



ORIGINAL

Argha Protim Dey · Fabian Welschinger · Matti Schneider  · Sebastian Gajek · Thomas Böhlke

# Training deep material networks to reproduce creep loading of short fiber-reinforced thermoplastics with an inelastically-informed strategy

Received: 18 January 2022 / Accepted: 12 June 2022 / Published online: 21 July 2022  
© The Author(s) 2022

**Abstract** Deep material networks (DMNs) are a recent multiscale technology which enable running concurrent multiscale simulations on industrial scale with the help of powerful surrogate models for the micromechanical problem. Classically, the parameters of the DMNs are identified based on linear elastic precomputations. Once the parameters are identified, DMNs may process inelastic material models and were shown to reproduce micromechanical full-field simulations with the original microstructure to high accuracy. The work at hand was motivated by creep loading of thermoplastic components with fiber reinforcement. In this context, multiple scales appear, both in space (due to the reinforcements) and in time (short- and long-term effects). We demonstrate by computational examples that the classical training strategy based on linear elastic precomputations is not guaranteed to produce DMNs whose long-term creep response accurately matches high-fidelity computations. As a remedy, we propose an inelastically informed early stopping strategy for the offline training of the DMNs. Moreover, we introduce a novel strategy based on a surrogate material model, which shares the principal nonlinear effects with the true model but is significantly less expensive to evaluate. For the problem at hand, this strategy enables saving significant time during the parameter identification process. We demonstrate that the novel strategy provides DMNs which reliably generalize to creep loading.

**Keywords** Multiscale methods · Computational homogenization · Deep material networks · Creep loading · Short fiber-reinforced thermoplastics

## 1 Introduction

### 1.1 State of the art

We are interested in characterizing short fiber-reinforced thermoplastics under creep loading. This class of materials combines attractive features like high stiffness and strength along with the ability to mass produce components via injection molding. Specifically, we aim to characterize the composite polybutylene terephthalate (PBT) reinforced with 30% (by weight) E-glass fibers. The classical way to characterize the material would require the manufacturing of test plates via injection molding, the milling of samples and the physical testing. For the case of long-term creep loading, this would take several months. Instead, we opt for an efficient multiscale virtual testing procedure by decomposing the problem into two scales. Due to the spatially varying microstructure, the microscale problem needs to be solved repeatedly but with changing input

---

A. P. Dey · F. Welschinger  
Corporate Sector Research and Advance Engineering, Robert Bosch GmbH, Renningen, Germany

M. Schneider (✉) · S. Gajek · T. Böhlke  
Institute of Engineering Mechanics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany  
E-mail: matti.schneider@kit.edu

parameters. Traditionally, mean-field approaches or variational estimates [1–4] were used to approximate the effective properties of composites. However, such methods rely upon simplifying assumptions on the considered microstructures and material models, and may come with a significant error. Alternatively, computational homogenization approaches may be used, which compute the effective response of a material by solving the balance of linear momentum on a spatially resolved representation of the material's microstructure. In the last decades, different rather efficient computational strategies were developed for this purpose [5]. Particularly successful are strategies based on the fast Fourier transform (FFT) [6,7], which operate on a regular grid and exploit the quality of modern FFT implementations. We refer to the recent review articles [8–10] for a discussion of the latest developments.

Computational homogenization methods may be used for computing the response of a material under a specific loading quite efficiently. However, when a large number of such analyses is required, for instance when the effective model is used as a material model on component scale [11–13], this approach faces limitations. Based on the realization that, in a multiscale context, the problems to be solved actually form variations of each other, strategies were sought which are able to reuse simulation data to derive a simplified effective model whose evaluation is significantly less costly. The transformation field analysis (TFA) of Dvorak and coworkers [14–16] clusters the internal variables into regions of spatial homogeneity. In this way, mean-field-type models are created which are informed by the actual location of the individual phases of the material. Unfortunately, restricting to piecewise homogeneous internal variables limits the predictive capabilities of such models. As a remedy, Michel and Suquet [17–19] generalized the method to permit superpositions of spatially heterogeneous fields of internal variables. To identify these fields, techniques from model order reduction were used [20–22]. Unfortunately, due to the nonlinear dependence of the stress on the internal variables, the difficulty was shifted to finding expressive closed-form expressions of the effective material [23–25]. The method was extended to reduce mechanics at interfaces [26,27], at finite strains [28,29] and in space-time [30], as well. Further data-driven approaches were exploited, for instance based on clustering [31–33] or proper orthogonal decomposition [34,35].

Alternatively, artificial neural networks (ANNs) may be used for approximating the effective elastic energy of nonlinear elastic media [36–38] or the stress–strain relationship of inelastic materials [39–41]. Moreover, ANNs and reduced-order models may be combined on the fly [42,43]. ANN-based approaches typically suffer when evaluated far away from the training set. Moreover, as a result of the neural network approximation, the underlying physical principles may be violated.

Liu et al. [44,45] proposed deep material networks (DMN) as surrogate models for micromechanical computations. Instead of trying to learn the mechanical response of a material, DMNs learn the *microstructure* of the material, i.e., they seek a reduced model of the geometrical interactions inherent to the microstructure.

Classically, DMNs are trained on linear elastic data, i.e., on tuples of input stiffness tensors, one for each phase, and corresponding apparent stiffness tensors, obtained from direct numerical simulations. Subsequently, nonlinear and inelastic material models may be used in the DMN framework and were demonstrated to replace full-field computations with high fidelity. It was shown [46,47] that DMNs inherit thermodynamical consistency from their phases, respect the classical micromechanical bounds and satisfy the Hill–Mandel condition.

Deep material networks come in different flavors. For a material with  $K$  phases, Liu et al. [44,45] use a  $K$ -ary tree of laminates with fixed direction of lamination and intermittent rotations. Gajek et al. [46,48] showed that using  $K$ -ary trees of laminates with variable direction of lamination, but no intermittent rotations, led to similar accuracy with fewer parameters to be trained. Recently, Nguyen and Noels [47,49] have studied more general building blocks inspired by polyhedral finite element methods.

DMNs were shown to work for modeling interface damage [50], strain localization [51], thermomechanically coupled materials [52] and porous materials [49]. Moreover, fully coupled FE-DMN methods were realized [47,48,52,53].

## 1.2 Contributions

Classically, deep material networks are trained on linear elastic data alone. However, when considering long-term creep loading of SFRT composites on an industrial scale, the framework needs to be re-evaluated with care. Indeed, for such a scenario, multiple scales are involved both in space (due to the reinforcements) and in time (due to the long-term loading). We demonstrate by computational experiments that the identified DMNs may or may not lead to good creep predictions. We show that this is caused by overfitting the linear elastic data relative to the creep response. In particular, an early stopping strategy, a classic in machine learning [54, § 7.8], may be used to resolve this issue.

Section 2 outlines the basic construction principles of DMNs, together with the classical elasticity-based training and the novel inelastically informed training strategy. Section 3 reports on the considered setup (E-glass fiber-reinforced PBT) and examines the various training procedures and their inelastic generalizations with care. Particularly powerful turns out to be a hybrid strategy, where the early stopping is itself based on an inelastic surrogate material model which shares the basic phenomena with the “real” material model but may be evaluated with less expense.

## 2 Deep material networks

### 2.1 Basic concepts

A deep material network (DMN) [44,46,47] with  $N$  phases in three spatial dimensions consists of the following data.

1. A set  $\mathcal{N}$  of nodes with indices  $1, \dots, n$ .
2. A set  $\mathcal{G}$  of (formal) integration points with indices  $1, \dots, m$ .
3. A partition of the set of integration points  $\mathcal{G}$  into  $N$  subsets  $\mathcal{G}_i$  ( $i = 1, \dots, N$ ) which are disjoint and cover the set  $\mathcal{G}$ . The set  $\mathcal{G}_i$  contains all integration points which are occupied by the  $i$ th material.
4. A set of positive weights  $w_1, \dots, w_m$ , which sum to unity.
5. A symmetrized gradient operator  $\mathbf{D} \in \mathbb{R}^{6m \times 3n}$ .

The weights may be collected in the diagonal matrix  $\mathbf{W} \in \mathbb{R}^{6m \times 6m}$  with

$$\mathbf{W}_{ii} = w_{\lfloor i/6 \rfloor} \quad (\text{floor division}). \tag{2.1}$$

To express the compatibility conditions, we introduce the averaging matrix  $\mathbf{A} \in \mathbb{R}^{6 \times 6m}$ , an  $m$ -fold copy of the  $6 \times 6$ -identity matrix. Then, the symmetrized gradient operator  $\mathbf{D}$  and the weights  $w_i$  should satisfy the condition

$$\mathbf{A}\mathbf{W}\mathbf{D} = 0, \tag{2.2}$$

which states that the weighted average of the compatible strains vanishes.

Suppose a nonlinear stress–strain relationship  $\mathbf{f}_i : \mathbb{R}^6 \rightarrow \mathbb{R}^6$  at small strains is given for each material  $i = 1, \dots, N$ . Then, for prescribed strain  $\bar{\boldsymbol{\varepsilon}}$ , which we consider as an element of  $\mathbb{R}^6$ , we seek a displacement field  $\bar{\mathbf{u}} \in \mathbb{R}^{3n}$ , s.t. the balance of linear momentum

$$\mathbf{D}^T \mathbf{W} \mathbf{f}(\mathbf{A}^T \bar{\boldsymbol{\varepsilon}} + \mathbf{D} \bar{\mathbf{u}}) = 0 \tag{2.3}$$

holds, where the function  $\mathbf{f}$  applies the nonlinearity  $\mathbf{f}_i$  at the  $i$ th integration point and the vector  $\mathbf{A}^T \bar{\boldsymbol{\varepsilon}}$  contains  $m$  identical copies of the average strain  $\bar{\boldsymbol{\varepsilon}}$ . The effective stress is subsequently computed via

$$\bar{\boldsymbol{\sigma}} = \mathbf{A}\mathbf{W}\mathbf{f}(\mathbf{A}^T \bar{\boldsymbol{\varepsilon}} + \mathbf{D} \bar{\mathbf{u}}). \tag{2.4}$$

Deep material networks serve as a high-level abstraction for a finite element discretization of an  $N$ -phase microstructure. Indeed, for such a microstructure  $Y$ , decomposed into  $N$  subdomains  $Y_i$ , any finite element discretization gives rise to a node set  $\mathcal{N}$ , a set of integration points  $\mathcal{G}$  with associated quadrature weights  $w_i$ . The gradient operator  $\mathbf{D}$  arises by evaluating the FE  $B$ -matrix at the integration points, and a nodal decomposition  $\mathcal{G}_i$  emerges by associating any integration point  $y_j \in Y$  to the material domain which it lies in,  $y_j \in \mathcal{G}_i \iff y_j \in Y_i$ . Then, the classical weak form of the finite element problem is equivalent to the equilibrium equation (2.3) for any type of nonlinearity  $\mathbf{f}_i$ .

Deep material networks give rise to an effective material behavior which automatically satisfies the Hill–Mandel condition, inherits thermodynamical consistency from its phases, preserves elementary micromechanical bounds and gives rise to a uniquely solvable nonlinear system of equations (2.3) (provided the kernel of  $\mathbf{D}$  is factored out and the nonlinearities  $\mathbf{f}_i$  are strictly monotone) [46,47].

Deep material networks serve as an abstraction of finite element discretizations of micromechanical problems. They inherit the positive characteristics, but dispense with the constraints of a physical realization of the finite element discretization. In particular, the node sets  $\mathcal{N}$  and  $\mathcal{G}$  have no intrinsic physical meaning. Still,

DMNs may be regarded as statistically similar representative volume elements [55,56], which reflect—on an abstract level—the topology of the microstructure which they should serve as a surrogate model for.

To start with, DMNs typically fix the node set  $\mathcal{N}$  and the integration points  $\mathcal{G}$  (with the decomposition  $\mathcal{G}_i$ ) a priori. Then, a model for the gradient and the weights

$$\mathbf{D} \equiv \mathbf{D}(\vec{\mathbf{p}}), \quad \mathbf{W} \equiv \mathbf{W}(\vec{\mathbf{p}}) \quad (2.5)$$

in terms of a suitable (hyper)parameter vector  $\vec{\mathbf{p}}$  is postulated. In the literature, different possibilities to choose the parametrized matrices  $\mathbf{D}(\vec{\mathbf{p}})$  and  $\mathbf{W}(\vec{\mathbf{p}})$  were proposed. Liu and coworkers [44,45,50,51] considered a hierarchy of rotated laminates. Gajek et al. [46] introduced a hierarchical construction with laminates of variable direction of lamination, see Sect. 2.2. This construction has fewer parameters than rotated laminates, but was shown to provide similar accuracy. Nguyen and Noels [49] introduced polytopal DMNs which permit more general rank-one jumps than pure laminates to deal with porous microstructures. All these approaches have in common that the matrix  $\mathbf{D}$  is sparse, reflecting a further characteristic of finite element discretizations.

In any case, once the construction plan (2.5) is fixed, DMNs are operated in two stages. In the first stage, the parameters  $\vec{\mathbf{p}}$  are fitted to a suitable objective

$$J(\vec{\mathbf{p}}) \equiv \sum_{s=1}^{n_{\text{obs}}} J_s(\vec{\mathbf{p}}) + \psi(\vec{\mathbf{p}}) \longrightarrow \min_{\vec{\mathbf{p}}} \quad (2.6)$$

with contributions  $J_s$  that measure data fidelity to  $n_{\text{obs}}$  observations and, possibly, a regularization term  $\psi$ . This so-called offline training is typically based on linear elastic precomputations obtained from full-field simulations on a high-fidelity representation of the microstructure [44,45], giving rise to the contributions  $J_s$ . Once the parameters  $\vec{\mathbf{p}}$  are identified, the deep material network may be used as a high-fidelity full-field surrogate model for inverse parameter identification or for concurrent multiscale simulations [48,52,53]. For this purpose, equation (2.3) is solved for nonlinearities that arise from a time discretization of inelastic constitutive laws.

As an alternative to a parameter identification based on linear elastic data alone, the nonlinear effective behavior may be taken into consideration [49]. However, such a procedure may be delicate, as modern gradient-based learning techniques are based on automatic differentiation, which may prove rather computationally expensive for sophisticated material models. The work at hand proposed an alternative strategy based on early stopping [54, § 7.8], see Sect. 2.3.

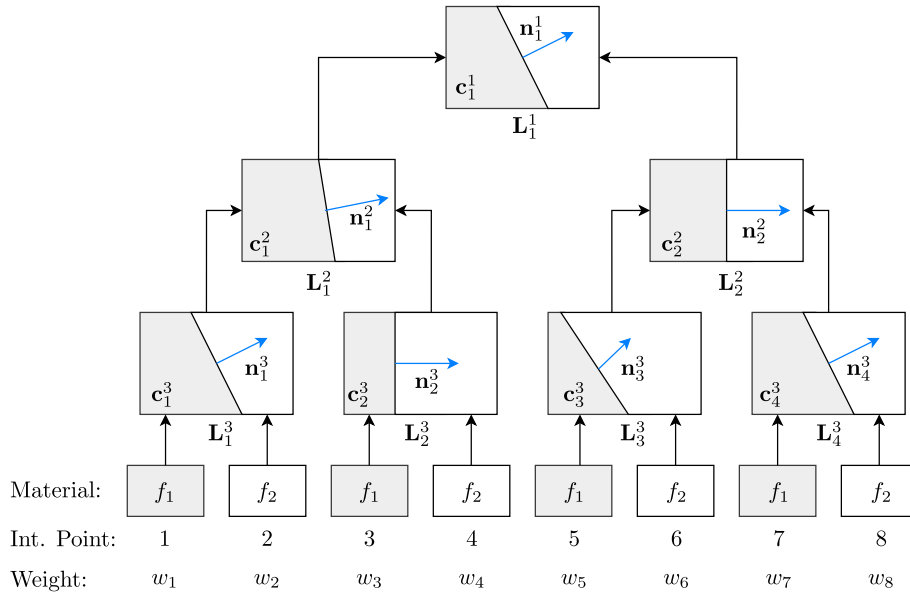
## 2.2 Direct deep material networks

As a construction strategy for the symmetrized gradient operator  $\mathbf{D}(\vec{\mathbf{p}})$  and the weight matrix  $\mathbf{W}(\vec{\mathbf{p}})$ , we use a hierarchy of  $N$ -phase laminates where a scale separation is assumed at each level of the hierarchy. For  $K - 1$  such scale separations, we obtain a perfect, ordered  $N$ -ary tree of depth  $K$  comprising

$$1 + N + N^2 + \dots + N^{K-1} = \frac{N^K - 1}{N - 1} \quad (2.7)$$

individual  $N$ -phase laminates. We call such an  $N$ -ary tree a *direct DMN* [46,48,52]. Figure 1 shows an example of a direct DMN with  $K = 3$  layers and  $N = 2$  phases.

To build the DMN, we first consider a single  $N$ -phase laminate with lamination direction  $\mathbf{n} \in \mathbb{R}^3$  and a vector  $\mathbf{c} \in \mathbb{R}^N$  of volume fractions, which are positive and sum to one. The elastic behavior of such a laminate may be expressed in terms of  $N - 1$  displacement jump vectors  $\vec{\mathbf{a}} = [\mathbf{a}_1, \dots, \mathbf{a}_{N-1}]^T \in \mathbb{R}^{3(N-1)}$ , see Ospald et al. [57]. Classically, a  $N$ -phase laminate has  $N$  displacement jump vectors whose weighted average vanishes. In our representation, this constraint has been explicitly resolved by expressing the last displacement jump vector in terms of the previous  $N - 1$  vectors.



**Fig. 1** Example of a two-phase direct DMN with depth  $K = 3$

For fixed kinematics, the  $N$  strain fluctuations, one for each phase, emerge by applying the symmetrized gradient operator  $\mathbf{L}(\mathbf{c}, \mathbf{n}) \in \mathbb{R}^{6N \times 3(N-1)}$ , given by the formula

$$\mathbf{L}(\mathbf{c}, \mathbf{n}) = \begin{bmatrix} \left( \sum_{i=1}^1 c_i - 1 \right) \mathbf{B}(\mathbf{n}) & \left( \sum_{i=1}^2 c_i - 1 \right) \mathbf{B}(\mathbf{n}) & \dots & \left( \sum_{i=1}^{N-1} c_i - 1 \right) \mathbf{B}(\mathbf{n}) \\ \sum_{i=1}^1 c_i \mathbf{B}(\mathbf{n}) & \left( \sum_{i=1}^2 c_i - 1 \right) \mathbf{B}(\mathbf{n}) & \dots & \left( \sum_{i=1}^{N-1} c_i - 1 \right) \mathbf{B}(\mathbf{n}) \\ \sum_{i=1}^1 c_i \mathbf{B}(\mathbf{n}) & \sum_{i=1}^2 c_i \mathbf{B}(\mathbf{n}) & \dots & \left( \sum_{i=1}^{N-1} c_i - 1 \right) \mathbf{B}(\mathbf{n}) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^1 c_i \mathbf{B}(\mathbf{n}) & \sum_{i=1}^2 c_i \mathbf{B}(\mathbf{n}) & \dots & \left( \sum_{i=1}^{N-1} c_i - 1 \right) \mathbf{B}(\mathbf{n}) \\ \sum_{i=1}^1 c_i \mathbf{B}(\mathbf{n}) & \sum_{i=1}^2 c_i \mathbf{B}(\mathbf{n}) & \dots & \sum_{i=1}^{N-1} c_i \mathbf{B}(\mathbf{n}) \end{bmatrix}, \quad (2.8)$$

to the vector of displacement jump vectors. Here, for any vector  $\mathbf{a}$ , the symmetrized tensor product  $\mathbf{n} \otimes_s \mathbf{a}$  may be expressed by the matrix–vector product  $\mathbf{B}(\mathbf{n})\mathbf{a}$  in Mandel’s notation. Explicitly, the strain–displacement–jump matrix  $\mathbf{B} : \mathbb{R}^3 \rightarrow \mathbb{R}^{6 \times 3}$  reads

$$\mathbf{B}(\mathbf{n}) = \begin{bmatrix} n_1 & 0 & 0 \\ 0 & n_2 & 0 \\ 0 & 0 & n_3 \\ \frac{1}{\sqrt{2}}n_2 & \frac{1}{\sqrt{2}}n_1 & 0 \\ \frac{1}{\sqrt{2}}n_3 & 0 & \frac{1}{\sqrt{2}}n_1 \\ 0 & \frac{1}{\sqrt{2}}n_3 & \frac{1}{\sqrt{2}}n_2 \end{bmatrix}. \quad (2.9)$$

With these expressions for a single laminate at hand, let us consider an  $N$ -ary tree of such  $N$ -phase laminates. To each laminate, a unit direction of lamination  $\mathbf{n}^k$  is associated. We collect these normals in a large vector

$$\vec{\mathbf{n}} = \left[ \mathbf{n}_1^1, \mathbf{n}_1^2, \dots, \mathbf{n}_N^2, \dots, \mathbf{n}_1^K, \dots, \mathbf{n}_{N^{K-1}}^K \right]^T \in \mathbb{R}^{3 \frac{N^K - 1}{N - 1}} \quad (2.10)$$

with an ordering which traverses the tree from the root to the leafs through each level from left to right (like a breadth-first search of the corresponding tree). Moreover, to each laminate,  $N$  volume fractions are associated. We parametrize those in terms of a collection  $\vec{\mathbf{w}} = [w_1, \dots, w_{N^K}] \in \mathbb{R}^{N^K}$  of positive weights (formally) residing on level  $K + 1$ . The volume fractions are then computed by a weighted average [46, eq. (3.8)–(3.9)].

The set of  $m = N^K$  integration points  $\mathcal{G}$  corresponds to the individual phases of the laminates on level  $K$ , and is partitioned into  $N$  subsets  $\mathcal{G}_i$  ( $i = 1, \dots, N$ ) in an alternating manner

$$\mathcal{G}_i = \left\{ m \in \mathcal{G} \mid m = i + lN, \quad l = 0, 1, \dots, N^{K-1} \right\}, \tag{2.11}$$

see Fig. 1 for an illustration for the special case of  $N = 2$ . Thus, each laminate on the lowest level receives one of the  $N$  phases as input.

The kinematics of each of the laminates is governed by  $N1$  (local) displacement jumps

$$\vec{a}_j^k = [\mathbf{a}_{1+(j-1)(N-1)}^k, \dots, \mathbf{a}_{j(N-1)}^k]^T \in \mathbb{R}^{3(N-1)}. \tag{2.12}$$

Here, the upper index  $k = 1, \dots, K$  labels the depth and the lower index  $j = 1, \dots, N^{k-1}$  corresponds the horizontal position in the  $N$ -ary tree. We collect these individual displacement jumps in a (global) displacement vector

$$\vec{u} = \left[ \mathbf{a}_1^1, \dots, \mathbf{a}_{N-1}^1, \mathbf{a}_1^2, \dots, \mathbf{a}_{N(N-1)}^2, \dots, \mathbf{a}_1^K, \dots, \mathbf{a}_{N^{K-1}(N-1)}^K \right]^T \in \mathbb{R}^{3(N^K-1)} \tag{2.13}$$

with the same ordering as for the lamination directions. These serves as the displacement-type degrees of freedom of the deep material network. Thus, the direct DMN comprises  $n = N^K - 1$  nodes, one for each displacement jump.

The (global) symmetrized gradient operator  $\mathbf{D} \in \mathbb{R}^{6N^K \times 3(N^K-1)}$  may be assembled from the local symmetrized gradient operators (2.8) in a similar fashion as finite elements [58]. More precisely, we obtain a representation

$$\mathbf{D} = \sum_{k=1}^K \sum_{j=1}^{N^{k-1}} \mathbf{J}_j^k \mathbf{L}(\mathbf{c}_j^k, \mathbf{n}_j^k) \mathbf{E}_j^k \tag{2.14}$$

in terms of suitable extraction  $\mathbf{E}_j^k$  and prolongation matrices  $\mathbf{J}_j^k \in \mathbb{R}^{6N^K \times 6N}$ . These matrices contain only zeros and ones and establish the connection between local quantities and global quantities. Moreover, the matrices  $\mathbf{E}_j^k$  and  $\mathbf{J}_j^k$  depend only on the topology of the tree and are independent of the DMN parameters  $\vec{p}$ . The extraction matrix  $\mathbf{E}_j^k \in \mathbb{R}^{3(N-1) \times 3(N^K-1)}$  picks out the (local) displacement jumps  $\vec{a}_j^k$  from the global vector  $\vec{u}$ , i.e.,  $\vec{a}_j^k = \mathbf{E}_j^k \vec{u}$  holds, and may be expressed in terms of the Kronecker product [59]

$$\mathbf{E}_j^k = \mathbf{e}_{j-1+N^{k-1}}^T \otimes \underbrace{\text{diag}(\mathbf{1}, \dots, \mathbf{1})}_{N-1 \text{ times}} \quad \text{with} \quad \mathbf{1} \equiv \text{diag}(1, 1, 1) \in \mathbb{R}^{3 \times 3} \tag{2.15}$$

and the Cartesian basis vector  $\mathbf{e}_{j-1+N^{k-1}} \in \mathbb{R}^{\frac{N^K-1}{N-1}}$ . Similarly, the injection matrix  $\mathbf{J}_j^k$  admits the Kronecker product representation

$$\mathbf{J}_j^k = \mathbf{e}_j^T \otimes \underbrace{\text{diag}(1, \dots, 1)}_{N \text{ times}} \otimes \vec{\mathbf{1}}^k \otimes \mathbf{I} \quad \text{with} \quad \mathbf{I} \equiv \text{diag}(1, 1, 1, 1, 1, 1) \in \mathbb{R}^{6 \times 6}, \tag{2.16}$$

the Cartesian basis vector  $\mathbf{e}_j \in \mathbb{R}^{N^{k-1}}$  and the column vector of ones  $\vec{\mathbf{1}}^k = [1, \dots, 1] \in \mathbb{R}^{N^{K-k}}$ .

To illustrate the procedure, we record these matrices for the two-phase DMN of depth  $K = 3$  shown in Fig. 1. The extraction matrices read

$$\mathbf{E}_1^1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], \mathbf{E}_1^2 = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0], \dots, \mathbf{E}_4^3 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1], \tag{2.17}$$

whereas the prolongation operators take the form

$$\mathbf{J}_1^1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{J}_1^2 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{J}_2^2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{J}_1^3 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \dots \quad \mathbf{J}_4^3 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{2.18}$$



We obtain the symmetrized gradient operator

$$\mathbf{D} = \begin{bmatrix} -c_{1,2}^1 \mathbf{B}(\mathbf{n}_1^1) & -c_{1,2}^2 \mathbf{B}(\mathbf{n}_1^2) & 0 & -c_{1,2}^3 \mathbf{B}(\mathbf{n}_1^3) & 0 & 0 & 0 \\ -c_{1,2}^1 \mathbf{B}(\mathbf{n}_1^1) & -c_{1,2}^2 \mathbf{B}(\mathbf{n}_1^2) & 0 & +c_{1,1}^3 \mathbf{B}(\mathbf{n}_1^3) & 0 & 0 & 0 \\ -c_{1,2}^1 \mathbf{B}(\mathbf{n}_1^1) & +c_{1,1}^2 \mathbf{B}(\mathbf{n}_1^2) & 0 & 0 & -c_{2,2}^3 \mathbf{B}(\mathbf{n}_2^3) & 0 & 0 \\ -c_{1,2}^1 \mathbf{B}(\mathbf{n}_1^1) & +c_{1,1}^2 \mathbf{B}(\mathbf{n}_1^2) & 0 & 0 & +c_{2,1}^3 \mathbf{B}(\mathbf{n}_2^3) & 0 & 0 \\ +c_{1,1}^1 \mathbf{B}(\mathbf{n}_1^1) & 0 & -c_{2,2}^2 \mathbf{B}(\mathbf{n}_2^2) & 0 & 0 & -c_{3,2}^3 \mathbf{B}(\mathbf{n}_3^3) & 0 \\ +c_{1,1}^1 \mathbf{B}(\mathbf{n}_1^1) & 0 & -c_{2,2}^2 \mathbf{B}(\mathbf{n}_2^2) & 0 & 0 & +c_{3,1}^3 \mathbf{B}(\mathbf{n}_3^3) & 0 \\ +c_{1,1}^1 \mathbf{B}(\mathbf{n}_1^1) & 0 & +c_{2,1}^2 \mathbf{B}(\mathbf{n}_2^2) & 0 & 0 & 0 & -c_{4,2}^3 \mathbf{B}(\mathbf{n}_4^3) \\ +c_{1,1}^1 \mathbf{B}(\mathbf{n}_1^1) & 0 & +c_{2,1}^2 \mathbf{B}(\mathbf{n}_2^2) & 0 & 0 & 0 & +c_{4,1}^3 \mathbf{B}(\mathbf{n}_4^3) \end{bmatrix} \in \mathbb{R}^{48 \times 21}. \quad (2.19)$$

Returning to the general case, we observe that the symmetrized gradient operator  $\mathbf{D}$  is uniquely parameterized by the individual volume fractions  $c_j^k$  as well as the directions of lamination  $\mathbf{n}_j^k$  of the collection of laminates. Thus, the vectors  $\vec{\mathbf{n}}$  and  $\vec{\mathbf{w}}$  of lamination directions and weights could serve to parametrize such a direct DMN.

In the implementation [44,46], it is convenient to ensure non-negativity of the weights  $\vec{\mathbf{w}} \in \mathbb{R}^{N^K}$  by expressing them in terms of unconstrained weights  $\vec{\mathbf{v}} \in \mathbb{R}^{N^K}$  via applying the Macauley bracket (or ReLU activation function in machine learning)

$$\langle \cdot \rangle_+ : \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}, \quad x \mapsto \max(0, x), \quad (2.20)$$

in a component-wise manner

$$\vec{\mathbf{w}} = \langle \vec{\mathbf{v}} \rangle_+. \quad (2.21)$$

Thus, the parameter vector which uniquely defines both the gradient operator  $\mathbf{D}(\vec{\mathbf{p}})$  and the weight matrix  $\mathbf{W}(\vec{\mathbf{p}})$  of a direct DMN is given by  $\vec{\mathbf{p}} = (\vec{\mathbf{n}}, \vec{\mathbf{v}})$ .

### 2.3 An inelastically informed training strategy

Classically, DMNs are trained on objective functions of the form (2.6)

$$J(\vec{\mathbf{p}}) \equiv \sum_{s=1}^{n_{\text{obs}}} J_s(\vec{\mathbf{p}}) + \psi(\vec{\mathbf{p}}) \longrightarrow \min_{\vec{\mathbf{p}}}, \quad (2.22)$$

where the functions  $J_s(\vec{\mathbf{p}})$  measure the proximity of the current DMNs predictions to precomputed effective elastic properties. The term  $\psi(\vec{\mathbf{p}})$  serves as a regularizing term and is explicitly defined in Sect. 3.2.

Although DMNs, which are identified based on elastic data, show accurate predictions for large classes on nonlinear and inelastic constitutive laws for the microstructural phases, the developed theory [46] hints that such a close agreement may be lost if the constitutive laws show a high degree of nonlinearity. Therefore, it appears a good idea to add nonlinear constitutive laws to the training, encoded by an additional term  $H(\vec{\mathbf{p}})$ . Then, instead of the original problem (2.22), an augmented problem

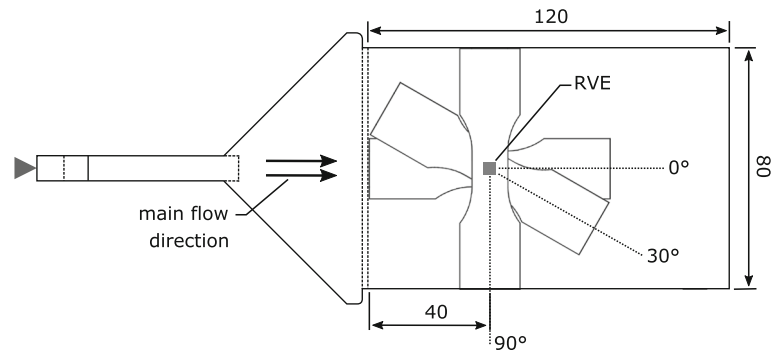
$$J(\vec{\mathbf{p}}) + H(\vec{\mathbf{p}}) \longrightarrow \min_{\vec{\mathbf{p}}} \quad (2.23)$$

may be used to identify the parameters. Unfortunately, such an approach is restricted to rather simple inelastic material models [49], as the computational effort in evaluating complex constitutive laws, including computing the derivatives necessary for the gradient descent for a multitude of different loading conditions and time steps, quickly becomes prohibitive.

For this reason, we propose to use an alternative strategy which is similar in spirit to the commonly used *early stopping* [54, § 7.8] technique in machine learning. More precisely, in our strategy, the iterates  $\vec{\mathbf{p}}_k$  of a conventional solver for the original problem (2.22) are stored in a set  $\mathcal{P}$ . Then, the additional term  $H(\vec{\mathbf{p}})$ , which encodes the performance of our model on the nonlinear inelastic constitutive laws is evaluated using the stored iterates  $\vec{\mathbf{p}}_k$  during training. Finally, the best parameter vector

$$\vec{\mathbf{p}}_{\text{best}} = \operatorname{argmin}_{\vec{\mathbf{p}} \in \mathcal{P}} H(\vec{\mathbf{p}}) \quad (2.24)$$

is selected. It is common to store only each 50th or 100th iterate  $\vec{\mathbf{p}}_k$  of the learning method, so that the runtime of such an early stopping strategy will be reduced by one or two orders of magnitude compared to a coupled strategy (2.23) if the effort of evaluating the inelasticity-aware objective  $H$  is significantly larger than for the original objective (2.22).



**Fig. 2** Prepared Becker [60] samples from composite plates with different directions

**Table 1** Identified material parameters for PBT-GF30

Matrix	Elastic	$E = 2399.3 \text{ MPa}$	$\nu = 0.4$	$y_0 = 20.9 \text{ MPa}$	$y_\infty = 51.9 \text{ MPa}$
	Plastic	$h = 346.9 \text{ MPa}$	$\omega = 384.9$		
	Creep	$A_1 = 0.014 \text{ MPa}^{-1}$	$n = 32.4$		
		$k = 0.016$	$\dot{\epsilon}_0 = 1.0 \text{ s}^{-1}$		
Fibers	Elastic	$E = 72,000 \text{ MPa}$	$\nu = 0.22$		

### 3 Computational investigations

#### 3.1 Setup

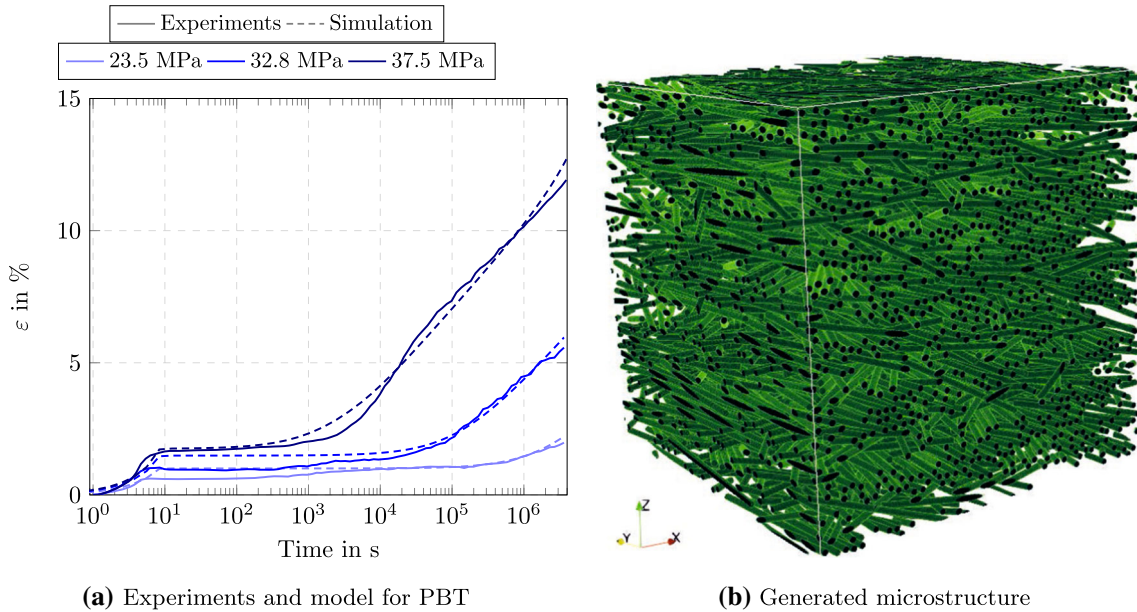
We consider a polybutylene terephthalate (PBT), reinforced by 30% E-glass fibers (PBT-GF30). To characterize the matrix material, experiments were performed on a dog-bone shaped Becker sample [60], shown in Fig. 2, at the three load levels 23.5 MPa, 32.8 MPa and 37.5 MPa. The load was ramped up to the respective stress levels in 8.5 s and held constant for different time intervals depending on the load level. The results are shown in Fig. 3a. For all three load levels, we observe the typical three different creep phases [61]: primary, secondary and tertiary creep. The creep–strain rate starts with a high value at the beginning of the primary creep stage and reaches an approximately constant value during the secondary creep stage. The tertiary state is characterized by an increase in the creep strain rate, leading to fracture of the specimen, eventually [62]. The first phase ends at roughly the same time for all three considered load levels. The onset of the third phase and the inclination of the strain at this onset, however, differ significantly for the three load levels.

For the matrix, we use a coupled plasticity–creep model, where the elastoplastic part accounts for short-term effects, whereas a viscoplastic augmentation accounts for long-term creep effects. We refer to Appendix A for details. To identify the free parameters, we used a two-step procedure. In a first step, the evolution of the creep strain was deactivated, and the elastic constants as well as the plasticity parameters were determined from uniaxial, monotone, tensile experiments up to 5% strain, performed for four different samples. For the parameter identification using OptiSlang [63], we set up a single-element unit–hexahedron model in Abaqus [64], and fixed the Poisson’s ratio to  $\nu = 0.4$  and the viscosity to  $\eta = 0.001 \text{ MPa s}$ , emulating strain rate independence.

Once the elasticity and plasticity coefficients were determined, the remaining creep parameters were identified from long-term creep tests carried out at different load levels, see Fig. 3a, based on OptiSlang [63], wrapping an Abaqus [64] simulation. The model predictions with the identified parameters are compared to the experimental results in Fig. 3a. The model captures the three creep phases rather well and represents the three experimental results to a sufficient accuracy. The final list of identified parameters is given in Table 1. For the E-glass fibers, we use standard parameters [65].

It is well known that the mechanical properties of fiber-reinforced composites strongly depend on the microstructure characteristics like fiber orientation and fiber length distribution [66,67]. To quantify the microstructure characteristics, we rely upon microcomputed tomography [68–70].





**Fig. 3** A comparison of experimental results and model predictions for the PBT matrix and generated periodic fiber-filled microstructure

More precisely, we determined the length-weighted average fiber length of  $269 \mu\text{m}$ , a fiber diameter of  $10 \mu\text{m}$  and the second-order fiber orientation tensor

$$\mathbf{A} = \begin{bmatrix} 0.213 & 0.006 & 0.001 \\ & 0.770 & -0.004 \\ \text{sym} & & 0.017 \end{bmatrix} \quad (3.1)$$

by the method discussed in Hessman et al. [71] for a specimen milled out from a  $120 \text{ mm} \times 80 \text{ mm} \times 2 \text{ mm}$  injection-molded plate, see Fig. 2. We used these data to reconstruct a digital microstructure with the SAM algorithm [72], see Fig. 3b. We generated a cubic unit cell with an edge length of  $625 \mu\text{m}$  containing 2056 cylindrical fibers with a length of  $269 \mu\text{m}$  and a diameter of  $10 \mu\text{m}$  with a minimum distance of  $3 \mu\text{m}$  between the fibers. The second-order fiber orientation tensor (3.1) was prescribed, combined with the exact closure approximation [73,74].

The microstructure was discretized by  $512^3$  voxels. For the subsequent FFT computations, we used the composite-voxel technique [75–77] with a resampling factor of two. Thus, after resampling, the unit cell consists of  $256^3$  voxels. The computation time when performing a linear elastic FFT simulation using a phase contrast of 10,000 and a unit cell discretized by  $512^3$  voxels was 19.9 h. In comparison, the resampled microstructure with  $256^3$  voxels and composite voxels required just 2.38 h. The resampling error between the stiffness tensors is just 1.12% in the Frobenius norm in Mandel notation. As, in the case of a lower phase contrast, an even smaller error is expected, the composite-voxel method serves as a reasonable way to speed up the computations.

Last but not least, let us discuss the used software and hardware. The micromechanical computations were performed with the software FeelMath [78] on a HPC cluster with nodes that have 16 CPUs each and 4 GB of memory per CPU. FeelMath [78] implements FFT-based computational homogenization methods. More precisely, we used the staggered grid discretization [79,80] and linear as well as nonlinear conjugate gradient methods [81–83] for linear and nonlinear constitutive behavior, respectively.

The DMN was trained on a single CPU with 60GB of memory. The DMN online phase code was run on a single computing node.

Both the material model and the DMN were implemented as user-defined material subroutines (UMATs) in Abaqus [64], which may also be used in FeelMath [78]. We use the LAPACK [84] libraries for the linear algebra operations.

### 3.2 Performance of classical sampling

We consider a microstructure with  $N = 2$  phases, i.e., the PBT matrix and the E-glass fibers. We use the direct DMN described in Sect. 2.2, and initialize the parameters  $(\vec{n}, \vec{v})$  as proposed in Gajek et al. [46].

For the training, we generated training and validation data based on full-field FFT-based computational homogenization. These data consist of triples  $(\mathbb{C}_1^s, \mathbb{C}_2^s, \bar{\mathbb{C}}_{\text{FFT}}^s)$ , where  $\mathbb{C}_1^s$  and  $\mathbb{C}_2^s$  serve as the linear elasticity tensors of the two materials, and  $\bar{\mathbb{C}}_{\text{FFT}}^s$  refers to the effective elasticity tensor. The tuples  $(\mathbb{C}_1^s, \mathbb{C}_2^s)$  were sampled by the method proposed by Liu and Wu [45] utilizing orthotropic elasticity tensors. The effective properties  $\bar{\mathbb{C}}_{\text{FFT}}^s$  were computed with FeelMath [78]. We generated 1 000 samples as mentioned, split into 800 training and 200 validation points.

For fixed DMN parameters  $(\vec{n}, \vec{v})$ , we denote by  $\bar{\mathbb{C}}_{\text{DMN}}^s(\vec{n}, \vec{v})$  the effective stiffness computed by the DMN with phase stiffnesses  $\mathbb{C}_1^s$  and  $\mathbb{C}_2^s$ . We refer to Gajek et al. [46] for details how to evaluate the effective stiffness  $\bar{\mathbb{C}}_{\text{DMN}}^s(\vec{n}, \vec{v})$  efficiently.

For the offline training with linear elastic data, we minimize the objective function (2.6), i.e.,

$$J(\vec{n}, \vec{v}) \longrightarrow \min_{\vec{n}, \vec{v}} \quad \text{with} \quad J(\vec{n}, \vec{v}) = \sum_{s=1}^{n_b} J_s(\vec{n}, \vec{v}) + \psi(\vec{n}, \vec{v}), \quad (3.2)$$

with the batch size  $n_b = 40$ , the contributions

$$J_s(\vec{n}, \vec{v}) = \frac{1}{n_b} \frac{\|\bar{\mathbb{C}}_{\text{FFT}}^s - \bar{\mathbb{C}}_{\text{DMN}}^s(\vec{n}, \vec{v})\|_1}{\|\bar{\mathbb{C}}_{\text{FFT}}^s\|_1} \quad (3.3)$$

and

$$\psi(\vec{n}, \vec{v}) = \lambda \left( \mathbf{a}_1^T \langle \vec{v} \rangle_+ - c_1 \right) + \lambda \left( \mathbf{a}_2^T \langle \vec{v} \rangle_+ - c_2 \right), \quad (3.4)$$

where  $c_1 = 0.822$  and  $c_2 = 0.178$  denote the volume fractions of the respective phases, and  $\mathbf{a}_1$  is a vector that has ones at all odd indices and is zero; otherwise,  $\mathbf{a}_2$  is a vector that has ones at all even indices and is zero otherwise, and  $\lambda$  refers to a penalty factor which we set to 100. The term  $\psi$  enforces the respective volume fractions of both phases [85]. This approach differs from previous works [44–46], where the total volume fraction of the DMN is enforced to unity. Instead, we enforce the respective volume fractions of each of the phases. We found empirically that such a strategy increases the reliability of the offline training.

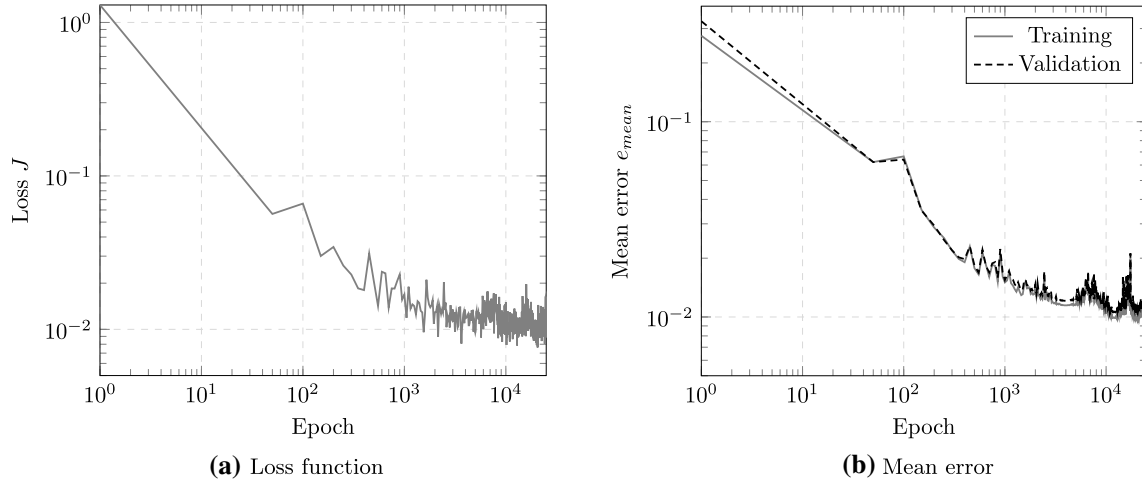
We use the  $\ell^1$ -norm in Eq. (3.3) since an independent study with  $\ell^p$ -norms and variable exponents  $p$  revealed that the subsequently trained DMNs capture the inelastic creep response most closely for  $p = 1$ . For reasons of conciseness, we chose not to report this study here.

The training, i.e., the minimization of the objective function (3.2), proceeds as for (more general) neural networks, i.e., via a stochastic batch-gradient-descent-type algorithm based on automatic differentiation and a grouping into batches. (We use batches of size 40.) We implemented the procedure in PyTorch [86] and used the AMSGrad method [87] for training the network along with a learning rate modulation using cosine annealing [88].

$$\beta(m) = \beta_{\min} + \frac{1}{2}(\beta_{\max} - \beta_{\min}) \left( 1 + \cos \left( \pi \frac{m}{M} \right) \right), \quad (3.5)$$

where  $\beta$  and  $m$  refer to the learning rate and epoch, respectively. We use a maximum learning rate  $\beta_{\max} = 0.0007$  and a minimum learning rate  $\beta_{\min} = 0$ . The parameter  $M$  is set to 4000. The training of the DMN was performed up to 25,000 epochs. The typical progress during training is shown in Fig. 4a. For the first 100 epochs, the objective function decreases monotonically. Thereafter, the loss function shows some fluctuations. These are triggered by the learning rate modulation which enables escaping local minima of the objective function. Indeed, on a large scale, a further decrease of the objective function up to around epoch 20,000 is observed. For the considered example, the smallest loss occurred at epoch 19,962. The training ran for 18.8 h. After training, we used the DMN with the smallest *realized* loss for further use.

We investigated the effect of depth on the online evaluation of creep and found that the performance of the DMN in the online phase increases up to a depth of seven. For higher depth, no further improvement in accuracy for the evaluation of creep in the online phase was observed. We trained our seven-layer DMN on



**Fig. 4** Offline training results and mean training and validation errors of DMN #1 from Table 2

**Table 2** Elastic training results of seven-layer DMNs

DMN #	Smallest loss	$e_{\text{mean}}^{\text{tr}}$ (%)	$e_{\text{max}}^{\text{tr}}$ (%)	$e_{\text{mean}}^{\text{val}}$ (%)	$e_{\text{max}}^{\text{val}}$ (%)
1	0.00591	0.97	12.78	1.08	5.10
2	0.00567	0.96	12.90	1.07	6.36
3	0.00612	1.01	13.03	1.11	6.17
4	0.00600	1.02	13.41	1.15	5.90
5	0.00592	0.98	13.66	1.07	4.19
6	0.00502	0.96	13.09	1.04	4.76

800 training samples. To assess the reproducibility of the results, we trained multiple seven-layer DMNs with the same hyperparameters but with random starting values of the parameters ( $\vec{n}$ ,  $\vec{v}$ ).

To assess the generalization capabilities of the DMN, we introduce the sample-wise error

$$e_s = \frac{\|\bar{\mathbb{C}}_{\text{FFT}}^s - \bar{\mathbb{C}}_{\text{DMN}}^s\|_1}{\|\bar{\mathbb{C}}_{\text{FFT}}^s\|_1}, \quad (3.6)$$

involving the relative  $\ell^1$ -difference of the actual stiffness tensor  $\bar{\mathbb{C}}_{\text{FFT}}^s$  and the predicted stiffness tensor  $\bar{\mathbb{C}}_{\text{DMN}}^s$  in Mandel's notation. Moreover, we define the maximum and the mean errors over all the samples

$$e_{\text{max}} = \max_s e_s \quad \text{and} \quad e_{\text{mean}} = \frac{1}{N_s} \sum_{s=1}^{N_s} e_s, \quad (3.7)$$

where  $N_s$  is the number of samples in the training or the validation set, respectively.

After the training, we studied the generalization capabilities of the model with the help of linear elastic validation data comprising 200 data points. The results of the training and the validation errors are detailed in Table 2.

We observe that the reached loss value is comparable for all six considered DMNs. Also, the mean training and validation error is rather close for all considered networks. We find that the mean training error is slightly smaller than the mean validation error for all the networks. The maximum training error for all the networks is around 13%. The maximum validation error ranges from slightly above 4% to almost 7%, i.e., we find a larger variation (roughly by a factor of two). Still, the elastic training and validation errors are on a reasonable level, in particular in view of the mean errors.

The mean training and validation errors for DMN #1 over the course of epochs is shown in Fig. 4b. We observe that the mean training and validation errors follow a similar trend as the loss function. Moreover, the training and validation error decrease simultaneously during training indicating no overfitting occurs in the offline training phase with respect to the linear elastic training data.

**Table 3** Maximum relative errors for elastoplastic (3.9) and creep loading (3.11)

DMN #	$\max_{ij} e_{ij}^p$ (%)	$e_{0^\circ}^c$ (%)	$e_{30^\circ}^c$ (%)	$e_{90^\circ}^c$ (%)	$e^c$ (%)
1	2.97	5.57	5.67	4.36	5.67
2	9.87	17.40	3.06	3.62	17.40
3	8.41	13.51	23.03	6.02	23.03
4	4.41	1.52	3.04	1.91	3.04
5	5.55	5.77	4.80	5.68	5.77
6	5.21	5.35	9.33	3.56	9.33

We use our six trained DMNs to evaluate the online phase. We briefly comment on the elastoplastic response of the DMNs. The online phase of the DMN, implemented as an UMAT, is evaluated on a single four-node unit tetrahedron element in Abaqus [64]. We use the load increment control of Abaqus [64] for applying the load, and compare the output to full-field simulations obtained by an FFT-based solver [83]. We applied hysteretic, uniaxial strain loadings  $\bar{\varepsilon}_{ij}$  for a period of 4 s with a strain amplitude of 2.5% for 40 load steps in all six loading directions. For each instance of time  $t$ , the relative error in each stress component

$$e_{ij,t}^p = \frac{|\bar{\sigma}_{ij,t}^{\text{DMN}} - \bar{\sigma}_{ij,t}^{\text{FFT}}|}{\max_{\tau} |\bar{\sigma}_{ij,\tau}^{\text{FFT}}|} \quad (3.8)$$

is considered, together with the maximum relative error

$$e_{ij}^p = \max_{\tau} e_{ij,\tau}^p \quad (3.9)$$

over the entire simulation window.

In addition to the elastoplastic response, we also investigate the creep behavior. To account for the anisotropy of the microstructure, we investigate the performance of the DMN in three loading directions with angles  $0^\circ$ ,  $30^\circ$ , and  $90^\circ$ . Here, the  $0^\circ$  direction indicates the flow direction and the  $90^\circ$  direction is perpendicular to the flow direction during the manufacturing of the test plate. The experimental analysis of the creep loading was performed using the samples cut out from the injection-molded plate in these specific angles as shown in Fig. 2. The creep response of the DMN was evaluated on a single voxel microstructure using an FFT-based solver [83]. The implemented DMN online phase UMAT can be flexibly integrated into the FFT-based solver FeelMath [78]. We consider 32 time steps, equally spaced in the logarithmic timescale. The DMN response was compared with the effective strain computed by full-field simulations performed with the same FFT-based solver [83].

In the  $0^\circ$  direction, we applied a uniaxial tensile stress  $\bar{\sigma}_{11}$  in the flow direction which is ramped up to 65.4 MPa in 8.5 s and subsequently held constant for  $3.78 \times 10^6$  s. We evaluated the creep strain component  $\bar{\varepsilon}_{11}$  in the flow direction over the entire simulation window. In the  $0^\circ$  direction, we refer to the evaluated strain by the name  $\bar{\varepsilon}_{0^\circ}$ . We apply a lower uniaxial stress  $\bar{\sigma}_{22}$  of 35.7 MPa until  $3.78 \times 10^6$  s for the creep response in the  $90^\circ$  direction. We evaluate the strain component  $\bar{\varepsilon}_{22}$  during loading and assign the results to  $\bar{\varepsilon}_{90^\circ}$ .

For the  $30^\circ$  direction, we follow a similar strategy using mixed boundary conditions [89] for a uniaxial tensile stress of 69 MPa, held constant up to  $4.45 \times 10^5$  s. During the entire simulation time, we evaluated the creep strain component  $\bar{\varepsilon}_{11}$ , and refer to it as  $\bar{\varepsilon}_{30^\circ}$ .

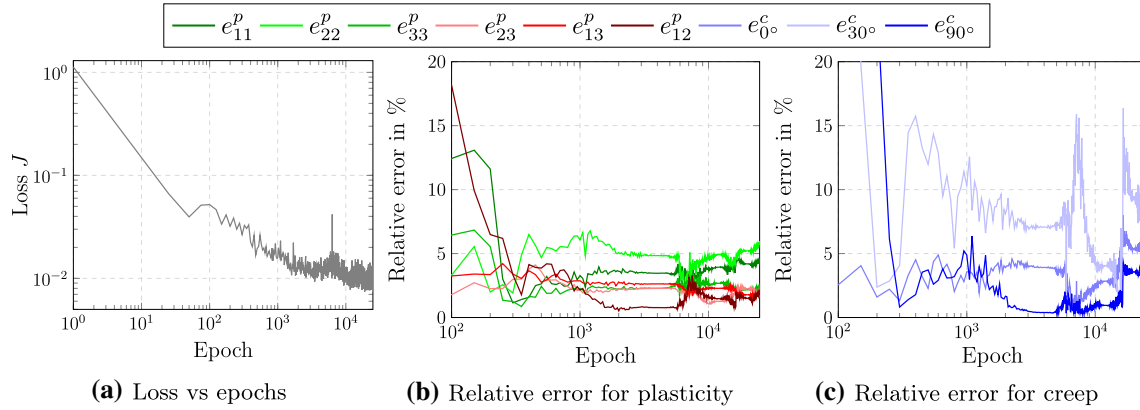
For the creep loading, we consider the relative strain errors

$$e_{\theta,t}^c = \frac{|\bar{\varepsilon}_{\theta,t}^{\text{DMN}} - \bar{\varepsilon}_{\theta,t}^{\text{FFT}}|}{\max_{\tau} |\bar{\varepsilon}_{\theta,\tau}^{\text{FFT}}|} \quad \text{for } \theta = 0^\circ, 30^\circ, 90^\circ \quad (3.10)$$

and the maximum errors

$$e_{\theta}^c = \max_{\tau} e_{\theta,\tau}^c \quad \text{as well as} \quad e^c = \max_{\theta} e_{\theta}^c. \quad (3.11)$$

The trained DMNs whose elastic validation results are shown in Table 2 are used for the online evaluation of plasticity and creep. Table 3 contains the results for the inelastic evaluation.



**Fig. 5** A closer look at DMN #6 from Table 3 in terms of the loss function and the plasticity (3.9) as well as creep errors (3.11) (every 50 epochs), evaluated in post-processing

For the maximum error during the elastoplastic loading, we observe a variation of the results by a factor of three. DMN #1 leads to a low relative error slightly below 3%. Three DMNs give rise to about 5% error, and two DMNs are characterized by a relative error slightly below 10%. Taking a glance at the elastic validation errors, see Table 2, we observe that the DMN with the best elastoplastic online error does not correspond to the DMN with the best elastic validation error.

Even more striking are the variations in the creep response. Whereas DMNs #1, #4 and #5 provide a rather small creep error, two DMNs give rise to relative creep errors on the order of 20%, which is certainly not acceptable for engineering accuracy. Moreover, we notice that the maxima are realized in different directions. Indeed, for DMN #2, the  $0^\circ$ -direction is worst, but the  $30^\circ$ -direction is matched rather accurately. In contrast, for DMN #3, both the  $0^\circ$  and the  $30^\circ$ -direction are inaccurate. Thus, there appears to be no systematic cause for this phenomenon, e.g., related to the fiber orientation. We observe that the DMN #1, #4 and #5 can represent both the elastoplastic loading and creep accurately. In contrast, the DMNs #2 and #3 are incapable of representing the creep loading with sufficient fidelity.

To sum up, the results of Table 3 reveal that good results achieved during the elasticity-based training results do not necessarily lead to accurate predictions for the creep response. We will study this phenomenon more thoroughly in the next section.

### 3.3 Accurate creep predictions with early stopping and a surrogate material model

In the previous section, we observed that being a minimizer of the elasticity-based loss function (3.2)–(3.3) is not necessarily related to providing an accurate creep prediction for the DMN under consideration. To get a deeper insight into this phenomenon, we took a closer look at DMN #6 from Table 3. More precisely, we stored the DMN parameters  $(\vec{n}_k, \vec{v}_k)$  every 50 epochs during the training. As a post-processing step, we evaluated the elastoplastic (3.9) and the creep errors (3.11) at these parameter states  $(\vec{n}_k, \vec{v}_k)$ . The results, alongside the loss function, are shown in Fig. 5. In Fig. 5a, we observe that the loss function decreases in a similar way as for DMN #1, see Fig. 4a. The decrease is monotonic on large epoch scales, but shows variations up to half an order of magnitude during training as a result of the learning rate modulation. Taking a look at the plasticity errors (3.9) for the six considered loading directions, see Fig. 5b, we observe that the errors decrease consistently up to epoch 300. Then, only the error of the 12-component decreases further, whereas the other five errors more or less remain on the same level. At epoch 6000, the 12-error starts to increase significantly, whereas the 22-error—the largest up to this epoch—starts to decrease. For epochs exceeding 10,000, the maximum error  $\max_{i,j} e_{ij}^p$  starts to increase again.

Similar observations may be made for the creep error (3.11), shown in Fig. 5c. For the first 300 epochs, the error decreases. For higher epochs, the errors in the considered directions show a non-monotonic behavior. However, the creep error is much larger than the plasticity error, in a range up to 15%. Interestingly, there is a range of epochs, approximately at 10,000, where the creep error is low, below 5%. This smallness is neither reflected in the loss function nor the plasticity error.



**Table 4** Identified material parameters for the Norton-type model (3.13)

Matrix	Elastic	$E = 2399.3 \text{ MPa}$	$\nu = 0.4$	
	Creep	$A = 0.014 \text{ MPa}^{-1}$	$n = 26.96$	$\dot{\varepsilon}_0 = 1.0 \text{ s}^{-1}$

In the terminology of deep learning, the observed phenomenon may be interpreted as “overfitting,” i.e., producing parameters that correspond too closely to a specific set of data, but fail to represent additional data or to predict future observations in a reliable way.

Classically, there are (at least) two ways to rectify this shortcoming. The first strategy consists of adding further data points to the considered set. In our case, we could augment the classical elasticity-based offline training by more cleverly chosen stiffness triples  $(\mathbb{C}_1^s, \mathbb{C}_2^s, \mathbb{C}_{\text{FFT}}^s)$  or by including inelastic data into the loss function [49]. Unfortunately, such a strategy does not protect us from overfitting relative to *other cases* which were not considered before. In the case of inelastic training for reproducing creep loading, training on inelastic data for the  $0^\circ$  direction does not guarantee us a good fit in the  $30^\circ$  and  $90^\circ$  directions. Therefore, an alternative way to circumvent overfitting is to stop early [54, § 7.8], i.e., to stop at a stage where the fitting quality of the considered data in the linear elastic regime (2.6) and an additional independent nonlinear validation data set are both good. In the former case, the training and validation error can be monitored so that there is no overfitting during the offline elastic training. In the latter case, the performance of the DMN in the inelastic regime should be within engineering accuracy when compared with inelastic full-field simulations. Indeed, by monitoring the additional data, we may detect when overfitting happens in the inelastic regime.

The analysis at the beginning of this section, in particular given in Fig. 5c, shows that early stopping based on the additional term (3.11)

$$H(\vec{n}, \vec{v}) = e^c(\vec{n}, \vec{v}) \equiv \max_{\theta} e_{\theta}^c(\vec{n}, \vec{v}) \quad (3.12)$$

does indeed work for DMN #6. The DMN #6 trained on linear elastic data can accurately predict the online inelastic response of highly nonlinear material models, providing a maximum relative creep error below 5%.

Thus, the strategy proposed in Sect. 2.3 works and, in theory, we can perform the evaluation of the creep model during training for identification of the best parameter vector.

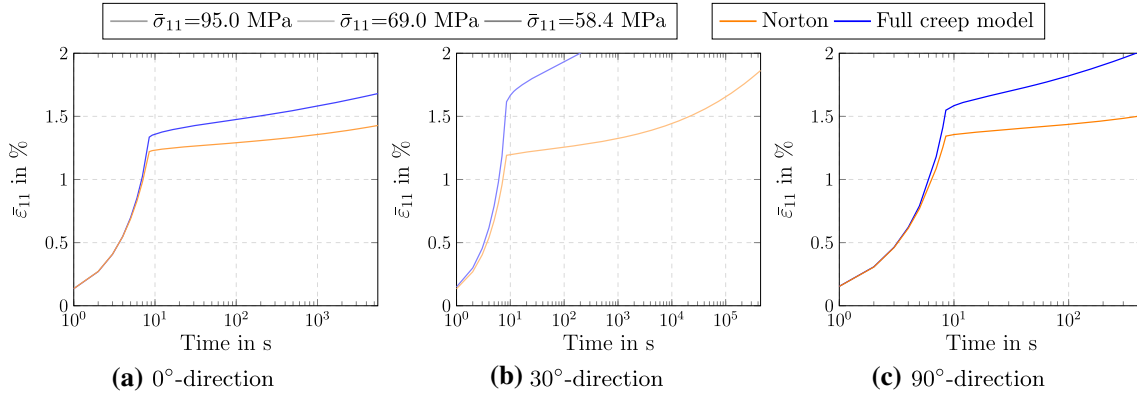
However, we are interested in a further improvement of the strategy. Due to the complexity of the considered creep model, the post-processing for evaluating the creep error  $e^c(\vec{n}, \vec{v})$  ran for 24.4 h. In particular, the post-processing took 30% longer than the previous elasticity-based training! Hence, performing the evaluation of the creep model during the elasticity-based training will lead to a cumbersome and time-intensive training procedure. Clearly, the long-term creep behavior demands certain characteristics of the DMNs to be accurate, and isolating the proper elastic training data appears difficult. To reduce the computational effort of evaluating the full creep model, we came up with the following idea. Instead of the full creep model, it might be sufficient to consider another creep model, which is, on the one hand, less expensive to evaluate, but, on the other hand, identifies the same weaknesses in generalization to long-term loading as the full-field model. For this purpose, we consider Hooke’s law in combination with a Norton-type creep law [62]

$$\begin{aligned} \boldsymbol{\sigma} &= \kappa \text{tr}[\boldsymbol{\varepsilon}] \mathbf{I} + 2\mu \text{dev}[\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^c] \\ \dot{\varepsilon}^c &= \dot{\varepsilon}_0 (A \sqrt{3/2} \|\text{dev} \boldsymbol{\sigma}\|)^n \frac{\text{dev}[\boldsymbol{\sigma}]}{\|\text{dev}[\boldsymbol{\sigma}]\|} \end{aligned} \quad (3.13)$$

with creep parameters  $A$  and  $n$ .

We chose the parameters as detailed in Table 4. The elastic parameters coincide with the full creep model, as does the parameter  $A \equiv A_1$ , see Table 1. The creep exponent  $n$  and the parameter  $A$  were identified with the help of the long-term creep experiments carried out at different load levels and an OptiSlang [63] procedure based on an Abaqus [64] simulation. We conducted full-field simulations on the considered microstructure, see Fig. 3b, with the Norton model (3.13) for the matrix material. For the three considered directions, see Fig. 6, we observe that both the elastic response and the first creep phase are captured rather accurately by the simplified model. Moreover, the onset of the secondary creep phase is also represented quite well. However, the effective strains are underestimated by the Norton-type model for the creep phase two and beyond.





**Fig. 6** Full-field results for the creep strain over time, comparing the Norton-type model (3.13) for the matrix and the full creep model, see “Appendix A”

In analogy to the true creep error (3.11), we define an error measure for the creep response of the Norton model

$$e_{\theta,t}^n = \frac{\left| \bar{\varepsilon}_{\theta,t}^{\text{DMN, Norton}} - \bar{\varepsilon}_{\theta,t}^{\text{FFT, Norton}} \right|}{\max_{\tau} \left| \bar{\varepsilon}_{\theta,\tau}^{\text{FFT, Norton}} \right|} \quad \text{for } \theta = 0^\circ, 30^\circ, 90^\circ \quad (3.14)$$

and the maximum errors

$$e_{\theta}^n = \max_{\tau} e_{\theta,\tau}^n \quad \text{as well as} \quad e^n = \max_{\theta} e_{\theta}^n. \quad (3.15)$$

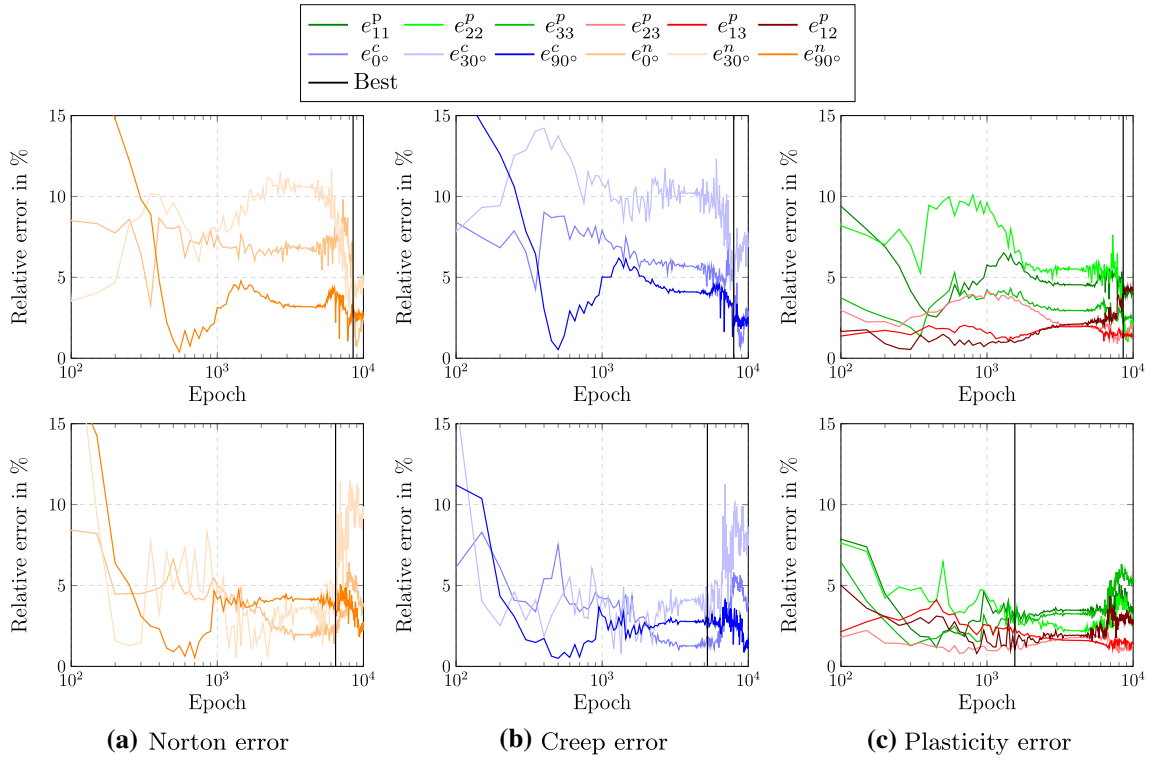
To assess the predictive quality of the Norton error  $e^n$ , we trained two additional DMNs with a maximum learning rate  $\beta_{\max} = 0.0005$ . During the offline training of the DMN based on the linear elastic data, we introduce an *inelastic validation data set* based on the Norton errors for  $0^\circ$ ,  $30^\circ$  and  $90^\circ$  directions. We consider this validation data set *in addition* to the linear elastic validation data set which is monitored during the training to prevent the overfitting in the elastic regime, see Fig. 4b.

The linear elasticity-based training proceeds as in Sect. 3.2, via a stochastic batch-gradient-descent-type algorithm. We evaluate the loss function  $J(\vec{p})$  3.2 involving the parameters  $\vec{p} = (\vec{n}, \vec{v})$ , determine the gradients  $\partial J / \partial \vec{n}$  and  $\partial J / \partial \vec{v}$  via automatic differentiation and finally update the fitting parameters. The parameter vector  $\vec{p}$  is stored every 50 epochs to evaluate the *inelastic validation data set*, wherein the Norton error  $e^n(\vec{p})$  (3.15) is calculated. This serves as the basis for identifying the best parameter vector  $\vec{p}_{\text{best}}$ .

In a classical early stopping approach, the validation data set error is monitored and the training is stopped when the error does not improve for a predefined number of states. A copy of the model parameters are stored every time the error on the validation set improves. The parameters with the best validation error rather than the most recent parameters are returned once the training algorithm is terminated [54, § 7.8].

In our case, the *inelastic validation error* does not show a monotonic decrease and has persistent fluctuations, see Fig. 5b, c, which is a consequence of the learning rate modulation (3.5) used during training. Thus, terminating the training once the *inelastic validation error* has not improved for a predefined number of epochs (which results in an additional hyperparameter introduced into the procedure) is not appropriate. As an alternative, we monitor the Norton error  $e^n(\vec{p})$  on the fly every 50 epochs and store the parameters each time the Norton error improves, for the complete training. More precisely, after the end of 10,000 epochs, we return the parameter vector  $\vec{p}_{\text{best}}$  from the state (evaluated every 50 epochs) having the best *inelastic validation error*, i.e., the least Norton error. This *quasi-early stopping* approach for the inelastically informed training methodology is outlined in Algorithm 1.

The results of the evaluation of the *inelastic validation data set* over the course of training is shown in Fig. 7a. We also performed the creep and plasticity evaluation using the fully coupled plasticity–creep law as a post-processing step, see Fig. 7b, c respectively. The creep and plasticity errors are not used for identifying the best DMN and the plots are presented in Fig. 7 to highlight the correlation between evolution of the Norton error during training with the post-processed results.



**Fig. 7** Comparison of the three inelastic error measures over epochs for DMN #1 (top) and DMN #2 (bottom) from Table 5

**Table 5** Results of online evaluation of DMN with the inelastically informed offline training method—the training time includes evaluating the Norton model

DMN #	Training time (h)	$e^p$ (%)	$e^c$ (%)	$e^n$ (%)	Stored at epoch	Evaluation $e^c$ time (h)
1	12.07	4.36	5.32	3.57	8500	8.7
2	12.37	3.19	3.43	3.33	6450	8.1

### Algorithm 1 Quasi-early stopping training approach

```

 $\vec{p} \leftarrow (\vec{n}, \vec{v})$  ▷ Initialize parameter vector randomly
 $e_{\text{best}}^n \leftarrow 1000000$  ▷ Inelastic validation error initialized with a high value before optimization start
for  $epoch \leftarrow 1, 10000$  do
  Calculate loss function
   $\mathbb{C}_{\text{DMN}}^s(\vec{p}) \leftarrow \mathbb{C}_{\text{DMN}}^s(\vec{n}, \vec{v})$ 
   $J(\vec{p}) \equiv \sum_{s=1}^{n_b} J_s(\vec{p}) + \psi(\vec{p})$  ▷ Calculated for every batch  $n_b$  (3.3)
  Update parameter vector
   $\vec{n} \leftarrow \vec{n} - \beta \partial J / \partial \vec{n}(\vec{p})$  ▷  $\beta$  is the learning rate
   $\vec{v} \leftarrow \vec{v} - \beta \partial J / \partial \vec{v}(\vec{p})$ 
   $\vec{p} \leftarrow (\vec{n}, \vec{v})$ 
  if  $epoch \bmod 50 = 0$  then
    Calculate error measure for Norton model (3.15) for on-the-fly inelastic validation
     $e^n(\vec{p}) \leftarrow \max_{\theta} e_{\theta}^n(\vec{p})$ 
    if  $e^n(\vec{p}) < e_{\text{best}}^n$  then
       $e_{\text{best}}^n \leftarrow e^n(\vec{p})$ 
      Update best parameter vector to be returned at end of training with the current parameters
       $\vec{p}_{\text{best}} \leftarrow (\vec{n}, \vec{v})$ 
    end if
  end if
end for

```

**Table 6** Maximum relative errors for creep loading using *test* data with the DMNs identified using the inelastically informed offline training method

DMN #	$e_{15^\circ}^c$ (%)	$e_{45^\circ}^c$ (%)	$e_{60^\circ}^c$ (%)
1	2.53	3.74	3.79
2	3.84	7.61	3.08

Again we observe differences in the error measures between the considered DMNs. Indeed, taking a look at the plasticity error shown in Fig. 7c, we observe  $e_{22}^p$  is largest for DMN #1 and  $e_{11}^p$  is largest for DMN #2 at the end of the training. We also observe differences between the plasticity error, see Fig. 7c, and the creep error, see Fig. 7b, for both DMNs. For instance, the large dip of  $e_{90^\circ}^c$  for DMN #1 is not reflected by particularly low values of the plasticity error in any component. Comparing the Norton and the creep error, see Fig. 7a, b, respectively, leads to a better resemblance. Indeed, after epoch 1000 there is a good agreement between the trends of the Norton errors and the creep errors for the individual loading directions. This correspondence is also underlined by the identified epochs where the respective error is lowest during training. For DMN #1, the best epoch is very similar for all three considered error measures. For DMN #2, the best plasticity error is reached a few thousand epochs before the best creep-error epoch, which is, in turn, closely matched by the best Norton error epoch. For both DMNs, the errors are reported in Table 5. We observe that the DMNs identified by early stopping based on the Norton error  $e^n$  provide both a good creep and plasticity error, slightly above 3%. Actually, the minimum reached Norton error exceeds both other considered error measures.

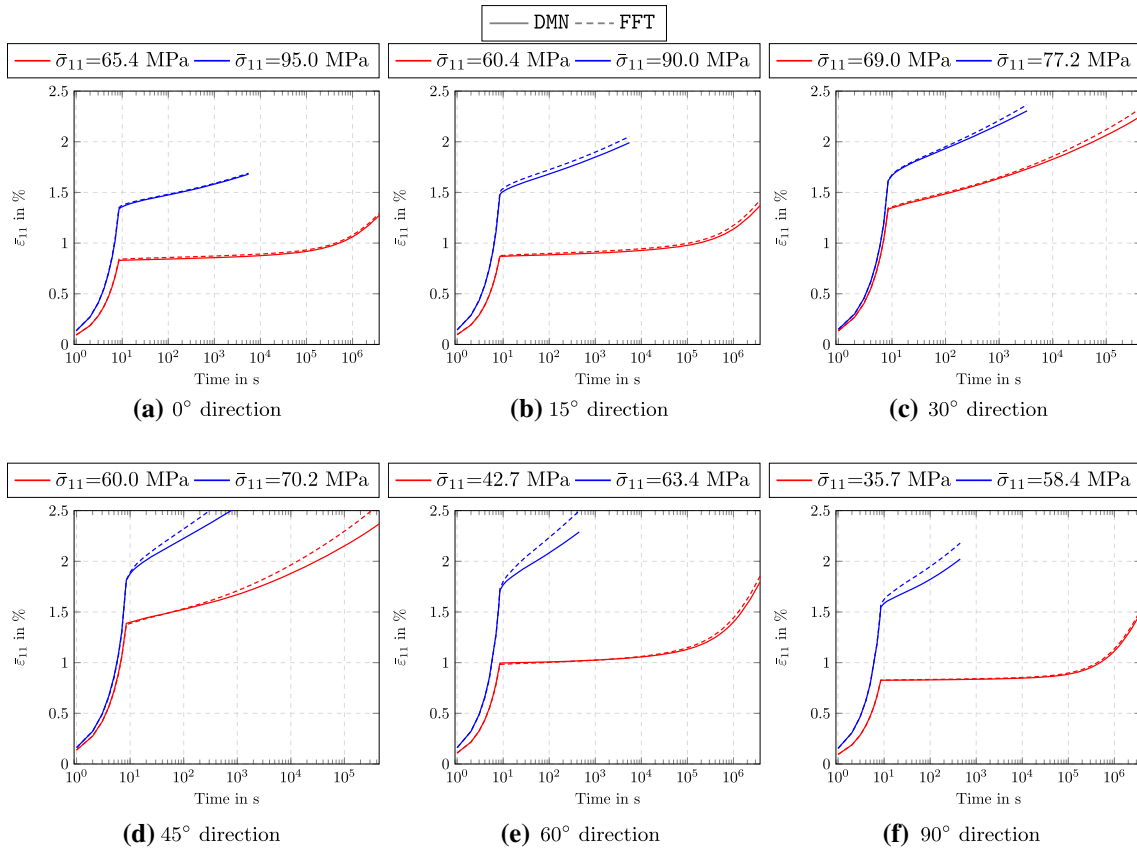
The minimum creep error  $e^c$  for DMN #1 over 10,000 epochs is 4.49% captured at epoch 7950. This is rather close to the creep error  $e^c$  of 5.32% captured using the Norton-type model as shown in Table 5. Similarly, the minimum creep error  $e^c$  for DMN #2 using the plasticity-creep coupled model is 2.33%. Moreover, just the evaluation of the coupled plasticity-creep law lasts for more than 8 h as described in Table 5, whereas the training time along with the evaluation of the Norton model requires only slightly more than 12 h. Thus, we save considerable computation time with only a slight loss of performance when using the Norton model.

To assess the generalization capabilities of the DMNs identified using the Norton error, we a posteriori evaluate their performance on an inelastic *test* data set. The *test* data sets consist of the creep response evaluated using the fully coupled plasticity-creep law at three different directions of 15°, 45° and 60° with respect to the flow direction. Uniaxial stresses of 60.4 MPa, 60.0 MPa and 42.7 MPa were used for the 15°, 45° and 60° directions, respectively, for obtaining the creep strain response. The loading is performed in a similar manner as for the 30° direction outlined in Sect. 3.2. We extend the true creep error evaluated for 0°, 30° and 90° to these directions and the maximum relative errors for both DMNs are recorded in Table 6. The relative errors for all the test directions for DMN #1 are rather close with a maximum error of 3.79% in the 60° direction. The relative errors for DMN#2 vary by a factor of roughly two with a maximum of 7.61% in the 45° direction. It is interesting to note that a smaller inelastic Norton validation error does not necessarily lead to a smaller test creep error, as evident from Tables 5 and 6. Although the errors in the test directions are not monitored during the inelastic validation, both DMNs are able to represent the a posteriori creep loading in the test data with sufficient fidelity.

Thus, we conclude that the novel early stopping-based training strategy serves as a low-cost methodology for reliably identifying the parameters of deep material networks suitable for creep loading.

Last but not least, we report on the performance of the DMNs, identified via the novel method, for predicting the creep response compared to high-fidelity simulations. For this purpose, we worked with DMN #2 from Table 5, and considered two different stress levels for each of the three directions from the *inelastic validation data set* and from the *inelastic test data set*. The results, shown in Fig. 8, reveal a close agreement between the DMN predictions and the full-field computations for the stress levels (shown in red) in the directions that the network was trained on. This comes as no surprise. By the way, it becomes apparent that for more than  $10^4$  s, the 30° direction leads to inaccurate predictions. For the additional stress loading and the directions, which had not been monitored, the DMN closely matches the full-field predictions for strain levels below 1.5%. Actually, the agreement is quite remarkable. For higher effective strains, the agreement between DMN predictions and full-field simulation is worse. It remains to be studied whether this is caused by a lack of suitable training or results from insufficient material modeling. Indeed, for effective strains exceeding 1.5%, the local strains in the matrix exceed 5%. In particular, a material model at small strains appears questionable.

The runtimes of DMN and full-field simulations are compared in Table 7. The full-field simulations operate on a microstructure with  $256^3$  voxels. Moreover, the composite-voxel technique [76] is used. In contrast, the



**Fig. 8** Comparison of FFT simulation results with DMN #2

**Table 7** Runtime comparison between full-field results and deep material networks

Direction	Stress level (MPa)	FFT runtime (min)	DMN runtime (min)	Speedup factor
0°	65.4	295	0.5	590
	95.0	348	0.5	696
15°	60.4	255	0.5	510
	90.0	425	0.5	850
30°	69.0	331	0.5	662
	77.0	310	0.5	620
45°	60.0	352	0.5	704
	70.2	343	0.5	686
60°	42.7	237	0.5	474
	63.4	290	0.5	580
90°	35.7	192	0.5	384
	58.4	236	0.5	472

DMN is integrated on a single voxel. The full-field simulations take between 3.2 and 7.0 h, whereas the DMN finishes after half a minute. Thus, we observe speedup factors between slightly less than 400 up to almost 850. Thus, we confirm the startling speedup factors reported by other authors [45,48,49].

## 4 Conclusion

Scientific progress consists of several phases. Initially, new territory gets explored and claimed. More often than not, these new islands of knowledge are established independently, lacking a proper connection to more established grounds. Moreover, the new territory needs to be charted and guarded. We consider the introduction of deep material networks by Liu and coworkers [44,45] as such a groundbreaking moment, which opened the eyes of the micromechanical community to the possibilities of the deep learning technology. Indeed, finding accurate yet computationally tractable surrogate models of microstructures materials on component scale for wide classes of industrially relevant materials has been an objective of micromechanics for a long time, and different strategies emerged, as discussed in Introduction.

Several years have gone by, and the basic inner workings of DMNs were uncovered, and new variants of DMNs were proposed. The purpose of the work at hand was to critically review the concept of elasticity-based training of DMNs when a high degree of nonlinearity and rather larger timescales are involved. The available theory [46] is based on a perturbation argument around linear viscoelastic constitutive laws. In particular, it is conceivable that training based on linear elastic data alone might not be sufficient to prepare the DMN for the harsh conditions of creep loading. In this work, we provided (more or less explicit) examples of DMNs that fail to generalize from linear elasticity to creep. Due to a lack of a mathematical convergence theory for DMNs, it is not clear whether this is a defect of the linear elasticity-based training *per se* or a clever choice of sampled stiffnesses or appropriately designed loss functions may resolve this issue. Notwithstanding this lack of understanding from our side, we worked out a remedy for this shortcoming based on the early stopping strategy which is classical in deep learning. We demonstrated that such an approach leads to reproducible and reliable training results for DMNs that may be used with confidence. The introduced technology serves as a low-cost alternative to inelastic training [49] and is expected to enter the standard toolbox of the deep material networker.

As already indicated, we consider the state of affairs surrounding the DMNs in the second, contemplating, phase following the initial exploration phase. The underpinnings of DMNs need to be understood more thoroughly, and guidelines for its use, including preferred tools, need to be identified.

**Acknowledgements** MS, SG and TB acknowledge partial support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 255730231. We thank the anonymous reviewers for their helpful comments and their interest in our work.

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

**Data availability** The data that support the findings of this study are available from the corresponding author upon reasonable request.

### Declarations

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

## A Details on the plasticity–creep model

We consider a material model at small strains with an additive decomposition of the strain tensor  $\varepsilon = \nabla_s \mathbf{u}$

$$\varepsilon = \varepsilon^e + \varepsilon^p + \varepsilon^c \quad (\text{A.1})$$

into an elastic, plastic and a creep contribution. We consider the plastic strain and the creep strain as the internal variables, together with an isotropic hardening variable  $\alpha$ . It is assumed that plastic and creep deformations are volume preserving. We consider an isotropic material model with stress tensor

$$\boldsymbol{\sigma} = \kappa \operatorname{tr}[\varepsilon] \mathbf{I} + 2\mu \operatorname{dev}[\varepsilon - \varepsilon^p - \varepsilon^c], \quad (\text{A.2})$$

where  $\kappa$  and  $\mu$  denote the compression and shear modulus, respectively. The evolution of the internal variables is governed by the equations

$$\begin{aligned}\dot{\varepsilon}^p &= \frac{\langle \phi \rangle_+}{\eta} \frac{\text{dev}[\boldsymbol{\sigma}]}{\|\text{dev}[\boldsymbol{\sigma}]\|}, \\ \dot{\alpha} &= \sqrt{2/3}/\eta \langle \phi \rangle_+, \\ \dot{\varepsilon}^c &= \dot{\varepsilon}_0 \left[ 1 + C e^{-\|\varepsilon^c\|/k} \right] \left[ \left( A_1 \sqrt{3/2} \|\text{dev}[\boldsymbol{\sigma}]\| \right)^n + \left( A_2 \sqrt{3/2} \|\text{dev}[\boldsymbol{\sigma}]\| \right) \right] \frac{\text{dev}[\boldsymbol{\sigma}]}{\|\text{dev}[\boldsymbol{\sigma}]\|},\end{aligned}\tag{A.3}$$

in terms of the yield function which combines a linear hardening with a Voce-type hardening,

$$\phi = \|\text{dev}[\boldsymbol{\sigma}]\| - \sqrt{2/3} (y_0 + h \alpha + (y_\infty - y_0)(1 - e^{-\omega \alpha})).\tag{A.4}$$

and the Macauley bracket

$$\langle \phi \rangle_+ = \max(0, \phi).\tag{A.5}$$

The material parameters encompass a yield stress  $y_0$ , a limiting stress  $y_\infty$ , a plastic hardening modulus  $h$ , an exponential hardening factor  $\omega$ , a viscosity  $\eta$ , the creep constants  $C$  and  $k$ , creep prefactors  $A_1$  as well as  $A_2$ , a reference creep rate  $\dot{\varepsilon}_0$ , and the creep exponent  $n$ . Details for the derivation of the material model [90,91], which may be cast in the framework of generalized standard material (GSM) [92], will be discussed elsewhere. The material model is discretized with an implicit Euler method in time. The discretized evolution equations are solved by a classical return mapping algorithm based on a creep–plastic predictor and a plastic corrector step [93].

## References

- Mori, T., Tanaka, K.: Average stress in matrix and average elastic energy of materials with misfitting inclusions. *Acta Metall.* **21**(5), 571–574 (1973)
- Willis, J.R.: Variational and related methods for the overall properties of composites. *Adv. Appl. Mech.* **21**, 1–78 (1981)
- Benveniste, Y.: A new approach to the application of Mori-Tanaka's theory in composite materials. *Mech. Mater.* **6**(2), 147–157 (1987)
- Hill, R.: Elastic properties of reinforced solids: some theoretical principles. *J. Mech. Phys. Solids* **11**(5), 357–372 (1963)
- Matouš, K., Geers, M.G.D., Kouznetsova, V.G., Gillman, A.: A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. *J. Comput. Phys.* **330**, 192–220 (2017)
- Moulinec, H., Suquet, P.: A fast numerical method for computing the linear and nonlinear mechanical properties of composites. *C. R. Acad. Sci. Sér. II* **318**(11), 1417–1423 (1994)
- Moulinec, H., Suquet, P.: A numerical method for computing the overall response of nonlinear composites with complex microstructure. *Comput. Methods Appl. Mech. Eng.* **157**, 69–94 (1998)
- Lebensohn, R.A., Rollett, A.D.: Spectral methods for full-field micromechanical modelling of polycrystalline material. *Comput. Mater. Sci.* **173**, 109336 (2020)
- Schneider, M.: A review of non-linear FFT-based computational homogenization methods. *Acta Mech.* **232**, 2051–2100 (2021)
- Lucarini, S., Upadhyay, M.V., Segurado, J.: FFT based approaches in micromechanics: fundamentals, methods and applications. In: *Modelling and Simulation in Materials Science and Engineering*, vol. online, pp. 1–86 (2021)
- Feyel, F.: Multiscale FE<sup>2</sup> elastoviscoplastic analysis of composite structures. *Comput. Mater. Sci.* **16**(1), 344–354 (1999)
- Feyel, F., Chaboche, J.-L.: FE<sup>2</sup> multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Comput. Methods Appl. Mech. Eng.* **183**(3–4), 309–330 (2000)
- Feyel, F.: A multilevel finite element method (FE<sup>2</sup>) to describe the response of highly non-linear structures using generalized continua. *Comput. Methods Appl. Mech. Eng.* **192**(28–30), 3233–3244 (2003)
- Dvorak, G., Bahei-El-Din, Y., Wafa, A.: Implementation of the transformation field analysis. *Comput. Mech.* **14**(14), 201–228 (1994)
- Dvorak, G., Bahei-El-Din, Y., Wafa, A.: The modeling of inelastic composite materials with the transformation field analysis. *Model. Simul. Mater. Sci. Eng.* **2**(2), 571–586 (1994)
- Chaboche, J.L., Kanouté, P., Roos, A.: On the capabilities of mean-field approaches for the description of plasticity in metal matrix composites. *Int. J. Plast.* **21**, 1409–1434 (2005)
- Michel, J.C., Suquet, P.: Nonuniform transformation field analysis. *Int. J. Solids Struct.* **40**, 6937–6955 (2003)
- Michel, J., Suquet, P.: Computational analysis of nonlinear composite structures using the nonuniform transformation field analysis. *Comput. Methods Appl. Mech. Eng.* **193**, 5477–5502 (2004)
- Michel, J.-C., Suquet, P.: A model-reduction approach in micromechanics of materials preserving the variational structure of constitutive relations. *J. Mech. Phys. Solids* **90**, 254–285 (2016)
- Fritzen, F., Böhlke, T.: Nonuniform transformation field analysis of materials with morphological anisotropy. *Compos. Sci. Technol.* **71**(4), 433–442 (2011)



21. Fritzen, F., Böhlke, T.: Reduced basis homogenization of viscoelastic composites. *Compos. Sci. Technol.* **76**, 84–91 (2013)
22. Largeton, R., Michel, J.-C., Suquet, P.: Extension of the nonuniform transformation field analysis to linear viscoelastic composites in the presence of aging and swelling. *Mech. Mater.* **73**, 76–100 (2014)
23. Fritzen, F., Leuschner, M.: Reduced basis hybrid computational homogenization based on a mixed incremental formulation. *Comput. Methods Appl. Mech. Eng.* **260**, 143–154 (2013)
24. Michel, J.-C., Suquet, P.: A model-reduction approach to the micromechanical analysis of polycrystalline materials. *Comput. Mech.* **57**(3), 483–508 (2016)
25. Michel, J.-C., Suquet, P.: Effective potentials in nonlinear polycrystals and quadrature formulae. *Proc. R. Soc. A* **473**, 20170213 (2017)
26. Leuschner, M., Fritzen, F., van Dommelen, J.A.W., Hoefnagels, J.P.M.: Potential-based constitutive models for cohesive interfaces: theory, implementation and examples. *Compos. B Eng.* **104**, 38–50 (2015)
27. Leuschner, M., Fritzen, F.: Reduced order homogenization for viscoplastic composite materials including dissipative imperfect interfaces. *Mech. Mater.* **104**, 121–138 (2017)
28. Kunc, O., Fritzen, F.: Finite strain homogenization using a reduced basis and efficient sampling. *Math. Comput. Appl.* **24**(2), 56 (2019)
29. Fritzen, F., Kunc, O.: Two-stage data-driven homogenization for nonlinear solids using a reduced order model. *Eur. J. Mech. A. Solids* **69**, 201–220 (2018)
30. Fritzen, F., Hassani, M.: Space-time model order reduction for nonlinear viscoelastic systems subjected to long-term loading. *Meccanica* **53**, 1333–1355 (2018)
31. Liu, Z., Bessa, M.A., Liu, W.K.: Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials. *Comput. Methods Appl. Mech. Eng.* **306**, 319–341 (2016)
32. Wulfinghoff, S., Cavaliere, F., Reese, S.: Model order reduction of nonlinear homogenization problems using a Hashin–Shtrikman type finite element method. *Comput. Methods Appl. Mech. Eng.* **330**, 149–179 (2018)
33. Schneider, M.: On the mathematical foundations of the self-consistent clustering analysis for non-linear materials at small strains. *Comput. Methods Appl. Mech. Eng.* **354**, 783–801 (2019)
34. Oliver, J., Caicedo, M., Huespe, A.E., Hernández, J.A., Roubin, E.: Reduced order modeling strategies for computational multiscale fracture. *Comput. Methods Appl. Mech. Eng.* **313**, 560–595 (2017)
35. Raschi, M., Lloberas-Valls, O., Huespe, A., Oliver, J.: High performance reduction technique for multiscale finite element modeling (HPR-FE2): towards industrial multiscale FE software. *Comput. Methods Appl. Mech. Eng.* **375**, 113580 (2021)
36. Shen, Y., Chandrashekhara, K., Breig, W.F., Oliver, L.R.: Neural network based constitutive model for rubber material. *Rubber Chem. Technol.* **77**(2), 257–277 (2004)
37. Le, B.A., Yvonnet, J., He, Q.-C.: Computational homogenization of nonlinear elastic materials using neural networks. *Int. J. Numer. Methods Eng.* **104**(12), 1061–1084 (2015)
38. Nguyen-Thanh, V.M., Nguyen, L.T.K., Rabczuk, T., Zhuang, X.: A surrogate model for computational homogenization of elastostatics at finite strain using the HDMR-based neural network approximator. *Int. J. Numer. Methods Eng.* **121**(21), 4811–4842 (2020)
39. Jadid, M.N.: Prediction of stress-strain relationships for reinforced concrete sections by implementing neural network techniques. *J. King Saud Univ. Eng. Sci.* **9**(2), 169–188 (1997)
40. Penumadu, D., Zhao, R.: Triaxial compression behavior of sand and gravel using artificial neural networks (ANN). *Comput. Geotech.* **24**(3), 207–230 (1999)
41. Srinivasu, G., Rao, R.N., Nandy, T.K., Bhattacharjee, A.: Artificial neural network approach for prediction of titanium alloy stress-strain curve. *Procedia Eng.* **38**, 3709–3714 (2012)
42. Fritzen, F., Fernández, M., Larsson, F.: On-the-fly adaptivity for nonlinear twoscale simulations using artificial neural networks and reduced order modeling. *Front. Mater.* **6**, 75 (2019)
43. Vijayaraghavan, S., Wu, L., Noels, L., Bordas, S.P.A., Natarajan, S., Beex, L.A.A.: Neural-network acceleration of projection-based model-order-reduction for finite plasticity: application to RVEs, pp. 1–8. [arXiv:2109.07747](https://arxiv.org/abs/2109.07747) (2021)
44. Liu, Z., Wu, C.T., Koishi, M.: A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials. *Comput. Methods Appl. Mech. Eng.* **345**, 1138–1168 (2019)
45. Liu, Z., Wu, C.T.: Exploring the 3D architectures of deep material network in data-driven multiscale mechanics. *J. Mech. Phys. Solids* **127**, 20–46 (2019)
46. Gajek, S., Schneider, M., Böhlke, T.: On the micromechanics of deep material networks. *J. Mech. Phys. Solids* **142**, 103984 (2020)
47. Nguyen, V.D., Noels, L.: Micromechanics-based material networks revisited from the interaction viewpoint; robust and efficient implementation for multi-phase composites. *Eur. J. Mech. A. Solids* **91**, 104384 (2022)
48. Gajek, S., Schneider, M., Böhlke, T.: An FE-DMN method for the multiscale analysis of short fiber reinforced plastic components. *Comput. Methods Appl. Mech. Eng.* **384**, 113952 (2021)
49. Nguyen, V.D., Noels, L.: Interaction-based material network: a general framework for (porous) microstructured materials. *Comput. Methods Appl. Mech. Eng.* **389**, 114300 (2021)
50. Liu, Z.: Deep material network with cohesive layers: multi-stage training and interfacial failure analysis. *Comput. Methods Appl. Mech. Eng.* **363**, 112913 (2020)
51. Liu, Z.: Cell division in deep material networks applied to multiscale strain localization modeling. *Comput. Methods Appl. Mech. Eng.* **384**, 113914 (2021)
52. Gajek, S., Schneider, M., Böhlke, T.: An FE-DMN method for the multiscale analysis of thermomechanical composites. *Comput. Mech.* **69**(5), 1087–1113 (2022)
53. Liu, Z., Wei, H., Huang, T., Wu, C.T.: Intelligent multiscale simulation based on process-guided composite database, pp. 1–15. [arXiv:2003.09491](https://arxiv.org/abs/2003.09491) (2020)
54. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, Cambridge (2016)
55. Balzani, D., Brands, D., Schröder, J.: Construction of Statistically Similar Representative Volume Elements, pp. 355–412. Springer, Vienna (2014)

56. Scheunemann, L., Balzani, D., Brands, D., Schröder, J.: Construction of Statistically Similar RVEs, pp. 219–256. Springer, New York (2015)
57. Ospald, F., Schneider, M., Kabel, M.: A model order reduction method for computational homogenization at finite strains on regular grids using hyperelastic laminates to approximate interfaces. *Comput. Methods Appl. Mech. Eng.* **309**, 476–496 (2016)
58. Fish, J., Belytschko, T.: *A First Course in Finite Elements*. Wiley, Hoboken (2008)
59. Brewer, J.W.: A note on Kronecker matrix products and matrix equation systems. *SIAM J. Appl. Math.* **17**(3), 603–606 (1969)
60. Becker, F.: Entwicklung einer Beschreibungsmethodik für das mechanische Verhalten unverstärkter Thermoplaste bei hohen Deformationsgeschwindigkeiten. PhD thesis, Martin-Luther University Halle-Wittenberg (2009)
61. Andrade, E.N.D.C.: On the viscous flow in metals, and allied phenomena. *Proc. R. Soc. Lond. Ser. A Contain. Pap. Math. Phys. Character* **84**(567), 1–12 (1910)
62. Naumenko, K., Altenbach, H.: *Modeling of Creep for Structural Analysis*. Foundations of Engineering Mechanics. Springer, Berlin (2007)
63. Will, J.: optislang - robust design optimization(rdo) - key technology for resource-efficient product development and performance enhancement. Accessed 2 Nov 2021
64. Simulia: “Abaqus CAE.” Accessed 11 Nov 2021
65. Doghri, I., Brassart, L., Adam, L., Gérard, J.-S.: A second-moment incremental formulation for the mean-field homogenization of elasto-plastic composites. *Int. J. Plast.* **27**(3), 352–371 (2011)
66. Breuer, K., Stommel, M.: RVE modelling of short fiber reinforced thermoplastics with discrete fiber orientation and fiber length distribution. *SN Appl. Sci.* **2**, 91 (2020)
67. Breuer, K., Stommel, M.: Prediction of short fiber composite properties by an artificial neural network trained on an RVE database. *Fibers* **9**(2), 8 (2021)
68. de Paiva, R.F., Bisiaux, M., Lynch, J., Rosenberg, E.: High resolution X-ray tomography in an electron microprobe. *Rev. Sci. Instrum.* **67**(6), 2251–2256 (1996)
69. Shen, H., Nutt, S., Hull, D.: Direct observation and measurement of fiber architecture in short fiber-polymer composite foam through micro-CT imaging. *Compos. Sci. Technol.* **64**(13–14), 2113–2120 (2004)
70. Garcea, S.C., Wang, Y., Withers, P.J.: X-ray computed tomography of polymer composites. *Compos. Sci. Technol.* **156**, 305–319 (2018)
71. Hessman, P.A., Riedel, T., Welschinger, F., Hornberger, K., Böhlke, T.: Microstructural analysis of short glass fiber reinforced thermoplastics based on X-ray micro-computed tomography. *Compos. Sci. Technol.* **183**, 107752 (2019)
72. Schneider, M.: The sequential addition and migration method to generate representative volume elements for the homogenization of short fiber reinforced plastics. *Comput. Mech.* **59**(2), 247–263 (2017)
73. Montgomery-Smith, S., He, W., Jack, D., Smith, D.: Exact tensor closures for the three-dimensional Jeffery’s equation. *J. Fluid Mech.* **680**, 321–335 (2011)
74. Montgomery-Smith, S., Jack, D., Smith, D.E.: The fast exact closure for Jeffery’s equation with diffusion. *J. Nonnewton Fluid Mech.* **166**, 343–353 (2011)
75. Kabel, M., Merkert, D., Schneider, M.: Use of composite voxels in FFT-based homogenization. *Comput. Methods Appl. Mech. Eng.* **294**, 168–188 (2015)
76. Kabel, M., Fink, A., Schneider, M.: The composite voxel technique for inelastic problems. *Comput. Methods Appl. Mech. Eng.* **322**, 396–418 (2017)
77. Charière, R., Marano, A., Gélébart, L.: Use of composite voxels in FFT based elastic simulations of hollow glass microspheres/polypropylene composites. *Int. J. Solids Struct.* **182–183**, 1–14 (2020)
78. Kabel, M.: *FeelMath - Mechanical and Thermal Properties of Microstructures*. Accessed 28 Oct 2021
79. Schneider, M., Ospald, F., Kabel, M.: Computational homogenization of elasticity on a staggered grid. *Int. J. Numer. Methods Eng.* **105**(9), 693–720 (2016)
80. Schneider, M.: On non-stationary polarization methods in FFT-based computational micromechanics. *Int. J. Numer. Methods Eng.* **122**(22), 6800–6821 (2021)
81. Zeman, J., Vondřejc, J., Novák, J., Marek, I.: Accelerating a FFT-based solver for numerical homogenization of periodic media by conjugate gradients. *J. Comput. Phys.* **229**(21), 8065–8071 (2010)
82. Brisard, S., Dormieux, L.: FFT-based methods for the mechanics of composites: a general variational framework. *Comput. Mater. Sci.* **49**(3), 663–671 (2010)
83. Schneider, M.: A dynamical view of nonlinear conjugate gradient methods with applications to FFT-based computational micromechanics. *Comput. Mech.* **66**(1), 239–257 (2020)
84. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: *LAPACK Users’ Guide*, 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia (1999)
85. Schmelzle, L.: Implementierung und Bewertung eines Deep Material Networks zur Effektiven Beschreibung des Deformationsverhaltens Kurzglasfaserverstärkter Thermoplaste. Master’s thesis, Karlsruhe Institute of Technology (KIT) (2020)
86. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. *NIPS Autodiff Workshop*, p. 4 (2017)
87. Reddi, S.J., Kale, S., Kumar, S.: On the Convergence of Adam and Beyond, pp. 1–23. [arXiv:1904.09237](https://arxiv.org/abs/1904.09237) (2019)
88. Loshchilov, I., Hutter, F.: SGDR: Stochastic Gradient Descent with Warm Restarts, pp. 1–16. [arXiv:1608.03983](https://arxiv.org/abs/1608.03983) (2017)
89. Kabel, M., Flieger, S., Schneider, M.: Mixed boundary conditions for FFT-based homogenization at finite strains. *Comput. Mech.* **57**(2), 193–210 (2016)
90. Kostenko, Y., Naumenko, K.: Power plant component design using creep and fatigue damage analysis. In: *Proceedings of the 5th Australasian Congress on Applied Mechanics*, pp. 89–94 (2007)
91. Gorash, Y., Altenbach, H., Naumenko, K.: Modeling of primary and secondary creep for a wide stress range. *PAMM* **8**(1), 10207–10208 (2008)

92. Halphen, N., Nguyen, Q.: Sur les Matériaux standards généralisés. *J. Méc.* **14**, 508–520 (1975)
93. Simo, J.C., Hughes, T.J.R.: *Computational Inelasticity*. Springer, New York (1998)

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