Physically Recurrent Neural Networks for computational homogenization of composite materials with microscale debonding

Kovács, N.; Maia, M. A.; Rocha, I. B.C.M.; Furtado, C.; Camanho, P. P.; van der Meer, F. P.

DOI
10.1016/j.euromechsol.2025.105668

Publication date
2025

Document Version
Final published version

Published in
European Journal of Mechanics, A/Solids

European Journal of Mechanics / A Solids 112 (2025) 105668

Contents lists available at ScienceDirect

European Journal of Mechanics / A Solids

journal homepage: www.elsevier.com/locate/ejmsol

Full length article

Physically Recurrent Neural Networks for computational homogenization of composite materials with microscale debonding

N. Kovács [a,b,c,*], M.A. Maia [a], I.B.C.M. Rocha [a], C. Furtado [b,c], P.P. Camanho [b,c], F.P. van der Meer [a]

[a] Delft University of Technology, Department of Civil Engineering and Geosciences, P.O. Box 5048, 2600GA, Delft, The Netherlands
[b] DEMec, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal
[c] INEGI, Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial, Rua Dr. Roberto Frias 400, 4200-465, Porto, Portugal

ARTICLE INFO

Keywords:
Multiscale modeling
Machine learning
Composite materials

ABSTRACT

In this work, we extend a recent surrogate modeling approach, the Physically Recurrent Neural Network (PRNN), to include the effect of debonding at the fiber–matrix interface of composite materials. The core idea of the PRNN is to implement the exact material models from the micromodel into one of the layers of the network to capture path-dependent behavior implicitly. For the case of debonding, additional material points with a cohesive zone model are integrated within the network, along with the bulk points associated to the fibers and/or matrix. The limitations of the existing architecture are discussed and taken into account for the development of novel architectures that better represent the stress homogenization procedure. In the proposed layout, the history variables of cohesive points act as extra latent features that help determine the local strains of bulk points. Different architectures are evaluated starting with small training datasets. To maximize the predictive accuracy and extrapolation capabilities of the network, various configurations of bulk and cohesive points are explored, along with different training dataset types and sizes.

1. Introduction

Composite materials such as fiber-reinforced polymers (FRPs) can be tailored to reach optimal mechanical properties and enhance structural performance by design at different length scales. In practice, however, while modeling techniques have the potential to reduce the experimental testing costs and accelerate product development, the multi-scale nature and nonlinear behavior of these materials pose challenges for the reliable prediction of their global response. A suitable method to tackle such challenges is computational homogenization, or FE^2, a concurrent multiscale approach that offers a high-fidelity representation of the material behavior. Nevertheless, this flexibility and generality is followed by excessively high computational costs.

Several methods have been proposed to alleviate the computational bottleneck associated with FE^2. One alternative is to solve the equilibrium problem at the microscale, where a Representative Volume Element (RVE) is used to describe the material geometry and properties, in low-dimensional spaces. For instance with Proper Orthogonal Decomposition (POD) (Monteiro et al., 2008) or Proper Generalized Decomposition (PGD) (Halabi et al., 2013; Cremonesi et al., 2013). Although both techniques can effectively reduce computational time, obtaining a meaningful set of basis functions/modes is not trivial. With PGD, the construction of the basis functions based on the separation of the variables (*e.g.*, space, time, geometry, etc.) requires careful formulation and involves a large number of parameters as non-linearity increases. On the other hand, POD relies on a meaningful coverage of potential loading paths in the snapshots used for mode extraction, which can become a difficult task when dealing with path-dependent materials.

With higher potential to accelerate the microscopic problem, another popular approach is to replace the full-order microscale BVP with a *surrogate model*. In Ghavamian and Simone (2019) and Logarzo et al. (2021), the full-order micromodel is replaced by a recurrent neural network (RNN) at each integration point. The network is then used to predict the average stress from the micromodel, as well as history variables such as the equivalent plastic strain. A drawback of this method is related to the black-box nature of RNNs. Since the network maps the input strain to output stress values without being grounded on physical assumptions, extrapolation to unseen loading scenarios is rather limited. This issue is usually tackled by training with a large amount of data.

* Corresponding author at: Delft University of Technology, Department of Civil Engineering and Geosciences, P.O. Box 5048, 2600GA, Delft, The Netherlands.
E-mail address: n.kovacs-1@tudelft.nl (N. Kovács).

https://doi.org/10.1016/j.euromechsol.2025.105668
Received 20 December 2024; Received in revised form 29 March 2025; Accepted 4 April 2025
Available online 16 April 2025

A different approach was presented in a recent work using mechanics-informed machine learning (Li et al., 2023). In this approach, *a priori* knowledge of mechanics was implemented in the network to predict elastoplasticity for composites with pressure-independent yielding behavior by physics-based decomposition of stresses and strains. The network showed accurate predictions trained on a small dataset. A newly emerging method of combining the mechanical basis with the flexibility of neural networks is using physics-augmented neural networks (Rosenkranz et al., 2024; Klein et al., 2023; Schommartz et al., 2024). The essence of these networks is that they are tailored for a specific material model to fulfill mechanical conditions by construction. Although these networks provide accurate predictions, their complexity increases significantly with more complex material models.

Another alternative not limited to a specific type of material model is using physically recurrent neural networks (PRNNs) (Maia et al., 2023). In PRNNs, the constitutive relations in the full-order micromodel are directly implemented in the hidden layer of the network, bypassing the need for a complex network while giving physical interpretation to the hidden layer. In that work, the performance of the PRNN was first assessed in several loading cases at the microscale and compared to state-of-the-art RNNs. The PRNN was able to predict with the same accuracy as the RNN, but with much less training data, and notably, it was able to extrapolate to non-monotonic test data even though it only saw data from monotonic loading scenarios during training. Then, in a multiscale setting, the proposed network was used to replace the microsctructure with elastic and elastoplastic phases. It was observed that replacing the full-order micromodel with the network resulted in a reduction of computational time by more than 20,000 times.

A large part of recent literature on surrogate modeling focuses on predicting (hyper)elastic or elastoplastic behavior. Surrogate modeling for damage and fracture mechanics applications is a much less explored field. For example, Wang and Sun (2019) used deep reinforcement learning techniques to create traction–separation laws, but did not apply this in a multiscale setting. In the works of Liu (2020) and Liu (2021), a deep material network (DMN) was developed, which describes the RVE with a network that is built up from physics-based building blocks. In the first work (Liu, 2020), debonding effects in the RVE were successfully captured by the adaptable cohesive building blocks included in the network. In a multi-stage training strategy, one of the phases from the DMN trained for the bulk material is enriched with the interface interaction learned from a second DMN built on top of cohesive building blocks. As a result, the accuracy of the final network is somehow limited by the original bulk DMN. In Liu (2021) this method was extended to localization problems with a cell-division scheme, which overcame the difficulties related to selecting the proper size of the RVE. While these networks excel in extrapolating from linear elastic RVE data to nonlinear and path-dependent behavior, training and online evaluation are not straightforward. These two stages involve different input spaces and an iterative Newton–Raphson scheme is required for the online stage (Liu, 2020, 2021). Further, a probabilistic machine learning approach using Bayesian regression was proposed in Rocha et al. (2021) and also applied to active learning of traction–separation relations in a multiscale setting, but this approach was not made suitable for capturing unloading behavior.

In fracture mechanics problems, the computational cost involved with FE (without a multiscale framework) can also be extreme. A noteworthy approach to deal with that is proposed by Kerfriden et al. (2013) through the use of a domain separation strategy to focus the computational power on the fracture region, which requires most of the attention. The domain separation strategy can also be used in multiscale settings, as proposed by Oliver et al. (2017). To further reduce the sampling points, it is combined with another key technique based on model order reduction, specifically the Reduced Optimal Quadrature (ROQ). However, these methods are highly dependent on snapshots and the complexity of the problem increases with nonlinear and path-dependent materials.

In this work, the ability of the PRNN to describe the effect of microscale damage is investigated. The study is restricted to diffuse damage in the form of microscale debonding in composite materials. The aim is to describe the stiffness degradation that is the result of diffuse damage, without the occurrence of macroscopic softening. Section 2 outlines the FE^2 framework and briefly describes the PRNN as detailed in Maia et al. (2023). Section 3 provides the data generation method for training and testing the networks. In Section 4, the performance of the existing PRNN on cases with stiffness degradation is tested, motivating the development of new architectures. Section 5 introduces the proposed architectures, while Section 6 evaluates their performance in achieving the research objective.

2. Theoretical background

In the following sections, the foundational aspects of the methods used in this work are discussed. This includes an overview of the FE^2 method and the homogenization procedure, as well as the main features and limitations of the existing PRNN.

2.1. The FE^2 method

Computational homogenization with the FE^2 method allows for capturing the response of composite materials, for cases where it is difficult to do that on the macroscale due to the complex interaction between nonlinear constituents and microstructural geometry. In this approach, the structure is discretized as a homogeneous macrostructure, where a heterogeneous micromodel is nested into each macroscopic integration point of it (Schröder, 2014; Feyel and Chaboche, 2000; Feyel, 2003). The micromodel is assumed to be a representative volume element (RVE). The macroscopic strain values are downscaled as boundary conditions for the micromodel, where the microscopic boundary value problem (BVP) is solved. The microscopic stress values obtained from the BVP are then upscaled back to the macromodel after a homogenization operation. This bypasses the need for any assumptions on the constitutive relation at the macroscale.

The schematics of the FE^2 method is shown in Fig. 1. The macroscopic solid domain is denoted by Ω, and the surfaces where the Dirichlet and Neumann boundary conditions are applied are denoted as Γ_u^Ω and Γ_f^Ω, respectively. In this work, we simulate damage at the fiber–matrix interface, which precludes global softening of the micromodel and avoids the need for inserting a discontinuity on the macroscale. The discontinuity in the microscopic domain is denoted by Γ_d^ω. At the fracture surface, the two opposite sides of the crack are differentiated by a + and a − sign.

The displacement field in the micromodel with domain denoted as ω is approximated for the boundary conditions imposed from the macroscale with a finite element discretization of the RVE geometry. The macroscale strain $\boldsymbol{\varepsilon}^\Omega$ is considered to be constant over the volume aside from a periodic microscale fluctuation field due to the assumption of separation of scales. Nonlinearity in the microscale problem comes from the constitutive models $\mathcal{D}^\omega$ and $\mathcal{T}^\omega$:

$$\boldsymbol{\sigma}^\omega, \boldsymbol{\alpha} = \mathcal{D}^\omega\left(\boldsymbol{\varepsilon}^\omega, \boldsymbol{\alpha}^{t-1}\right) \tag{1}$$

$$\mathbf{t}_d^\omega, d = \mathcal{T}^\omega\left([\![\mathbf{u}^\omega]\!], d^{t-1}\right), \tag{2}$$

where $\boldsymbol{\sigma}^\omega$ is the microscale stress obtained from the microscale strain $\boldsymbol{\varepsilon}^\omega$ and the internal variables $\boldsymbol{\alpha}$, and $\mathbf{t}_d^\omega$ is the cohesive traction computed from the displacement jump $[\![\mathbf{u}^\omega]\!]$ and internal variable d.

After the computation of the full-order solution at the microscale, the resulting stress field is homogenized and the homogenized stress is returned to the macroscale model. The homogenized stress is defined as the average over the volume of the RVE:

$$\boldsymbol{\sigma}^\Omega = \frac{1}{|\omega|}\int_\omega \boldsymbol{\sigma}^\omega \, d\omega \tag{3}$$

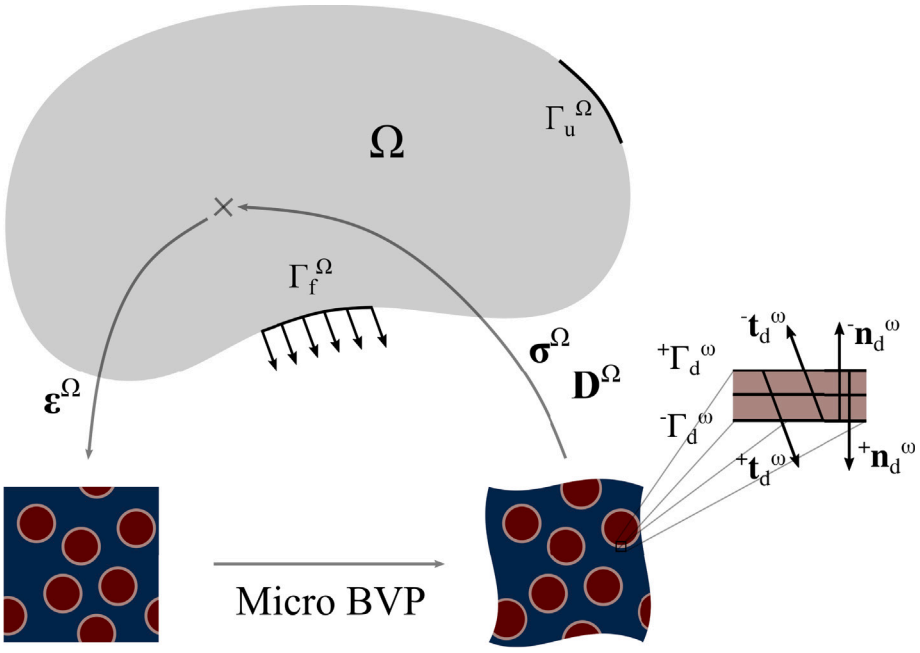


Fig. 1. FE^2 framework.

For accurate coupling between the two scales, the energy between them must be consistent. For this, the Hill–Mandel principle has to be satisfied (Hill, 1963) which is ensured when periodic boundary conditions are used. To solve the macroscale problem stress update, a nonlinear finite element solution procedure (e.g. based on the Newton–Raphson method) is needed, making the cost of solving the microscale BVP at each integration point and every macroscale iteration prohibitive for practical applications.

2.2. Physically recurrent neural network

To tackle the issues related to the black-box nature of neural networks, Physically Recurrent Neural Networks (PRNNs) developed by Maia et al. (2023) introduce a new way of embedding knowledge on the physics of a system in a surrogate model. Unlike in PINNs, where the physical constraints of the problem are incorporated in the loss function (Raissi et al., 2019), in PRNNs the actual material models used in the full-order microscopic BVP are implemented in the hidden layer of the network such that their state variables introduce a physics-based recurrency. Fig. 2 displays the PRNN in general terms for the case with two constitutive models on the microscale, D_1 and D_2.

The architecture consists of an input layer, a material layer, and an output layer. The macroscale strains $\boldsymbol{\varepsilon}^\Omega$ at the integration points of the macrostructure are the inputs to this network. In two dimensions assuming small strains, this corresponds to 3 input values. These macroscale strains are passed through an encoder, which is a single dense layer with linear activations. This encoder converts the macroscale strains to a set of values we interpret as fictitious microscopic strains, or local strains, which corresponds to the macro- to micro-scale transition in the FE^2 method. These local strains $\boldsymbol{\varepsilon}$ are given by:

$$\boldsymbol{\varepsilon} = \mathbf{W}_1 \boldsymbol{\varepsilon}^\Omega + \mathbf{b}_1, \tag{4}$$

where $\mathbf{W}_1$ are the weights connecting the input layer to the material layer and $\mathbf{b_1}$ is the bias associated with the encoder. There are no residual stresses considered in this work, which means that there is a zero stress state for when no strain is applied to the microstructure. Therefore, the network should also predict zero stresses when the strain inputs are zeros. This is achieved by setting the bias term $\mathbf{b_1} = \mathbf{0}$.

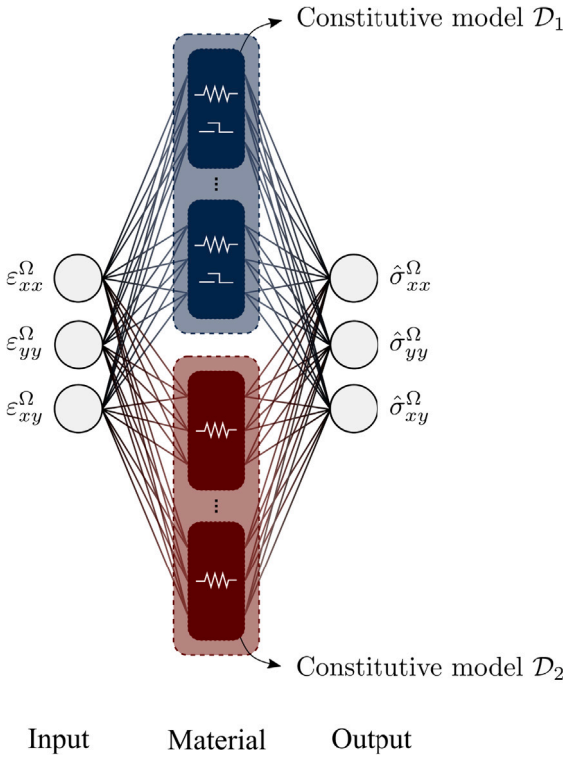


Fig. 2. Architecture of PRNN.

These local strains are passed through the material layer, which provides the essence of the physically recurrent neural network. The material layer consists of cells, each with three inputs and three outputs, and possibly internal variables, representing a fictitious integration point. In these points, a classical constitutive model D^ω converts the local strains $\boldsymbol{\varepsilon}$ to local stresses $\boldsymbol{\sigma}$. This constitutive model is the exact same model

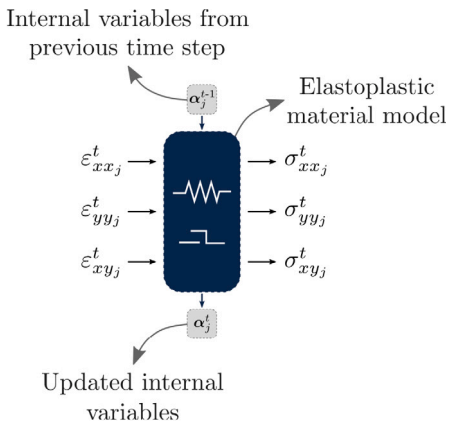


Fig. 3. Fictitious material point in material layer.

that is used to compute the stress in the integration points of the full-order micromodel. The number of fictitious material points in the layer is a model hyperparameter to be tweaked via model selection.

History dependence is naturally included in the PRNN by storing the internal variables $\boldsymbol{\alpha}$ of each material point, which for example in plasticity can be plastic deformation, as they are computed in the assigned constitutive model. Therefore, path-dependency does not need to be learned from data. This stands in contrast with regular recurrent neural networks, where the evolution of history variables is also learned through additional learnable parameters and standard activation functions (e.g. with LSTM or GRU cells). The operation in the fictitious material point j at time t is shown in Fig. 3 and can be described by:

$$\boldsymbol{\sigma}_j^t, \boldsymbol{\alpha}_j^t = D^{\omega}(\boldsymbol{\varepsilon}_j^t, \boldsymbol{\alpha}_j^{t-1}) \tag{5}$$

After the local stresses are computed in the material layer, a decoder is applied to these values. In the particular architecture shown in Fig. 2, the decoder consists of a dense layer with a SoftPlus activation function applied on the weights ($\phi_{sp}(\cdot)$). This is done to represent the homogenization process through numerical integration, in which weights are strictly positive. Therefore, the macroscale stress output of the network is obtained by:

$$\hat{\boldsymbol{\sigma}}^{\Omega} = \phi_{\text{sp}}(\mathbf{W}_2)\boldsymbol{\sigma} + \mathbf{b}_2, \tag{6}$$

where $\mathbf{W}_2$ are the weights connecting the material layer to the output layer, and $\mathbf{b_2}$ is the bias associated with the decoder. This bias term is again set to zero to ensure zero macroscale stresses for zero local stress values. Essentially, the network is tasked to learn how to best combine the response of a small number of material points into a representative macroscale response.

During training, the following loss function is minimized:

$$\text{L} = \frac{1}{N}\sum_{t=1}^{N} \|\boldsymbol{\sigma}^{\Omega}(\boldsymbol{\varepsilon}_t^{\Omega}) - \hat{\boldsymbol{\sigma}}^{\Omega}(\boldsymbol{\varepsilon}_t^{\Omega})\|^2, \tag{7}$$

where N is the number of stress–strain pairs in the dataset and $\boldsymbol{\sigma}^{\Omega}(\boldsymbol{\varepsilon}_t^{\Omega})$ is the target value, which in this case is obtained from full-order micromodel simulations followed by averaging stresses over the microscopic volume. Predicting the stress response in PRNNs consists of a simple forward pass, making them computationally efficient in the online phase and alleviating the computational bottleneck of multiscale modeling. This is one of the main differences compared to DMNs, where the online-phase is computationally heavier due to its iterative nature. Additionally, PRNNs offer a more flexible alternative by implementing the constitutive models directly into the network structure.

Because of the history variables $\boldsymbol{\alpha}$, back-propagation in time becomes necessary and stress/strain pairs are grouped in paths (time series). Data and gradient handling is therefore analogous to when training RNNs, with the key difference being that memory in PRNNs is physical and interpretable. It is worth emphasizing that if the material model is implemented with automatic differentiation support, gradients are handled automatically through general-use packages such as pytorch and tensorflow. Otherwise, a detailed implementation of how to incorporate these using finite differences is given in Maia et al. (2023).

In the full-order micromodel used in Maia et al. (2023), J_2 plasticity was used for the matrix and a linear elasticity model for the fibers. In the previous study, the PRNN was able to find a solution with only the elastoplastic model (i.e. the constitutive model used to describe the matrix) in the material layer with no loss of accuracy. The expected linear elastic behavior in the fibers is reproduced in elastoplastic material points when small enough strain values are passed by the encoder and stresses are amplified in the decoder, making one or more matrix material points effectively work as if they were linear elastic fiber points.

3. Data generation

To generate the data for assessing the PRNN's performance when predicting microscale damage, a micromodel with cohesive elements at the fiber–matrix interface is considered. This section introduces the micromodel used to create the training and test datasets and the different loading conditions that are considered.

3.1. Full-order micromodel

The FE model of the microscale is shown in Fig. 4 and consists of 25 periodically arranged fibers and diameter of 5 μm embedded in a matrix to result in a fiber volume fraction of 0.6. This single RVE is used to generate all datasets in this study. There are two bulk constitutive models: a plasticity model for the matrix, and a linear elastic model for the fibers. The geometry, mesh, and bulk material properties in the RVE are kept as in Maia et al. (2023) except that zero-thickness interface elements are positioned at the fiber–matrix interfaces. Limiting damage to the fiber–matrix interface means that no global failure can take place. Tractions at the interface elements are computed from displacement jumps with the bilinear cohesive zone model (CZM) by Turon et al. (2006).

For the CZM properties, we use equal normal and shear strengths $\tau_n^0 = \tau_s^0 = 60$ MPa, mode I and mode II fracture energy $G_{Ic} = 0.874$ kJ/m^2, $G_{IIc} = 1.717$ kJ/m^2, mode interaction parameter $\eta = 1$, and penalty stiffness $K = 5 \cdot 10^7$ N/mm^3. Plane stress conditions are assumed for the micromodel.

3.2. Load path generation

To generate data for training and testing of the network, the micromodel is subjected to different loading paths using periodic boundary conditions. The datasets can be separated into two categories: proportional and non-proportional loading.

3.2.1. Proportional loading

For proportional loading, we use a modified arc-length algorithm to enforce the directions of the applied stress, as described in Rocha et al. (2020). In this method, the proportionality of the stress response is enforced by considering a constant load vector with increments defined in terms of displacement magnitude, specifically the sum of the unsigned applied displacements. In this work, the increments are fixed at $\Delta s = 1.67 \times 10^{-3}$ mm. The loading directions are categorized as either fundamental or random. The fundamental directions contain 18 common loading cases often used for traditional material model calibration, shown in red in Fig. 5(a), and include pure tension, compression, shear, biaxial tension, and combinations thereof. On the other hand, the random directions are obtained by sampling three values, each corresponding to a component of the load vector, from

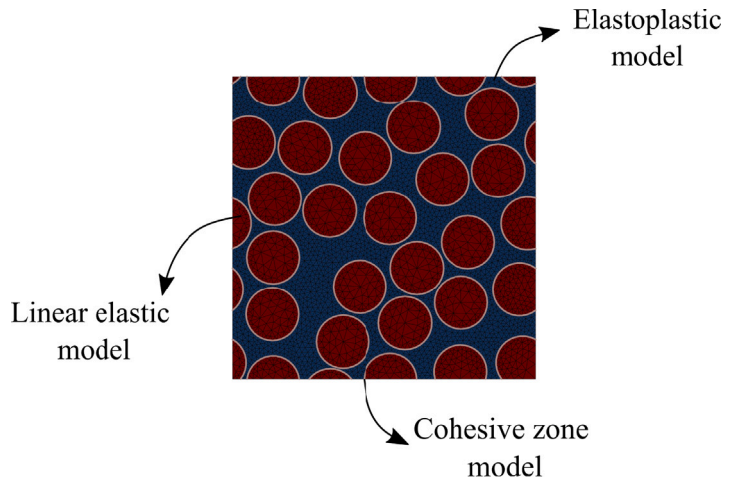
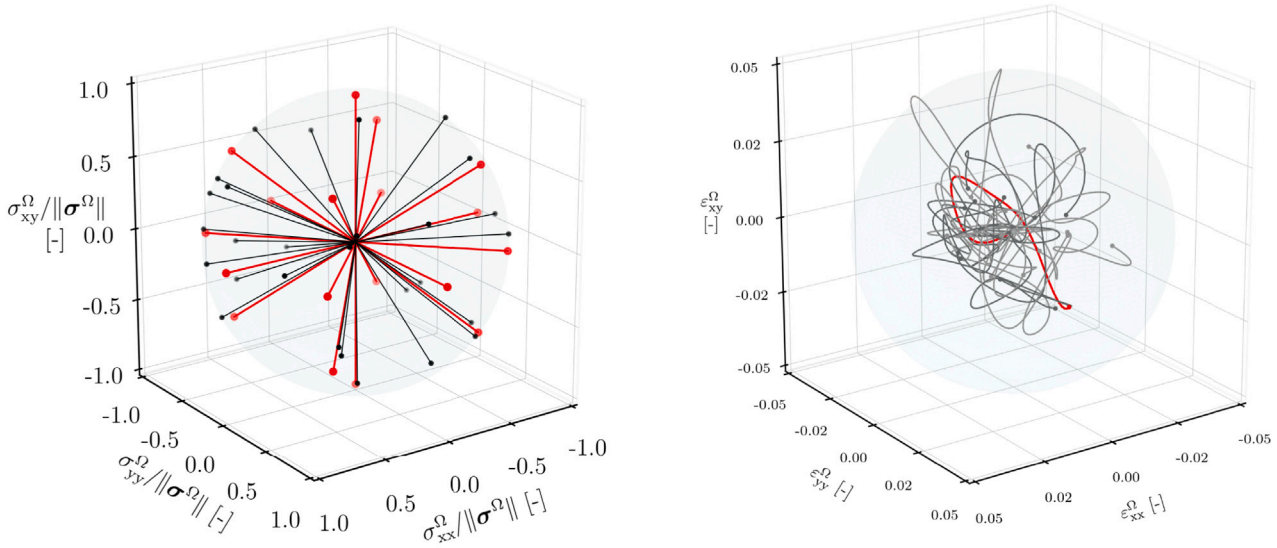


Fig. 4. Full-order micromodel used in this work.



(a) Proportional loading path directions

(b) Non-proportional loading paths

Fig. 5. Types of loading paths considered in this work. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

a normal distribution $\mathcal{N}(0,1)$ and normalizing them to a unit vector, with examples shown in black in Fig. 5(a).

In this work, only non-monotonic loading is considered. During non-monotonic loading, the direction in which the step size is kept fixed, but unloading takes place at different loading steps for a predefined amount of time. The loading functions that define the relation between t and the magnitude of loading for non-monotonic cases considered in this paper are shown in Fig. 6. In the arc-length formulation, this corresponds to the imposed value for the unsigned sum of the displacements at the controlling nodes.

3.3. Non-proportional loading

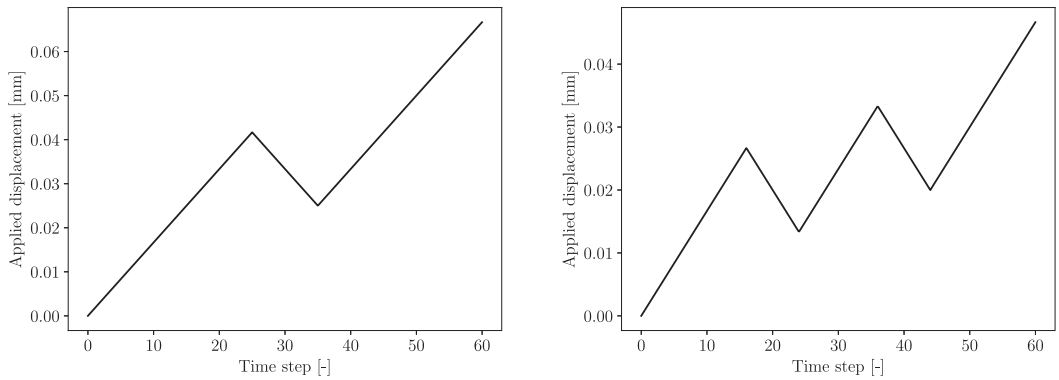
To create more diverse loading scenarios, non-proportional and non-monotonic loading paths are generated. Both the direction of loading and the step size are varied at each time step. This is achieved by sampling the strains from Gaussian Processes (GPs). Each strain component is drawn from an independent multivariate normal distribution given by:

$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{8}$$

where $\mathbf{X}$ represents a vector containing the strain values at the different time steps, $\boldsymbol{\mu}$ is the mean vector that specifies the expected value of strains, and $\boldsymbol{\Sigma}$ is the covariance matrix. The covariance matrix $\boldsymbol{\Sigma}$ describes the relationships between the samples in each of the components. The covariance function between two time steps i and j is given by:

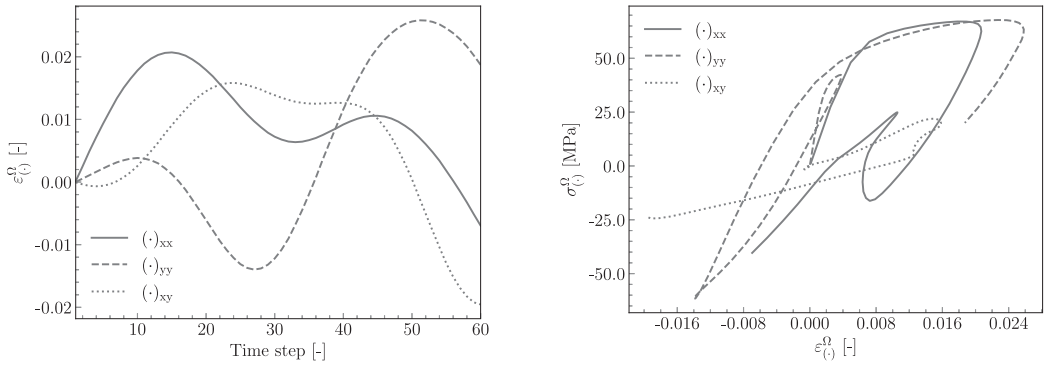
$$\Sigma_{ij} = k(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|x_i - x_j\|^2\right) \tag{9}$$

with σ_f^2 being the variance that determines the step size and ℓ being the length scale that controls the smoothness of the generated path. With increased variance σ_f^2 the strains are able to attain larger values, and with increased length scale the path becomes smoother. Values $\sigma_f^2 = 0.0001667$ and $\ell = 200$ are used in this work. A subset of the load paths generated by GPs is shown in Fig. 5(b), with one path highlighted in red for clarity. We also show in Fig. 7(a) the corresponding strain paths for the highlighted loading path and the corresponding stress–strain curves obtained from the full-order micromodel in Fig. 7(b).



(a) Function with one unloading cycle (b) Function with two unloading cycles

Fig. 6. Loading functions used to generate proportional non-monotonic loading curves.



(a) Strain paths (b) Stress-strain curves

Fig. 7. Example of non-proportional GP-based loading path.

4. Performance of PRNN with bulk model only

This section investigates whether the PRNN as proposed in Maia et al. (2023) is able to capture stiffness degradation due to microscale damage. The architecture consists of one input layer, one material layer with bulk integration points only, and one output layer, as depicted in Fig. 8. All bulk material points embed a J_2 plasticity model to convert 2D local strains to 2D local stresses.

In Maia et al. (2023), the network could find a way to make elastoplastic material points reproduce linear elasticity by appropriately scaling encoder and decoder weights. In the following, we demonstrate how such an approach does not work for distributed damage. To highlight this inability to learn as clearly as possible, the networks here are trained and tested on the same curves. Specifically, the non-monotonic, proportional dataset with one cycle of unloading in the 18 fundamental directions is used. Networks with different material sizes are trained by adding bulk points to the network until the mean value of the Mean Squared Errors (MSEs) no longer decreases with additional points. The training MSE across the different material layer sizes is shown in Fig. 9, with 10 networks with different initializations per size plotted as blue dots and the purple line representing the mean value for each material layer size. The best performing network with 7 fictitious material points and a xtraining MSE of 4.61 MPa is selected for further examination.

The prediction of the network on two fundamental loading scenarios is shown in Fig. 10: uniaxial tension and biaxial tension with shear. The network provides a somewhat accurate prediction on the monotonic region of the curve, however, the model is unable to reproduce the unloading/reloading region. The PRNN starts to predict unloading with the initial, linear elastic stiffness following the assumptions embedded in the J_2 model, and predicts erratically afterwards. This highlights the limitation of the PRNN for describing stiffness degradation in its original design. The network encodes plasticity through the presence of a plasticity model in the material layer. This design gave the network a good bias in earlier work, when it could predict unloading behavior in plasticity without seeing it during training (Maia et al., 2023).

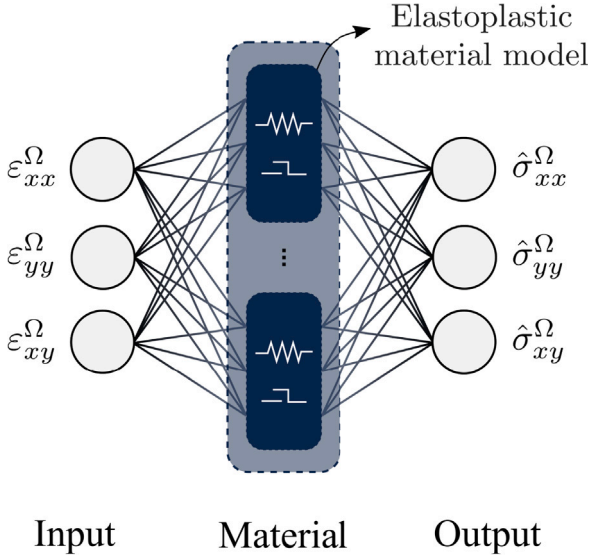


Fig. 8. PRNN with elastoplastic model only.

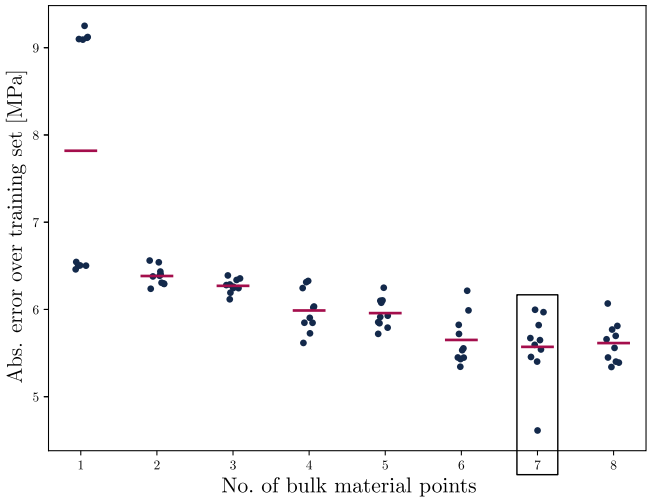


Fig. 9. Training error for PRNN trained on 18 fundamental curves with one cycle of unloading. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Here, however, the bias is too strong as it prevents the network from describing the stiffness loss that is present in the micromodel.

This observation is in line with the core idea of the PRNN to include a representation of all relevant physics by embedding the constitutive models from the micromodel in the network. This idea is violated by not including the cohesive zone model in the network. Therefore, the following sections of this work focus on the implementation of the cohesive zone model within the PRNN framework, along with evaluation of the proposed architectures.

5. Extending the network with cohesive material points

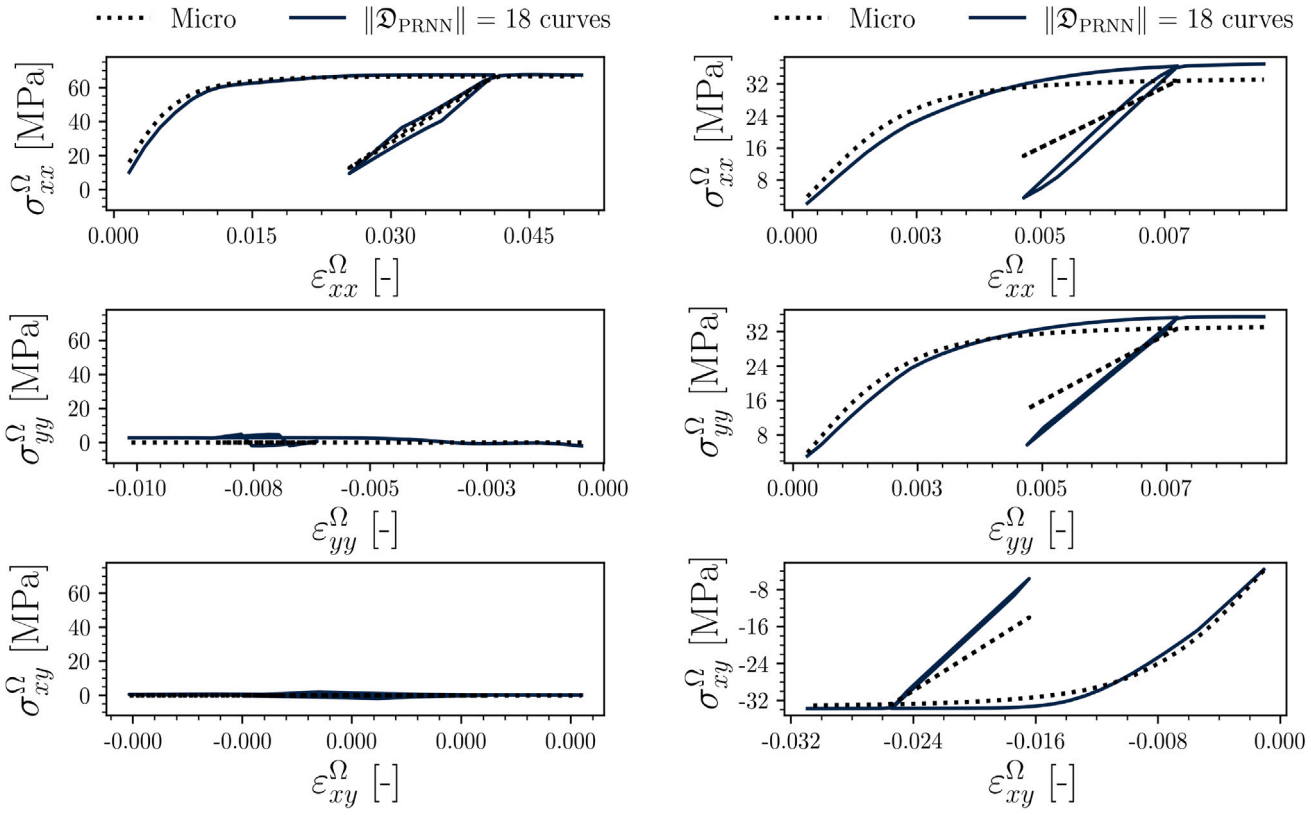
As shown in Section 4, the physically recurrent neural network cannot accurately predict the effect of debonding at the fiber–matrix interface without including all sources of nonlinearity present in the RVE. Therefore, the cohesive zone model from the full-order micromodel has to be implemented in the PRNN as well. This section details the network configurations considered in this study for implementing the CZM within the PRNN framework.

5.1. Cohesive points in the existing material layer

The first design option retains the architecture proposed in Maia et al. (2023) as much as possible. In this design, referred from now on as PRNN_1, there is one material layer containing bulk and cohesive fictitious points. The network is illustrated in Fig. 11 with bulk and cohesive points, represented in blue and pink, respectively. The cohesive points relate the local displacement jump vector, with normal and shear components, to a local traction vector, as illustrated in Fig. 12. Similar to the bulk points, the cohesive points also store internal variables to account for history, in this case the damage variable defined as the ratio between dissipated energy and critical energy release rate (Turon et al., 2018).

5.2. Cohesive points in separate layer

Instead of having both types of models in the same layer, we also investigate architectures with two material layers: a cohesive and a



(a) Uniaxial load case

(b) Biaxial load case with shear

Fig. 10. Prediction of PRNN trained with 18 fundamental curves with one cycle of unloading on training curves.

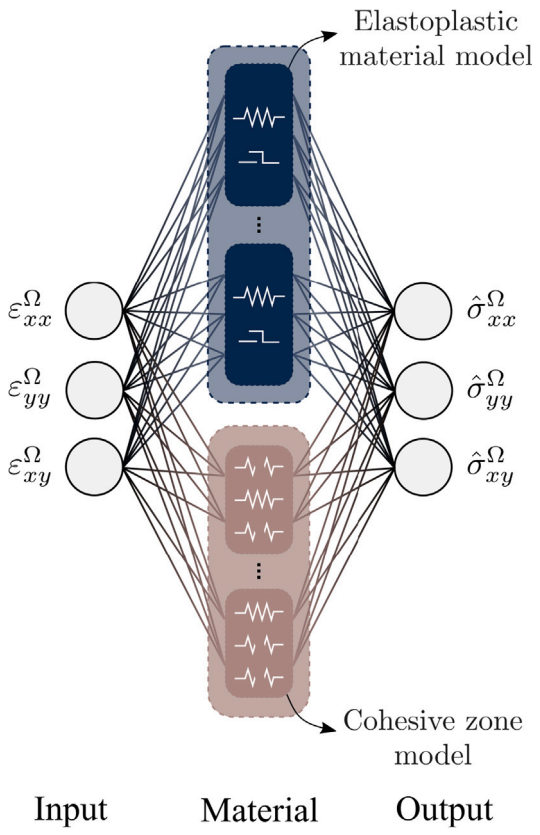


Fig. 11. PRNN$_1$ architecture: bulk and CZM in the same material layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

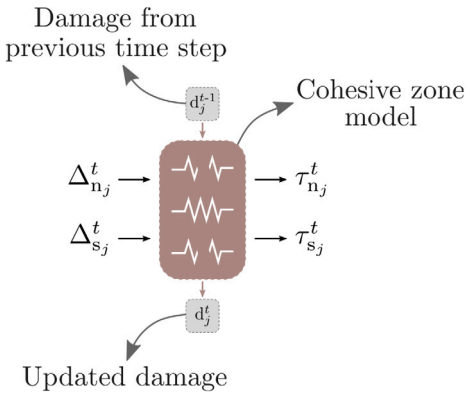


Fig. 12. Fictitious cohesive material point j.

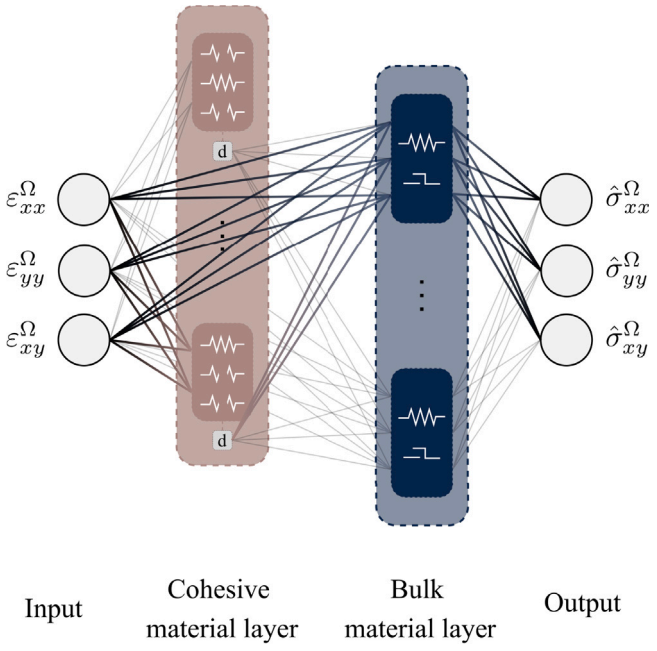


Fig. 13. Novel architecture PRNN$_2$ and PRNN$_3$ with damage as input to bulk point.

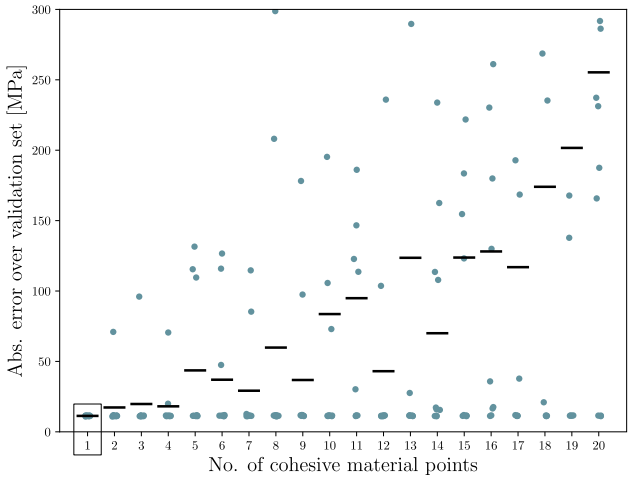


Fig. 14. Validation error for PRNN$_1$ trained on 192 GP curves.

bulk material layer, each with embedded models, as illustrated in Fig. 13. The two architectures considered here consist of one input layer that receives macroscopic strain, two material layers containing the nonlinear models, and one output layer yielding the macro-scale stress predictions. The state variables of the cohesive points are densely connected to the bulk points together with the macroscopic strains. To illustrate the connectivity of the layers, Fig. 13 highlights how one cohesive point and one bulk point are linked to each other, as well as to the input and output.

Rather than using the output traction values of the cohesive points, damage is used as input to the bulk points. Damage, the internal variable stored in the cohesive points, either increases or remains the same in case of unloading, providing a more monotonic influence on the overall response. This damage variable modifies the local strain value received by the bulk points, resulting in adjusted local stress values for the same level of macroscopic strain. The irreversibility of damage gives rise to a decrease in stiffness during unloading. This design ensures that only the bulk points contribute directly to the stress homogenization procedure, unlike in the architecture described in Section 5.1, where tractions rising from the cohesive points are directly connected to the output through the decoder layer. This is more consistent with the homogenization procedure in FE^2, where cohesive tractions do not contribute directly to the macroscopic stress (cf. Eq. (3)).

Two ways of connecting the damage variable from the cohesive points to the local strain at the bulk points are considered. The first method follows a more conventional approach, which involves densely connecting the damage values to the bulk points:

$$\boldsymbol{\epsilon} = \mathbf{W}_{\mathrm{d}} \cdot \mathbf{d} + \mathbf{W}_{\varepsilon \mathrm{b}} \cdot \boldsymbol{\epsilon}^{\Omega} \tag{10}$$

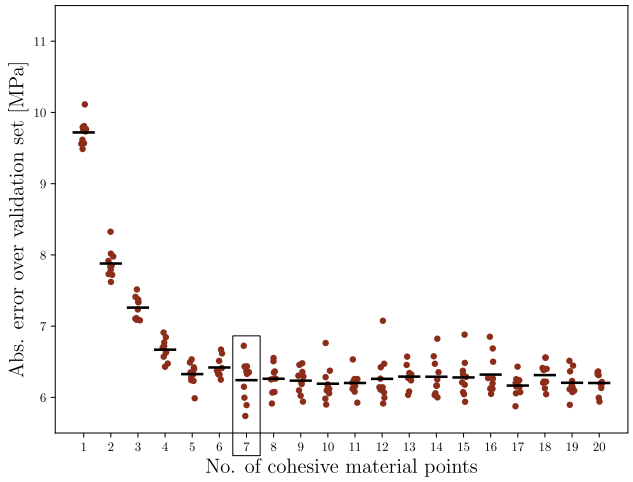


Fig. 15. Validation error for $PRNN_2$ trained on 192 GP curves.

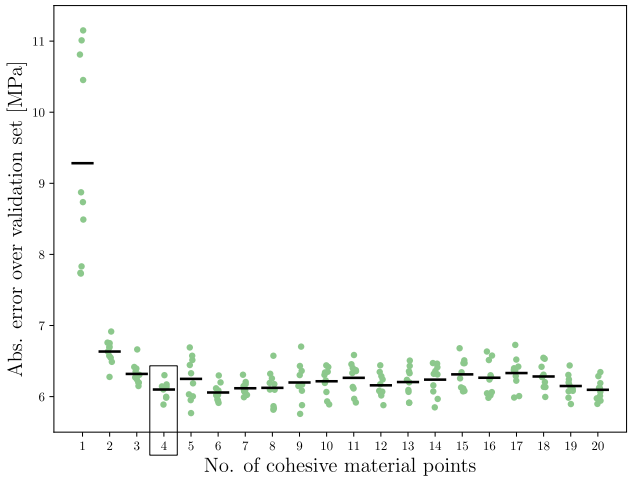


Fig. 16. Validation error for $PRNN_3$ trained on 192 GP curves.

where $\mathbf{W}_{\mathrm{d}}$ and $\mathbf{W}_{\epsilon\mathrm{b}}$ are the weight matrices connecting the damage values $\mathbf{d}$ from all the cohesive points and the macroscale strain, respectively, to the local strain values of the bulk points. The network with this approach will be referred to as $PRNN_2$ from now on.

In the second method, referred to as $PRNN_3$ from now on, the damage variables are used to modify the amplitude of the local strain input to the bulk points. This is achieved by multiplying the local strain input piece-wise by the term $\phi_{\mathrm{sp}}(\mathbf{1} + \mathbf{W}_{\mathrm{d}} \cdot \mathbf{d})$, which is forced to attain positive values by applying a SoftPlus activation function ($\phi_{\mathrm{sp}}(\cdot)$):

$$\boldsymbol{\epsilon} = \phi_{\mathrm{sp}}(\mathbf{1} + \mathbf{W}_{\mathrm{d}} \cdot \mathbf{d}) \odot (\mathbf{W}_{\epsilon\mathrm{b}} \cdot \boldsymbol{\epsilon}^{\Omega}) \tag{11}$$

6. Performance of PRNN with cohesive model

In this section, we assess the performance of the PRNNs with the architectures proposed in Section 5. The model selection process is presented by analyzing their performance across different training sets and material layer sizes. We evaluate and compare the PRNNs' ability to accurately capture the homogenized response, taking into account microscale damage, under various loading scenarios.

6.1. Model selection

First we perform model selection for the size of the material layer. For that purpose, networks are trained on 192 GP-based curves (non-monotonic and non-proportional loading) with varying numbers of bulk and cohesive points. The ratio of bulk to cohesive points is kept constant and equal to 4, mirroring the ratio of matrix to cohesive elements in the RVE. The size of the material layer ranges from a minimum configuration of four bulk and one cohesive points to 80 bulk and 20 cohesive points. Figs. 14–16 show the MSE on a validation set with 200 GP-based curves for 10 different initializations in each material layer size, for all three architectures considered in this work. Networks with lowest validation MSE are selected for optimal performance, while prioritizing small networks to avoid overfitting. For $PRNN_1$, networks with 4 bulk points and 1 cohesive point are selected, while for $PRNN_2$ a combination with 28 bulk points and 7 cohesive points is needed. Finally, for $PRNN_3$, 16 bulk and 4 cohesive points are selected. The observation that the validation error increases for increasing network sizes of $PRNN_1$ points at the tendency of this architecture to overfit.

The selected networks are then trained on different training set sizes, ranging from 4 to 192 GP-based curves (non-monotonic and non-proportional loading). Fig. 17 displays the validation MSE across the various training data sizes for the selected material layer sizes of the three architectures considered in this work. The solid lines in the figure represent the mean MSE values for each PRNN at different training data sizes. For the first architecture ($PRNN_1$), a training set size of 96 paths is selected. The plateau in Fig. 17 indicates that the network in this configuration has reached the limit of its representational power and is too rigid to capture the underlying physical behavior. On the other hand, $PRNN_2$ and $PRNN_3$ can represent a broader range of material non-linearity and therefore can reach lower MSE values as dataset size is further increased. Therefore, 192 curves are selected for the latest two architectures. The lowest validation error corresponding to these networks is 11.40 MPa, 5.74 MPa, and 5.89 MPa, achieved with training times of 10, 60, and 20 h, respectively.

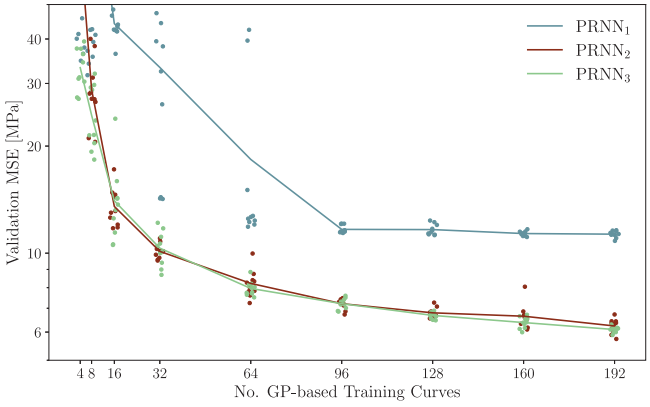
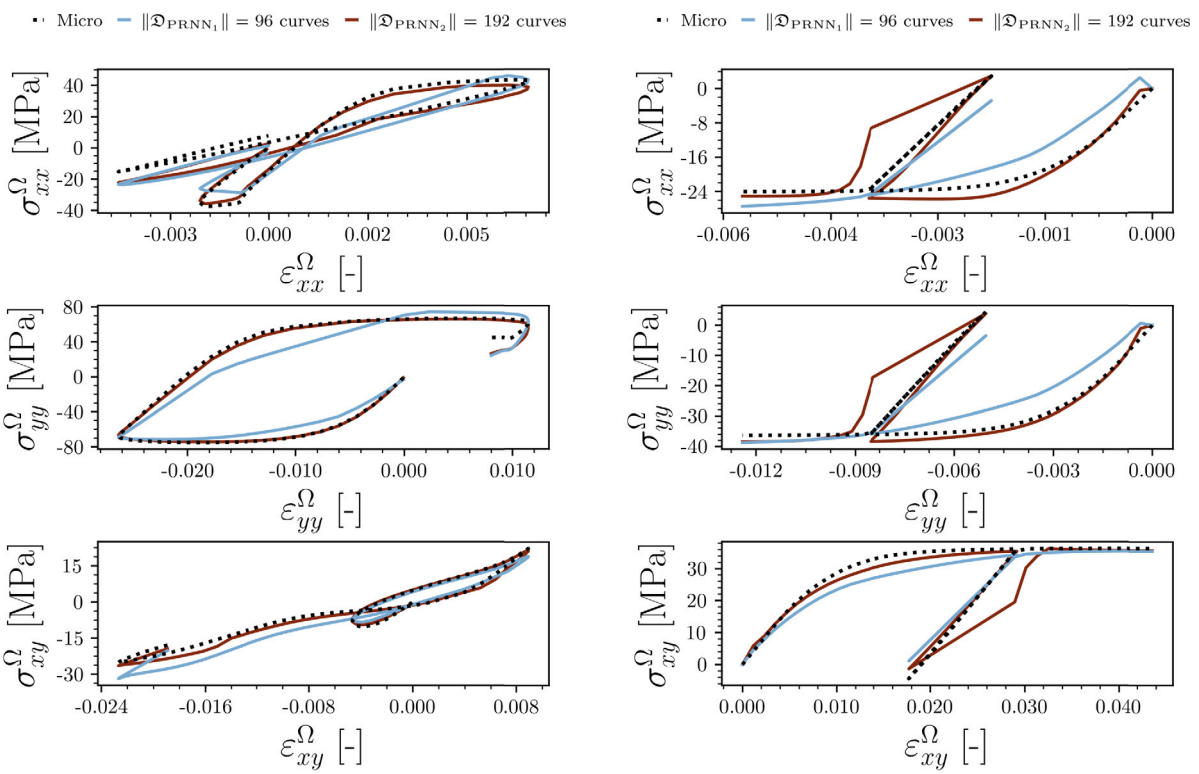


Fig. 17. Validation MSE for the PRNNs considered across various training data sizes.

6.2. Predicting micro-scale damage

The selected networks are tested on two different datasets: one containing 54 curves from the non-monotonic, non-proportional dataset (the same loading type used for training and validation but in different directions), and another with 54 curves from the random, non-monotonic, and proportional dataset with one cycle of unloading with. The average MSE values on these two test sets are presented in Table 1 for each network. It is observed that for the non-monotonic non-proportional test curves, $PRNN_2$ and $PRNN_3$ both outperform $PRNN_1$, with a small difference in accuracy between the two. However, when testing on curves of the proportional type, $PRNN_3$ offers significant additional accuracy over $PRNN_2$.

To illustrate the meaning of these numbers, in the remainder of this section, we compare the performance of the three networks in more



(a) Prediction of PRNN_1 and PRNN_2 on a representative GP curve

(b) Prediction of PRNN_1 and PRNN_2 on a representative curve with one cycle of unloading

Fig. 18. Prediction of PRNN_1 and PRNN_2 on representative test curves.

Table 1
Average MSE values for the two test datasets, in MPa.

Test set	PRNN_1	PRNN_2	PRNN_3
Non-monotonic, non-proportional	11.63	5.72	6.03
Non-monotonic, proportional	7.45	5.56	3.40

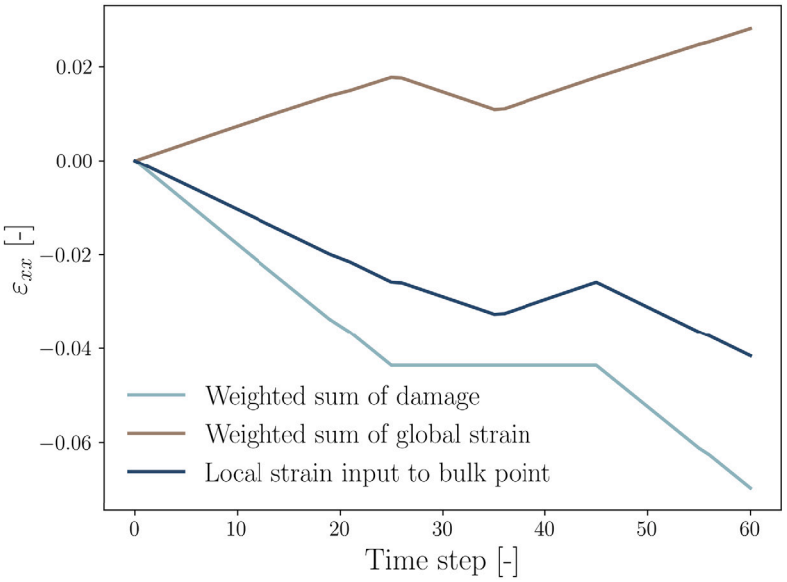
detail with stress predictions on individual curves, each time picking representative curves with MSE close to the average MSE from Table 1.

First, we illustrate in Fig. 18 the performance of PRNN_1 and PRNN_2. Note how the predictions of PRNN_1 on GP-based curves follow the overall trend but with significantly less accuracy compared to PRNN_2 (Fig. 18(a)). This observation aligns well with the results shown in Fig. 17, where the validation set is also comprised of GP-based curves, emphasizing the significant decrease in validation error when the cohesive points are implemented in a separate layer from the bulk points. The difference between PRNN_1 and PRNN_2 predictions becomes more pronounced when tested on the non-monotonic, proportional dataset. As shown in Fig. 18(b), PRNN_1 predicts poorly. The network not only fails to capture the decrease in stiffness during unloading but also loses accuracy in the monotonic part. It is observed that with PRNN_1 there is a preference towards networks with fewer cohesive points. Moreover, small weights connect the normal component of these cohesive points with the output, which is likely due to the large traction values output from the cohesive points in compression. These factors indicate that the network avoids utilizing the cohesive points implemented in the material layer, resulting in unloading with the initial linear stiffness.

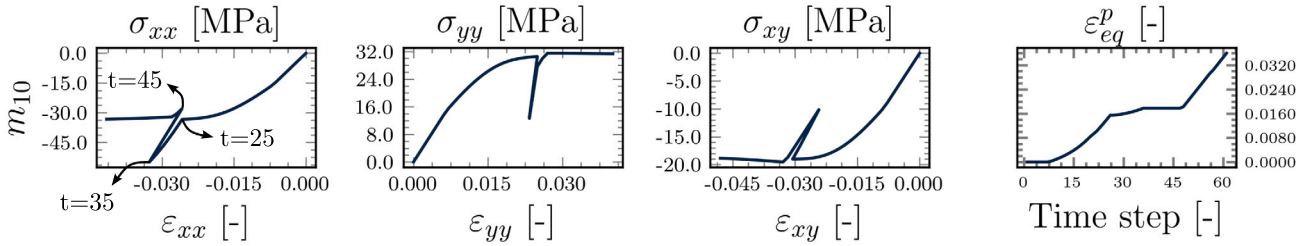
Besides the improved accuracy on the test sets, as shown in Table 1, PRNN_2 shows another advantage. Unloading occurs with a different slope than the initial linear phase (Fig. 18(b)), indicating that the network is able to account for the effect of microscale damage. However, a new problem arises: reloading follows a different path than unloading. Given the interpretable nature of PRNNs, this phenomenon can be investigated by closely examining the input to one of the bulk points. Fig. 19(a) illustrates the ε_{xx} component of a particular fictitious bulk material point and its two contributions, one that follows directly from the macroscopic strain ($\mathbf{W}_{\varepsilon\mathrm{b}} \cdot \boldsymbol{\varepsilon}^{\Omega}$) and the other that follows from the damage variables from all the cohesive points ($\mathbf{W}_{\mathrm{d}} \cdot \mathbf{d}$). Globally, the micromodel is unloading from time step 25, a trend represented by the weighted sum of the global strain values. Meanwhile, the weighted sum of damage is larger than the weighted sum of the global strain and has an opposite sign. Therefore, the final sum used as input to the bulk point prevents the point to follow the global unloading trend (from t = 25 to t = 35). Instead, the bulk point continues to load while the macroscopic strain is subjected to unloading and only starts unloading once the macroscopic strain is at the reloading branch (from t = 35 to t = 45). This mismatch between the loading phases leads to further evolution of plastic strain during the macroscopic unloading, as shown in Fig. 19(b), and causes the undesired change in slopes during unloading and reloading.

When damage is used as an amplifier rather than being simply densely connected to the bulk points, no significant difference in validation errors is found (Fig. 17). This is reflected on the prediction of PRNN_2 and PRNN_3 on a representative GP-based curve in Fig. 20(a). However, robustness improves significantly when predicting on curves from the non-monotonic, proportional set. Fig. 20(b) clearly shows that unloading/reloading now takes place along the same path and with a different slope than the initial linear phase, effectively capturing the effect of microscale damage.

To further demonstrate the network's predictive capabilities, Fig. 21 shows the prediction of PRNN_3 on a curve from a test set containing non-monotonic, proportional curves with two cycles of unloading. The evolution of damage over time is evident as the slope



(a) Input to the y-component of the bulk point



(b) Response of the bulk point

Fig. 19. Behavior of one of the bulk points of $PRNN_2$ when predicting on a representative curve from the non-monotonic, proportional test set.

of the unloading–reloading phase gradually decreases as the loading continues.

7. Conclusions

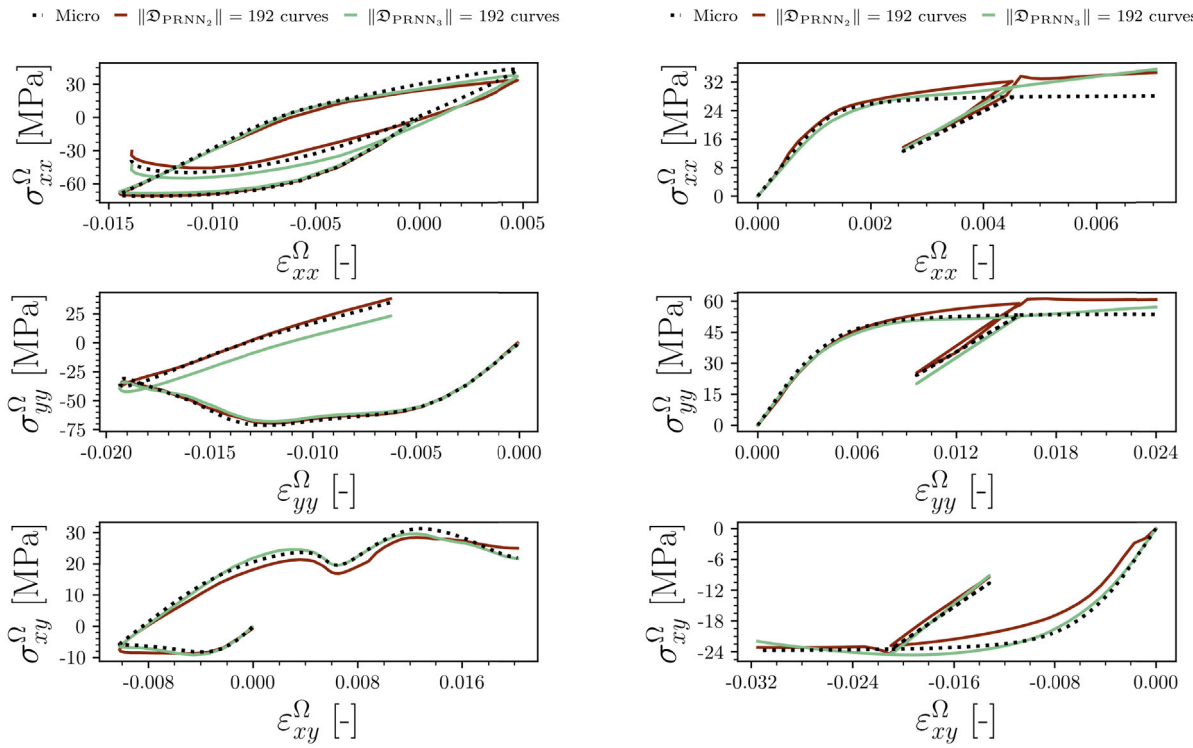
In this paper, we have proposed an extension to a recently proposed surrogate model, namely the Physically Recurrent Neural Network (PRNN), to account for the complex combination of plasticity and microscale damage. The PRNN's excellent ability to predict elastoplastic behavior motivated this study into its use as a surrogate model in a more challenging context where both plasticity and damage are present. Constitutive relations from the full-order micromodel are directly implemented into the hidden layer of the PRNN, creating a direct link to the micromodel. Path-dependency naturally arises from the material models in the network, resulting in accurate predictions with a significantly smaller training dataset compared to networks without physical interpretation.

As a first step to use the PRNN framework for microscale damage, a preliminary study was conducted using the network in its original form with bulk material points only. It was demonstrated that the original PRNN could not describe stiffness degradation, even when trying to overfit on a small set of training curves. These results align well with the general guideline in Maia et al. (2023) that all types of nonlinearities present in the RVE need to be included in the network, justifying the need for an extended PRNN architecture that integrates a cohesive zone model.

Next, three architectures of the PRNN with bulk and cohesive points were proposed. In the first design, $PRNN_1$, cohesive points with the CZM were incorporated into the same material layer of the PRNN as the plasticity model. Two material layers were used in the second and third design with the cohesive points implemented in a separate cohesive layer from the bulk material layer. Together with the global strain, the internal variable of the cohesive points, damage, was then used as input to the bulk points. This connection was defined in two different ways, either in a conventional way with a dense connection ($PRNN_2$) or by using the damage as an amplifier to the local strain of the bulk points ($PRNN_3$).

Afterwards, the performance of the proposed PRNNs was evaluated. The three networks were trained with data from non-monotonic, non-proportional (GP-based) curves and tested on the same type of curves, and on proportional, non-monotonic curves with one cycle of unloading.

When tested on GP-based curves, the results showed that while all three networks followed the general trend of the curves, $PRNN_2$ and $PRNN_3$ performed with significantly higher accuracy than $PRNN_1$. Additionally, $PRNN_1$ failed to accurately capture the loss of stiffness due to damage evolution when predicting on the non-monotonic, proportional dataset. Specifically, it unloaded with the initial linear stiffness, which is due to the network's preference towards not utilizing the cohesive points effectively. This limitation can be explained by the network's layout: when the cohesive points are implemented in the material layer together with the bulk points, the stress output of the network is given by a linear combination of both stresses coming from the bulk models and tractions coming from cohesive zone models. This layout of the $PRNN_1$ does not resemble the physics of the full-order solution, where only the bulk points contribute to the stress homogenization.



(a) Prediction of PRNN_2 and PRNN_3 on a representative GP curve

(b) Prediction of PRNN_2 and PRNN_3 on a representative curve with one cycle of unloading

Fig. 20. Prediction of PRNN_2 and PRNN_3 on representative test curves.

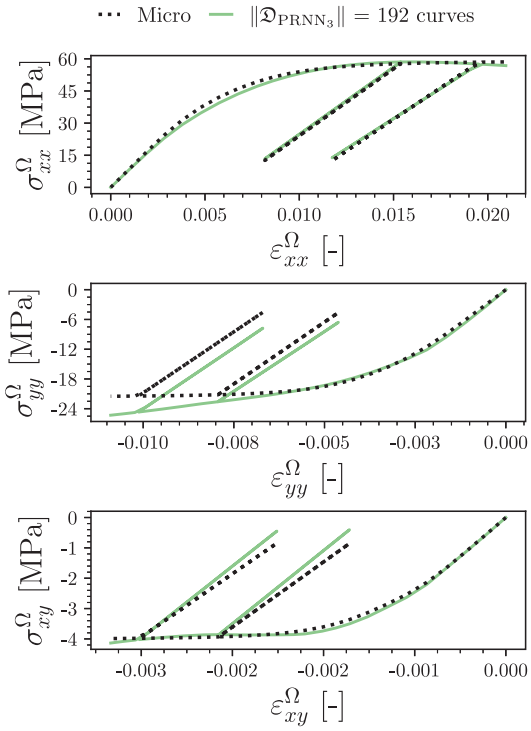


Fig. 21. Prediction of PRNN_3 on a representative curve with two cycles of unloading.

On the other hand, the modified architectures with the damage input to the bulk points did not have this problem. Adjusting the local strain by the damage input allowed for a modified tangent stiffness matrix able to capture the decrease in stiffness during unloading. This highlights the significance of designing the network's architecture with the knowledge of the underlying material behavior to achieve more accurate predictions.

When tested on non-monotonic, proportional curves with one cycle of unloading, PRNN_3 outperformed PRNN_2. It was observed that with a simple linear dense connection between the damage and bulk points, unloading and reloading occurred along different paths: while the RVE was unloading on the global scale, some bulk points in the network experienced further loading. This phenomenon occurred because the weighted sum of damage caused the input to the bulk point to have an opposite sign, leading these points to undergo further loading instead of unloading. This caused further plastic strain development during macroscopic unloading and led to the different slopes during unloading and reloading. The issue with the different unloading/reloading path was mitigated when damage was used as an amplifier in PRNN_3. This method ensured that the fictitious bulk points follow the global trend of unloading/reloading.

Lastly, PRNN_3 was tested on non-monotonic, proportional curves with two cycles of unloading. The network provided accurate predictions in this case as well, demonstrating a progressively decreasing stiffness in successive unloading/reloading phases. This significant result highlights the network's capability to capture damage evolution over time, and further reinforces that the PRNN with modifications to its architecture is capable of representing microscale damage.

CRediT authorship contribution statement

N. Kovács: Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **M.A. Maia:** Writing – review & editing, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **I.B.C.M. Rocha:** Writing – review & editing, Supervision,

Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **C. Furtado:** Writing – review & editing, Supervision. **P.P. Camanho:** Writing – review & editing, Supervision. **F.P. van der Meer:** Writing – review & editing, Supervision, Software, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

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

Acknowledgments

This work is supported by the TU Delft AI Labs programme through the SLIMM AI lab. FM acknowledges financial support from the Dutch Research Council (NWO) under Vidi grant 16464. We conrm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed.

Data availability

Data will be made available on request.

References

Cremonesi, M., Néron, D., Guidault, P.-A., Ladevèze, P., 2013. A PGD-based homogenization technique for the resolution of nonlinear multiscale problems. Comput. Methods Appl. Mech. Engrg. 267, 275–292. http://dx.doi.org/10.1016/j.cma.2013.08.009.

Feyel, F., 2003. A multilevel finite element method (FE2) to describe the response of highly non-linear structures using generalized continua. Comput. Methods Appl. Mech. Engrg. 192 (28), 3233–3244. http://dx.doi.org/10.1016/S0045-7825(03)00348-7, Multiscale Computational Mechanics for Materials and Structures.

Feyel, F., Chaboche, J.-L., 2000. FE2 multiscale approach for modelling the elasto-viscoplastic behaviour of long fibre SiC/Ti composite materials. Comput. Methods Appl. Mech. Engrg. 183 (3), 309–330. http://dx.doi.org/10.1016/S0045-7825(99)00224-8.

Ghavamian, F., Simone, A., 2019. Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network. Comput. Methods Appl. Mech. Engrg. 357, 112594. http://dx.doi.org/10.1016/j.cma.2019.112594.

Halabi, F.E., González, D., Chico, A., Doblaré, M., 2013. FE2 multiscale in linear elasticity based on parametrized microscale models using proper generalized decomposition. Comput. Methods Appl. Mech. Engrg. 257, 183–202. http://dx.doi.org/10.1016/j.cma.2013.01.011.

Hill, R., 1963. Elastic properties of reinforced solids: Some theoretical principles. J. Mech. Phys. Solids 11 (5), 357–372. http://dx.doi.org/10.1016/0022-5096(63)90036-X.

Kerfriden, P., Goury, O., Rabczuk, T., Bordas, S., 2013. A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics. Comput. Methods Appl. Mech. Engrg. 256, 169–188. http://dx.doi.org/10.1016/j.cma.2012.12.004.

Klein, D.K., Roth, F.J., Valizadeh, I., Weeger, O., 2023. Parametrised polyconvex hyperelasticity with physics-augmented neural networks. arXiv:2307.03463.

Li, Z., Li, X., Chen, Y., Zhang, C., 2023. A mechanics-informed machine learning approach for modeling the elastoplastic behavior of fiber-reinforced composites. Compos. Struct. 323, 117473. http://dx.doi.org/10.1016/j.compstruct.2023.117473.

Liu, Z., 2020. Deep material network with cohesive layers: Multi-stage training and interfacial failure analysis. Comput. Methods Appl. Mech. Engrg. 363, 112913. http://dx.doi.org/10.1016/j.cma.2020.112913.

Liu, Z., 2021. Cell division in deep material networks applied to multiscale strain localization modeling. Comput. Methods Appl. Mech. Engrg. 384, 113914. http://dx.doi.org/10.1016/j.cma.2021.113914.

Logarzo, H.J., Capuano, G., Rimoli, J.J., 2021. Smart constitutive laws: Inelastic homogenization through machine learning. Comput. Methods Appl. Mech. Engrg. 373, 113482. http://dx.doi.org/10.1016/j.cma.2020.113482.

Maia, M., Rocha, I., Kerfriden, P., van der Meer, F., 2023. Physically recurrent neural networks for path-dependent heterogeneous materials: Embedding constitutive models in a data-driven surrogate. Comput. Methods Appl. Mech. Engrg. 407, 115934. http://dx.doi.org/10.1016/j.cma.2023.115934.

Monteiro, E., Yvonnet, J., He, Q., 2008. Computational homogenization for nonlinear conduction in heterogeneous materials using model reduction. Comput. Mater. Sci. 42 (4), 704–712. http://dx.doi.org/10.1016/j.commatsci.2007.11.001.

Oliver, J., Caicedo, M., Huespe, A., Hernández, J., Roubin, E., 2017. Reduced order modeling strategies for computational multiscale fracture. Comput. Methods Appl. Mech. Engrg. 313, 560–595. http://dx.doi.org/10.1016/j.cma.2016.09.039.

Raissi, M., Perdikaris, P., Karniadakis, G., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707. http://dx.doi.org/10.1016/j.jcp.2018.10.045.

Rocha, I., Kerfriden, P., van der Meer, F., 2020. Micromechanics-based surrogate models for the response of composites: A critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks. Eur. J. Mech. A Solids 82, 103995. http://dx.doi.org/10.1016/j.euromechsol.2020.103995.

Rocha, I., Kerfriden, P., van der Meer, F., 2021. On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning. J. Comput. Phys.: X 9, 100083. http://dx.doi.org/10.1016/j.jcpx.2020.100083.

Rosenkranz, M., Kalina, K.A., Brummund, J., Sun, W., Kästner, M., 2024. Viscoelasticty with physics-augmented neural networks: Model formulation and training methods without prescribed internal variables. arXiv:2401.14270.

Schommartz, J.O., Klein, D.K., Cobo, J.C.A., Weeger, O., 2024. Physics-augmented neural networks for constitutive modeling of hyperelastic geometrically exact beams. arXiv:2407.00640.

Schröder, J., 2014. A numerical two-scale homogenization scheme: The FE2-method. Plast. Beyond http://dx.doi.org/10.1007/978-3-7091-1625-8_1.

Turon, A., Camanho, P., Costa, J., Dávila, C., 2006. A damage model for the simulation of delamination in advanced composites under variable-mode loading. Mech. Mater. 38 (11), 1072–1089. http://dx.doi.org/10.1016/j.mechmat.2005.10.003.

Turon, A., González, E., Sarrado, C., Guillamet, G., Maimí, P., 2018. Accurate simulation of delamination under mixed-mode loading using a cohesive model with a mode-dependent penalty stiffness. Compos. Struct. 184, 506–511. http://dx.doi.org/10.1016/j.compstruct.2017.10.017.

Wang, K., Sun, W., 2019. Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning. Comput. Methods Appl. Mech. Engrg. 346, 216–241. http://dx.doi.org/10.1016/j.cma.2018.11.026.