https://doi.org/10.1016/j.cma.2017.10.019.

[13] I. Rocha, P. Kerfriden, F. van der Meer, Micromechanics-based surrogate models for the response of composites: A critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks, European Journal of Mechanics - A/Solids 82 (2020) 103995 (2020). doi:https://doi.org/10.1016/j.euromechsol.2020.103995.

[14] T. Furukawa, G. Yagawa, Implicit constitutive modelling for viscoplasticity using neural networks, International Journal for Numerical Methods in Engineering 43 (2) (1998) 195–219 (1998).

[15] C. Settgast, G. Htter, M. Kuna, M. Abendroth, A hybrid approach to simulate the homogenized irreversible elasticplastic deformations and damage of foams by neural networks, International Journal of Plasticity 126 (2020) 102624 (2020). doi:https://doi.org/10.1016/j.ijplas.2019.11.003.

[16] J. Ghaboussi, D. Sidarta, New nested adaptive neural networks (nann) for constitutive modeling, Computers and Geotechnics 22 (1) (1998) 29 – 52 (1998). doi:https://doi.org/10.1016/S0266-352X(97)00034-7.

[17] M. Lefik, B. Schrefler, Artificial neural network as an incremental non-linear constitutive model for a finite element code, Computer Methods in Applied Mechanics and Engineering 192 (28) (2003) 3265 – 3283, multiscale Computational Mechanics for Materials and Structures (2003). doi:https://doi.org/10.1016/S0045-7825(03)00350-5.

[18] S. Jung, J. Ghaboussi, Neural network constitutive model for rate-dependent materials, Computers & Structures 84 (15) (2006) 955 – 963 (2006). doi:https://doi.org/10.1016/j.compstruc.2006.02.015.

[19] B. A. 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 (Dec. 2015). doi:10.1002/nme.4953.

[20] M. Bessa, R. Bostanabad, Z. Liu, A. Hu, D. W. 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 (2017). doi:https://doi.org/10.1016/j.cma.2017.03.037.

[21] F. Fritzen, M. Fernndez, F. Larsson, On-the-fly adaptivity for nonlinear twoscale simulations using artificial neural networks and reduced order modeling, Frontiers in Materials 6 (2019) 75 (2019). doi:10.3389/fmats.2019.00075.

[22] H. Yang, X. Guo, S. Tang, W. K. Liu, Derivation of heterogeneous material laws via data-driven principal component expansions, Computational Mechanics 64 (2) (2019) 365–379 (2019).

[23] M. B. Gorji, M. Mozaffar, J. N. Heidenreich, J. Cao, D. Mohr, On the potential of recurrent neural networks for modeling path dependent plasticity, Journal of the Mechanics and Physics of Solids 143 (2020) 103972 (2020). doi:https://doi.org/10.1016/j.jmps.2020.103972.

[24] D. Huang, J. N. Fuhg, C. Weienfels, P. Wriggers, A machine learning based plasticity model using proper orthogonal decomposition, Computer Methods in Applied Mechanics and Engineering 365 (2020) 113008 (2020). doi:https://doi.org/10.1016/j.cma.2020.113008.

[25] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M. A. Bessa, Deep learning predicts path-dependent plasticity, Proceedings of the National Academy of Sciences 116 (52) (2019) 26414–26420 (2019). doi:10.1073/pnas.1911815116.

[26] F. Ghavamian, A. Simone, Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network, Computer Methods in Applied Mechanics and Engineering 357 (2019) 112594 (2019). doi:https://doi.org/10.1016/j.cma.2019.112594.

[27] L. Wu, V. D. Nguyen, N. G. Kilingar, L. Noels, A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths, Computer Methods in Applied Mechanics and Engineering 369 (2020) 113234 (2020). doi:https://doi.org/10.1016/j.cma.2020.113234.

[28] H. J. Logarzo, G. Capuano, J. J. Rimoli, Smart constitutive laws: Inelastic homogenization through machine learning, Computer Methods in Applied Mechanics and Engineering 373 (2021) 113482 (2021). doi:https://doi.org/10.1016/j.cma.2020.113482.

[29] Z. Liu, C. Wu, M. Koishi, A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials, Computer Methods in Applied Mechanics and Engineering 345 (2019) 1138 – 1168 (2019). doi:https://doi.org/10.1016/j.cma.2018.09.020.

[30] Z. Liu, C. Wu, Exploring the 3d architectures of deep material network in data-driven multiscale mechanics, Journal of the Mechanics and Physics of Solids 127 (2019) 20 – 46 (2019). doi:https://doi.org/10.1016/j.jmps.2019.03.004.

[31] Z. Liu, Deep material network with cohesive layers: Multi-stage training and interfacial failure analysis, Computer Methods in Applied Mechanics and Engineering 363 (2020) 112913 (2020). doi:https://doi.org/10.1016/j.cma.2020.112913.

[32] S. Gajek, M. Schneider, T. Bhlke, On the micromechanics of deep material networks, Journal of the Mechanics and Physics of Solids 142 (2020) 103984 (2020). doi:https://doi.org/10.1016/j.jmps.2020.103984.

[33] L. Wu, L. Adam, L. Noels, Micro-mechanics and data-driven based reduced order models for multi-scale analyses of woven composites, Composite Structures (Revised version).

[34] D. Perić, E. A. de Souza Neto, R. A. Feijóo, M. Partovi, A. J. C. Molina, On micro-to-macro transitions for multi-scale analysis of non-linear heterogeneous materials: unified variational basis and finite element implementation, International Journal for Numerical Methods in Engineering 87 (15) (2011) 149–170 (2011). doi:10.1002/nme.3014.

[35] T. Kanit, S. Forest, I. Galliet, V. Mounoury, D. Jeulin, Determination of the size of the representative volume element for random composites: statistical and numerical approach, International Journal of Solids and Structures 40 (13) (2003) 3647–3679 (2003). doi:https://doi.org/10.1016/S0020-7683(03)00143-4.

[36] K. Terada, M. Hori, T. Kyoya, N. Kikuchi, Simulation of the multi-scale convergence in computational homogenization approaches, International Journal of Solids and Structures 37 (16) (2000) 2285–2311 (2000). doi:https://doi.org/10.1016/S0020-7683(98)00341-2.

[37] V.-D. Nguyen, E. Bchet, C. Geuzaine, L. Noels, Imposing periodic boundary condition on arbitrary meshes by polynomial interpolation, Computational Materials Science 55 (2012) 390 – 406 (2012). doi:https://doi.org/10.1016/j.commatsci.2011.10.017.

[38] V.-D. Nguyen, L. Wu, L. Noels, Unified treatment of microscopic boundary conditions and efficient algorithms for estimating tangent operators of the homogenized behavior in the computational homogenization method, Comput. Mech. 59 (3) (2017) 483505 (Mar. 2017). doi:10.1007/s00466-016-1358-z.

[39] S. Gajek, M. Schneider, T. Böhlke, An fe-dmn method for the multiscale analysis of fiber reinforced plastic components, arXiv preprint arXiv:2103.08253 (2021).

[40] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747 (2016).

[41] L. Wu, V.-D. Nguyen, L. Adam, L. Noels, An inverse micro-mechanical analysis toward the stochastic homogenization of nonlinear random composites, Computer Methods in Applied Mechanics and Engineering 348 (2019) 97–138 (2019). doi:https://doi.org/10.1016/j.cma.2019.01.016.

[42] V.-D. Nguyen, L. Noels, Data of Micromechanics-based material networks revisited from the interaction viewpoint; robust and efficient implementation for multi-phase composites (2021). doi:https://doi.org/10.5281/zenodo.4743654.

[43] A. Cuitino, M. Ortiz, A material-independent method for extending stress update algorithms from small-strain plasticity to finite plasticity with multiplicative kinematics, Engineering computations 9 (1992) 437–437 (1992).