Micromechanics-based surrogate models for the response of composites: A critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks

I.B.C.M. B C M Rocha, Pierre Kerfriden, F.P. P van Der Meer

Micromechanics-based surrogate models for the response of composites: A critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks

I. B. C. M. Rocha[1], P. Kerfriden[2,3], F. P. van der Meer[1]

[1] Delft University of Technology, Faculty of Civil Engineering, and Geosciences, P.O. Box 5048, 2600GA Delft, The Netherlands
email: i.barceloscarneiromrocha@tudelft.nl
[2] Mines ParisTech, PSL University, Centre des matériaux, 63-65 Rue Henri-Auguste Desbrueres BP87, F-91003 Évry, France
[3] Cardiff University, School of Engineering, Queen's Buildings, The Parade, Cardiff, CF24 3AA, United Kingdom

April 10, 2020

Abstract

Although being a popular approach for the modeling of laminated composites, mesoscale constitutive models often struggle to represent material response for arbitrary load cases. A better alternative in terms of accuracy is to use the FE^2 technique to upscale microscopic material behavior without loss of generality, but the associated computational effort can be extreme. It is therefore interesting to explore alternative surrogate modeling strategies that maintain as much of the fidelity of FE^2 as possible while still being computationally efficient. In this work, three surrogate modeling approaches are compared in terms of accuracy, efficiency and calibration effort: the state-of-the-art mesoscopic plasticity model by Vogler *et al*. [1], regularized feed-forward neural networks and hyper-reduced-order models obtained by combining the Proper Orthogonal Decomposition (POD) and Empirical Cubature Method (ECM) techniques. Training datasets are obtained from a Representative Volume Element (RVE) model of the composite microstructure with a number of randomly-distributed linear-elastic fibers surrounded by a matrix with pressure-dependent plasticity. The approaches are evaluated with a comprehensive set of numerical tests comprising pure stress cases and three different stress combinations relevant in the design of laminated composites. The models are assessed on their ability to accurately reproduce the training cases as well as on how well they are able to predict unseen stress combinations. Gains in execution time are compared by using the trained surrogates in the FE^2 model of an interlaminar shear test.

Keywords: Laminated composites, Reduced-order modeling, Hyper-reduction, artificial neural networks.

1 Introduction

Numerical analysis of fiber-reinforced composite materials is, by nature, a multiscale endeavor. Although most of the design effort in composites is concentrated at the structural level (macroscale), most of the material characterization effort is spent at the mesoscale (thin coupon-sized specimens) [2, 3]. At the same time, many of the current knowledge gaps in composite behavior stem from physical and chemical processes taking place at the much smaller microscale (individual fibers and surrounding matrix), where performing discerning experiments becomes a complex and delicate task [4, 5]. Bridging these scale gaps through high-fidelity numerical analysis

[6, 7, 8] and increasingly substituting real experiments by *virtual testing* campaigns [9] is seen as the way forward in the design of composite structures.

A popular modeling approach consists in using micromechanical models to calibrate mesoscale constitutive models [10, 1]. The appeal of this approach lies in allowing the use of realistic constitutive models for each microscopic constituent — fibers [4, 11], matrix [12, 13] and fiber/matrx interface [14, 15] — and using homogenization techniques to derive the mesoscopic behavior from a number of numerical microscopic experiments. However, the ability of mesoscopic models to correctly represent the composite material under general stress states is limited by assumptions made in order to minimize the number of parameters to be calibrated. This can be seen, for instance, in [7], where the state-of-the-art mesoscopic plasticity model by Vogler *et al*. [1] is put to the test by comparing its predictions with micromechanical results and found to be lacking in its ability to represent the influence of matrix plasticity in the fiber direction on the longitudinal shear behavior of the composite material, a loading scenario commonly encountered in practice.

An alternative to homogenized mesomodels is the concurrent multiscale (FE^2) approach [16, 17, 18]. FE^2 allows material behavior to be directly derived from embedded microscopic models without introducting any mesoscopic constitutive assumptions. However, even though the method effectively carries microscopic fidelity over to the mesoscale without loss of generality, the computational effort required by having an embedded micromodel at each and every mesoscopic integration point can be extreme [19]. It is therefore interesting to seek alternative strategies that improve computational efficiency without sacrificing the generality of FE^2.

One such strategy consists in reducing the computational complexity of the microscopic boundary-value problem through Model Order Reduction (MOR) techniques: through a series of analysis snapshots obtained before model deployment (*offline training*), reduced-order solution manifolds are computed both for displacements [20, 21] and internal forces [22, 23, 24]. During the many-query multiscale analysis, projection constraints ensure that only solutions belonging to these reduced manifolds are sought, resulting in dramatic reductions in the number of degrees of freedom and constitutive model computations. The advantage of using such dimensionality reduction techniques is that, although the amount of freedom the micromodel has to represent general stress states is reduced, it is still driven by the original high-fidelity microscopic material models and therefore still obeys basic physical assumptions made at the microscale (*e.g*. thermodynamic consistency, loading-unloading conditions). Furthermore, recent innovations allow the training process [25] and basis construction [26] to be optimized, leading to hyper-reduced models with increased accuracy and efficiency.

Alternatively, physics-based constitutive models may be altogether abandoned by employing artificial neural networks as surrogate models [27]. This approach is based on the fact that neural networks are *universal approximators* — *i.e*. capable of approximating any continuous function to an arbitrary level of precision provided that enough parametric freedom is given to the model [28]. A network can be trained with macroscopic stress-strain snapshots from a full-order micromodel and subsequently employed *online* to give predictions of stress and tangent stiffness. Since the early work of Ghaboussi *et al*. [29], a number of efforts have been made to improve predictions by restricting the parameter space by focusing on a fixed macroscopic strain distribution [30], using gated neural layers with memory in order to capture path dependency and unloading [31], including additional microscopic parameters such as material volume fractions in the network input [32] and attempting to infuse the network with physics-based constraints [33]. Nevertheless, the use of artificial neural networks as surrogate constitutive models is still far from widespread, and its applicability to model general stress states of complex micromodels is still an open issue.

In summary, three different alternatives to a fully-resolved micromodel have been discussed: physics-based mesoscale models, hyper-reduced micromodels and artificial neural networks. Conceptually, these three approaches can be seen as entities of the same nature: surrogate models that require an *offline* calibration phase and sacrifice part of the generality and accuracy of a micromodel in favor of computational efficiency. In this work, the three strategies are compared in terms of calibration effort, efficiency and generality of representation. In order to keep the focus on the surrogate modeling techniques, matrix plasticity is the only source of nonlinear microscopic behavior considered in the study. Firstly, the multiscale equilibrium problem to be solved is briefly described. Secondly, each of the three acceleration approaches is presented, starting with a brief description of a state-of-the-art mesoscale plasticity model for composites [1] followed by formulations of the hyper-reduced and neural surrogate models. Finally, the three strategies are put to the test in a number of numerical examples involving both pure stress cases and combined loading conditions.

2 Multiscale analysis of laminated composites

In order to introduce the context of the present discussion, the full-order concurrent multiscale equilibrium problem for which surrogate models are sought is presented. Two distinct spatial scales are identified. In the *mesoscale*, individual composite plies are modeled as homogeneous orthotropic media. Descending to the *microscale*, a Representative Volume Element (RVE) of the composite microstructure is modeled, consisting of a number of unidirectional fibers and surrounding matrix.

When coupling these two scales, the goal is to exploit the high-fidelity information obtained at the microscale to derive the constitutive behavior of a material point at the mesoscale. Before comparing the different approaches to perform this coupling through an *offline* training/calibration phase, this section outlines how an *online* scale coupling can be achieved without mesoscopic constitutive assumptions or loss of generality through the FE^2 technique. In the context of the present study, FE^2 is regarded as the reference solution that represents both the upper bound of model fidelity and the lower bound of computational efficiency. Formulating alternative strategies based on surrogate models entails significantly improving efficiency while retaining as much fidelity as possible.

2.1 Mesoscopic problem

Let Ω be the continuous and homogeneous mesoscopic domain being modeled and let it be bounded by the surfaces Γ_u and Γ_f on which Dirichlet and Neumann boundary conditions are applied, respectively ($\Gamma_\mathrm{u} \cap \Gamma_\mathrm{f} = \varnothing$). Stress equilibrium and strain-displacement relationships in Ω are given by:

$$\mathrm{div}\left(\boldsymbol{\sigma}^\Omega\right) = \mathbf{0} \qquad \boldsymbol{\varepsilon}^\Omega = \frac{1}{2}\left(\nabla\mathbf{u}^\Omega + \left(\nabla\mathbf{u}^\Omega\right)^\mathrm{T}\right) \tag{1}$$

where $\mathrm{div}\left(\cdot\right)$ is the divergence operator, $\nabla\left(\cdot\right)$ is the gradient operator, body forces are neglected and a small strain formulation is adopted. In order to solve for the displacements $\mathbf{u}^\Omega$, a constitutive relation between stresses and strains must be introduced:

$$\boldsymbol{\sigma}^\Omega = \mathcal{D}\left(\boldsymbol{\varepsilon}^\Omega, \boldsymbol{\varepsilon}^\Omega_\mathrm{h}\right) \tag{2}$$

where the dependency on the strain history $\boldsymbol{\varepsilon}^\Omega_\mathrm{h}$ accounts for the possibility of path dependency. For the moment, no assumptions on the behavior of the constitutive operator $\mathcal{D}$ are made. In a general sense, $\mathcal{D}$ should account for the information on material behavior coming from smaller scales that is lost when assuming that Ω is a continuous and homogeneous medium.

In a FE environment, the domain is discretized by a finite element mesh with N degrees of freedom and the equilibrium problem is solved by minimizing the force residual $\mathbf{r}^\Omega \in \mathbb{R}^N$:

$$\mathbf{r}^\Omega = \mathbf{f}^\Omega - \mathbf{f}^\Gamma = \mathbf{0} \quad \text{with} \quad \mathbf{f}^\Omega = \int_\Omega \mathbf{B}^\mathrm{T}\boldsymbol{\sigma}^\Omega \mathrm{d}\Omega \quad \mathbf{f}^\Gamma = \int_{\Gamma_\mathrm{f}} \mathbf{N}^\mathrm{T}\mathbf{t}^\Gamma \mathrm{d}\Gamma \tag{3}$$

where $\mathbf{N}$ and $\mathbf{B}$ contain the shape functions and their spatial derivatives, respectively, $\mathbf{t}^\Gamma$ are the tractions at surface Γ_f and the Dirichlet boundary conditions $\mathbf{u}\big|_{\Gamma_\mathrm{u}} = \overline{\mathbf{u}}^\Gamma$ are implicitly applied. The formulation is completed with the definition of the tangent stiffness matrix $\mathbf{K}^\Omega \in \mathbb{R}^{N\times N}$, used to compute the displacement update $\Delta\mathbf{u}_\mathrm{n} = -\left(\mathbf{K}^\Omega\right)^{-1}\mathbf{r}^\Omega$:

$$\mathbf{K}^\Omega = \int_\Omega \mathbf{B}^\mathrm{T}\mathbf{D}^\Omega\mathbf{B}\mathrm{d}\Omega \qquad \mathbf{D}^\Omega = \frac{\partial\boldsymbol{\sigma}^\Omega}{\partial\boldsymbol{\varepsilon}^\Omega} \tag{4}$$

with $\mathbf{D}^\Omega$ being the tangent material stiffness matrix. Although not explicit in the preceding equations, it is important to note that since composite laminates are anisotropic materials, constitutive computations are performed in a local material coordinate system and rotation operators are used to bring $\boldsymbol{\sigma}^\Omega$ and $\mathbf{D}^\Omega$ back to global coordinates.

2.2 Microscopic problem

Let ω define the microscopic domain of a Representative Volume Element (RVE) of the material where individual fibers and surrounding matrix are modeled. The domain is assumed to be continuous and bounded by the Dirichlet and Neumann surfaces γ_{u} and γ_{f} ($\gamma_{\mathrm{u}} \cap \gamma_{\mathrm{f}} = \varnothing$). Maintaining the small strain assumption and neglecting body forces, stress equilibrium and strains are given by:

$$\operatorname{div}\left(\boldsymbol{\sigma}^{\omega}\right) = \mathbf{0} \qquad \boldsymbol{\varepsilon}^{\omega} = \frac{1}{2}\left(\nabla \mathbf{u}^{\omega} + \left(\nabla \mathbf{u}^{\omega}\right)^{\mathrm{T}}\right) \tag{5}$$

At the microscale, constitutive operators for fibers and matrix are assumed *a priori*. Fibers are modeled as isotropic and linear-elastic and the matrix is modeled with the plasticity model proposed by Melro *et al*. [6]. The matrix response starts as linear-elastic and transitions to plasticity with pressure-dependent hardening until the response reaches a perfectly-plastic regime. The model is briefly described in the following, with most formulation details being omitted for compactness. For further details, the interested reader is referred to [6, 7].

The stress-strain relationship in tensor notation is given by:

$$\boldsymbol{\sigma} = \mathbf{D}_{\mathrm{e}}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\mathrm{p}}\right) \tag{6}$$

where $\mathbf{D}_{\mathrm{e}}$ is the fourth-order elastic stiffness matrix and an additive decomposition between elastic and plastic strains ($\boldsymbol{\varepsilon}_{\mathrm{p}}$) is assumed. The onset of plasticity is defined by a pressure-dependent paraboloidal yield surface:

$$f\left(\boldsymbol{\sigma}, \sigma_{\mathrm{c}}, \sigma_{\mathrm{t}}\right) = 6J_2 + 2I_1\left(\sigma_{\mathrm{c}} - \sigma_{\mathrm{t}}\right) - 2\sigma_{\mathrm{c}}\sigma_{\mathrm{t}} \tag{7}$$

with I_1 and J_2 being stress invariants and the yield stresses in compression (σ_{c}) and tension (σ_{t}) being functions of the equivalent plastic strain $\varepsilon_{\mathrm{p}}^{\mathrm{eq}}$ in order to allow for the occurrence of hardening:

$$\sigma_{\mathrm{c}} = \sigma_{\mathrm{c}}\left(\varepsilon_{\mathrm{p}}^{\mathrm{eq}}\right) \qquad \sigma_{\mathrm{t}} = \sigma_{\mathrm{t}}\left(\varepsilon_{\mathrm{p}}^{\mathrm{eq}}\right) \qquad \dot{\varepsilon}_{\mathrm{p}}^{\mathrm{eq}} = \sqrt{\frac{1}{1-2\nu_{\mathrm{p}}}\dot{\boldsymbol{\varepsilon}}_{\mathrm{p}} : \dot{\boldsymbol{\varepsilon}}_{\mathrm{p}}} \tag{8}$$

where ν_{p} is the plastic Poisson's ratio. The development of plastic strains is dictated by the non-associative flow rule:

$$\Delta\boldsymbol{\varepsilon}_{\mathrm{p}} = \Delta\gamma\left(3\mathbf{S} + \frac{1-2\nu_{\mathrm{p}}}{1+\nu_{\mathrm{p}}} I_1 \mathbf{I}\right) \tag{9}$$

where $\Delta\gamma$ is the plastic multiplier increment computed through a return mapping procedure [7] and $\mathbf{S}$ is the deviatoric stress tensor. The formulation is completed by the definition of the consistent tangent operator, obtained by differentiating Eq. (6) with respect to the strains [7].

With constitutive models in place, the equilibrium residual $\mathbf{r}^{\omega}$ to be minimized is computed as:

$$\mathbf{r}^{\omega} = \mathbf{f}^{\omega} - \mathbf{f}^{\gamma} = \mathbf{0} \quad \text{with} \quad \mathbf{f}^{\omega} = \int_{\omega} \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma}^{\omega} \mathrm{d}\omega \quad \mathbf{f}^{\gamma} = \int_{\gamma_{\mathrm{f}}} \mathbf{N}^{\mathrm{T}} \mathbf{t}^{\gamma} \mathrm{d}\gamma \tag{10}$$

2.3 Scale coupling

The basic idea behind the FE^2 approach consists in defining the mesoscopic constitutive operator $\mathcal{D}$ of Eq. (2) as the homogenized response of a finite element micromodel embedded at each integration point of the domain Ω (Fig. 1). Assuming the principle of separation of scales holds ($\omega \ll \Omega$) [16], a link between the two scales is enforced by satisfying:

$$\mathbf{u}^{\omega} = \boldsymbol{\varepsilon}^{\Omega}\mathbf{x}^{\omega} + \widetilde{\mathbf{u}} \tag{11}$$

where $\widetilde{\mathbf{u}}$ is a fluctuation displacement field subjected to $\widetilde{\mathbf{u}}^{\gamma+} = \widetilde{\mathbf{u}}^{\gamma-}$, where γ_- and γ_+ represent pairs of opposing microdomain boundaries. In practice, enforcing Eq. (11) entails converting the macroscopic strain $\boldsymbol{\varepsilon}^{\Omega}$ into prescribed displacements at the corners of the micromodel, tying nodes at γ_- and γ_+ through periodic boundary conditions and solving the resultant boundary-value problem [18].

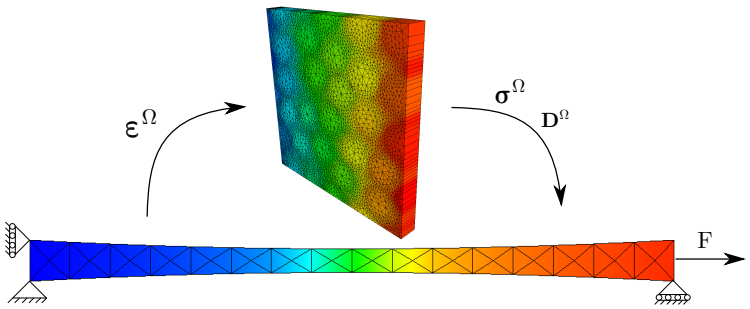

Figure 1: The FE^2 approach: A concurrent link is established between meso and microscales.

After convergence of the microscopic nonlinear analysis, the Hill-Mandel principle is used to recover the mesoscopic stresses:

$$\boldsymbol{\sigma}^{\Omega} = \frac{1}{\omega}\int_{\omega} \boldsymbol{\sigma}^{\omega} \mathrm{d}\omega \tag{12}$$

while the tangent stiffness is obtained through a probing operator $\mathcal{P}$ based on the microscopic stiffness matrix $\mathbf{K}^{\omega}$ according to the procedure in [34]:

$$\mathbf{D}^{\Omega} = \mathcal{P}\left(\mathbf{K}^{\omega}\right) \tag{13}$$

which completes the formulation. The FE^2 approach effectively defines the operator $\mathcal{D}$ through an implicit procedure that involves no mesoscopic constitutive assumptions. However, the associated computational effort can be prohibitive even for simple applications. In the next sections, three alternative strategies for defining $\mathcal{D}$ are presented.

3 Mesoscale constitutive model

The mesoscopic constitutive model proposed by Vogler *et al.* [1] and later revisited by Van der Meer [7] is briefly presented here as a way of defining the $\mathcal{D}$ operator of Eq. (2) through a physics-based model that effectively condenses the microscale material behavior into a small number of mesoscale constitutive parameters calibrated with micromechanical simulations.

A unidirectional composite lamina is modeled as an orthotropic material with pressure-dependent plasticity and assuming an additive decomposition of strains. The stress-strain relationship is similar to the one of Eq. (6) but the stiffness tensor $\mathbf{D}_{\mathrm{e}}$ is now orthotropic. The onset of plasticity is defined by the following yield surface, written in Voigt notation:

$$f = \frac{1}{2}\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{A}\boldsymbol{\sigma} + \mathbf{a}^{\mathrm{T}}\boldsymbol{\sigma} - 1 \tag{14}$$

where $\mathbf{A}$ is given by:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}\alpha_1 + 2\alpha_{32} & -\frac{1}{2}\alpha_1 + 2\alpha_{32} & 0 & 0 & 0 \\ 0 & -\frac{1}{2}\alpha_1 + 2\alpha_{32} & \frac{1}{2}\alpha_1 + 2\alpha_{32} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\alpha_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\alpha_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\alpha_2 \end{bmatrix} \tag{15}$$

and $\mathbf{a} = \begin{bmatrix} 0 & \alpha_3 & \alpha_3 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$. The α coefficients are piecewise-linear functions of the equivalent plastic strain $\varepsilon_{\mathrm{p}}^{\mathrm{eq}}$ and pressure-dependency is introduced by allowing for distinct values of α_{32} and α_3 to be defined depending on the sign of $\sigma_2 + \sigma_3$.

Plastic strain evolution is dictated by the flow rule:

$$\Delta\boldsymbol{\varepsilon}_{\mathrm{p}} = \Delta\gamma\mathbf{G}\boldsymbol{\sigma} \tag{16}$$

where $\Delta\gamma$ is the plastic multiplier computed by a return mapping procedure [7] and $\mathbf{G}$ is given by:

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -\nu_{\mathrm{p}} & 0 & 0 & 0 \\ 0 & -\nu_{\mathrm{p}} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\left(1+\nu_{\mathrm{p}}\right) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\left(1+\nu_{\mathrm{p}}\right) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\left(1+\nu_{\mathrm{p}}\right) \end{bmatrix} \tag{17}$$

with ν_{p} being the plastic Poisson's ratio.

Calibration of the mesomodel consists in determining ν_{p} and the α coefficients through a set of micromechanical numerical experiments. The procedure used here follows the one described in [7]. From the homogenized stress-strain curves obtained from the micromodels, the components of $\mathbf{D}_{\mathrm{e}}$ are obtained and with those the equivalent plastic strain histories. With values for $\boldsymbol{\sigma}$ and $\varepsilon_{\mathrm{p}}^{\mathrm{eq}}$, the model parameters are computed as:

$$\alpha_1\left(\varepsilon_{\mathrm{p}}^{\mathrm{eq}}\right) = \frac{1}{\sigma_{\mathrm{ts}}^2} \qquad \alpha_2\left(\varepsilon_{\mathrm{p}}^{\mathrm{eq}}\right) = \frac{1}{\sigma_{\mathrm{ls}}^2} \tag{18}$$

$$\alpha_{32}^{\mathrm{t}}\left(\varepsilon_{\mathrm{p}}^{\mathrm{eq}}\right) = \frac{1 - \frac{\sigma_{\mathrm{ut}}}{2\sigma_{\mathrm{bt}}} - \alpha_1 \frac{\sigma_{\mathrm{ut}}^2}{4}}{\sigma_{\mathrm{ut}}^2 - 2\sigma_{\mathrm{bt}}\sigma_{\mathrm{ut}}} \qquad \alpha_{32}^{\mathrm{c}}\left(\varepsilon_{\mathrm{p}}^{\mathrm{eq}}\right) = \frac{1 - \frac{\sigma_{\mathrm{uc}}}{2\sigma_{bc}} - \alpha_1 \frac{\sigma_{\mathrm{uc}}^2}{4}}{\sigma_{\mathrm{uc}}^2 - 2\sigma_{\mathrm{bc}}\sigma_{\mathrm{uc}}} \tag{19}$$

$$\alpha_3^{\mathrm{t}}\left(\varepsilon_{\mathrm{p}}^{\mathrm{eq}}\right) = \frac{1}{2\sigma_{\mathrm{bt}}} - 2\alpha_{32}^{\mathrm{t}}\sigma_{\mathrm{bt}} \qquad \alpha_3^{\mathrm{c}}\left(\varepsilon_{\mathrm{p}}^{\mathrm{eq}}\right) = \frac{1}{2\sigma_{\mathrm{bc}}} - 2\alpha_{32}^{\mathrm{c}}\sigma_{\mathrm{bc}} \tag{20}$$

where *ts* stands for transverse shear, *ls* for longitudinal shear, *ut* and *uc* for uniaxial tension and compression, respectively, and *bt* and *bc* for biaxial tension and compression, respectively. With this relatively limited amount of calibration data, the model can be used to predict the behavior under general stress states.

4 Neural networks

An alternative to a physically-motivated mesoscopic model is the use of a purely data-driven approach, the idea consisting in the introduction of a parametric regression model $\mathcal{S}$ used to compute an approximation $\widehat{\boldsymbol{\sigma}}$ of the stresses:

$$\widehat{\boldsymbol{\sigma}} = \mathcal{S}\left(\boldsymbol{\varepsilon}^{\Omega}, \mathbf{W}\right) \tag{21}$$

where $\mathbf{W}$ are model parameters. In contrast to the parameters in Eqs. (18) to (20), parameters in $\mathbf{W}$ have no direct physical meaning, being instead calibrated through a fitting procedure based on observations of the actual micromechanical model:

$$\mathbf{W} = \underset{\overline{\mathbf{W}}}{\arg\min} \sum_{i \in \mathbf{X}} \left\| \widehat{\boldsymbol{\sigma}}_i\left(\boldsymbol{\varepsilon}_i^{\Omega}, \overline{\mathbf{W}}\right) - \boldsymbol{\sigma}_i^{\Omega}\left(\boldsymbol{\varepsilon}_i^{\Omega}\right) \right\|^2 \tag{22}$$

where $\mathbf{X} \in \mathbb{R}^{2n_\varepsilon \times P}$ is a snapshot matrix with P $\boldsymbol{\varepsilon}^{\Omega}$-$\boldsymbol{\sigma}^{\Omega}$ pairs obtained from micromodel executions. Given enough parametric freedom, the surrogate should be able to encapsulate the observed constitutive information ($\mathbf{X}$) and provide accurate stress predictions when presented with previously unseen values of $\boldsymbol{\varepsilon}^{\Omega}$.

Here, $\mathcal{S}$ is chosen to be the feed-forward artificial neural network shown in Fig. 2, being composed of a number of fully-connected neural layers (*dense layers*) followed by a *dropout layer* that regularizes the model. When used to make predictions, strains are fed to the first neural layer (*input layer*) and values are propagated until the final layer is reached (*output layer*), at which point the output neurons contain the predicted stress $\widehat{\boldsymbol{\sigma}}$. In the next sections, each component of the network is briefly described and further details are given on how training is performed.

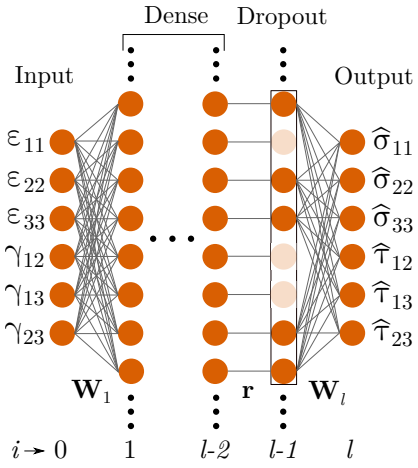

Figure 2: A neural network acting as a surrogate constitutive model. An arbitrary number of *dense* neural layers is combined with a single *dropout* layer that regularizes model response.

4.1 Dense layer

A dense neural layer i propagates neuron states ($\mathbf{a}$) from the previous layer $i-1$ and subsequently applies an activation function φ to the resulting values in order to introduce nonlinearity in the network response:

$$\mathbf{v}_i = \mathbf{W}_i \mathbf{a}_{i-1} + \mathbf{b}_i \quad \Rightarrow \quad \mathbf{a}_i = \varphi\left(\mathbf{v}_i\right) \tag{23}$$

where $\mathbf{W}_i \in \mathbb{R}^{n_i \times n_{i-1}}$ is a weight matrix and $\mathbf{b}_i \in \mathbb{R}^{n_i}$ is a bias term, with n_i being the number of neurons of layer i. The activation function φ here represents the element-wise application of the sigmoid function:

$$\varphi\left(v\right) = \frac{\mathrm{e}^v}{\mathrm{e}^v + 1} \tag{24}$$

on the neuron values, with the exception of the output layer which is left unactivated ($\mathbf{a}_l = \mathbf{v}_l$). Different activation functions are used depending on the intended application [35], with the sigmoid function being a popular choice for building regression models. In general, increasing n_i leads to a higher representational capability, following from the intuitive fact that the amount of fitting freedom of the model increases with the number of trainable parameters. In practice, however, models that are too large tend to exactly represent training data but fail to generalize to unseen inputs (*overfitting*) [36].

4.2 Dropout layer

Dropout is an increasingly popular regularization strategy used avoid the phenomenon of overfitting [37]. Here, a dropout layer is positioned immediately before the output layer and stochastically deactivates some of the neurons coming from the previous layer:

$$\mathbf{a}_{l-1} = \frac{1}{1 - r_\mathrm{d}} \left(\mathbf{r} \odot \mathbf{a}_{l-2}\right) \tag{25}$$

where $\odot$ indicates element-wise multiplication, $r_\mathrm{d} \in (0, 1]$ is the probability that a given neuron is set to zero and $\mathbf{r} \in \{0, 1\}^{n_{l-2}}$ is a boolean vector determined by drawing from a uniform unit distribution and comparing the value to r_d. If the drawn value is lower than the dropout rate, the correspondent element of $\mathbf{r}$ is set to zero. In order to keep the average of the neuron values unchanged after dropout, neurons that are not deactivated are scaled by $1 - r_\mathrm{d}$.

During training, $\mathbf{r}$ is redrawn each time the network is used to make a prediction. This means that, on average, neurons of layer $l-2$ will have been deactivated at least once. This introduces a regularizing effect

because the network cannot rely on the availability of any given neuron in order to make accurate predictions. When using the network model *online*, the dropout layer is removed — which is equivalent to setting r_d to zero — and all neurons contribute to the response.

4.3 Training

The objective of the training process is to minimize a *loss function* that represents how well predictions match actual model observations:

$$L = \frac{1}{P}\sum_{j=1}^{P}\frac{1}{2}\left\|\boldsymbol{\sigma}\left(\boldsymbol{\varepsilon}_j\right) - \widehat{\boldsymbol{\sigma}}\left(\boldsymbol{\varepsilon}_j\right)\right\|^2 \tag{26}$$

where P is the number of snapshots and the $1/2$ factor is added for convenience when computing the gradients of L. In order to keep track of how well the model generalizes to unseen data, it is common to remove part of the snapshots from the training process to act as a *validation set* and use them to compute a separate error measure to be used as stopping criterion for the optimization.

Based on this objective function, a Stochastic Gradient Descent (SGD) optimization algorithm is used to update the trainable parameters $\mathbf{W}$ and $\mathbf{b}$:

$$\mathbf{W}^\mathrm{n} = \mathbf{W}^\mathrm{o} - \mathcal{A}\left(\frac{1}{B}\sum_{j}^{B}\frac{\partial L_j}{\partial \mathbf{W}}\right) \qquad \mathbf{b}^\mathrm{n} = \mathbf{b}^\mathrm{o} - \mathcal{A}\left(\frac{1}{B}\sum_{j}^{B}\frac{\partial L_j}{\partial \mathbf{b}}\right) \tag{27}$$

where L_j is the loss term of the j-th sample, o indicates current values, n indicates updated values and B is the size of the sample *mini-batch* used in the update. The idea behind using a mini-batch instead of updating the parameters using either one sample at a time or all samples at once is that it provides a balance between speed of convergence and gradient variance. In any case, a complete solver iteration (*epoch*) is only complete after the model has seen every sample in the training set — *i.e.* after approximately P/B mini-batches. Finally, the operator $\mathcal{A}$ depends on the choice of solver. Here, the *Adam* solver proposed by Kingma and Ba [38] is adopted.

In order to compute the gradients appearing in Eq. (27), a *backpropagation* procedure is adopted: based on the network state ($\mathbf{v}$, $\mathbf{a}$ and $\mathbf{r}$) after computing each[1] training sample, the chain rule is used to propagate the derivative of the loss function starting from the output layer and progressively moving back through the network. For this, an auxiliary quantity $\mathbf{d}_i \in \mathbb{R}^{n_i}$ is defined for each layer. At the output layer l, it is simply defined as:

$$\mathbf{d}_l = \frac{\partial L}{\partial \mathbf{a}_l} = \widehat{\boldsymbol{\sigma}} - \boldsymbol{\sigma} \tag{28}$$

Next, the effect of the activation function is taken into account:

$$\overline{\mathbf{d}}_i = \mathbf{d}_i \odot \frac{\partial \varphi}{\partial v}\left(\mathbf{v}_i\right) \tag{29}$$

after which it is possible to compute the gradients of the trainable parameters:

$$\frac{\partial L}{\partial \mathbf{W}_i} = \overline{\mathbf{d}}_i \mathbf{a}_i^\mathrm{T} \qquad \frac{\partial L}{\partial \mathbf{b}_i} = \overline{\mathbf{d}}_i \tag{30}$$

Finally, the values of $\mathbf{d}$ of the previous layer (the next layer to be backpropagated) can be computed as:

$$\mathbf{d}_{i-1} = \mathbf{W}_i^\mathrm{T}\,\overline{\mathbf{d}}_i \tag{31}$$

and the algorithm moves to Eq. (29) for layer $i-1$. For the dropout layer, since it does not have any trainable parameters, the effect of the stochastic dropout is simply backpropagated to the previous layer:

$$\mathbf{d}_{i-1} = \overline{\mathbf{d}}_i = \frac{1}{1-r_\mathrm{d}}\mathbf{r} \odot \mathbf{d}_i \tag{32}$$

[1] In practice, since the model is only updated between mini-batches, the feed-forward and backpropagations of all samples in a mini-batch are performed at the same time, with $\mathbf{v}$, $\mathbf{a}$, $\mathbf{r}$ and $\mathbf{d}$ taking a matrix form. This reduces computational overhead and allows for fast GPU computations to be performed.

4.4 Use as constitutive model

To make new stress predictions, the input layer is set to the applied mesoscopic strain, a complete forward pass is performed and the final activated neuron values of the output layer give the predicted stress:

$$\mathbf{a}_0 = \boldsymbol{\varepsilon}^\Omega \qquad \widehat{\boldsymbol{\sigma}} = \mathbf{a}_l \tag{33}$$

For the consistent tangent stiffness, it is necessary to compute the jacobian $\mathbf{J}$ of the network:

$$\mathbf{D}^\Omega = \frac{\partial \widehat{\boldsymbol{\sigma}}}{\partial \boldsymbol{\varepsilon}} = \frac{\partial \mathbf{a}_l}{\partial \mathbf{v}_0} = \mathbf{J} \tag{34}$$

which is obtained with a backward pass through the network (from output to input):

$$\mathbf{J}_i = \mathbf{J}_{i+1} \mathbf{I}_i^{\varphi'} \mathbf{W}_i \quad \text{with} \quad \mathbf{J}_{l+1} = \mathbf{I} \tag{35}$$

where $\mathbf{I}_i^{\varphi'}$ is a matrix whose diagonal contains the derivatives of the activation function with respect to the neuron values $\mathbf{v}$:

$$\mathbf{I}_i^{\varphi'} = \operatorname{diag}\left(\frac{\partial \varphi}{\partial v} (\mathbf{v}_i) \right) \tag{36}$$

5 Hyper-reduced-order modeling

Instead of resorting to surrogate mesoscopic models, FE^2 can be made efficient by accelerating the associated microscopic boundary-value problems. In this section, two complexity reduction operations are applied to the equilibrium problem of Section 2.2. First, the number of degrees of freedom of the problem is drastically reduced, followed by a hyper-reduction phase on which a reduced global integration scheme for internal forces is defined. The techniques are only described briefly in order to keep the focus on their application to the problem at hand. More details on the underlying formulations can be found in [39].

5.1 Proper Orthogonal Decomposition (POD)

The first strategy consists in projecting the original equilibrium problem of size N onto a reduced solution manifold spanned by a basis matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N \times n}$:

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 \cdots \boldsymbol{\phi}_n \end{bmatrix} \tag{37}$$

where $\boldsymbol{\phi}_i$ are a set of orthonormal basis vectors that represent global displacement modes. By constraining the possible displacement configurations to the ones lying in the latent space defined by $\boldsymbol{\Phi}$, the number of degrees of freedom of the problem is reduced from N to $n \ll N$. The full-order displacement field is recovered as a linear combination of the latent variables $\boldsymbol{\alpha} \in \mathbb{R}^n$:

$$\mathbf{u}^\omega = \boldsymbol{\Phi} \boldsymbol{\alpha} \tag{38}$$

In order to solve for $\boldsymbol{\alpha}$, the full-order residual of Eq. (10) is constrained to lie on the reduced space through the Galerkin projection $\boldsymbol{\Phi}^{\mathrm{T}} \mathbf{r}^\omega = \mathbf{0}$, yielding reduced versions of the internal force vector and stiffness matrix:

$$\mathbf{f}_{\mathrm{r}}^\omega = \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{f}^\omega \qquad \mathbf{K}_{\mathrm{r}}^\omega = \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{K}^\omega \boldsymbol{\Phi} \tag{39}$$

5.2 Empirical Cubature Method (ECM)

Even though the POD-reduced problem has only a small number of degrees of freedom, solving for $\boldsymbol{\alpha}$ still involves computing stresses at every integration point in order to obtain $\mathbf{f}^\omega$ and $\mathbf{K}^\omega$ for use in Eq. (39). However, given the fact that $\mathbf{f}_{\mathrm{r}}^\omega$ is of small dimensionality, it is intuitive to surmise that the amount of constitutive information needed to define it is also significantly reduced.

This hypothesis may be posited more formally as follows: From the complete set of M integration points with original integration weights w_i, it is possible to define a reduced set of $m \ll M$ integration points with modified integration weights ϖ_j such that the approximation:

$$\mathbf{f}_{\mathrm{r}}^{\omega} = \boldsymbol{\Phi}^{\mathrm{T}} \left(\sum_{i=1}^{M} \mathbf{B}^{\mathrm{T}} \left(\mathbf{x}_i \right) \boldsymbol{\sigma}^{\omega} \left(\mathbf{x}_i \right) w_i \right) \approx \boldsymbol{\Phi}^{\mathrm{T}} \left(\sum_{j=1}^{m} \mathbf{B}^{\mathrm{T}} \left(\mathbf{x}_j \right) \boldsymbol{\sigma}^{\omega} \left(\mathbf{x}_j \right) \varpi_j \right) \tag{40}$$

leads to a negligible loss of accuracy. This idea is the basis for the Empirical Cubature Method (ECM) proposed by Hernández *et al*. [22]. The reduced set $\boldsymbol{\mathcal{Z}}$ of m integration points is chosen from among the original M points by using a Greedy least-squares procedure that solves:

$$(\boldsymbol{\beta}, \boldsymbol{\mathcal{Z}}) = \underset{\overline{\beta} \geq \mathbf{0}\,,\, \overline{\boldsymbol{\mathcal{Z}}}}{\arg\min} \left\| \mathbf{J}_{\overline{\boldsymbol{\mathcal{Z}}}}\, \overline{\boldsymbol{\beta}} - \mathbf{b} \right\|^2 \tag{41}$$

where $\mathbf{J}$ and $\mathbf{b}$ are given by:

$$\mathbf{J} = \begin{bmatrix} \boldsymbol{\Lambda} & \sqrt{\mathbf{w}} \end{bmatrix}^{\mathrm{T}} \qquad \mathbf{b} = \begin{bmatrix} \mathbf{0} & \omega \end{bmatrix}^{\mathrm{T}} \tag{42}$$

where $\boldsymbol{\Lambda}$ is a basis matrix for the contribution of each integration point to the global reduced force vector $\mathbf{f}_{\mathrm{r}}^{\omega}$. With $\boldsymbol{\beta}$, the modified integration weights of points in $\boldsymbol{\mathcal{Z}}$ are computed as $\varpi_i = \sqrt{w_i}\beta_i$. For details on the Greedy selection procedure, the reader is referred to [22].

During the *online* FE^2 analysis, the responses of integration points not included in $\boldsymbol{\mathcal{Z}}$ are never computed, leading to a full-order internal force vector composed almost solely by zeros. On the other hand, the homogenization procedure of Section 2.3 requires a complete assembly of $\mathbf{f}^{\omega}$ and $\mathbf{K}$. In order to bypass this issue, a tangent mode contribution matrix $\mathbf{H} \in \mathbb{R}^{n \times n_{\varepsilon}}$ is computed for each micromodel such as to satisfy:

$$\boldsymbol{\alpha} = \mathbf{H} \boldsymbol{\varepsilon}^{\Omega} \tag{43}$$

where $\boldsymbol{\alpha}$ are the latent variable values resulting from solving the equilibrium problem with applied macroscopic strains $\boldsymbol{\varepsilon}^{\Omega}$. With this operator, the homogenized stress and stiffness are computed as:

$$\boldsymbol{\sigma}^{\Omega} = \mathbf{H}^{\mathrm{T}} \mathbf{f}_{\mathrm{r}}^{\omega} \qquad \mathbf{D}^{\Omega} = \mathbf{H}^{\mathrm{T}} \mathbf{K}_{\mathrm{r}}^{\omega} \mathbf{H} \tag{44}$$

5.3 Training

Both reduction stages are constructed with mechanical behavior information that must be computed before model deployment, similar to the calibration procedure of Section 4.3. For POD, the basis matrix $\boldsymbol{\Phi}$ is computed from a series of P displacement snapshots $\mathbf{X}_{\mathrm{u}} \in \mathbb{R}^{N \times P}$ decomposed into elastic and inelastic parts:

$$\mathbf{X}_{\mathrm{u}} = \begin{bmatrix} \mathbf{X}_{\mathrm{e}} & \mathbf{X}_{\mathrm{i}} \end{bmatrix} \tag{45}$$

where a snapshot is considered inelastic if at least one integration point in ω has non-zero equivalent plastic strain. Following the elastic/inelastic training strategy presented in [22], the basis $\boldsymbol{\Phi} \in \mathbb{R}^{N \times (n_{\mathrm{e}} + n_{\mathrm{i}})}$ is given by:

$$\boldsymbol{\Phi} = \begin{bmatrix} \overline{\mathbf{U}}_{\mathrm{e}} & \overline{\mathbf{U}}_{\mathrm{i}} \end{bmatrix} \tag{46}$$

where each portion of the basis (n_{e} elastic and n_{i} inelastic modes) is obtained through a truncated Singular Value Decomposition (SVD) operation:

$$\overline{\mathbf{X}}_{\mathrm{e}} \approx \overline{\mathbf{U}}_{\mathrm{e}} \overline{\mathbf{S}}_{\mathrm{e}} \overline{\mathbf{T}}_{\mathrm{e}}^{\mathrm{T}} \qquad \overline{\mathbf{X}}_{\mathrm{i}} \approx \overline{\mathbf{U}}_{\mathrm{i}} \overline{\mathbf{S}}_{\mathrm{i}} \overline{\mathbf{T}}_{\mathrm{i}}^{\mathrm{T}} \tag{47}$$

with the modified snapshot matrices

$$\overline{\mathbf{X}}_{\mathrm{e}} = \mathbf{Y} \left(\mathbf{Y}^{\mathrm{T}} \mathbf{X} \right) \qquad \overline{\mathbf{X}}_{\mathrm{i}} = \mathbf{X} - \overline{\mathbf{X}}_{\mathrm{e}} \tag{48}$$

and $\mathbf{Y}$ being a basis matrix computed from the SVD of $\mathbf{X}_\mathrm{e}$. In order to guarantee that every possible stress state in the elastic regime is exactly reproduced by the reduced model, the decomposition that generates $\overline{\mathbf{U}}_\mathrm{e}$ is truncated at n_ε components ($n_\varepsilon = 6$ for three-dimensional micromodels). For $\overline{\mathbf{U}}_\mathrm{i}$, the basis includes all basis vectors whose associated singular values satisfy the condition:

$$\frac{\overline{S}_\mathrm{i}^j}{\overline{S}_\mathrm{i}^1} > \epsilon_\mathrm{sv} \tag{49}$$

with $\overline{S}_\mathrm{i}^1$ being the first (and highest) singular value and ϵ_sv a truncation tolerance.

For ECM, training consists in running the POD-reduced model for the same original training cases[2] and collecting snapshots of stresses at every integration point. Following again the elastic/inelastic strategy, a basis matrix for stresses $\mathbf{\Psi} \in \mathbb{R}^{Mn_\epsilon \times q}$ is computed, with $q = n_\mathrm{e} + n_\mathrm{i}$ in order to keep the truncations consistent with the ones from the first reduction phase.

With $\mathbf{\Phi}$, the basis matrix for internal forces used in Eq. (42) can be obtained:

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_1 & \mathbf{\Lambda}_2 & \cdots & \mathbf{\Lambda}_q \end{bmatrix} \tag{50}$$

with each of the q submatrices $\mathbf{\Lambda}_j \in \mathbb{R}^{M \times n}$ being given by:

$$\mathbf{\Lambda}_j = \begin{bmatrix} \sqrt{w_1}\left(\mathbf{f}_{\mathrm{r}j}^1\left(\mathbf{x}_1\right) - \frac{1}{\omega}\mathbf{f}_{\mathrm{r}j}^\omega\right) \\ \sqrt{w_2}\left(\mathbf{f}_{\mathrm{r}j}^2\left(\mathbf{x}_2\right) - \frac{1}{\omega}\mathbf{f}_{\mathrm{r}j}^\omega\right) \\ \vdots \\ \sqrt{w_M}\left(\mathbf{f}_{\mathrm{r}j}^M\left(\mathbf{x}_M\right) - \frac{1}{\omega}\mathbf{f}_{\mathrm{r}j}^\omega\right) \end{bmatrix} \tag{51}$$

and the contribution of each integration point being:

$$\mathbf{f}_{\mathrm{r}j}^i = \mathbf{\Phi}_i^\mathrm{T}\mathbf{B}_i^\mathrm{T}\mathbf{s}_j\boldsymbol{\psi}_j \tag{52}$$

where $\mathbf{\Phi}_i$ is the submatrix of $\mathbf{\Phi}$ that contains the degrees of freedom of the finite element that contains point i, $\mathbf{B}_i$ is the matrix of shape function derivatives evaluated at point i and $\mathbf{s}_j$ and $\boldsymbol{\phi}_j$ are respectively the singular value and left-singular vector associated with the j-th mode of $\mathbf{\Psi}$.

6 Comparing the strategies

The surrogate modeling strategies have been implemented in an in-house Finite Element code based on the Jem/Jive C++ numerical analysis library [40]. All models were executed on a single core of a Xeon E5-2630V4 processor on a cluster node with 128 GB RAM running CentOS 7.

The micromodel used as a basis for training the reduced-order models is the one shown in Fig. 1. This is the same RVE adopted by Van der Meer in [7] and is assumed to be sufficiently representative of the mechanical response of a mesoscopic material point. Material properties for both the micromodel and the calibrated meso-model of Section 3 are also adopted from [7]. In order to guarantee constant stress ratios in biaxial scenarios while avoiding large strain steps during the perfect plasticity regime, a special arc-length constraint a is adopted:

$$a = \left(\sum_i \mathrm{sign}\left(f_i^\Gamma\right) u_i\right) - \overline{u} = 0 \quad \text{with} \quad \frac{\partial a}{\partial \lambda} = 0 \quad \frac{\partial a}{\partial u_i} = \mathrm{sign}\left(f_i^\Gamma\right) \tag{53}$$

with which the load factor λ that scales unit forces applied at the corner nodes of the RVE is controlled so as to guarantee that the unsigned sum of displacements at the same locations is equal to a prescribed value $\overline{u}$. All snapshots used for training come from models loaded monotonically with a constant stress ratio (proportional loading) until the norm of the strain at controlled nodes reaches a value of 0.1. To test the trained surrogates,

[2] Since ECM is built as an approximation of the POD-reduced model response, this second training phase is performed for consistency.

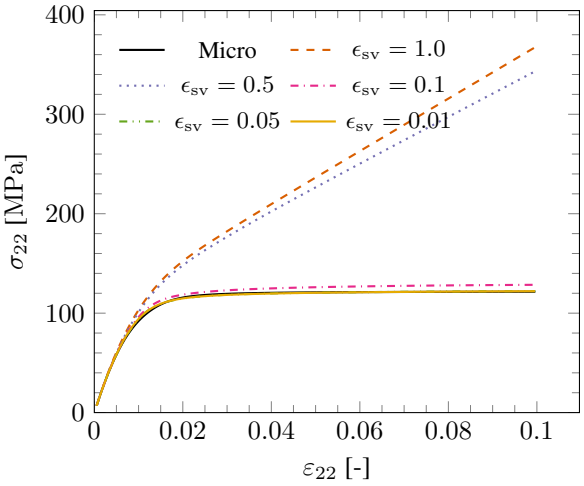

Figure 3: Hyper-reduced model trained with pure stress states. Predictions improve as the truncation tolerance ϵ_{sv} is reduced.

a homogeneous mesoscopic 1-element model[3] with a single integration point and the same dimensions as the original micromodel is used, with the fiber direction (1-axis) aligned with the mesoscopic x-axis.

Neural networks with a single hidden dense layer are considered. Deeper networks with up to 5 hidden layers have also been investigated, but were found to provide lower accuracy than shallow networks with a similar number of parameters. Results from these deeper networks are therefore not included in the discussion. Unless otherwise specified, training sets are formed by randomly drawing $80\,\%$ of the samples of the original dataset without replacement, with the remaining $20\,\%$ serving as a validation set. At the beginning of training, network biases are initialized as zero and weights are initialized with draws from an uniform distribution in the interval $[-1, 1]$ and scaled with the factor $\sqrt{\frac{6}{n_i+n_{i-1}}}$ [41]. The dropout rate is fixed at $r_{\mathrm{d}} = 0.05$ for all models. Although this is a much lower rate than the one adopted for instance in [26], it is found to provide sufficient regularization for the network and dataset sizes treated in this study. For the SGD solver, the default values recommended in [38] are used for all hyperparameters. All models are trained for a total of $200\,000$ epochs and the final model parameters are the ones associated with the lowest historical validation error. The only hyper-parameter to be studied is therefore the width n_1 of the hidden dense layer.

6.1 Pure stress states

First, reduced models are trained to reproduce the material behavior of a single unidirectional composite layer under isolated stress components, *i.e.* uniaxial cases in the parameter space. Here the training dataset consists of twelve stress-strain curves, two for each of the $n_\varepsilon = 6$ mesoscopic strain components (positive and negative directions). From this point on, strain and stress components are expressed in the local mesoscale coordinate system — *i.e.* $\{\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \gamma_{12}, \gamma_{13}, \gamma_{23}\}$, where the 1-axis is the fiber direction and the superscript Ω is dropped for compactness.

Hyper-reduced models are trained with different values of the inelastic SVD tolerance ϵ_{sv} (Eq. (49)). The resultant model predictions for the transverse stress σ_{22} are shown in Fig. 3. For high values of ϵ_{sv} — *i.e.* with a small number of inelastic modes — the plasticity response is not correctly captured, with predictions improving as the tolerance is lowered and more modes are added. Note that the snapshot decomposition of Section 5.3 effectively guarantees an exact response during the elastic regime. A similar response is observed for the remaining five strain components.

[3]For hyper-reduction, this is actually a 1-element FE^2 problem with a hyper-reduced micromodel

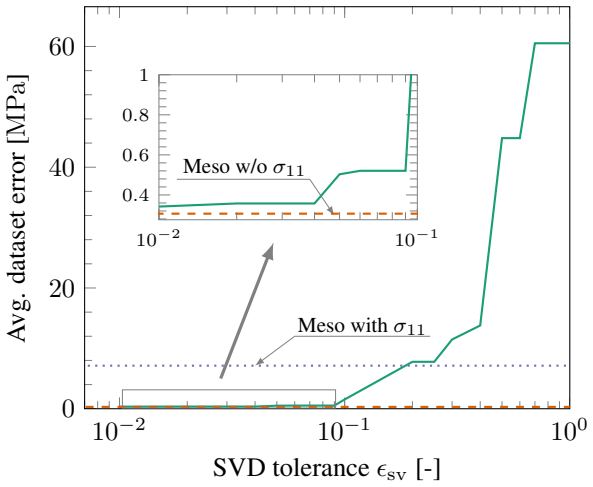

Figure 4: Average absolute errors of the hyper-reduced model for the pure stress dataset.

Using the surrogate models to reproduce stresses at the same strain values used for training, an average error over the complete dataset comparing the training targets $\boldsymbol{\sigma}$ with the surrogate responses $\widehat{\boldsymbol{\sigma}}$ can be defined:

$$\mathcal{E} = \frac{1}{n_\varepsilon} \sum_{i}^{n_\varepsilon} \left(\frac{1}{n_i^{\text{t}}} \sum_{j}^{n_i^{\text{t}}} \left| \sigma_i\left(t_j\right) - \widehat{\sigma}_i\left(t_j\right) \right| \right) \tag{54}$$

with n_i^t being the number of load steps comprising the stress-strain curve associated with each strain component i. Errors are computed for different values of ϵ_{sv}, with results being shown in Fig. 4. As with Fig. 3, the error starts at a high value when only elastic modes are used and decreases to values as low as $0.4\,\text{MPa}$ for $\epsilon_{\text{sv}} = 0.01$. Fig. 4 also includes the average error of predictions made with the mesoscopic model of Section 3. Since that model explicitly ensures no plasticity occurs in the fiber direction while the actual microscopic response in that direction is slightly nonlinear, the average absolute error over the dataset appears to be high[4] even though all the other directions are very well captured. For this reason, Fig. 4 shows two accuracy levels for the mesomodel, with and without including σ_{11}.

Since controlling the tolerance only influences the number of modes n indirectly, the error tends to decrease in discrete steps. This can also be observed in Fig. 5, which shows how the number of modes n and integration points m increases as ϵ_{sv} is reduced. Since the reduction in the number of integration points is made possible by the POD reduction, maintaining a low ECM integration error for higher values of n requires a larger set of cubature points. In any case, the reduction remains relatively efficient even for the lowest ϵ_{sv} considered here — with compression factors $N/n \approx 1284$ and $M/m \approx 65$.

The same dataset is used to train neural networks with a number of hidden units n_1 ranging from 10 to 1000. In order to track the training process, the evolution of the average absolute error over the validation set (20 % of the complete dataset) is plotted in Fig. 6. The monotonic error decrease observed for all curves suggests that no overfitting to the data is occurring. Increasing the size of the hidden layer improves the obtained predictions but with diminishing returns for n_1 larger than 100. Indeed, doubling the size of the hidden layer from 500 to 1000 leads to a negligible decrease in the error.

The same trend can be observed in Fig. 7, where *online* predictions are computed from a one-element model loaded in the 2-direction (transverse direction). Although accurate predictions of the perfect plasticity plateau can be obtained by using sufficiently large networks, both the initial stiffness and the response leading up to the plasticity plateau are still slightly inaccurate even for $n_1 = 1000$. The important observation to be made here is

[4]Due to the stiffness gradient between fiber and matrix, σ_{11} is the stress component with the highest order of magnitude. Even small relative differences in this direction lead to high absolute errors.

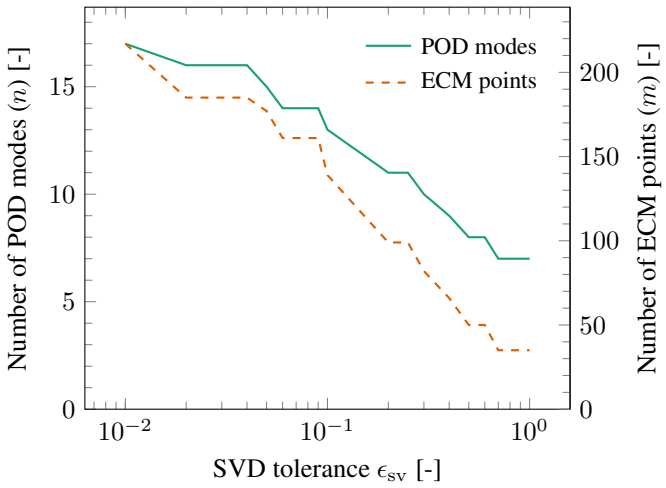

Figure 5: Number of modes and integration points of the hyper-reduced model for different tolerances ϵ_{sv} ($N = 21828$, $M = 14176$).

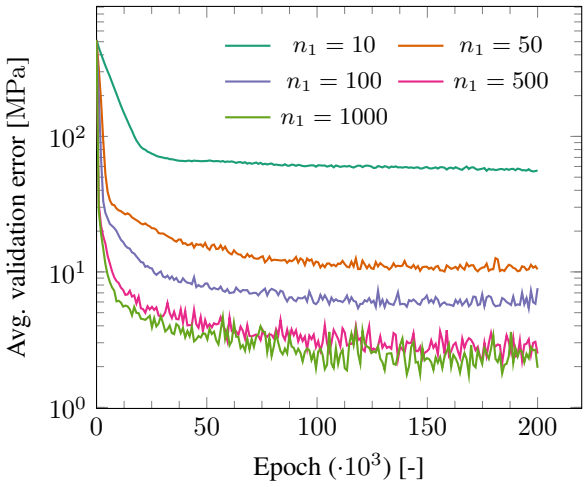

Figure 6: Evolution of the average validation error during training of networks with different hidden layer widths (n_1).

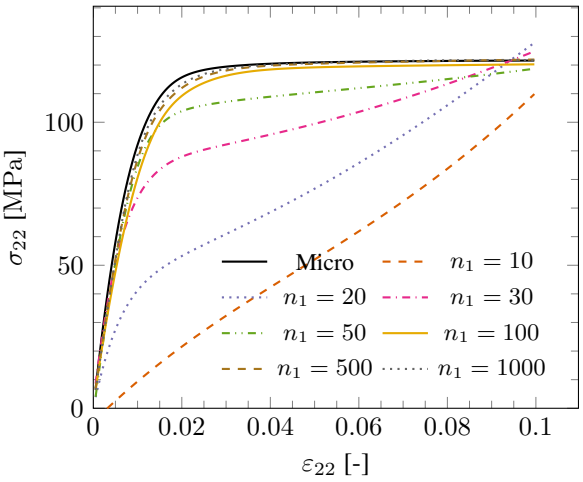

Figure 7: Predictions of transverse stress made by neural network models with different hidden layer sizes (n_1).

that even though neural networks are regarded as universal function approximators, the regularization brought by the dropout layer has the adverse effect of making an exact fit with the training data very difficult to achieve.

The average absolute error for the complete dataset obtained with networks of different sizes is plotted in Fig. 8. Although showing a similar trend as Fig. 6, two important differences between the errors in these two cases should be noted. Firstly, errors in Fig. 8 take into account the whole dataset, while Fig. 6 only shows errors computed for samples in the validation set. Secondly, while errors in Fig. 6 are computed by feeding the network with the exact strain vectors coming from micromodels, Fig. 8 is obtained by using the trained network *online* in a one-element model that includes numerical noise intrinsic to the Newton-Raphson procedure used to solve it.

The presence of numerical noise combined with the fact that data-driven models lack any sort of physical constraint to their behavior can lead to substantial error accumulation as the analysis progresses: wrong stress predictions lead to wrong solutions for the displacements which in turn become wrong strains to be fed to the network. After a few time steps, the network will be operating well outside of its training space and making nonsensical predictions.

In order to demonstrate how the inclusion of a dropout layer increases model robustness against noise, two networks — one of size $n_1 = 500$ with dropout and the other of size $n_1 = 100$ without dropout[5] — are used to predict the response of a model loaded in transverse tension (2-direction) with and without the inclusion of small perturbations to all three shear components, $\epsilon_{12} = -\epsilon_{13} = \epsilon_{23} = 0.01\epsilon_{22}$. Results are shown in Fig. 9. While the regularized response remains unchanged after the introduction of noise, the unregularized model branches off into an unphysical softening regime. Note how the unregularized model actually gives better predictions than the regularized one before it starts to lose precision: training a robust and accurate model entails finding a balance between the bias introduced by regularization and the variance introduced by allowing the model to become overly complex (this is also known as the *bias-variance tradeoff*).

Before moving on to more complex stress states, an interesting conclusion can be drawn by letting the reduced models make predictions on a strain range beyond the one used during training. Fig. 10 shows the straightforward case of tension in the fiber direction (σ_{11}). The training snapshots teach the models how the stress response should behave for strains in the range $[0, 0.1]$, but in the range $(0.1, 0.2]$ the models must rely on their extrapolation capabilities. Owing to its stronger physical foundation, the hyper-reduced model correctly predicts a nearly linear stress response, while the network deviates from linearity after only a few time steps and transitions to an unphysical perfectly-plastic response. For hyper-reduced models, it is enough to stop training after the material response stabilizes. For neural networks the requirement is slightly stronger, as the complete

[5] Unregularized networks need less parameters to fit the training data to any given level of precision when compared to regularized ones. The size of the unregularized network is chosen by gradually increasing n_1 until a validation error lower than $1\,\mathrm{MPa}$ is obtained.

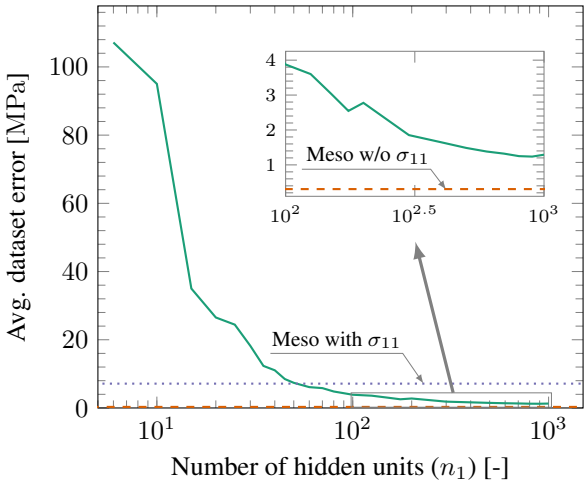

Figure 8: Average absolute errors over the entire pure stress dataset for network models with different hidden layer sizes (n_1).

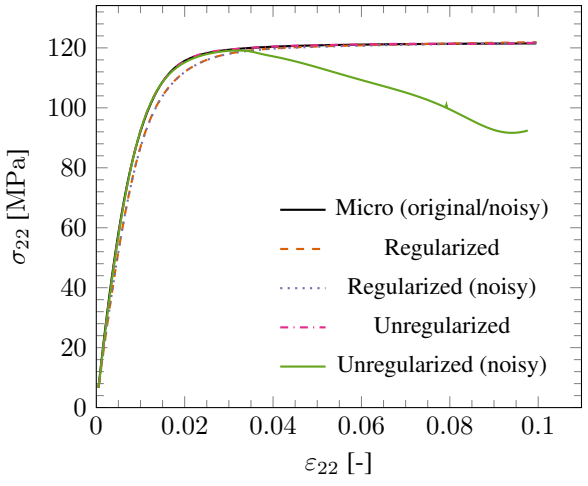

Figure 9: Erroneous predictions by an unregularized neural network when making predictions on noisy strain values. The robustness introduced by the dropout layer alleviates the issue.

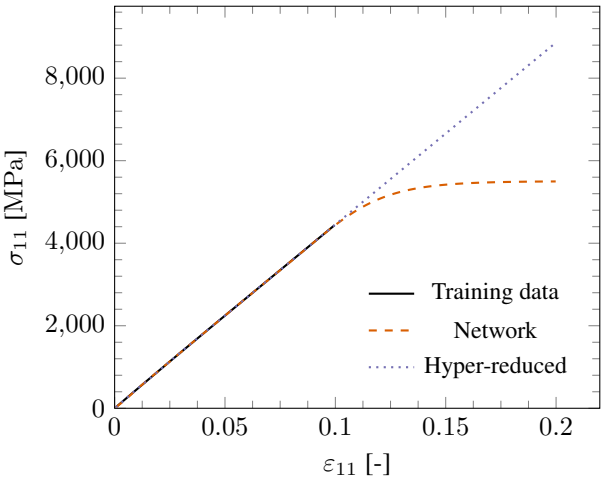

Figure 10: Surrogate models used to predict material behavior outside of the strain range seen during training. The hyper-reduced model predicts the correct response, while the network shows an unphysical perfectly-plastic behavior.

strain range to be encountered *online* should be seen by the model during training.

Finally, the impact on computational efficiency of increasing the size of the reduced models is investigated. Execution times are related to model size (number of POD modes n or size of the hidden neural layer n_1) in Fig. 11, where the smallest model of each type ($\epsilon_{\text{sv}} = 1.0$ or $n_1 = 10$) is used to normalize the curves. For the neural model, increasing the size of the model 100 times only leads to an execution time approximately twice as long (0.09 s), indicating that other operations related to the 1-element FE model (*e.g.* solving the 24-DoF equilibrium system) are more expensive than the very efficient neural network computations. For the hyper-reduced model, an increase of only 2.5 times on the number of POD modes leads to a 5 times longer computation (20.70 s). In any case, both models are still significantly faster than the full-order one (3167 s)

For linear materials, a simple linear combination of the pure stress states considered in this section would be enough to describe any combined stress state. Unfortunately, the material behavior being learned here is highly nonlinear and path dependent. In the next sections, the accuracy impact incurred by using pure stress combinations to approximate combined stress scenarios is investigated. Furthermore, the ability of surrogate models to incorporate new information coming from additional micromechanical simulations (*retraining*) is assessed.

6.2 Biaxial transverse tension

For the next set of examples, the trained models of Section 6.1 are used to predict material response under biaxial transverse tension loading (a combination of σ_{22} and σ_{33}). A common design practice when dealing with plasticity is to compute a yield stress envelope by plotting the final stress levels for different stress ratios. Fig. 12 shows an illustration of such an envelope, where the angle $\theta = \arctan\left(\frac{\sigma_{22}}{\sigma_{33}}\right)$ defines the stress ratio.

Recalling that models in Section 6.1 are trained on pure stress states for all stress components, they are already capable of predicting both the lower ($\theta = 0°$) and upper ($\theta = 90°$) bounds of the tension-tension envelope of Fig. 12. In order to investigate the accuracy of the models upon extrapolation from the training set, they are used to predict the response for $\theta = 45°$. The models are also retrained by including extra training cases that gradually approach the center of the envelope from both sides — with the limit of the new training sets being represented by the angle θ_{lim} (Fig. 12) — and used to predict $\theta = 45°$. For these new trainings, $\epsilon_{\text{sv}} = 0.01$ is adopted and the size of the hidden neural layer is fixed at $n_1 = 500$. Error levels over the training set similar to the ones in Figs. 4 and 8 are obtained for the retrained models.

Fig. 13 shows ε_{33}-σ_{33} curves obtained with hyper-reduced models. The obtained responses are very accurate

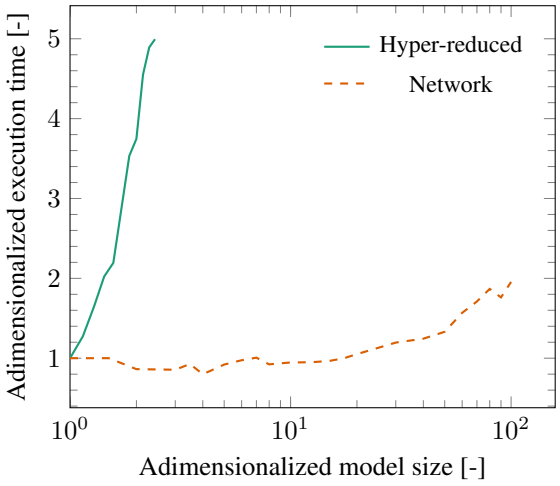

Figure 11: Increases in execution time when model size (n for hyper-reduced models and n_1 for network models) is increased.

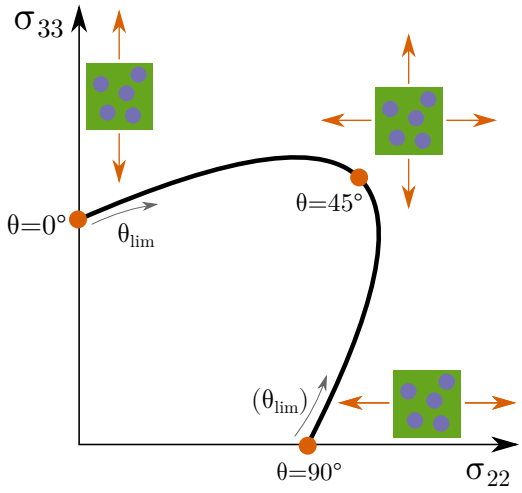

Figure 12: Illustration of a biaxial yield envelope. The angle θ defines the ratio between the two stress components. When training surrogates, θ_{lim} is used to define the bounds of the training space.

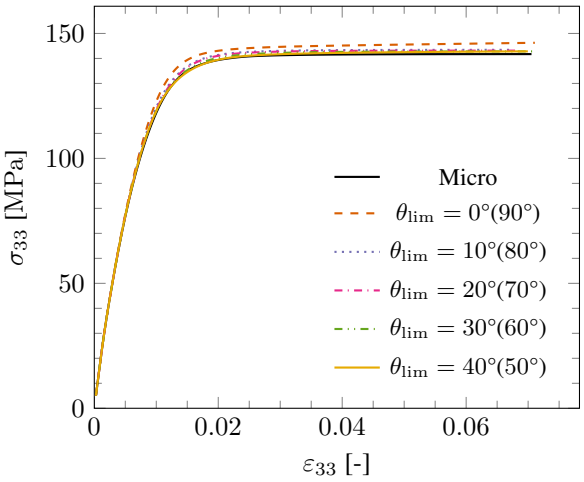

Figure 13: Hyper-reduced model predictions of the biaxial transverse tension response when $\theta = 45°$. Curves from models trained only on pure stress states ($\theta_{\text{lim}} = 0$) as well as models retrained with additional biaxial cases ($\theta_{\text{lim}} > 0$) are shown.

even with no additional retraining ($\theta_{\text{lim}} = 0°$). This is an interesting feature of the projection-based reduction: an accurate response at $\theta = 45°$ hinges on correctly accounting for pressure-dependent yielding, which the POD model does in an approximate way by using information obtained from pure compression snapshots. A similar level of accuracy is obtained for σ_{22}.

The network model does not perform as well. With no additional retraining, the stress stabilizes at a value approximately $50\,\%$ lower than the reference one. Adding training cases closer to the one being predicted brings the response closer to the target, but even with training points at $\theta = 40°$ and $\theta = 50°$ the maximum stress is still approximately $10\,\text{MPa}$ off. On the other hand, the regularization applied to the network does ensure a stable response with physically-sound shape (linear, plastic hardening and perfect plasticity) even upon significant extrapolation from the training set.

Although the robustness of the network model is an advantageous feature when working with nonlinear solvers at the mesoscale, the model outputs the expected curve shape even when the actual stress values are far from being correct and therefore does not provide any clue that it is operating outside of its training space. Ideally, the analyst should be provided not only with a prediction but also with a measure of how much confidence the model has in giving it.

The next example explores the *bootstrap* strategy, a popular approach for estimating uncertainty in neural networks [42]. Instead of relying on the prediction of a single[6] network, 50 different networks are trained with all pure stress cases and one extra case with $\theta = 45°$ and used to predict the complete envelope. Each network has different initial weights and different training sets obtained through a *bagging* process [43]: from the complete bag of 3500 stress-strain pairs, samples are randomly drawn, included in the training set and placed back in the bag until the training set has 3500 pairs. This process leads to sets that see approximately $63.2\,\%$ of the original sample pool, with some pairs appearing more than once. The samples that remain unseen are used as a validation set.

Fig. 15 shows the envelopes predicted by each of the 50 networks as well as the average prediction. Following [7], the stresses that define the envelope are computed at a strain level of $\sqrt{\varepsilon_{22}^2 + \varepsilon_{33}^2} = 0.04$. Close to trained points ($0°$, $45°$ and $90°$), predictions from all networks are close to the average one, indicating a high level of confidence in the prediction. Moving away from the trained points, the level of disagreement between networks gradually increases, indicating that predictions in those ranges of θ should be used with care. Natu-

[6]Technically, a network with dropout can be seen as a combination of 2^{n_1} slightly different networks sharing the same parameters, this being the total number of possible dropout combinations [37]. However, since dropout is only applied during training, the average behavior of this network ensemble is accessible *online* but its variance is not.

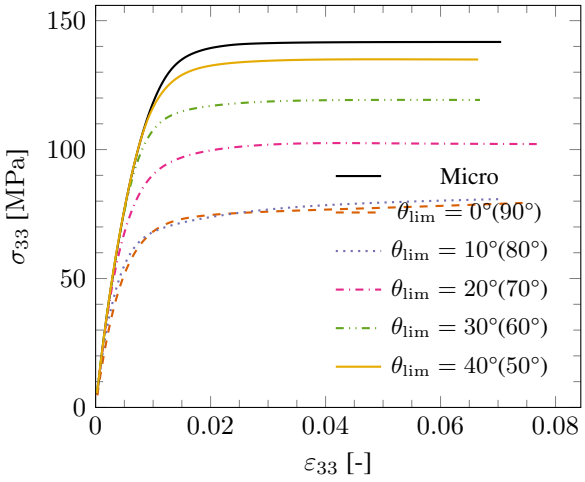

Figure 14: Network model predictions of the biaxial transverse tension response when $\theta = 45°$. Curves from models trained only on pure stress states ($\theta_{\text{lim}} = 0$) as well as models retrained with additional biaxial cases ($\theta_{\text{lim}} > 0$) are shown.

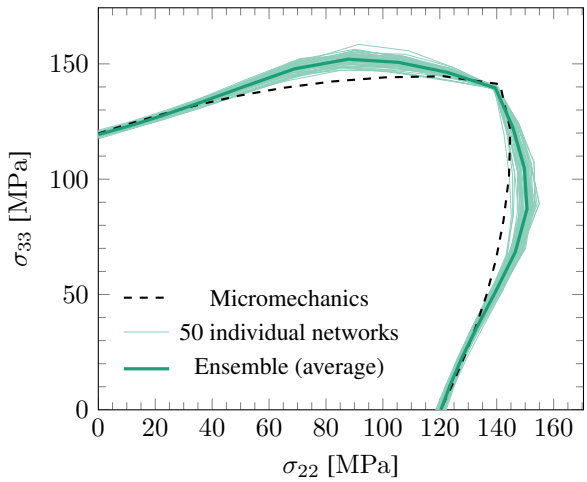

Figure 15: Biaxial yield envelopes obtained by 50 different bootstrapped networks trained with pure stress states plus the biaxial case $\theta = 45°$.

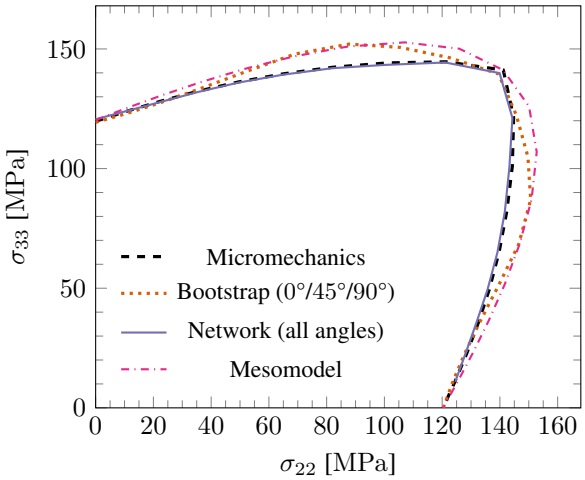

Figure 16: Biaxial yield envelopes obtained with the bootstrapped network ensemble trained on pure stress cases plus $\theta = 45°$ and with a single network trained with all values of θ. The mesomodel envelope is shown for comparison.

rally, this additional piece of information comes at the cost of computing 50 network responses instead of one, but more efficient techniques such as Bayesian neural networks can also be used to derive network responses with uncertainty intervals [42].

Plotting the ensemble response together with predictions obtained with the mesomodel of Section 3 in Fig. 16, it can be seen that both give predictions with roughly the same level of accuracy, with errors of up to 10 MPa. The advantage of the network model over the mesomodel lies in the possibility of retraining. Fig. 16 also shows the prediction of a single network trained with all values of θ used to construct the envelope. Even though this network is now trained on two complete datasets (pure stress states and biaxial transverse tension), the size $n_1 = 500$ of the network is kept unchanged. Nevertheless, the same level of accuracy shown in Fig. 8 is achieved.

Finally, an analogous study is performed with the hyper-reduced model. The response of models trained with pure stress cases plus a single biaxial case ($\theta = 45°$) and with all envelope points are shown in Fig. 17. With only a single biaxial training point, the hyper-reduced model already outperforms the mesomodel. Expanding the training set leads to an almost perfect agreement with the full-order model, but a price is paid in terms of efficiency: the model including all stress ratios has a reduced space of size $n = 30$ and $m = 714$ cubature points (compare with $n = 18$ and $m = 241$ for the model trained with only 0°, 45° and 90°). In practice and depending on the application, it might be more advantageous to accept a relatively small loss of accuracy in order to keep the surrogate model efficient.

6.3 Longitudinal shear and transverse tension

The next set of examples considers the combination of longitudinal shear (σ_{12}) and transverse tension (σ_{22} or σ_{33}). This is a loading scenario commonly encountered by laminated composites in service. It is therefore an important stress combination to consider when training surrogate models. Here, the relevant stress ratio is $\theta = \arctan\left(\frac{\sigma_{12}}{\sigma_{\mathrm{tt}}}\right)$, where σ_{tt} can be either σ_{22} of σ_{33}. Changing the direction of this transverse stress leads to different micromodel responses, a distinction that is lost in the invariant-based mesomodel.

First, models are trained with a combination of pure stress states and a number of extra cases defined by the limit stress ratio $\theta_{\mathrm{lim}} \in [0°, 90°]$ (analogous to Fig. 12) and used to predict the response of $\theta = 45°$. For this first part, $\sigma_{\mathrm{tt}} = \sigma_{22}$. Fig. 18 shows results for hyper-reduced models. For this load combination, information gathered from only pure stress cases ($\theta_{\mathrm{lim}} = 0$) is not enough to properly reproduce the response at $\theta = 45°$, with a relative error of 13 % for the maximum stress level. Adding extra training cases quickly reduces the

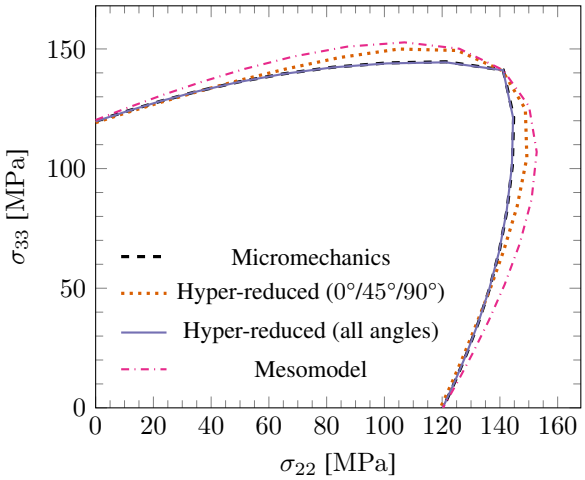

Figure 17: Biaxial yield envelopes obtained with a hyper-reduced model trained on pure stress cases plus $\theta = 45°$ and with another one trained with all values of θ. The mesomodel envelope is shown for comparison.

error, as expected. Although not shown in Fig. 18, a similar accuracy level is obtained for σ_{22}. Interestingly, predictions by the network model for this load combination are significantly better than the ones obtained for biaxial transverse tension. With the addition of relatively few extra training cases (from $\theta_{\text{lim}} = 30°$), the network converges to the micromodel solution, as can be seen in Fig. 19.

For the next test, the network and hyper-reduced model of Figs. 18 and 19 trained with $\theta_{\text{lim}} = 40°$ and $\sigma_{\text{tt}} = \sigma_{22}$ are used to predict the curve with $\theta = 45°$ but this time with $\sigma_{\text{tt}} = \sigma_{33}$. The obtained results can be seen in Fig. 20. None of the surrogates is able to correctly predict the shear response when the direction of the transverse stress is shifted. The hyper-reduced model is the one with the lowest error, being able to correctly predict the response up to the perfect plasticity regime and overshooting the maximum stress by about $5\,\%$. Interestingly, the mesomodel is the one with the largest discrepancy. Since the model is invariant-based, no distinction is made between σ_{22} and σ_{33} when combining them with τ_{12}, leading to excellent agreement for the σ_{22}-τ_{12} combination but not for σ_{33}-τ_{12}.

Fig. 20 illustrates the high level of complexity of the parameter space being treated here and raises the issue of how to best sample this parameter space in order to ensure accuracy under general stress states. For the mesomodel, sampling is a simple task that consists of a small pre-defined amount of micromechanical experiments (Section 3). But the underlying assumptions that allow for such a simple calibration process lead to highly inaccurate predictions for this specific loading scenario which is still a relatively simple one. The biggest drawback of the mesomodel is that there is no straightforward way to substitute these prior assumptions by posterior knowledge coming from additional micromodel simulations.

For hyper-reduction and neural networks, the problem is the opposite: these models can readily incorporate new epistemic information but must contend with sampling a potentially infinite parameter space. Although the question of sampling is much simplified here by focusing on monotonic loading along a number of load paths defined *a priori*, it is an open issue that should be addressed in tandem with the development of new surrogate modeling techniques [25, 31].

Models trained with pure stress cases plus two combined stress cases — $\theta = 45°$ for $\sigma_{\text{tt}} = \sigma_{22}$ and $\sigma_{\text{tt}} = \sigma_{33}$ — are used to predict the complete stress envelopes for σ_{22}-τ_{12} and σ_{33}-τ_{12}. The bootstrap strategy is once again employed in order to obtain the average and variance of a combination of 50 different network models. Results are shown in Fig. 21, with each envelope point corresponding to predictions at a strain level $\sqrt{\varepsilon_{\text{tt}}^2 + \gamma_{12}^2} = 0.04$.

It is interesting to note that the network ensemble gives more accurate and more confident predictions for the region of the envelope dominated by shear than for the one dominated by transverse stresses. The average response is compared with the one obtained from a single network trained on the complete dataset as well as

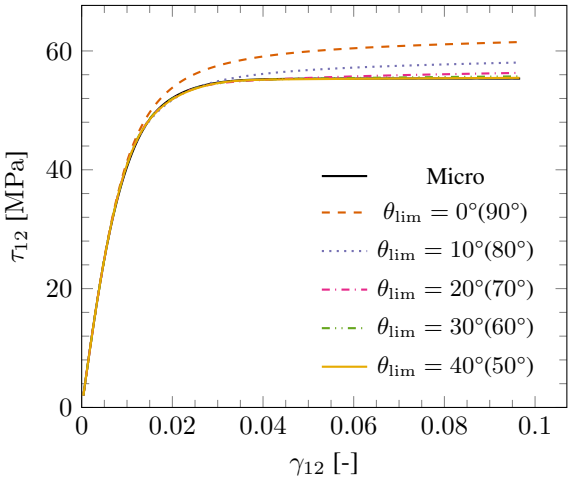

Figure 18: Hyper-reduced model predictions of the biaxial σ_{22}-τ_{12} response when $\theta = 45°$. Curves from models trained only on pure stress cases ($\theta_{\text{lim}} = 0$) as well as models retrained with additional biaxial cases ($\theta_{\text{lim}} > 0$) are shown.

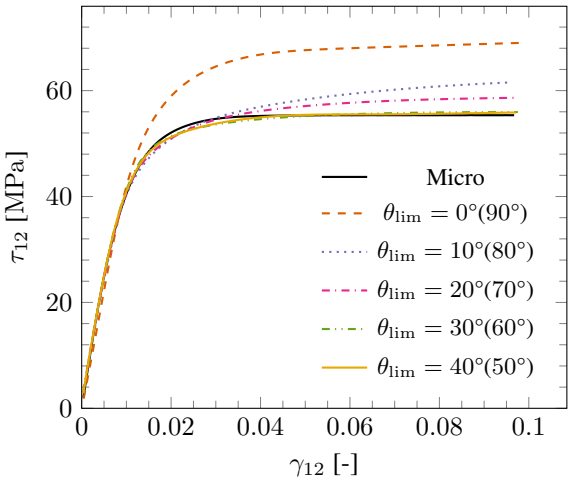

Figure 19: Network model predictions of the biaxial σ_{22}-τ_{12} response when $\theta = 45°$. Curves from models trained only on pure stress cases ($\theta_{\text{lim}} = 0$) as well as models retrained with additional biaxial cases ($\theta_{\text{lim}} > 0$) are shown.

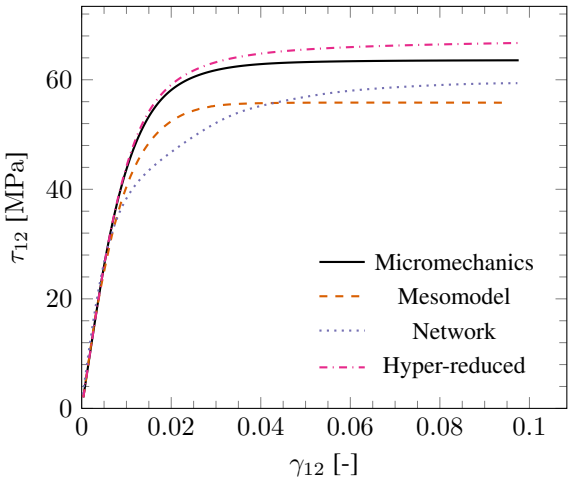

Figure 20: Surrogate model predictions for $\sigma_{\mathrm{tt}} = \sigma_{33}$ after being trained with $\sigma_{\mathrm{tt}} = \sigma_{22}$ ($\theta_{\mathrm{lim}} = 40°$). The curves show the predicted responses for $\theta = 45°$.

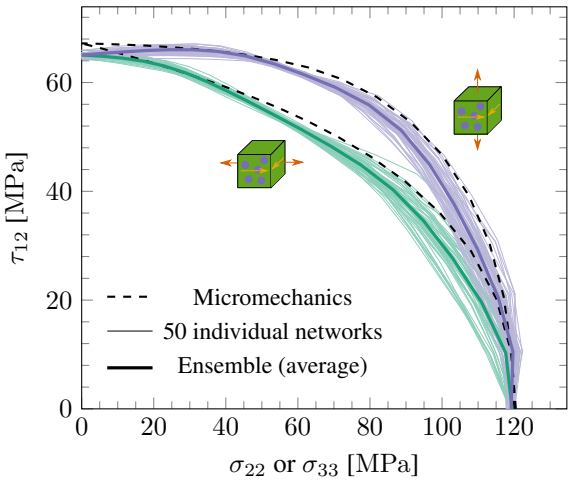

Figure 21: Biaxial yield envelopes for the $\sigma_{22(33)}$-τ_{12} combination obtained by 50 bootstrapped networks trained with pure stress cases plus the biaxial case $\theta = 45°$.

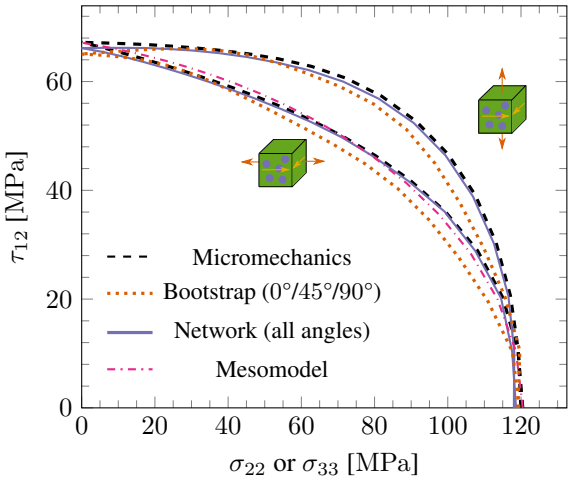

Figure 22: Biaxial yield envelopes ($\sigma_{22(33)}$-τ_{12}) obtained with the bootstrapped network ensemble trained on pure stress states plus $\theta = 45°$ and with a single network trained with all values of θ. The mesomodel envelope is shown for comparison.

with mesomodel predictions in Fig. 22. As in Section 6.2, adding extra training cases improves predictions. Once again the same model size n_1 used for pure stress cases is enough to learn the larger dataset considered here without loss of accuracy.

For the hyper-reduced model, envelopes obtained with $\theta = 0°/45°/90°$ and with all angles are shown in Fig. 23. The partially-trained model already gives excellent predictions for σ_{22}-τ_{12} but fail to reproduce part of the σ_{33}-τ_{12} envelope. Regarding model size, the one trained with only pure stress cases has $n = 17$ and $m = 217$. Adding the biaxial case for $\theta = 45°$ leads to a model with $n = 19$ and $m = 317$. Finally, adding the remaining angles results in $n = 23$ and $m = 509$. In both Figs. 22 and 23, note that the mesomodel is only capable of capturing the σ_{22}-τ_{12} envelope.

6.4 Axial stress and longitudinal shear

One last stress combination is briefly examined, namely longitudinal shear (τ_{12}) with tension in the fiber direction (σ_{11}). For high σ_{11}/τ_{12} ratios, the longitudinal shear response is heavily affected by the presence of plastic strains in the fiber direction. Since the mesomodel of Section 3 explicitly eliminates the possibility of plasticity developing under axial loading, its effect on the shear behavior is not captured. Van der Meer [7] points to this as being a major weakness of Vogler's mesomodel, so it is interesting to investigate how well the other surrogate strategies can handle this scenario.

The hyper-reduced model trained only on pure stress cases is used to predict shear response for a set of ratios $\sigma_{11}/\tau_{12} \in [57, 29, 11, 6, 0]$. Results are shown in Fig. 24. Without any additional training, the hyper-reduced model reproduces the curves for all ratios remarkably well. On the other hand, a network without additional retraining gives poor predictions (Fig. 25). This example illustrates the advantage of reduction methods that, although constrained to a reduced solution manifold, are still driven by the original constitutive laws of the full-order micromodel (see [44] for an interesting alternative involving neural networks infused with actual constitutive laws).

The neural network is retrained by including every curve in Fig. 25 in addition to the pure stress curves. The resultant curves are shown in Fig. 26. Although providing better predictions, the retrained network is still not able to accurately capture the response leading up to the perfect plasticity plateau. This is consistent with the observed, for instance, in Fig. 7 and seems to be a side effect introduced when regularizing the network.

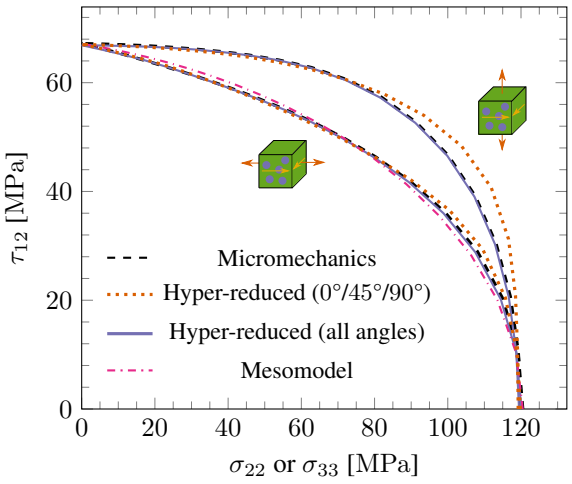

Figure 23: Biaxial yield envelopes ($\sigma_{22(33)}$-τ_{12}) obtained with a hyper-reduced model trained on pure stress cases plus $\theta = 45°$ and with one trained with all values of θ. The mesomodel envelope is shown for comparison.

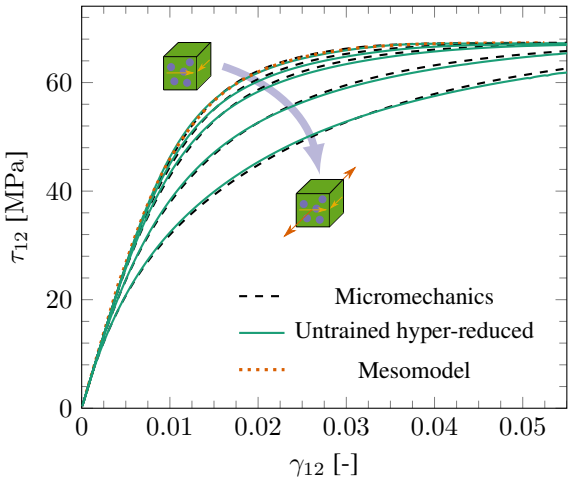

Figure 24: Hyper-reduced model predictions for the biaxial σ_{11}-τ_{12} response under various stress ratios. The model trained with only pure stress cases predicts these unseen scenarios remarkably well.

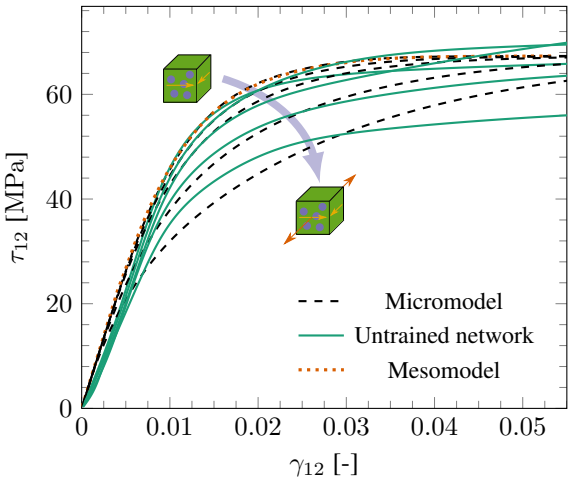

Figure 25: Network model predictions for the biaxial σ_{11}-τ_{12} response under various stress ratios. The curves are not reproduced well without additional network retraining.

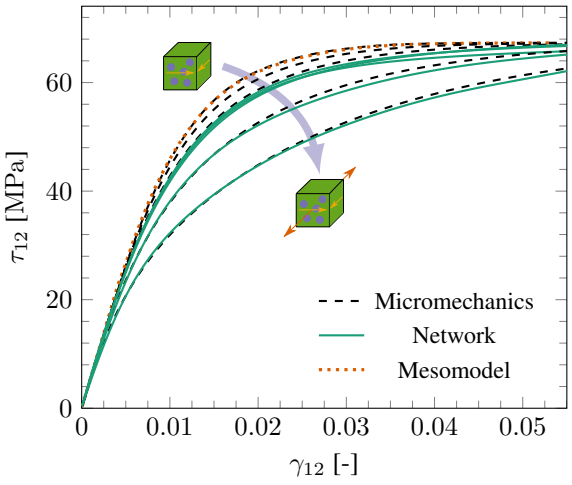

Figure 26: Biaxial σ_{11}-τ_{12} predictions for the retrained network model. The predictions improve but are still not as accurate as the ones obtained with the untrained hyper-reduced model.

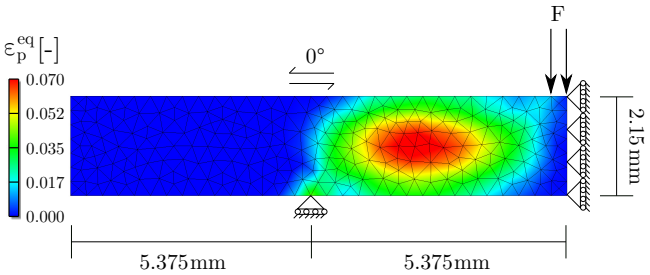

Figure 27: Short-beam FE^2 example. Loads and boundary conditions are shown as well as the plastic strain field at the final time step.

6.5 FE^2 example

As one final illustrative example, the surrogate models are used to simulate the interlaminar shear test shown in Fig. 27. The model consists of a short beam composed of unidirectional composite layers with fibers aligned in the $0°$ direction shown in Fig. 27. Symmetry is exploited by modeling only half of the span of the beam and the problem is simplified by modeling the beam in 2D with a plane strain assumption. The model is discretized with 484 constant-strain triangles each with a single integration point. For models requiring an embedded RVE (full-order FE^2 and hyper-reduced), the same 3D micromodel used for training the surrogate models is adopted and only in-plane stress and stiffness components are upscaled. Due to the short span between supports, strain localizes at mid-thickness (Fig. 27) in a region dominated by longitudinal shear (τ_{12}).

The models of Section 6.1, trained only with pure stress cases, are used as surrogates ($\epsilon_{\text{sv}} = 0.01$, $n_1 = 500$). The full-order FE^2 problem is also solved as reference. This is a challenging scenario for the surrogates since the model experiences a complex combination of longitudinal shear, fiber stress and transverse tension and compression close to the load and support. Furthermore, the plane strain assumption at the macroscale leads to stress combinations not covered during training under pure stress states. The analysis is executed for 118 time steps, after which global convergence cannot be obtained for the full-order FE^2 model. None of the surrogates show this lack of robustness, but for the sake of comparison with the full model they are also stopped after 118 time steps.

The resultant load-displacement curves are shown in Fig. 28. Despite operating under a complex scenario not covered during training, all surrogates predict the response well. The network model is the one showing the highest discrepancy, with predictions for the load factor approximately $5\,\%$ lower than the reference ones. This lack of precision during the hardening regime is consistent with previous observations (*c.f.* Figs. 7 and 26).

Execution times and speedups are shown in Table 1. Even with a coarse mesoscopic mesh with only 484 embedded micromodels, the full-order model takes more than one week to run. Without additional techniques such as parallelization or the construction of surrogates, FE^2 is effectively unsuitable for any practical application. Among the surrogate models, the mesomodel is the most efficient, followed by the neural network and the more expensive hyper-reduced model.

	Full	Mesomodel	Network	Hyper-reduced
Runtime [s]	726 500	2.2	10.8	2692
Speedup [-]	N/A	329 478	67 393	270

Table 1: Execution times and speedups for the FE^2 examples. Full-order values are used as reference.

There is, however, no clear-cut recommendation to be made as to which strategy should be chosen. The mesomodel is fast and robust but fails in predicting relevant loading combinations. The neural network is fast, can be retrained to incorporate new information and its efficiency scales well with model size, but it has poor extrapolation capabilities and grapples with the *bias-variance tradeoff*. The hyper-reduced model

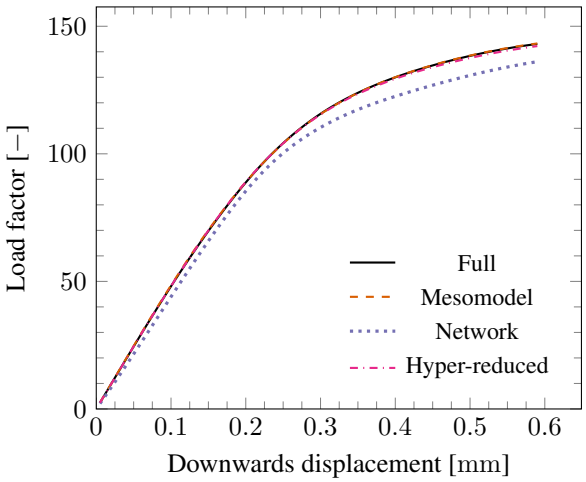

Figure 28: Load-displacement curves for the short beam FE^2 example. Predictions made with all three surrogate modeling strategies are shown. The analysis is stopped before the perfect plasticity plateau due to non-convergence of the full-order FE^2 model.

retains relevant physical information, extrapolates well to unseen data and readily handles unloading and path-dependency, but is inherently slower than the other options and scales poorly with the size of its latent space.

7 Conclusions

Three different approaches for constructing surrogate models for multiscale analysis of laminated composites have been compared through an extensive series of numerical tests. Comparisons involved a state-of-the-art orthotropic mesoscale model with pressure-dependent plasticity, feed-forward neural networks with dropout regularization and hyper-reduced models combining the POD and ECM techniques. Even though substantial computational efficiency gains could be obtained with all of the approaches, each comes with a particular set of advantages and drawbacks:

- *Vogler's plasticity mesomodel* is the fastest among the strategies and enjoys a robust physical foundation. Simplifying assumptions adopted in its formulation allow for a simple calibration procedure and a reduced number of model parameters. But these same assumptions lead to poor predictions for a number of realistic loading scenarios (Figs. 23 and 24). Once formulated, it is not possible to easily include in the model new epistemic information gained from running additional micromechanical models.

- *Neural networks* are fast, can be trained to reproduce general stress states and can be retrained to incorporate additional data (*c.f.* Figs. 25 and 26). However, their extrapolation capabilities are limited, which makes using them away from their training sets risky (Fig. 14). Furthermore, unregularized networks can lead to high errors and nonsensical predictions by feeding on their own inaccuracy (Figs. 9 and 10). Finally, conventional feed-forward networks assume a unique relationship between stresses and strains and therefore cannot handle unloading or strain path dependency.

- *Hyper-reduction* tends to give better predictions with a lower training effort by retaining physical information from the original full-order model. Hyper-reduced models tend to generalize well to unseen data, albeit with varying degrees of success (*c.f.* Figs. 13, 18 and 24), and can be retrained on new observations (Fig. 23). On the other hand, they are significantly slower than the other surrogates and their efficiency does not scale well as more precision is sought or as more training cases are added (Fig. 11).

Although none of the techniques were found to be optimally efficient and accurate in every situation, they could be employed in combination in order to leverage their strengths and minimize their weaknesses. For

instance, for a given mesoscopic structure to be modeled, one could first use the mesomodel to quickly solve the problem, gather a number of representative strain histories from multiple integration points and inject those in a single micromodel in order to generate highly-tailored training data for hyper-reduced models or neural networks. This can be used to efficiently solve the issue of sampling over an extremely large space of possible strain combinations without having to run full-order FE^2 models. Alternatively, an adaptive approach could be used to switch between surrogates: an ensemble of neural networks could be used to compute the response at all points but predictions with low confidence would be substituted by those coming from a hyper-reduced micromodel. In any case, the present in-depth investigation on the advantages and limitations of each technique may serve as a valuable starting point for building smarter multiscale analysis frameworks for laminated composites.

Acknowledgements

The authors gratefully acknowledge financial support from the Netherlands Organization for Scientific Research (NWO) under Vidi grant nr. 16464.

Data availability

The data used to generate the graphs in this article is available at the 4TU.ResearchData repository through http://doi.org/10.4121/uuid:45d11acd-3906-4474-ace4-e1073b45b8d2

References

References

[1] M. Vogler, R. Rolfer, and P. P. Camanho. Modeling the inelastic deformation and fracture of polymer composites - Part I: plasticity model. *Mech Mater*, 59:50–64, 2013.

[2] S. Ciutacu, P. Budrugeac, and I. Niculae. Accelerated thermal aging of glass-reinforced epoxy resin under oxygen pressure. *Polym Degrad Stabil*, 31:365–372, 1991.

[3] S. A. Grammatikos, M. Evernden, J. Mitchels, B. Zafari, J. T. Mottram, and G. C. Papanicolaou. On the response to hygrothermal ageing of pultruded FRPs used in the civil engineering sector. *Mater Des*, 96:283–295, 2016.

[4] C. Qian, T. Westphal, and R. P. L. Nijssen. Micro-mechanical fatigue modelling of unidirectional glass fibre reinforced polymer composites. *Comput Mater Sci*, 69:62–72, 2013.

[5] F. Naya, C. González, C. S. Lopes, S. van der Veen, and F. Pons. Computational micromechanics of the transverse and shear behaviour of unidirectional fiber reinforced polymers including environmental effects. *Compos Part A-Appl S*, 92:146–157, 2016.

[6] A. R. Melro, P. P. Camanho, F. M. Andrade Pires, and S. T. Pinho. Micromechanical analysis of polymer composites reinforced by unidirectional fibres: Part I - Constitutive modelling. *Int J Solids Struct*, 50:1897–1905, 2013.

[7] F. P. van der Meer. Micromechanical validation of a mesomodel for plasticity in composites. *Eur J Mech A-Solid*, 60:58–69, 2016.

[8] A. Gagani, Y. Fan, A. H. Mulian, and A. T. Echtermeyer. Micromechanical modeling of anisotropic water diffusion in glass fiber epoxy reinforced composites. *J Compos Mater*, pages 1–15, 2017.

[9] B. N. Cox and Q. D. Yang. In quest of virtual tests for structural composites. *Science*, 314(5802):1102–1107, 2006.

[10] F. P. van der Meer and L. J. Sluys. Continuum models for the analysis of progressive failure in composite laminates. *J Compos Mater*, 43(20):2131–2156, 2009.

[11] S. Pimenta, R. Gutkin, S. T. Pinho, and P. Robinson. A micromechanical model for kink-band formation: Part I - Experimental study and numerical modelling. *Compos Sci Technol*, 69(7-8):948–955, 2009.

[12] A. Krairi and I. Doghri. A thermodynamically-based constitutive model for thermoplastic polymers coupling viscoelasticity, viscoplasticity and ductile damage. *Int J Plas*, 60:163–181, 2014.

[13] X. Poulain, A. A. Benzerga, and R. K. Goldberg. Finite-strain elasto-viscoplastic behavior of an epoxy resin: Experiments and modeling in the glassy regime. *Int J Plas*, 62:138–161, 2014.

[14] G. Alfano and E. Sacco. Combining interface damage and friction in a cohesive-zone model. *Int J Numer Meth Eng*, 68:542–582, 2006.

[15] A. Turon, P. P. Camanho, J. Costa, and C. G. Dávila. A damage model for the simulation of delamination in advanced composites under variable-mode loading. *Mech Mater*, 38:1072–1089, 2006.

[16] M. G. D. Geers, V. G. Kouznetsova, and W. A. M. Brekelmans. Multi-scale computational homogenization: Trends and challenges. *J Comput Appl Math*, 234:2175–2182, 2010.

[17] C. Miehe, J. Schotte, and J. Schröder. Computational micro-macro transitions and overall moduli in the analysis of polycrystals at large strains. *Comput Mater Sci*, 16:372–382, 1999.

[18] V. Kouznetsova, W. A. M. Brekelmans, and F. P. T. Baaijens. An approach to micro-macro modeling of heterogeneous materials. *Comput Mech*, 27:37–48, 2001.

[19] I. B. C. M. Rocha, F. P. van der Meer, S. Raijmaekers, F. Lahuerta, R. P. L. Nijssen, L. P. Mikkelsen, and L. J. Sluys. A combined experimental/numerical investigation on hygrothermal aging of fiber-reinforced composites. *Eur J Mech A-Solid*, 73:407–419, 2019.

[20] P. Kerfriden, P. Gosselet, S. Adhikari, and S. P. A. Bordas. Bridging proper orthogonal decomposition methods and augmented newton-krylov algorithms: An adaptive model order reduction for highly nonlinear mechanical problems. *Comput Method Appl M*, 200:850–866, 2011.

[21] M. Chevreuil and A. Nouy. Model order reduction based on proper generalized decomposition for the propagation of uncertainties in structural dynamics. *Int J Numer Meth Eng*, 89:241–268, 2012.

[22] J. A. Hernández, M. A. Caicedo, and A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Comput Method Appl M*, 313:687–722, 2017.

[23] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J Sci Comput*, 32:2737–2764, 2010.

[24] R. A. van Tuijl, J. J. C. Remmers, and M. G. D. Geers. Integration efficiency for model reduction in micro-mechanical analyses. *Comput Mech*, pages 1–19, 2017.

[25] O. Goury, D. Amsallem, S. P. A. Bordas, W. K. Liu, and P. Kerfriden. Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: from dissipation-driven random selection to Bayesian optimization. *Comput Mech*, 58:213–234, 2016.

[26] F. Ghavamian, P. Tiso, and A. Simone. POD-DEIM model order reduction for strain-softening viscoplasticity. *Comput Method Appl M*, 317:458–479, 2017.

[27] M. Lefik, D. P. Boso, and B. A. Schrefler. Artificial neural networks in numerical modelling of composites. *Comput Method Appl M*, 198:1785–1804, 2009.

[28] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math Control Signals Syst*, 2:303–314, 1989.

[29] J. Ghaboussi, J. H. Garrett Jr, and X. Wu. Knowledge-based modeling of material behavior with neural networks. *J Eng Mech*, 117:132–153, 1991.

[30] J. Ghaboussi, D. A. Pecknold, M. Zhang, and R. M. Haj-ali. Autoprogressive training of neural network constitutive models. *Int J Numer Meth Eng*, 42:105–126, 1998.

[31] F. Ghavamian and A. Simone. Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network. *Comput Method Appl M*, 357:23p, 2019.

[32] B. A. Le, J. Yvonnet, and Q. -C He. Computational homogenization of nonlinear elastic materials using neural networks. *Int J Numer Meth Eng*, 104:1061–1084, 2015.

[33] X. Lu, D. G. Giovanis, J. Yvonnet, V. Papadopoulos, F. Detrez, and J. Bai. A data-driven computational homogenization method based on neural networks for the nonlinear anisotropic electrical response of graphene/polymer nanocomposites. *Comput Mech*, 64(2):307–321, 2018.

[34] V. P. Nguyen, O. Lloberas-Valls, M. Stroeven, and L. J. Sluys. Computational homogenization for multiscale crack modelling. implementation and computational aspects. *Int J Numer Meth Eng*, 89:192–226, 2012.

[35] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans Pattern Anal Mach Intell*, 35(8):17981828, 2013.

[36] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[37] N. Srivastava, G Hinton, A. Krizhevsky, A. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res*, 15:1929–1958, 2014.

[38] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv e-prints, arXiv:1412.6980*, 2014.

[39] I. B. C. M. Rocha, F. P. van der Meer, and L. J. Sluys. Efficient micromechanical analysis of fiber-reinforced composites subjected to cyclic loading through time homogenization and reduced-order modeling. *Comput Method Appl M*, 345:644–670, 2019.

[40] Jive - Software development kit for advanced numerical simulations. http://jive.dynaflow.com. Accessed: 08-11-2019.

[41] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In: Proceedings of AISTATS*, volume 9, pages 249–256, 2010.

[42] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya. Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Trans Neur Netw Learn Syst*, 22:1341–1356, 2011.

[43] L. Breiman. Bagging predictors. *Mach Learn*, 24:123–140, 1996.

[44] Z. Liu, C. T. Wu, and M. Koishi. A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials. *Comput Method Appl M*, 345:1138–1168, 2019.