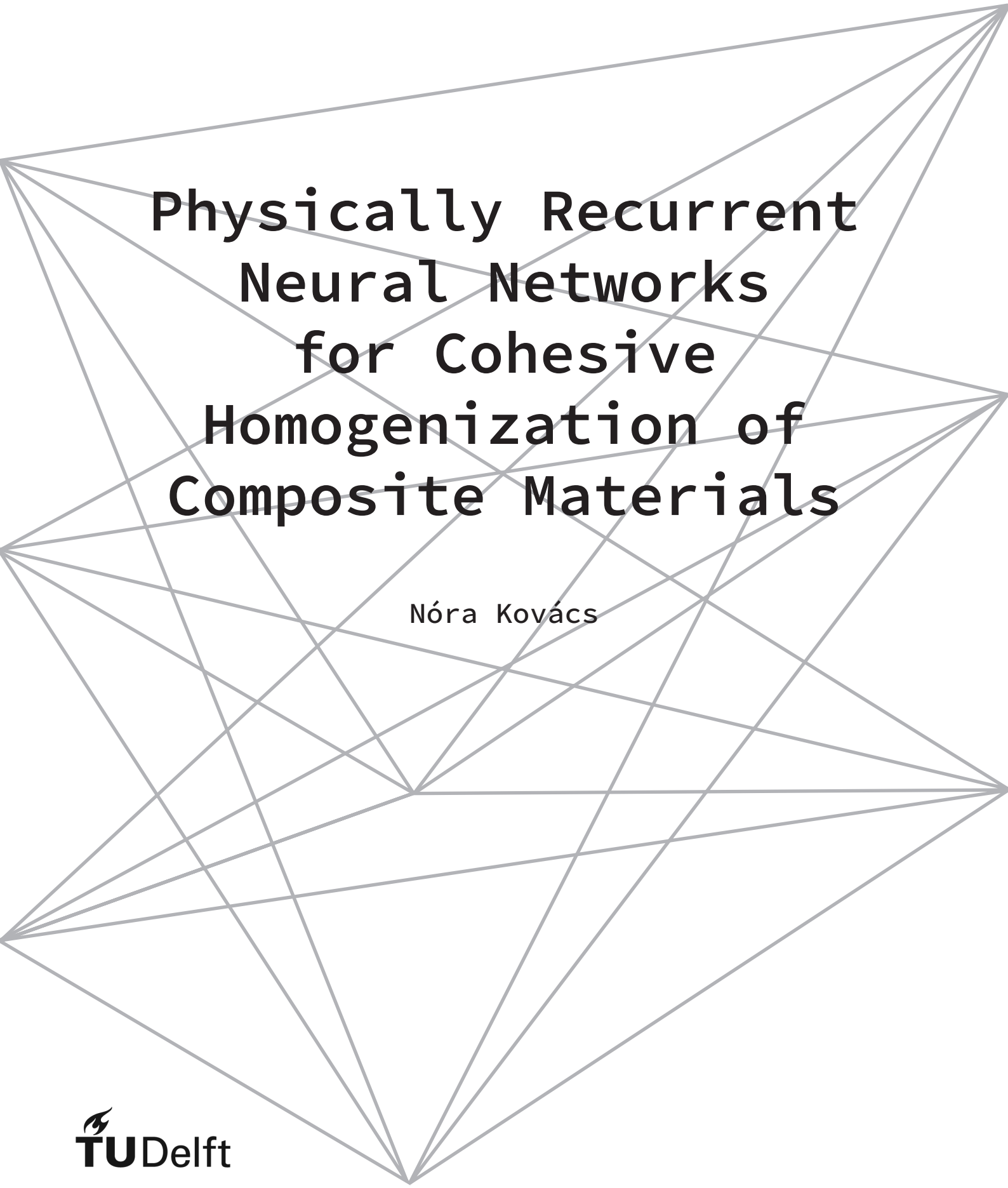


Physically Recurrent Neural Networks for Cohesive Homogenization of Composite Materials

Nóra Kovács

TU Delft

Physically Recurrent Neural Networks for Cohesive Homogenization of Composite Materials

by

Nóra Kovács

to obtain the degree of Master of Science
at the Delft University of Technology,
Faculty of Civil Engineering and Geosciences.

Student number: 5503698
Project duration: January, 2023 – September, 2023
Thesis committee:

Dr. F. P. van der Meer,	TU Delft, chair
Dr. I. B. C. M. Rocha,	TU Delft, supervisor
Dr. D. Zarouchas,	TU Delft, supervisor
ir. M. A. Maia,	TU Delft, daily supervisor

TU Delft

Acknowledgements

As my journey of being a master's student at TU Delft comes to an end, I would like to express my most sincere gratitude to my support system which helped me along the way.

First, I would like to thank my committee for their academic support and for giving me the opportunity to work on this exciting thesis topic. Dimitrios for providing great insight in the last stretch, and for joining the team on short notice. Frans, Iuri, and Marina for constantly pushing me to be the best researcher version of myself and for teaching me resilience. Trusting me to represent our team and show our work at CFRAC inspired me to continue my path in academia. Special thanks to Marina, whose constant support and feedback helped me greatly to keep me on track and to be more consistent.

I would like to express my gratitude to the ZJUI Institute for giving me a solid background in engineering that allowed me to succeed during my master's studies. I would like to thank all my professors from both UIUC and ZJU for inspiring me to pursue a graduate degree after my undergraduate studies.

I would like to thank:

The Computational Mechanics group for letting me sit with them. Seeing their research was a constant inspiration. I'd like to especially thank Suman, Til, and Fanxiang for celebrating the small victories with me.

The MUDE team for giving me the opportunity to experience being a student assistant. Especially Robert, for bearing with my constructive criticism.

The Tailored Materiality Research group led by Dr. Mariana Popescu, for peer support during thesis work. Sharing our progress each week and exchanging ideas were valuable times of the thesis period.

I would like to extend my gratitude to Mariana, whose summer project reinstated a passion for research in me. Her persistent enthusiasm for her exceptional research work is a great source of inspiration. Involving me in her research group not only created a sense of belonging to TU but also allowed me to have a bit of fun with knit.

I would like to express my gratitude to Dr. Florentia Kavoura, whose mentorship was a constant during my whole master's studies. Her invaluable academic guidance helped me establish short- and long-term goals for myself, and I learned how to focus on these during challenging times. I continuously draw inspiration from her, learning how to grow confidence within my work and the dedication I invest in it.

I could not have done any of this without the support of my friends, who became my second family over the past years. The coffee breaks, the brunches, the trips, and each and every activity with them made the time I spent here special. Many thanks to my roommates, old and new, who helped me through the past year. I am grateful to Paolo, Alberto, and Mathijs for their extra support during the final stages of my thesis. Special thanks to my friends who provided a home for me when in need. Amey, Firas, Andres, Anass, and Anders, I will never be able to express how grateful I am for you. I am incredibly thankful for my best friends who have been by my side through all of it, without whom I would not even be at TU Delft: Csilla, Nolan, Filippo, and last but certainly not least Camille. In the same country or not, you will always be my family.

Most importantly, I would like to express my gratitude to my family. To my brother Bálint, who always sets an exceptionally high bar for me and inspires me to keep pushing my limits. To my parents who provided me with everything I needed to follow my dreams and to be able to do what I am good at. Anya, Apa, nem tudom szavakba foglalni mennyire hálás vagyok nektek, nélkületek ez nem sikerült volna.

Thank you all for making this journey unforgettable.

Nóra Kovács
Delft, September 2023

Abstract

The growing use of composite materials in engineering applications accelerated the demand for computational methods, such as multiscale modeling, to accurately predict their behavior. Combining different materials with target mechanical properties helps achieve optimal structural performance. Nevertheless, the complex nature of composite materials poses several challenges. Current multiscale methods, such as the FE^2 method, are hindered by a computational bottleneck, limiting their widespread industrial adoption. Most of the existing surrogate modeling techniques that address this bottleneck are limited to predicting homogeneous materials or require an extensive dataset. The application of surrogate models in the fracture mechanics field is largely unexplored, where the existing models are highly convoluted.

The goal of this thesis is to apply surrogate modeling to predict cohesive damage in the fracture mechanics field. It focuses on one of the existing techniques using Physically Recurrent Neural Networks (PRNN). The core idea behind PRNNs is to implement the exact material models from the micromodel into the material layer of the network. The PRNN, which incorporates an elastoplastic model in what is referred to as bulk material points has resulted in exceptional performance when predicting elastoplastic behavior in composite materials. The primary objective of this thesis is to extend the existing PRNN to predict the effect of debonding at the fiber-matrix interface while capturing path-dependent behavior and minimizing the size of the training dataset with excellent extrapolation ability.

The fundamental capabilities of the existing PRNN with bulk material points only are evaluated in the microscale cohesive damage framework, particularly when interface elements are implemented at the fiber-matrix interface of the micromodel. This initial step reveals the limitations of the existing architecture and it becomes apparent that all types of nonlinearities present in the micromodel must also be implemented in the network.

This thesis extends the PRNN by incorporating a Cohesive Zone Model (CZM) within the existing material layer. This new architecture introduces cohesive integration points with the CZM along with the bulk integration points. Through model selection, various configurations of bulk and cohesive points are explored, along with different training dataset types and sizes, to maximize predictive accuracy and extrapolation capabilities. It is observed that training with non-monotonic data is required for the network to learn both types of nonlinearities. The limitations of the network's prediction are noted, which are due to the fact that its architecture does not represent the stress homogenization step of the multiscale method. This realization highlights the importance of the layout of the PRNN.

Further study investigates new PRNN architectures to improve the physical representation of the micromodel. The networks are trained on a single curve to select the optimal architecture. The most promising option is discussed in detail, in which the history parameter of the cohesive points is input to the bulk points. The network is proven to provide accurate prediction on a small training dataset when tested on the training dataset. Constraints of the PRNN are discussed and further improvements are recommended to extend the modified PRNN to a larger dataset.

This research contributes to the field of surrogate modeling for composite materials by investigating the predictive capabilities of the PRNNs and exploring new architectures. The results provide a promising outlook for accurately predicting the complex behavior of composite materials, specifically in the context of cohesive microscale damage considering debonding at the fiber-matrix interface. The proposed PRNN has the potential to increase computational efficiency of multiscale modeling in engineering applications.

Contents

1 Introduction **1**
1.1 Background and motivation . . . 1
1.2 Research objective and scope . . . 3
1.3 Methodology . . . 3
1.4 Thesis outline . . . 3

2 Theoretical framework **5**
2.1 The FE^2 method . . . 5
2.1.1 Macroscale problem formulation . . . 5
2.1.2 Microscale RVE . . . 6
2.1.3 Homogenization procedure . . . 7
2.2 Feed-forward neural networks . . . 7
2.3 Recurrent neural networks . . . 9
2.4 Physically recurrent neural network . . . 10

3 Data generation **13**
3.1 Full-order micromodel . . . 13
3.2 Load path generation . . . 15
3.2.1 Proportional loading . . . 16
3.2.2 Non-proportional loading . . . 17

4 PRNN with bulk models only **19**
4.1 Training with monotonic and proportional data . . . 20
4.2 Training with non-monotonic and non-proportional data . . . 22

5 Extending the PRNN with cohesive material points **25**
5.1 Training with monotonic and proportional data . . . 26
5.2 Training with non-monotonic and non-proportional data . . . 29
5.2.1 Training size selection . . . 29
5.2.2 Varying the material layer configuration . . . 30
5.2.3 Modified model with lower K . . . 33

6 Modified architectures **35**
6.1 Bulk and cohesive points in the same material layer . . . 35
6.2 Input to bulk points from cohesive points . . . 36
6.2.1 Tractions as input to bulk points . . . 36
6.2.2 Damage as input to bulk points . . . 37
6.2.3 Damage as input to bulk points with a hidden layer before cohesive points . . . 42
6.2.4 Damage as input to bulk points with Leaky ReLU at cohesive points . . . 43

7 Conclusions and recommendations **47**
7.1 Conclusions . . . 47
7.2 Limitations and recommendations . . . 49

Bibliography **51**

1 Introduction

1.1. Background and motivation

The use of composite materials as structural elements has been increasing over the years due to their superior qualities. The overall idea behind designing composite materials is to achieve a material with better mechanical performance than what its components individually could obtain. For example, fiber-reinforced polymers (FRPs) used in aircraft structures contribute to a significant weight-reduction [1]. In the civil engineering context, using composites enables tailoring the materials to obtain desired mechanical properties for a target structural performance [2].

Combining different materials into a heterogeneous material while utilizing them in the most optimal way possible has clear advantages. Obtaining these desired mechanical properties requires testing the composite materials. While modeling techniques mitigate the costs of experimental testing, obtaining a global response from these composite materials is challenging. These challenges arise from the complex nature of these composite materials. Additionally, it is difficult to capture the non-linear behavior of the material on multiple scales. Consider the case of laminates, where a macroscale model is required to model the overall response of the structure. However, the overall response depends on the plies that the composite consists of, such as their orientation and thickness, which can be modeled in mesoscale. The behavior of the piles is dependent on the fiber-to-matrix ratio and their exact material properties for instance, which are modeled in microscale.

An appealing but expensive computational solution to overcome the difficulties posed by the heterogeneous nature of the composite materials is the use of multiscale modeling. In this method, the structure is modeled in different scales. Multiscale modeling methods can be categorized into two approaches based on the connection between the scales: the hierarchical and the concurrent multiscale method [3]. In the hierarchical approach, the model is evaluated on the different scales separately, one at a time. On the contrary, concurrent models incorporate the different scales into one model, and the scales are evaluated simultaneously. The concurrent method is able to overcome the difficulties posed by strain localization that the hierarchical method cannot, which makes it a more advantageous method.

A specific case of concurrent multiscale modeling method is the FE^2 method. In this method, a given structure is analyzed on two scales: on micro- and macroscale [3]. On the macroscale, the composite structure is modeled as a homogeneous structure with a micromodel nested at each of its material points. This heterogeneous micromodel contains all the material complexities along with the constitutive relations and is referred to as the representative volume element (RVE). In the FE^2 framework, scale transitions between micro- and macroscale are established. For the RVE, a microscale boundary value problem (BVP) is constructed which provides the transition from macro- to microscale. This is referred to as downscaling. After the full-order solution is evaluated on the microscale, an upscaling process takes place, in which the microscopic stresses are homogenized to obtain a macroscopic stress value. The FE^2 method is an appealing computational method since it bypasses directly computing a complex macroscale constitutive relation between the stresses and strains. However, the downscaling process, computing the full-order solution for the RVE, and the upscaling process are repeated for each integration point of the macrostructure, which is a rigorous process and has a high computational cost.

There are several methods proposed to alleviate the computational bottleneck associated with the FE^2 method. One way is to reduce the computational time required to solve the boundary value problem in the RVE by using dimensionality reduction techniques. Proper orthogonal decomposition (POD) [4] and proper generalized decomposition (PGD) [5, 6] are both techniques that create a reduced-order model of the RVE to solve the micromodel in a low-dimension space. POD and PGD are effectively reducing the computational time, however, these techniques can still involve a large number of parameters. To overcome this problem, a new approach is emerging in the field which involves surrogate models replacing the full-order microscale BVP. For instance in [7] and [8], the full-order micromodel is replaced by a recurrent neural network (RNN) at each integration point. The network predicts the microscale constitutive model while storing history parameters to account for the path-dependency of composite materials. A drawback of this method can be related to the black-box nature of these RNNs. Since the network only sees the input strain and output stress values during training with no physical interpretation, it is not able to extrapolate to unseen loading scenarios. This means that a large amount of data is required to train the network to cover the loading space. To improve on the extrapolation capabilities of RNNs, a portion of the literature focuses on using physics-informed neural networks (PINNs) as surrogate models [9, 10]. In PINNs, physical constraints are implemented in the loss function of the network. In [9] and [10] the accuracy of the PINNs applied for elastoplastic materials is showcased while allowing for a more accurate extrapolation based on the actual material properties. A different approach was presented in a recent work using mechanics-informed machine learning [11]. In this approach, a priori knowledge of mechanics was implemented in the network to predict elastoplasticity for composites with pressure-independent yielding behavior by mechanics-based decomposition of stresses and strains. The network showed accurate predictions trained on a small dataset.

Another novel approach applied for composites with elastoplastic behavior showed great accuracy when trained on a small dataset by using physically recurrent neural networks (PRNNs) [12]. 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. The performance of the PRNN was evaluated on a microsctructure and compared to the performance of a Bayesian Recurrent Neural Network (RNN) [12]. Trained on monotonic data, the PRNN was able to predict with the same accuracy as the RNN, but with one sixteenth of the training data required for the RNN. Moreover, the PRNN was able to extrapolate to non-monotonic test data even though it only saw monotonic training data. This level of accuracy could be achieved by the RNN only when adding non-monotonic data during training and with a much larger dataset. One of the trained PRNNs was applied within the FE^2 framework on a macrostructure, and it was observed that replacing the full-order solution with the network resulted in a reduction of computational time by more than 20,000 times.

Although an increasing number of research has been devoted to modeling the behavior of heterogeneous materials with path-dependency in the surrogate modeling field, reducing model complexity while keeping the training data size small is still quite a challenging endeavor. For instance, the RNNs applied in the FE^2 framework discussed above require a large sampling effort when it comes to predicting path-dependent behavior. Moreover, a large part of the literature focuses on predicting elastic or elastoplastic behavior. In fracture mechanics, surrogate modeling is even more of an unexplored field. For example, surrogate models were used to create traction-separation laws using deep reinforcement learning [13], though this work does not attempt to accelerate multiscale modeling. In the works of Liu [14, 15], 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 [14], debonding effects in the RVE were successfully captured by the adaptable cohesive building blocks included in the network, where a multi-stage training strategy was explored. In [15] 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 and the length-scale. A probabilistic machine learning approach using Bayesian regression was proposed in [16], in which full-order models are nested in anchor material points of the structure. These anchor models are fully solved and used for training the models online, bypassing the offline training stage.

Another solution to reduce computational time in the FE^2 framework applied in fracture mechanics is to implement model order reduction with a domain separation strategy [17, 18]. In these works, a domain separation strategy is introduced to focus the computational power on the fracture, which requires most of the attention.

These innovative approaches reduce the high computational cost associated with the repeated evaluation of the full-order micromodel at the integration points. However, a critical gap persists in the existing literature on surrogate modeling in the fracture mechanics field. Although advanced methods are capable of capturing elastoplastic behavior, they do not incorporate softening constitutive laws required for modeling cohesive damage, and the works that do incorporate them are highly convoluted such as the DMNs. Along this line, the physically recurrent neural networks introduced in [12] showcase great potential in capturing the underlying physics of material behavior. Therefore, it is interesting to investigate the capabilities of PRNNs in the fracture mechanics field.

There are different ways of incorporating softening behavior of composites using interface elements with a cohesive zone model (CZM). When debonding between fibers and matrix is considered, these interface elements are introduced at the fiber-matrix interface in the micromodel. When strain localization is considered, these interface elements can be implemented at a predefined crack path, or distributed along the whole micromodel to allow for an arbitrary crack path to form.

1.2. Research objective and scope

In order to address the computational challenge posed in Section 1.1, the following research objective was established:

The goal of this research is to design a surrogate model based on the current version of physically recurrent neural network [12] that is able to predict cohesive damage of composite materials. To achieve this, the PRNN should ideally meet the following criteria:

- is able to capture path-dependency
- does not require a large training set
- extrapolates accurately to unseen loading scenarios.

This thesis is limited to the case of accounting for debonding at the fiber-matrix interface is considered. Application for strain localization is out of the scope of this work.

1.3. Methodology

The training, validation, and test dataset for the PRNNs are created to be able to evaluate their performance. This consists of generating the load paths and applying the loading on the full-order FE^2 micromodel implemented in the in-house code using the Jem/Jive library of C++. The high-fidelity solution to which the predictions of the networks are compared is obtained and the different datasets are created.

The parameter study conducted on the existing network developed in [12], implemented in the Jem/Jive library of C++, is done by an external Python file defining the values of the parameters which allows for modification of the PRNN. The implementation of the cohesive zone model in the material layer of the existing PRNN is done by modifying the in-house code in Jem/Jive C++. For the modified architectures of the PRNN, the PyTorch library of Python is used for ease of implementation. The cohesive zone model written in C++ is converted to Python, and custom layers are created to accommodate the changes in network layout. For all visualization of results, Python is used.

For each PRNN configuration, 10 initializations are created during training to account for variations in network performance. The variables that are changed during the study for the different architectures consist of the size of the training data and the number of fictitious material points with plasticity and/or with the cohesive zone model. During training, stochastic gradient descent is used with Adam optimizer.

1.4. Thesis outline

First, a theoretical background is presented for multiscale modeling and neural networks in Chapter 2, with more emphasis on the physically recurrent neural network. [12]. In Chapter 3, a description is given of the full-order micromodel, and the material models used in this thesis. Additionally, the data generation method for training and testing the networks is described. Chapters 4-6 address the performance of the PRNNs proposed to achieve the research objective. In Chapter 4, a brief parameter study is conducted on the PRNN in its current state [12]. In Chapter 5, the architecture of the existing network is kept but enhanced with cohesive material points. Based on the conclusions drawn from preceding chapters, new architectures are proposed in Chapter 6. Conclusions are drawn and recommendations are established in Chapter 7.

2

Theoretical framework

This chapter is dedicated to discussing the theoretical background behind this thesis work. First, a detailed description of the FE^2 method is presented. Then, the use of surrogate models to accelerate computational time caused by the bottleneck associated with the FE^2 method is discussed. Particularly, a brief introduction to neural networks is presented followed by a detailed description of the physically recurrent neural network, and its main difference from standard, recurrent, and physics-informed neural networks is highlighted.

2.1. The FE^2 method

The FE^2 method is a popular concurrent multiscale finite element analysis used to obtain a homogenized response of heterogeneous materials. In this method, the structure is discretized into a homogeneous macrostructure, and a full-order micromodel is nested into each integration point of it. This full-order micromodel is called a representative volume element (RVE), and it includes all material heterogeneity. The macroscopic strain values are downscaled to microscopic strain values for the full-order 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, where equilibrium is solved.

The schematics of the FE^2 method is shown in Figure 2.1. 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. 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 following key elements of FE^2 applied to fracture mechanics are discussed in this chapter: the macroscale problem formulation, the microscale RVE problem formulation along with the boundary conditions received from the macroscale, and the micro- to macroscale transition.

2.1.1. Macroscale problem formulation

The macroscopic BVP, without body forces of the macrostructure, is described by the following equilibrium equations:

$$\nabla \cdot \boldsymbol{\sigma}^\Omega = \mathbf{0}, \tag{2.1}$$

where ∇ is the divergence operator, $\boldsymbol{\sigma}^\Omega$ are the macroscale stresses. The Dirichlet and Neumann boundary conditions are given in Eqs. (2.2) and (2.3), respectively.

$$\boldsymbol{\sigma}^\Omega \mathbf{n} = \mathbf{t}^{\Gamma_f} \quad \text{on} \quad \Gamma_f^\Omega \tag{2.2}$$

$$\mathbf{u}^\Omega = \mathbf{u}^{\Gamma_u} \quad \text{on} \quad \Gamma_u^\Omega \tag{2.3}$$

The prescribed traction forces and displacements are denoted by $\mathbf{t}^{\Gamma_f}$ and $\mathbf{u}^{\Gamma_u}$, respectively, and $\mathbf{n}$ is the unit outward normal vector to the boundary. Since no macroscopic cracks are allowed in the

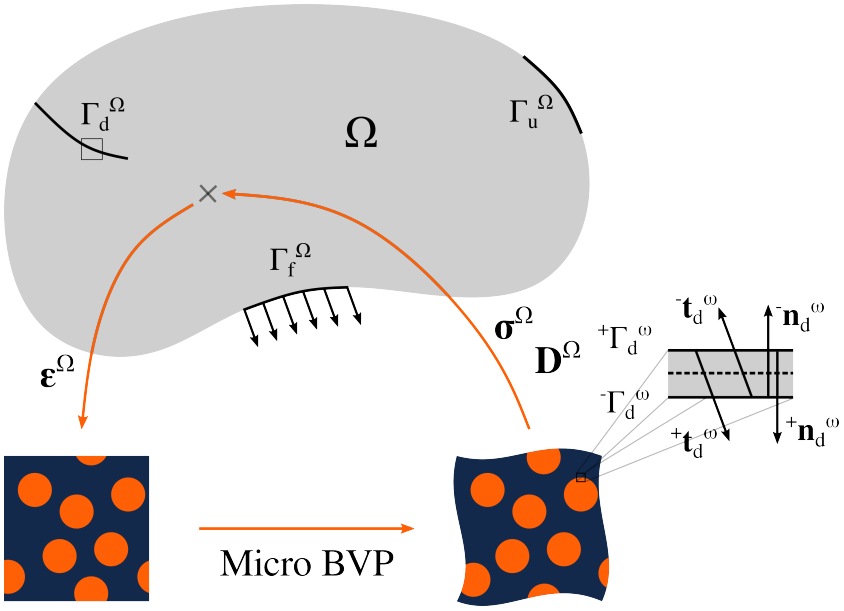


Figure 2.1: FE^2 framework

applications considered in this work, the formulation of macroscale strains in the continuous domain can be described by:

$$\boldsymbol{\varepsilon}^\Omega = \frac{1}{2}\left(\nabla \mathbf{u}^\Omega + \left(\nabla \mathbf{u}^\Omega\right)^{\mathrm{T}}\right) \tag{2.4}$$

Next, the constitutive law is defined to relate strains and stresses:

$$\boldsymbol{\sigma}^\Omega = \boldsymbol{\sigma}^\Omega\left(\boldsymbol{\varepsilon}^\Omega, \boldsymbol{\alpha}\right) \tag{2.5}$$

where $\boldsymbol{\alpha}$ are internal variables storing the previous states.

Note, however, that the constitutive relations on the macro level are not defined explicitly, due to the complexity of the material. This is why full-order micromodels are nested at the integration points, which bypasses the need for such model.

2.1.2. Microscale RVE

The macroscale strains at the material points are passed to the RVE, where they are used as boundary conditions. The displacement field in the micromodel with domain denoted as ω can be computed using these strains:

$$\mathbf{u}^\omega = \boldsymbol{\varepsilon}^\Omega \cdot \mathbf{x}^\omega + \tilde{\mathbf{u}}^\omega, \tag{2.6}$$

where $\tilde{\mathbf{u}}^\omega$ is the displacement fluctuation, which vanishes if proper boundary conditions are applied to the RVE. The the macroscale strain is considered to be constant over the volume due to the assumption of separation of scales, and it can be computed by:

$$\boldsymbol{\varepsilon}^\Omega = \frac{1}{\|\omega\|}\int_\omega \boldsymbol{\varepsilon}^\omega \, d\omega \tag{2.7}$$

On the lower scale, the following equilibrium equation needs to be solved with $\boldsymbol{\sigma}^\omega$ being the microscale stresses:

$$\nabla \cdot \boldsymbol{\sigma}^\omega = \mathbf{0} \tag{2.8}$$

The additional boundary conditions posed by the discontinuity along the fiber-matrix interface can be described with equations:

$$ {}^{+}\mathbf{t}_d^{\omega} + {}^{-}\mathbf{t}_d^{\omega} = \mathbf{0}; \quad \boldsymbol{\sigma}_d^{\omega}\mathbf{n}_d = {}^{-}\mathbf{t}_d^{\omega} \quad \text{on} \quad \Gamma_d^{\omega}, \tag{2.9} $$

where Γ_d^{ω} is the fracture surface at the fiber-matrix interface, ${}^{+}\mathbf{t}_d^{\omega}$ and ${}^{-}\mathbf{t}_d^{\omega}$ are the traction forces on the opposite sides of the microcrack, $\mathbf{n}_d$ is the unit outward normal vector to the fracture surface and σ_d^{ω} is the stress in the bulk point at the fracture surface.

As mentioned before, the exact constitutive models are included directly in the RVE. These are expressed as the following:

$$ \boldsymbol{\sigma}^{\omega} = \boldsymbol{\sigma}^{\omega}\left(\boldsymbol{\varepsilon}^{\omega}, \boldsymbol{\alpha}\right) \tag{2.10} $$

$$ \mathbf{t}_d^{\omega} = \mathbf{t}_d^{\omega}\left(\llbracket \mathbf{u}^{\omega} \rrbracket, \mathbf{d}\right), \tag{2.11} $$

where $\mathbf{t}_d^{\omega}$ is the cohesive traction computed from the displacement jump $\llbracket \mathbf{u}^{\omega} \rrbracket$ and internal variables $\mathbf{d}$. The displacement jumps are also interpolated from the nodal values in each interface element and contribute to the global stiffness matrix of the micromodel as:

$$ \boldsymbol{K}_{\llbracket \mathbf{u}^{\omega} \rrbracket} = \int_{\gamma_i} \mathbf{N}_{\llbracket \mathbf{u}^{\omega} \rrbracket}^{T} \mathbf{T} \mathbf{N}_{\llbracket \mathbf{u}^{\omega} \rrbracket} \, d\gamma_i \tag{2.12} $$

where γ_i represents the interface surfaces, $\mathbf{N}_{\llbracket \mathbf{u}^{\omega} \rrbracket}$ is the shape function matrix, and $\mathbf{T}$ refers to the tangent constitutive matrix of the interface element given by:

$$ \mathbf{T} = \frac{\partial \mathbf{t}}{\partial \llbracket \mathbf{u} \rrbracket} \tag{2.13} $$

As for the regular bulk models, the tangent constitutive matrix is given by:

$$ \mathbf{D}_{\mathbf{u}} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} \tag{2.14} $$

and their contributions to the global tangent stiffness matrix of the micromodel follows:

$$ \mathbf{K}_{\mathbf{u}} = \int_{\Omega} \mathbf{B}_{\mathbf{u}}^{T} \mathbf{D}_{\mathbf{u}} \mathbf{B}_{\mathbf{u}} \, d\Omega \tag{2.15} $$

where $\mathbf{B}_{\mathbf{u}}$ is the matrix containing the derivatives of shape functions.

Section 3.1 provides a brief overview of the material models used in this thesis.

2.1.3. Homogenization procedure

After the exact computation of the full-order solution in the micro-scale, the obtained stress values across the RVE have to go through an averaging procedure called homogenization, so that a final stress value can be sent back to the macroscale model. The stress homogenization is obtained by integrating over the volume of the RVE:

$$ \boldsymbol{\sigma}^{\Omega} = \frac{1}{||\omega||} \int_{\omega} \boldsymbol{\sigma}^{\omega} \, d\omega \tag{2.16} $$

For accurate coupling between the two scales, the energy between them must be consistent. For this, the Hill-Mandel principle has to be satisfied [19].

After the macroscale model receives the stress values at each of the material points of the structure, global equilibrium can be solved by a chosen computational method (e.g. Newton-Raphson method).

2.2. Feed-forward neural networks

To understand the concept behind the physically recurrent neural network used in this thesis to accelerate the computational time of multiscale modeling, first, an introduction to neural networks is presented. The simplest type are the feed-forward neural networks, which are a popular foundation for machine learning and are widely used in pattern recognition. The goal of neural networks is to create a computational framework that is able to learn a certain pattern/behavior given observed data and is able to utilize this information to generalize and make new predictions at a lower computational cost. It attempts to simulate the act of learning in biological systems, such as the human brain, in which neurons are connected by axons, resulting in a framework that is able to process information [20].

A feed-forward neural network is built up by an input layer, hidden layers, and an output layer. To illustrate that, Figure 2.3 shows a particular architecture choice with 2 hidden layers. Each layer consists of neurons that are connected by axons to the neurons in neighboring layers, with a weight and a bias parameter corresponding to each axon. The input values $\mathbf{v}$ to the neurons in layer l are a linear combination of the values of the neurons in the previous layer, and are computed by using these weight and bias parameters ($\mathbf{W}$ and $\mathbf{b}$):

$$\mathbf{v}_l = \mathbf{W}_l \mathbf{a}_{l-1} + \mathbf{b}_l \tag{2.17}$$

where in each neuron, an activation function ϕ is applied, introducing non-linearity into the network:

$$\mathbf{a}_l = \phi(\mathbf{v}_l) = \phi(\mathbf{W}_l \mathbf{a}_{l-1} + \mathbf{b}_l) \tag{2.18}$$

This gives the output $\mathbf{a}_l$ of a layer l. The activation functions considered in this work are the Linear, the SoftPlus, the Tanh, and the Leaky ReLU activation functions, and they are shown in Figure 2.2 when applied to arbitrary values.

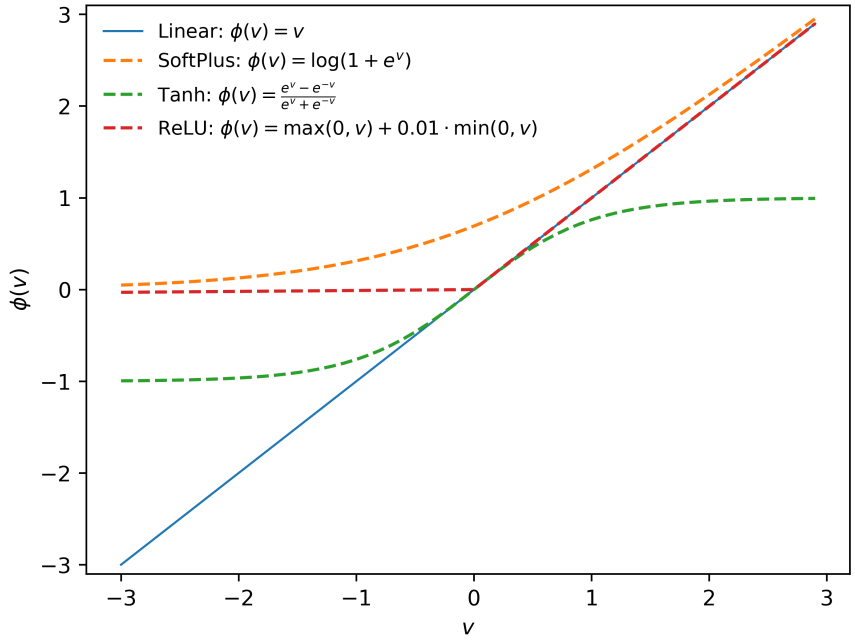


Figure 2.2: Activation functions used in this work

The parameters of the network are initialized, and updated during training. The objective of training the network is to find the optimal set of parameters that give the maximum likelihood of the training data. To obtain these optimal values, forward and backward passes are required. During the forward pass, Eq. (2.17) is evaluated for a given batch for every layer until the final predictions are obtained at the output layer. The fit of the prediction on the data is evaluated by computing an error function, for example, the mean squared error (MSE):

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^{N} ||\mathbf{y}(\mathbf{x_n}) - \hat{\mathbf{y}}(\mathbf{x_n})||^2, \tag{2.19}$$

where $\mathbf{y}(\mathbf{x_n})$ is the true target value, and $\hat{\mathbf{y}}(\mathbf{x_n})$ is the prediction of the network.

This loss function is minimized during training by modifying the parameters. This is done during the backward pass by using gradient information on the parameters. Detailed descriptions and derivations on gradient descent optimization and error backpropagation can be found in [21]. In the standard case, one epoch contains one forward and one backward pass. To accelerate learning, minibatching can be applied. When minibatching, the dataset is grouped into smaller batches that are passed forward and backward at the same time, leading to a more frequent update of the parameters.

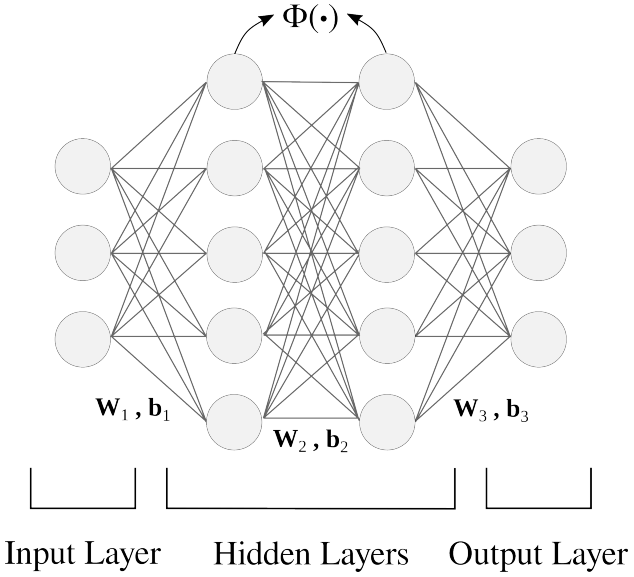


Figure 2.3: Feed-forward neural network

The training continues until a maximum number of iterations is reached, or when no significant decrease in the training error is observed after a fixed set of epochs, thus a local minimum is obtained. The model parameters are then stored, and the network is ready to predict unseen data. On an important note, the network can lose its capacity to obtain a generalized response as it tries to fit its predictions too closely, which is commonly known as overfitting. To avoid this, a validation set is often considered, which consists of a certain portion of the data selected randomly prior to training. The error is computed on this validation set and while minimizing the training error, the model is saved only if the validation error is smaller than the historical best.

In the case of the simple feed-forward neural network, the only input to the layers is the linear combination of nonlinear functions of the values in the previous layer. This becomes an issue when predicting the response of materials with nonlinear behavior, where the stress state of the material point depends on the loading history. Considering elastoplastic behavior, once the material undergoes plastic deformation, its response not only depends on the current load but also on its prior plastic deformation. During cyclic loading, there are multiple stress states corresponding to a strain level. The setup of the network does not allow for it to keep track of its previous state and, therefore cannot account for path-dependency. For that matter, recurrence can be added to the networks which are discussed in the next section.

2.3. Recurrent neural networks

In order to use the networks as surrogate models in the FE^2 framework for path-dependent heterogeneous materials, they have to be capable of keeping track of previous states of the material. This could be done by using recurrent neural networks (RNNs), in which the state of a layer depends not only on its input but also on its previous state. The previous state of the neuron is stored in a history variable, which also has corresponding parameters to optimize. The current hidden state is calculated by:

$$\mathbf{h}^t = \phi(\mathbf{W}_{hh}\mathbf{h}^{t-1} + \mathbf{W}_{vh}\mathbf{v}^t + \mathbf{b}_h) \tag{2.20}$$

This relation is represented by Figure 2.4. The hidden state is then used to calculate the output of the neuron:

$$\mathbf{a}^t = \phi(\mathbf{W}_{ha}\mathbf{h}^t + \mathbf{b}_a) \tag{2.21}$$

Although improvements were made to this architecture in order to account for long-term memory [22], the ability of RNNs to extrapolate to unseen scenarios is limited. In the context of the FE^2 framework, a large amount of data is required to cover the possible loading scenarios [12].

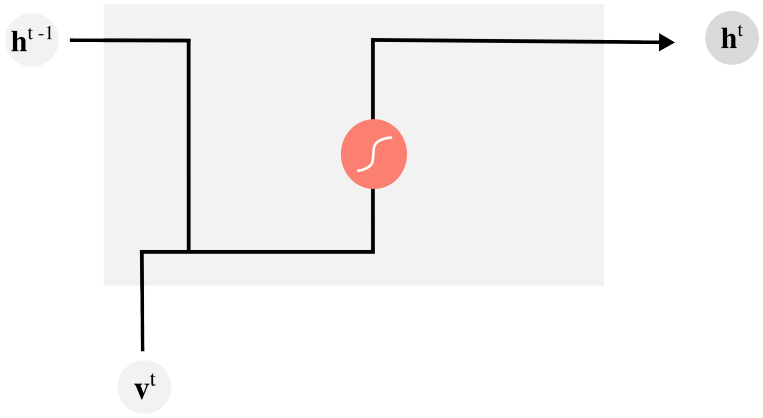


Figure 2.4: Simple recurrent neural network neuron[12]

When applied as a surrogate model at the integration points of the macrostructure, the network can be viewed as a computational tool that inputs the observed strain values into a black box and outputs the approximation of the stress values. An important thing to note is that this black box of hidden layers has no physical meaning behind it.

2.4. 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. [12] introduce a new way of implementing the physical properties present in the microstructure. Unlike in PINNs, where the physical constraints of the problem are incorporated in the loss function [23], in PRNNs the actual material models used in the full-order solution are implemented in the hidden layer of the network. Figure 2.5 displays the PRNN in general terms and with the simplest architecture studied in the reference work.

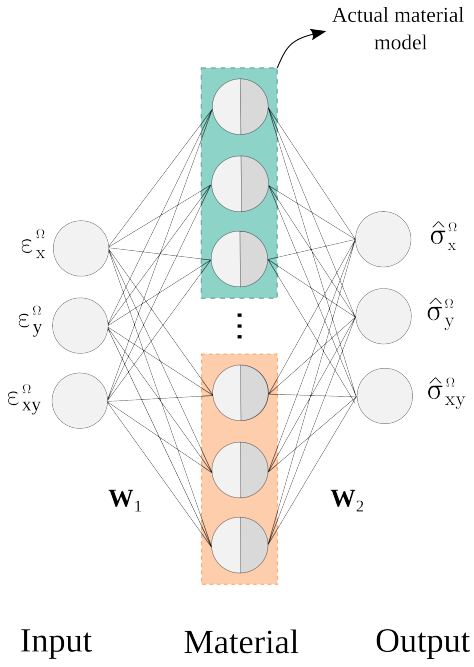


Figure 2.5: Physically recurrent neural network [12]

The architecture consists of an input, a material, 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 layer, which is a dense layer with linear activation functions. This encoder converts

the macroscale strains to microscale strains, which corresponds to the macro- to micro-scale transition in the FE^2 method. Combining the Linear activation function with Eq. (2.17), the microscale strains $\boldsymbol{\varepsilon}^{\omega}$ are then given by:

$$\boldsymbol{\varepsilon}^{\omega} = \mathbf{W}_1 \boldsymbol{\varepsilon}^{\Omega} + \mathbf{b}_1, \tag{2.22}$$

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 $\mathbf{b_1} = \mathbf{0}$, which ensures zero microscale strain inputs to the material models for zero macroscale strain values.

Then, these microscale strains are passed through the material layer, which provides the essence of the physically recurrent neural network. The material layer consists of nodes grouped in threes (in the two-dimensional case), and each of these groups represents a fictitious integration point. In these points, a constitutive model D^{ω} is nested which converts the microscale strains $\boldsymbol{\varepsilon}^{\omega}$ to microscale stresses $\boldsymbol{\sigma}^{\omega}$. This constitutive model is the exact same model in the full-order micromodel. This way, the black box idea of the hidden layer is deciphered, and true physical meaning is given to the layer.

As discussed in the previous sections, an ideal network keeps track of its state in previous time steps in order to be able to predict path-dependent materials. This is done in the PRNN by storing the internal variables $\boldsymbol{\alpha}$ of each material point, which for example in plasticity can be plastic deformation, in a history vector ($\mathbf{h}^t$) as they are computed in the assigned constitutive model. Therefore natural path-dependency arises. This stands in contrast with regular recurrent networks, where the history variables are also learned through additional learnable parameters and standard activation functions (e.g. Tanh and Sigmoid). The operation in the neurons at the fictitious material points can be described by Eq. (2.23) and is shown in Figure 2.6.

$$\boldsymbol{\sigma}^{\omega}, \mathbf{h} = D^{\omega}(\boldsymbol{\varepsilon}^{\omega}, \mathbf{h}^{t-1}) \tag{2.23}$$

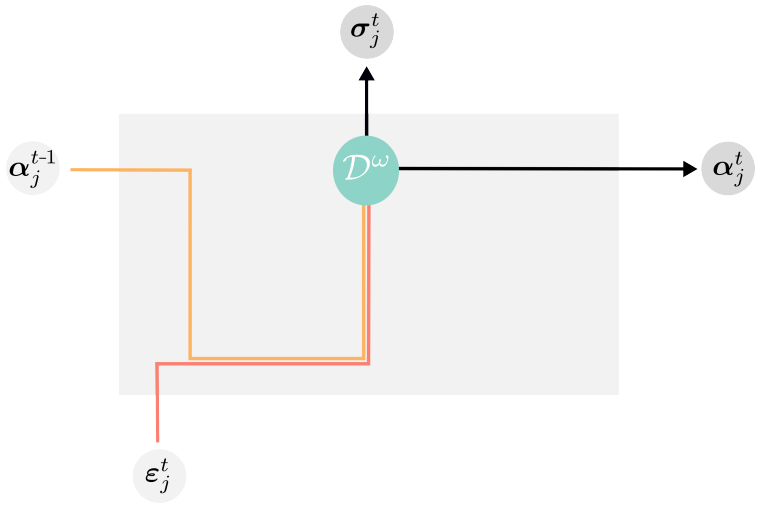


Figure 2.6: Neuron in material layer of physically recurrent neural network [12]

After the microscale stresses are computed in the material layer, these stress values are passed through a decoder. This step corresponds to the stress homogenization step of the FE^2 method. Essentially, the macro-scale response of the microstructure is a combination of the fictitious material points present in the material layer. In the particular achitecture shown in Fig. 2.5, the decoder consists of a dense layer with a SoftPlus activation function applied on the weights only. This is done to represent the homogenization process through numerical integration, in which weights are strictly positive. The macroscale stress output of the network is obtained by combining the Softplus activation function and Eq. (2.17):

$$\hat{\boldsymbol{\sigma}}^{\Omega} = \phi_{sp}(\mathbf{W}_2)\boldsymbol{\sigma}^{\omega} + \mathbf{b}_2, \tag{2.24}$$

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 however is also neglected to ensure zero macroscale stresses for zero microscale stress values, as a stress-free state is assumed at zero strain level.

During training, the following loss function is minimized:

$$\mathrm{L} = \frac{1}{N}\sum_{t=1}^{N} ||\boldsymbol{\sigma}^{\Omega}(\boldsymbol{\varepsilon}_t^{\Omega}) - \hat{\boldsymbol{\sigma}}^{\Omega}(\boldsymbol{\varepsilon}_t^{\Omega})||^2, \tag{2.25}$$

where N is the number of timesteps during loading and $\boldsymbol{\sigma}^{\Omega}(\boldsymbol{\varepsilon}_t^{\Omega})$ is the target value. Stochastic gradient descent is used with Adam optimizer.

The full-order micromodel used in [12] is shown in Figure 2.7, where J2 plasticity model was used for the matrix and a linear elasticity model for the fibers. It is important to highlight a conclusion drawn in that 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. This was possible due to the presence of the linear elastic behavior expected by the fiber's model when small enough strain values are passed by the encoder and stresses are amplified in the decoder, making the matrix model effectively work like the linear elastic model of the fiber.

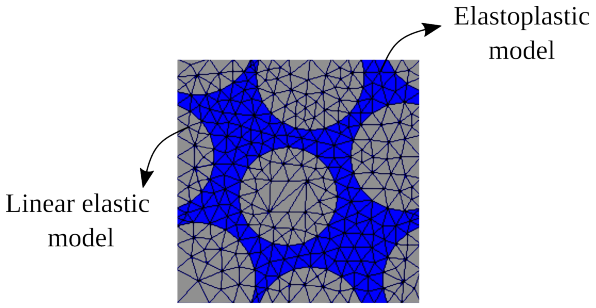


Figure 2.7: Full-order micromodel used in [12]

In the following chapters, the performance of the PRNN and potential modifications are evaluated in the fracture mechanics context. The next chapter presents the full-order micromodel with interface elements used in the FE^2 method. This micromodel is subjected to different loading scenarios to obtain high-fidelity solutions used to train and assess the network's performance. The loading path generation is discussed in the next chapter as well.

3

Data generation

This thesis work is dedicated to predicting microscale damage using the PRNN [12]. To generate the data required for assessing its performance, the full-order micromodel considered in the reference work is modified to include a cohesive zone model (CZM) at the fiber-matrix interface to allow for debonding to take place. This chapter focuses on introducing the micromodel used to create the training and test datasets, the material models that are included, and different loading conditions considered.

3.1. Full-order micromodel

The micromodel considered in this work consists of a bulk matrix with fibers as inclusions and interface elements at the boundary of the two constituents. In the FE model of the micromodel shown in Figure 3.1. There are two types of bulk points: the ones corresponding to the matrix which are described by a plasticity material model, and the ones corresponding to the fibers which are described by a linear elastic material model. Finally, the cohesive points are described by a cohesive zone model that accounts for microscale damage of the material. Limiting damage to the fiber-matrix interface means that no global failure can take place. The material properties of the bulk points are kept as in [12].

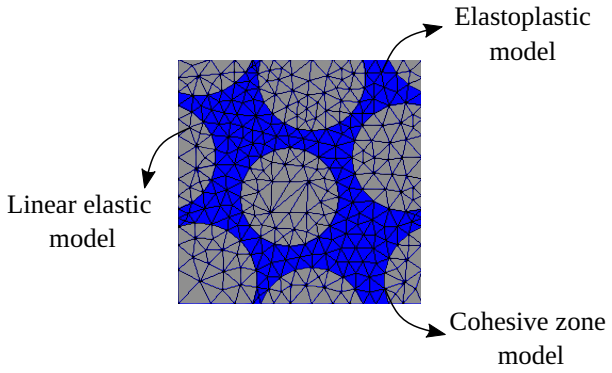


Figure 3.1: Full-order micromodel used in this work

The linear elastic fibers have the following properties:

$$E = 74000 \quad \text{MPa} \tag{3.1}$$

$$\nu = 0.2 \tag{3.2}$$

J2 plasticity was used for describing the matrix material and is commonly known as von Mises plasticity. The von Mises stress σ_{vm} is defined as:

$$\sigma_{\text{vm}} = \sqrt{\frac{3}{2}\text{tr}(\boldsymbol{\sigma}_d^2)} = \sqrt{3\text{J}_2}, \tag{3.3}$$

with $\mathrm{tr}(\boldsymbol{\sigma}_d^2)$ being the trace of the deviatoric stress tensor and J_2 a deviatoric stress invariant, which the J2 model got its name from. The yield function $f(\sigma, \kappa)$ is given by

$$f(\sigma, \kappa) = \sigma_{\text{vm}} - \sigma_y(\kappa), \tag{3.4}$$

which states that the material enters plastic deformation after a certain yield stress $\sigma_y(\kappa)$ is reached. The yield stress depends on the hardening parameter κ, which drives the hardening (or softening) of the material. The plastic deformation progress $\dot{\boldsymbol{\varepsilon}}^{\boldsymbol{p}}$ is defined by the flow rule:

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda}\mathbf{m}, \tag{3.5}$$

where $\dot{\lambda}$ is the plastic multiplier that represents the magnitude of the plastic flow, and $\mathbf{m}$ represents the direction of the plastic flow.

The properties of the elastoplastic matrix are:

$$E = 3130 \quad \text{MPa} \tag{3.6}$$

$$\nu = 0.3 \tag{3.7}$$

$$\sigma_y = 64.8 - 33.6e^{-\varepsilon_{eq}^p/0.0003407} \tag{3.8}$$

$$\varepsilon_{eq}^p = \sqrt{\frac{2}{3}\boldsymbol{\varepsilon}^p : \boldsymbol{\varepsilon}^p}, \tag{3.9}$$

with yield function σ_y with isotropic hardening, and plastic and equivalent plastic strain $\boldsymbol{\varepsilon}^{\boldsymbol{p}}$ and ε_{eq}^p, respectively.

The cohesive zone model used in the micromodel is developed by Turon et al. [24]. The constitutive model between traction and displacement jump is defined by a bilinear equation and is shown in Figure 3.2.

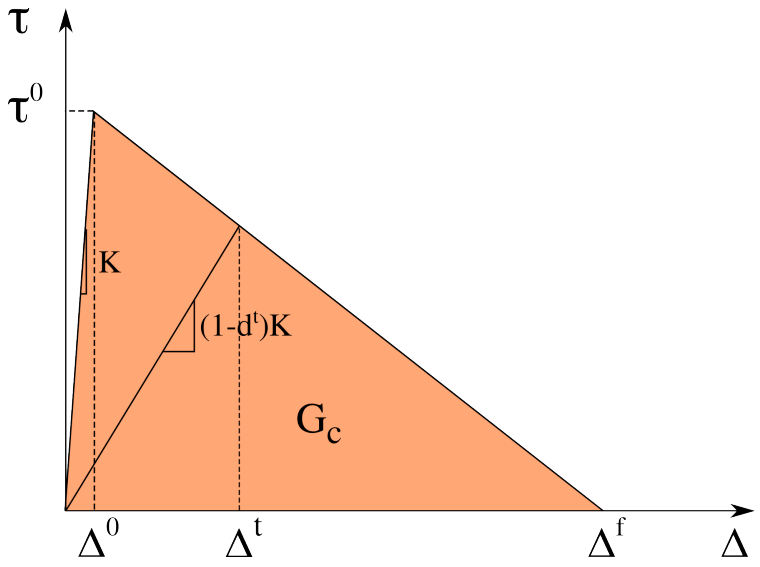


Figure 3.2: Bilinear constitutive relation between traction and displacement jump

The displacement jump and traction force in local coordinates are denoted by Δ and τ, respectively. Damage d is initiated once the traction force reaches a critical value τ_0. Ideally, there is no displacement jump before damage onset, so the value Δ_0 should be zero. However, the displacement jump should ideally also be zero for traction forces smaller than the critical value before damage is initiated. This complexity is overcome by including a linear function for $0 \leq \Delta \leq \Delta_0$ with a high stiffness K, which is referred to as dummy stiffness. It is a dummy variable, as it does not arise from material properties but it is an artificial value to ensure $\Delta \approx 0$ for the stage with no damage. After reaching the peak traction force, the force capacity of the point decreases as the microcrack forms. If the microstructure

is unloaded at time t, the unloading takes place with the secant stiffness $(1 - d^t)K$ where d^t is the damage at time t. This is unlike the plasticity model, where the unloading takes place with the linear elastic stiffness. The damage parameter d increases from 0 to 1, with 0 meaning no damage and 1 meaning fully damaged material (*i.e.* there is no load-bearing capacity of the material point).

In the normal component of the local coordinate of the interface element, no displacement jump is allowed in compressive forces, since negative displacement is not possible (i.e. interpenetration would take place). This is achieved by using a linear curve with the dummy stiffness in the negative values. The shear component of the local coordinate allows slip in the negative direction, therefore the plot in Figure 3.2 is symmetric with respect to its origin. The constitutive relations for the local normal and shear components are shown in Figure 3.3.

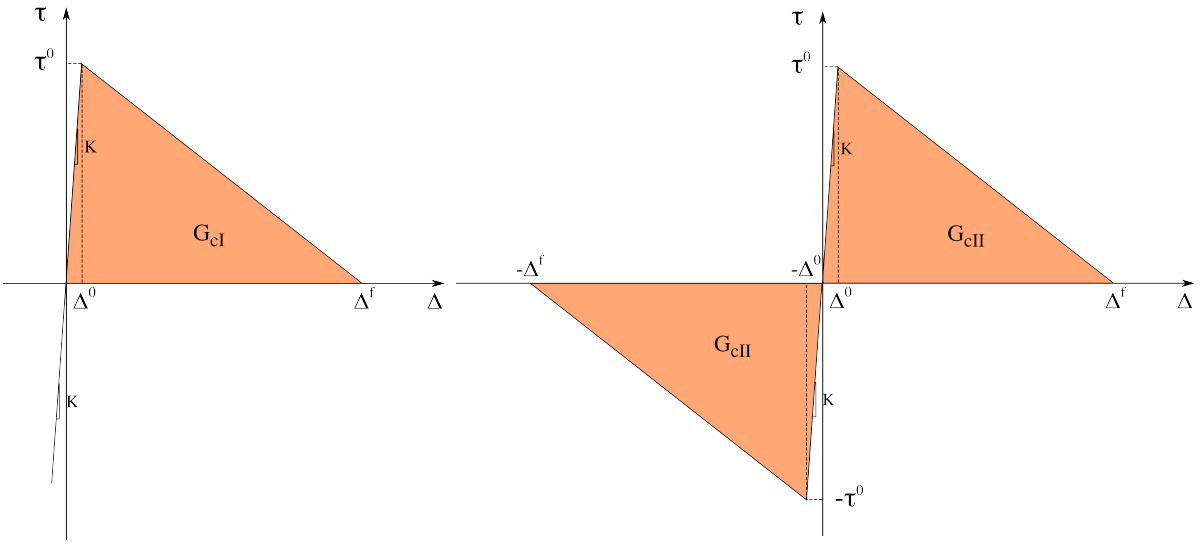


Figure 3.3: Turon constitutive law between traction and displacement jump in local normal and shear components

K	5×10^7 MPa
τ_n^0	60 MPa
τ_s^0	60 MPa
G_{Ic}	0.874 kJ/m^2
G_{IIc}	1.717 kJ/m^2

Table 3.1: Properties of cohesive points

The properties considered for the cohesive zone model are summarized in Table 3.1, in which τ_n^0 and τ_s^0 are the critical traction forces and G_{Ic} and G_{IIc} are the fracture energy values that define the area under the bilinear curve in normal and shear directions. The damage evolution law is discussed in [24]. In this work, plane stress conditions are assumed for the micromodel.

3.2. Load path generation

The micromodel discussed in Section 3.1 is subjected to different loading paths using periodic boundary conditions. To prevent rigid body rotations, the micromodel was fixed in x and y direction at corner 0 and fixed in x direction at corner y, while the loading was applied in x and y direction at corner x, and in y direction at corner y, as illustrated in Figure 3.4. The datasets used for training and testing the network can be separated into two categories: proportional and non-proportional loading.

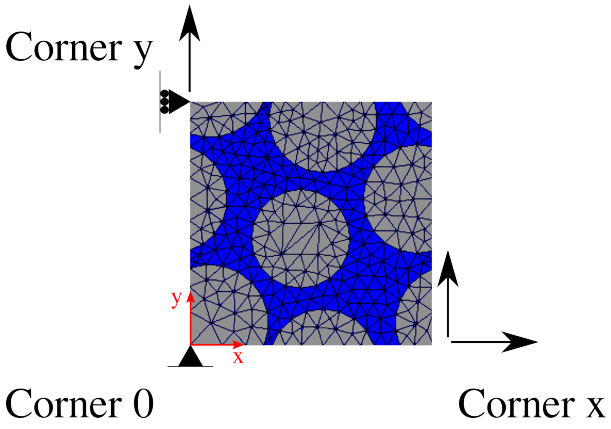


Figure 3.4: Boundary conditions for full-order micromodel

3.2.1. Proportional loading

The loading takes place over a fictitious time t. During proportional loading, the applied strain increments have a constant step size $s = 1.67 \times 10^{-3}$. These increments are applied in the predefined directions of the strain space, which are represented by ε_x, ε_y, and ε_{xy}. A subset of the predefined directions can be seen in Figure 3.5, where the blue curves display the load paths in 18 directions from the fundamental loading cases, containing pure tension, compression in x, y, and xy direction, biaxial tension, and mixed mode loadings. This dataset generated from the known loading directions is referred to as the canonical dataset. The orange curves represent load paths in random directions and the dataset generated from these curves is referred to as the random directions dataset.

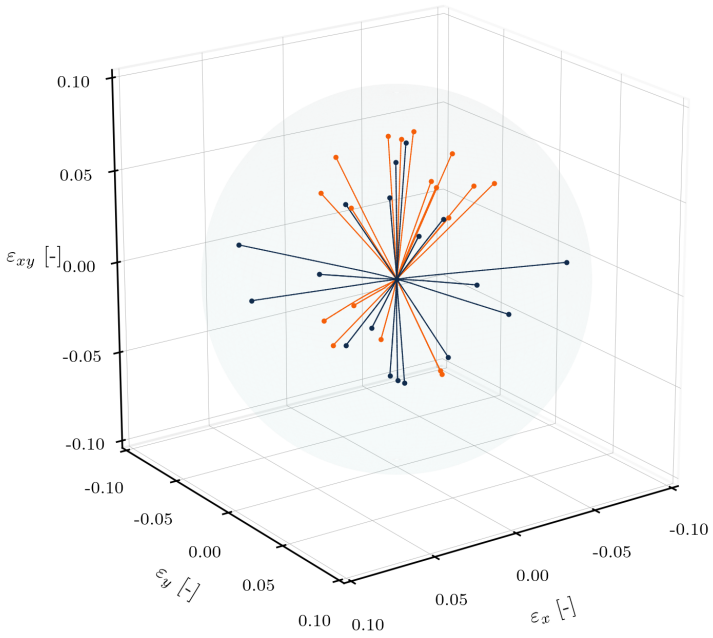
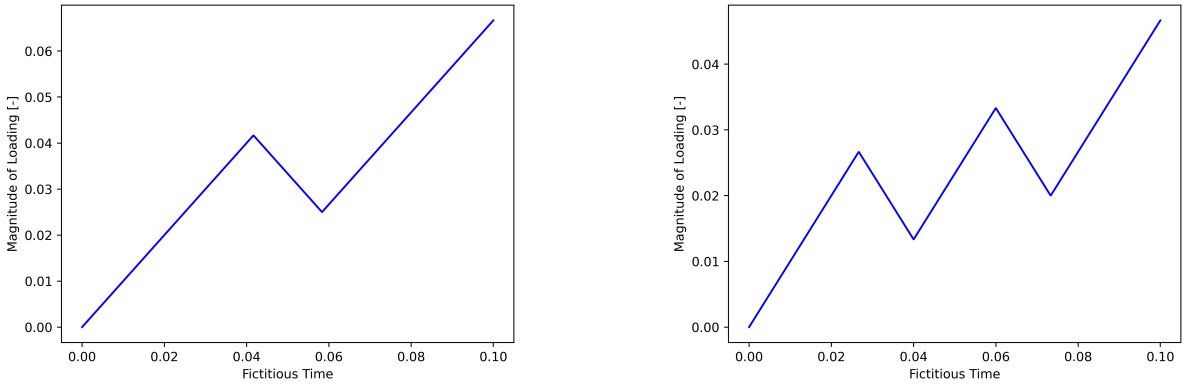


Figure 3.5: Proportional loading path directions

Monotonic and non-monotonic loading can be differentiated in the proportional loading set. During monotonic loading, the step size is always applied in one direction, and the load will only increase. During non-monotonic loading, however, while the direction in which the step size is kept fixed, 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 monotonic and non-monotonic cases considered in this thesis are shown in Figure 3.6.



(a) Function with one unloading cycle

(b) Function with two unloading cycles

Figure 3.6: Loading functions used to generate proportional loading curves

3.2.2. Non-proportional loading

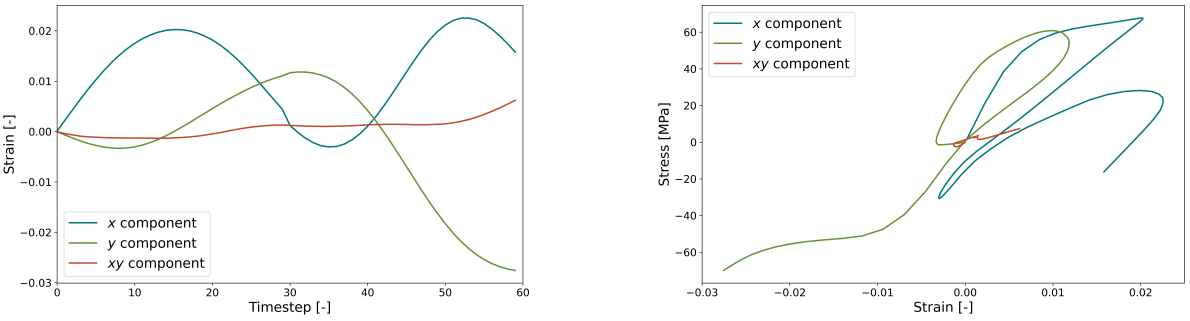
To create a more diverse loading scenario, non-proportional and non-monotonic loading paths are generated. Both the direction of loading and step size are varied at each time step. This is achieved by sampling the strains from Gaussian Processes (GPs). Each strain component is drawn from an indepndent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

$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{3.10}$$

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:

$$\Sigma_{ij} = k(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}||x_i - x_j||^2\right) \tag{3.11}$$

with σ_f^2 being the variance that determines the step size and ℓ being the lengthscale that controls the smoothness of the generated path. With increased variance σ_f^2 the strains are able to obtain larger values, and with increased lengthscale the curve becomes smoother. Values $\sigma_f^2 = 0.0001667$ and $\ell = 200$ were used in this work. The strain values are sampled step by step, which results in the same stress-strain path if sampling the entire sequence at once for a fixed random seed. The strain paths



(a) Strain paths

(b) Stress-strain curves

Figure 3.7: Example of GP-based loading path

are generated independently in the x, y, and xy components, which are illustrated in Figure 3.7. Figure

3.7b displays the resulting stress-strain curve from the corresponding generated strain paths. A subset of the load paths generated by GPs is shown in Figure 3.8, with a curve highlighted in red for clarity.

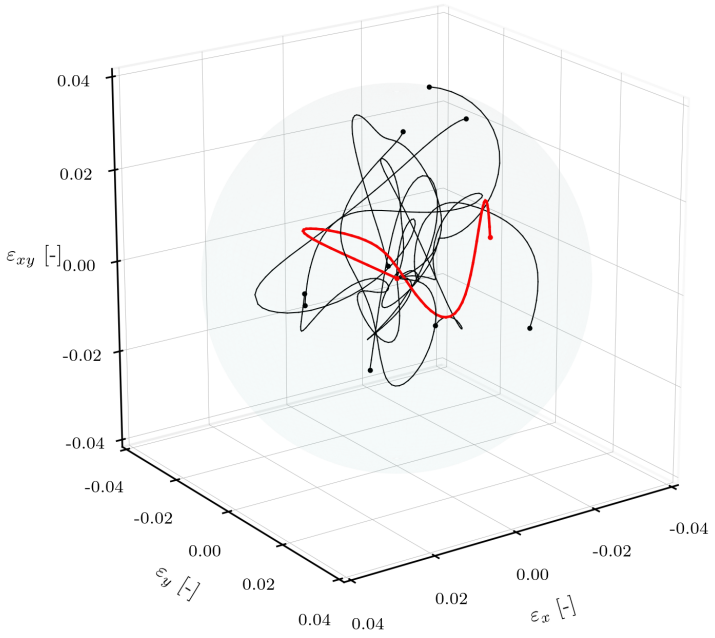


Figure 3.8: Non-proportional loading path directions

4

PRNN with bulk models only

This chapter addresses the research objective by investigating whether the physically recurrent neural network as proposed in [12] (Figure 4.1) is able to capture microscale damage. Recall that in that work, the fibers and the matrix in the micromodel are assumed to be fully bonded. Consequently, the constitutive models in the material layer of the network do not describe softening. Here, however, full bondage is no longer assumed in the full-order micromodel. Since not all nonlinearities in the full-order micromodel (Figure 3.1) are implemented in the material layer of the PRNN, this study is an initial step to understand the network attempts to predict debonding.

In this initial verification, the architecture explored consists of one input, one material, and one output layer with only bulk integration points in the material layer. The bulk integration points consist of 3 neural units, in which J2 plasticity model is nested to convert 2D local strains to 2D local stresses. Note that no damage model is considered at this stage in the network other than in the training data itself.

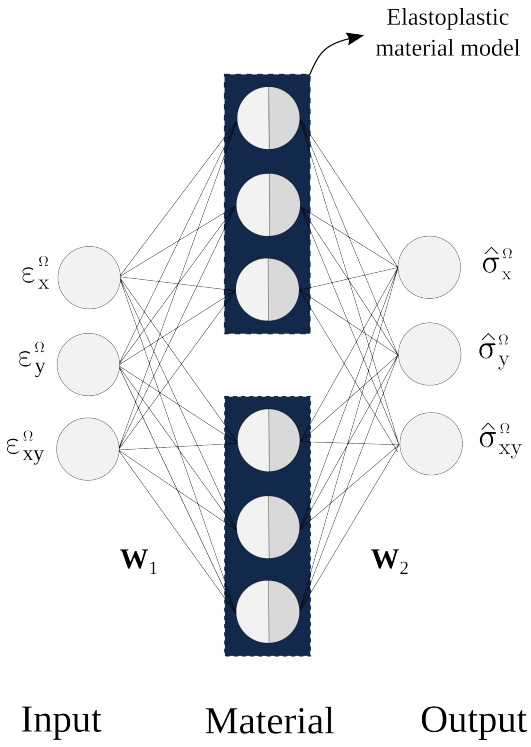


Figure 4.1: PRNN with elastoplastic model only

4.1. Training with monotonic and proportional data

The network was trained on 18 curves from the canonical, monotonic, and proportional dataset. The validation data consisted of 54 curves from the monotonic and proportional dataset but with random directions. The size of the training and validation dataset are based on [12], and are not varied in this chapter due to the assumption that increasing the training data will not influence the results qualitatively. Different material layer sizes were studied, which was kept to be a multiple of 3 to represent the fictitious material points consisting of 3 neurons. The validation error for these material layer sizes are shown in Figure 4.2, where the number of material layer units n_u corresponds to $n_u/3$ bulk material points. For each material layer size, the box plot displays the first quartile (Q1), the median, the third quartile (Q3), and the interquartile range (IQR) which spans between the first and third quartiles. The whiskers extend to the minimum and maximum values, excluding the outliers. Data lying outside the range $[Q1 - 1.5IQR, Q3 + 1.5IQR]$ are marked with a diamond, which represents the outlier data points. It is not recommended to consider the outliers when selecting the optimal network size. The sample mean is marked with a white cross.

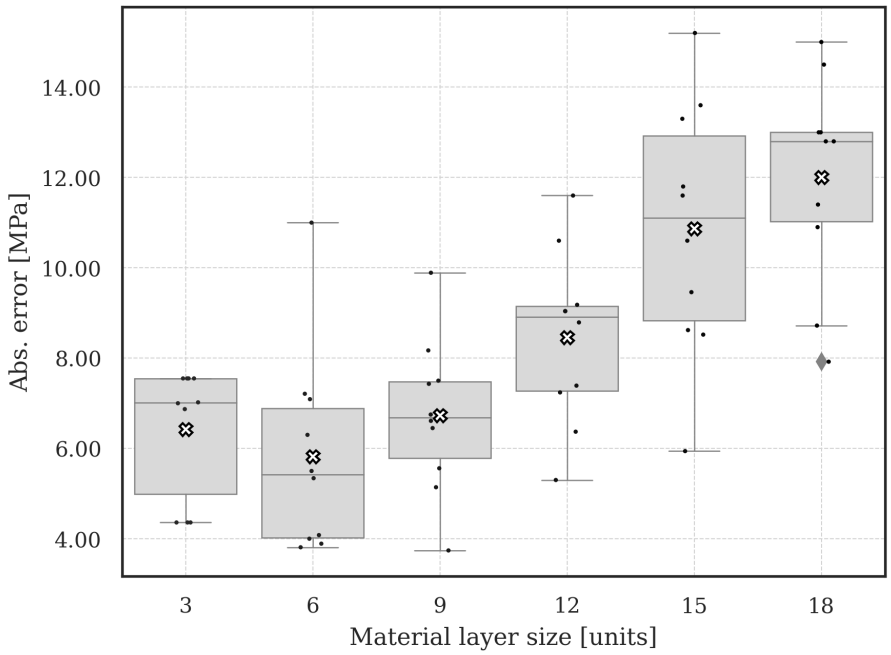
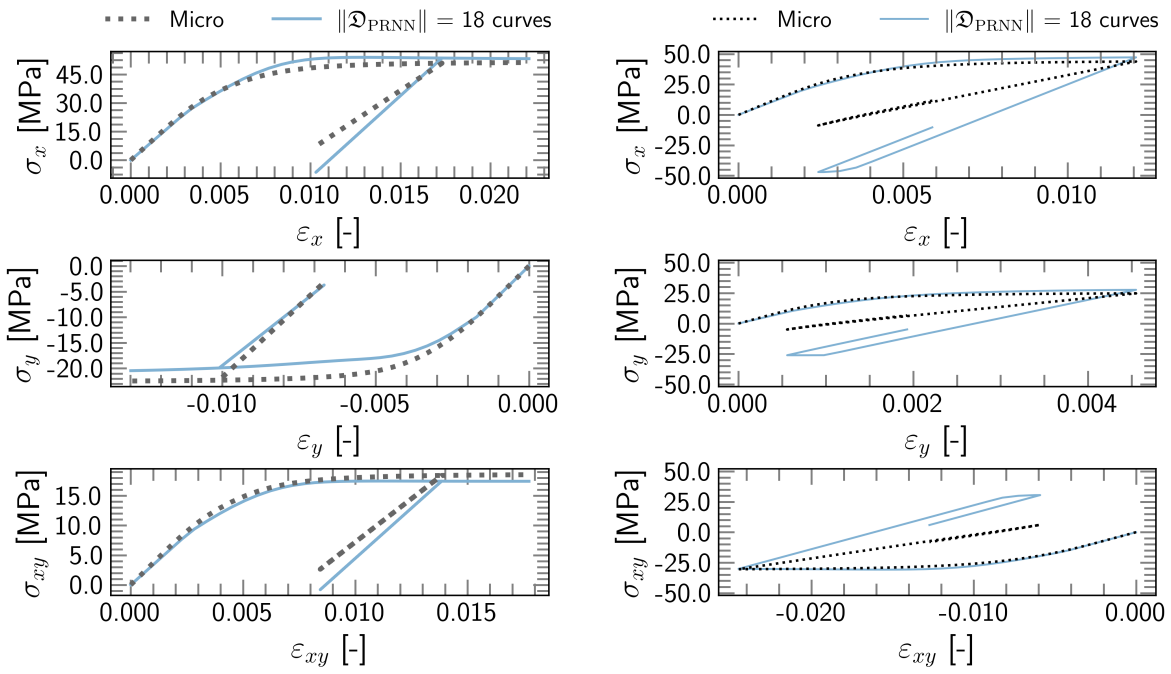


Figure 4.2: Validation Error for PRNN trained on monotonic data for different material layer sizes.

The lowest validation error was found for a material layer size of 9 with an MSE of 3.74 MPa. However, a pattern of increasing average MSE can be observed for a material layer size of 6 and higher, with a minimum at 6 units. Since the lowest validation error for 6 units (3.81 MPa) is not significantly higher than of 9 units, a material layer size of 6 was selected which corresponds to two fictitious material points in the network.

The selected network was tested with 54 curves from the random, non-monotonic, and proportional dataset, with one cycle of unloading. The lowest average MSE of the test set is around 5.22 MPa. The prediction of the network on one of the test curves from this test set is shown on Figure 4.3a, with a MSE close to the average MSE of the test set (4.53 MPa). To obtain this prediction, the network outputs a linear combination of the microscopic stresses that were computed using the constitutive model embedded in the fictitious points in the material layer. Figure 4.4 displays the microscopic stress-strain relation along with equivalent plastic strain in each fictitious point in the network for the same representative curve shown in 4.3a.

The network predicts accurately on the monotonic region, which aligns with observations made in [12]. The material layer with the elastoplastic model is able to learn the nonlinear behavior that the monotonic training data encloses. However, once unloading is initiated, the network loses its accuracy. The PRNN predicts unloading with the initial, linear elastic stiffness following the assumptions



(a) Regular unloading

(b) Longer unloading

Figure 4.3: Network prediction on representative curves from non-monotonic test set

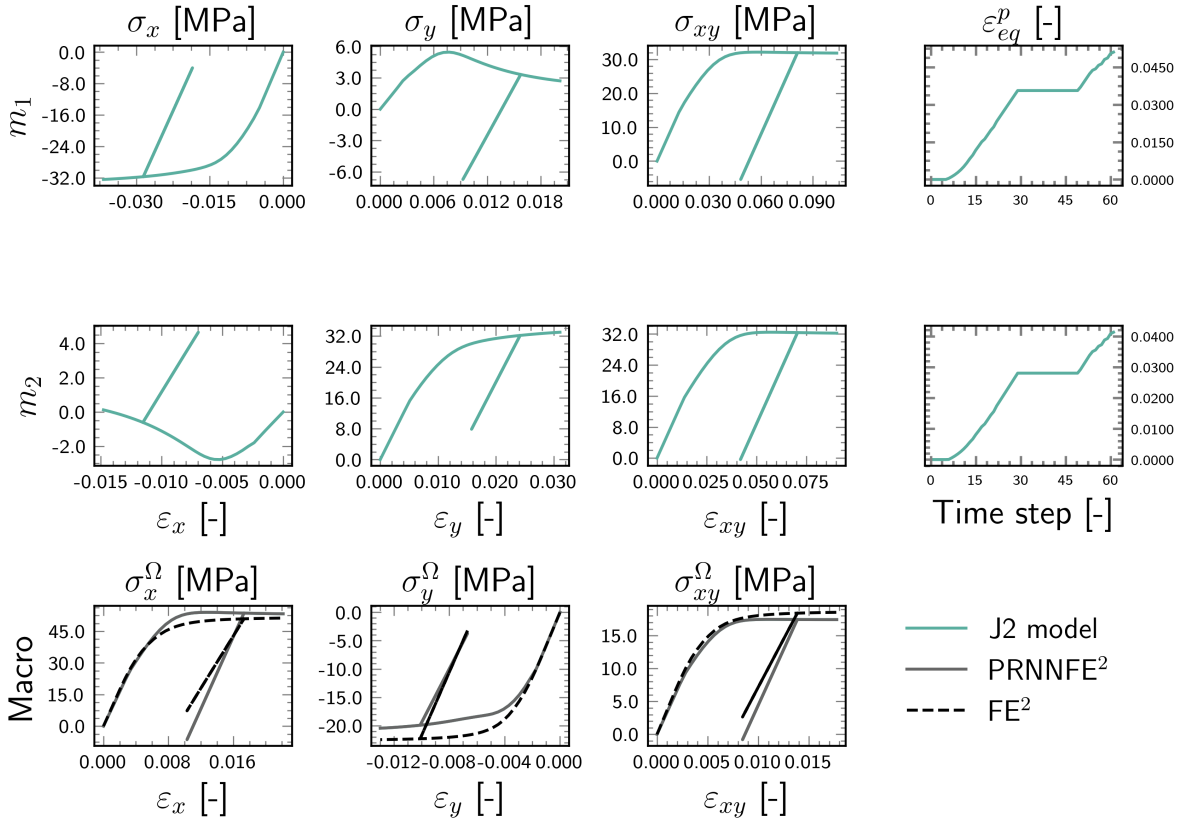


Figure 4.4: Micro and macro response of PRNN on a representative curve from non-monotonic test set

embedded in the elastoplastic model. In this case, the error is relatively low for a short period of unloading. However, this error increases largely when the micromodel is subjected to unloading for a longer period of time. Figure 4.3b shows the prediction of the network on a test curve that was generated by unloading the micromodel for 20 timesteps instead of 10. Even though excellent agreement is observed between the curves on the monotonic part, the error in the unloading region is more apparent, resulting in a high error of around 16.64 MPa. Since the training data consists of monotonic curves only, the network is not given a chance to learn the effect of the second type of nonlinearity that comes from the result of the full-order micromodel. The fact that the PRNN with bulk points only, trained on curves that do not contain information on the effect of the cohesive points, is not able to capture microscale damage is a reasonable conclusion.

In order to develop a model that can extrapolate well to all types of load cases, the network has to be improved to capture loss of stiffness due to damage. On that note, it is important to verify whether including non-monotonic data during training improves the network's accuracy in the non-monotonic region. This way, the network is exposed to the effect of the cohesive zone model in the RVE by information coming from the training data.

4.2. Training with non-monotonic and non-proportional data

To observe if training with curves that undergo unloading improves accuracy of the network in the non-monotonic region of the test set, for training and validation of the network 18 and 54 curves from the random, non-monotic, non-proportional dataset were used, respectively. The validation error for different material layer sizes of the network are shown in Figure 4.5.

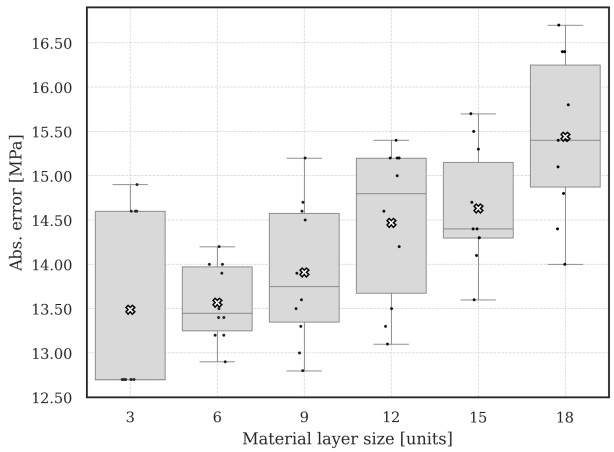


Figure 4.5: Validation Error for PRNN trained on non-monotonic data for different material layer sizes

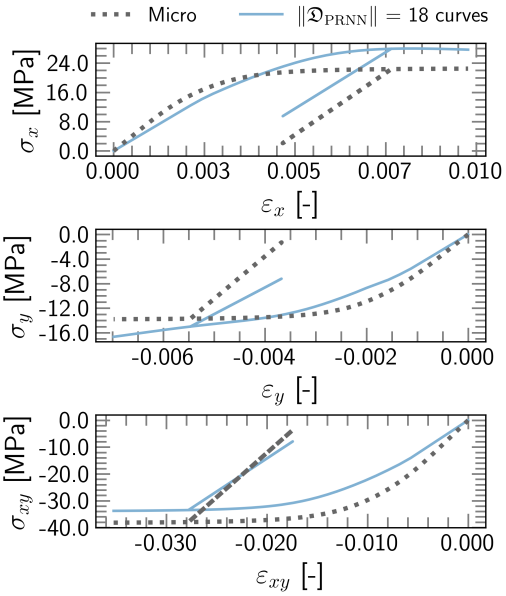


Figure 4.6: Network prediction on a representative curve from test set

The network with a material layer size of 3 resulted in the lowest validation MSE (12.7 MPa). Having a network with only one fictitious material point tends to underfit the data, therefore choosing this size should be avoided. There is a trend of increase in the average validation MSE as the material layer size is increased, therefore the network with the lowest validation error is selected, other than the network with 3 units. This corresponds to a network with 9 units of 3 fictitious points of a MSE of 12.8 MPa.

The same test set of 54 curves from the random, non-monotonic, and proportional dataset with one unloading was used as for the network that was trained on monotonic data, for fair comparison. The lowest average MSE of the test set was found to be 8.43 MPa. The prediction of the network on a representative curve from this test set is shown on Figure 4.6.

The network seems to learn that the unloading should occur with a lower stiffness than the initial linear elastic stiffness. On the stress-strain curve in the x component in Figure 4.6, it can be observed

that the network is able to accurately predict the stiffness on the unloading branch. However, it does so while compromising its accuracy on the monotonic region. In order to achieve a lower stiffness value in the unloading branch, the network also decreases the initial linear elastic stiffness to match the unloading stiffness. Due to this decrease in initial stiffness, the network either under or overshoots the stress values in the monotonic region.

The fact that the network predicts unloading with the initial linear elastic stiffness suggests that the PRNN in its current state, with the only source of nonlinearity being the elastoplastic model in the material layer, is not able to learn the effect of both types of nonlinearities that are present in the RVE. The network learns only one type of nonlinearity, even with a richer training set. That being the case, it is concluded that in order for the PRNN to learn all types of nonlinearities that are included in the RVE, all nonlinear material models that are present in the RVE also have to be included in the network.

The conclusions drawn in this chapter are not surprising, since there is no implementation in the network that can account for cohesive damage. Moreover, the core idea of the PRNN is to include all material models in the RVE as they are in the network itself as well. This idea is violated by not including the cohesive zone model in the network. In spite of these facts, this initial study on the PRNN with bulk points only represents the foundation for this work and clearly motivates why it is necessary to dedicate resources to further investigate the case of cohesive damage. Therefore, the following chapters of this study focus on the implementation and its challenges of the cohesive zone model within the PRNN framework.

5

Extending the PRNN with cohesive material points

As discussed in Chapter 4, the physically recurrent neural network cannot predict the effect of debonding at the fiber-matrix interface without including all sources of nonlinearity in the network that are present in the RVE (Figure 3.1). Therefore, the cohesive zone model (CZM) in the full-order micromodel has to be implemented in the physically recurrent neural network as well. This chapter investigates whether the physically recurrent neural network with CZM in the existing material layer is able to capture micro-scale damage.

As a starting point, including the CZM in the network is achieved by adding cohesive integration points to the same material layer that has bulk integration points. This design choice preserves the same network architecture as previously discussed in Chapter 4, and is the most straightforward extension of the network. Therefore, the PRNN proposed in this chapter consists of one input layer that receives macroscopic strains, one material layer with the nonlinear models, and one output layer yielding the macro-scale stress predictions (Figure 5.1).

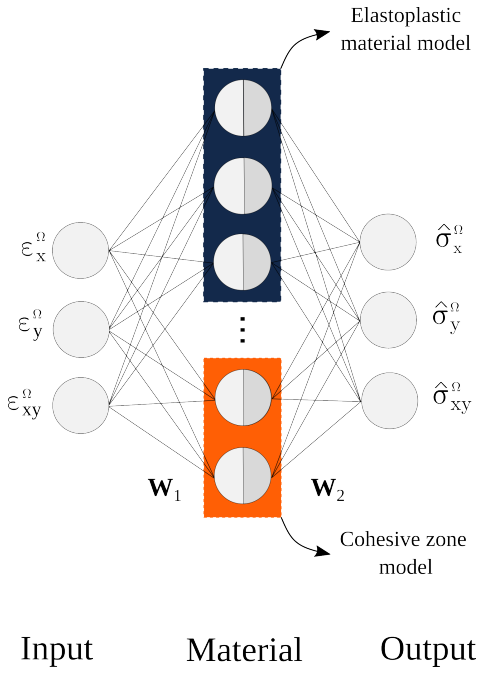


Figure 5.1: PRNN with CZM in material layer

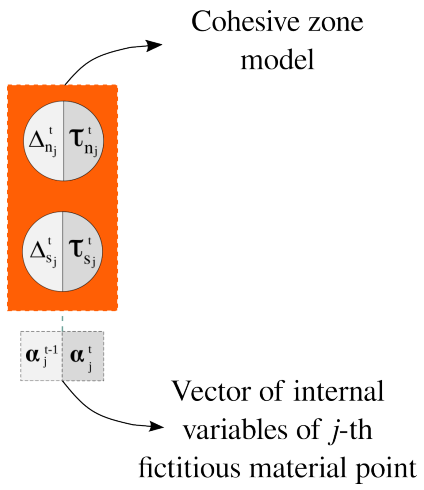


Figure 5.2: Fictitious cohesive material point

The nonlinear material models present in the RVE are incorporated in the material layer by nesting them in bulk and cohesive fictitious integration points. Recall that 3 neurons represent one bulk integra-

tion point. In the bulk points, local strains are converted into local stresses, while also storing internal variables to account for path-dependency. The cohesive integration points, on the other hand, consist of 2 neurons, which are associated with a cohesive zone model to convert the local displacement jumps with normal and shear components to local tractions. The cohesive points also store internal variables to keep track of history. Along with the local strains and displacement jumps (depending on the type of material point), internal variables also serve as an input to the fictitious integration points. These material points are illustrated in Figure 5.2.

To assess the performance of the network in several loading scenarios, model selection was performed. The parameters being changed in this study, therefore, are the number of bulk integration points and the number of cohesive integration points in the material layer. Different training sizes were also considered to observe the effect of increasing the training data on the performance of the network. The objective of the model selection is to obtain a configuration that maximizes accuracy of the network prediction while being able to extrapolate to unseen loading cases.

5.1. Training with monotonic and proportional data

The PRNN with bulk points only in Chapter 4 trained with monotonic data was not able to extrapolate to cases with unloading and it was concluded that the network should learn both types of nonlinearities during training. The goal of training with monotonic data despite these conclusions is to observe whether they also hold to the case when cohesive points are included in the material layer. In addition to that, a brief investigation was also considered to examine the influence of the training size, as the assumption and the findings in [12] suggest that the size of the training set should have no significant impact on the accuracy of the prediction on the non-monotonic region.

The PRNN with CZM was trained on 18 curves from the canonical, monotonic, and proportional dataset. The validation set consisted of 54 curves from a dataset with the same type of loading but with random directions. First, a study on the size of the material layer was conducted by setting the number of cohesive points equal to the number of bulk points. The validation errors are plotted in Figure 5.3 for the various sizes, where the x axis represents the number of bulk and cohesive points in the network (i.e. 2 represents 2 bulk and 2 cohesive points corresponding to $2 \cdot 3 + 2 \cdot 2 = 2 \cdot 5$ units).

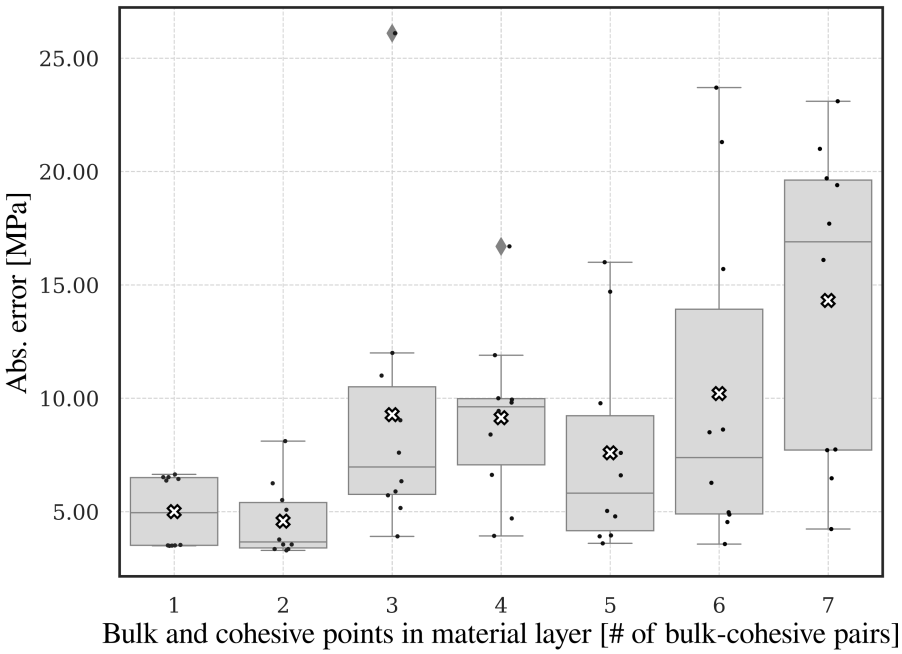
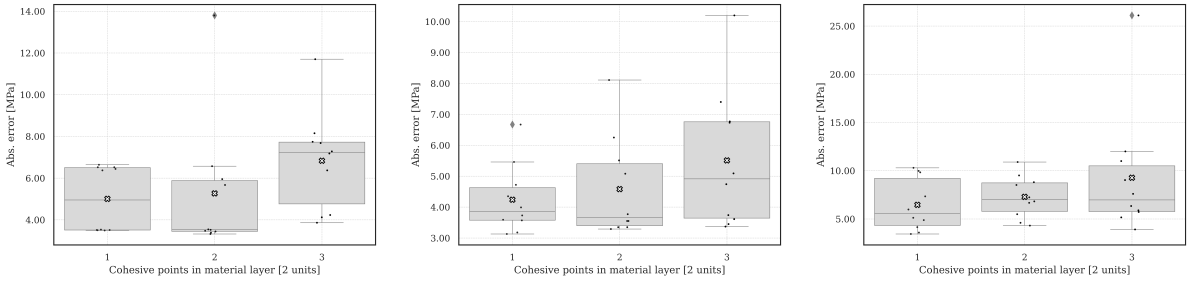


Figure 5.3: Validation error for PRNN trained on 18 monotonic curves from canonical dataset

Figure 5.3 indicates no clear trend in the lowest MSE across the different material layer sizes, it is consistent around 3.6 MPa. Networks with smaller material layer sizes seem to have the lowest average MSE, with 5.00 MPa and 4.58 MPa for 1 and 2 bulk and cohesive points (corresponding to

5 and 10 units), respectively. An increase in the average MSE is observed for networks larger than 3 bulk and cohesive points. Based on these findings, when different combinations of bulk and cohesive points are considered, the maximum number of integration points per material model is set to 3 for the remaining of this section. The goal is to understand whether the accuracy of the network can be improved with a combination of bulk and cohesive points other than the fixed ratio of points shown in Figure 5.3.



(a) 1 bulk point (b) 2 bulk points (c) 3 bulk points

Figure 5.4: Validation error for different number of bulk points

For this study, the number of bulk points was fixed and the number of cohesive points was changed to see the effect of increasing the number of cohesive points on the validation error. Figure 5.4 shows the validation error for the various sizes. A network with 1 cohesive point results in the lowest average MSE across the different bulk sizes. For testing, the network with 2 bulk and 1 cohesive points was selected, with the lowest validation error of 3.13 MPa.

The selected network was tested on 54 curves from the random, non-monotonic, and proportional dataset, with one cycle of unloading. This test set is the same as the one used in Chapter 4 for a straightforward comparison. The lowest error of the test set for a material layer with 2 bulk and 1 cohesive points was found to be 4.69 MPa. The prediction of the network on a representative curve from this test set is shown in Figure 5.5a.

This time, the network makes a fairly accurate prediction in the compressive region, where the unloading takes place with the initial linear elastic stiffness and has a minimal contribution from the cohesive zone models in the actual micro-model. However, no qualitative change can be observed between the prediction of the network with bulk points only from Section 4.1 and the prediction of this network trained on monotonic data, as the network is visually not accounting for the damage. This is verified by analyzing the weights associated with the cohesive points.

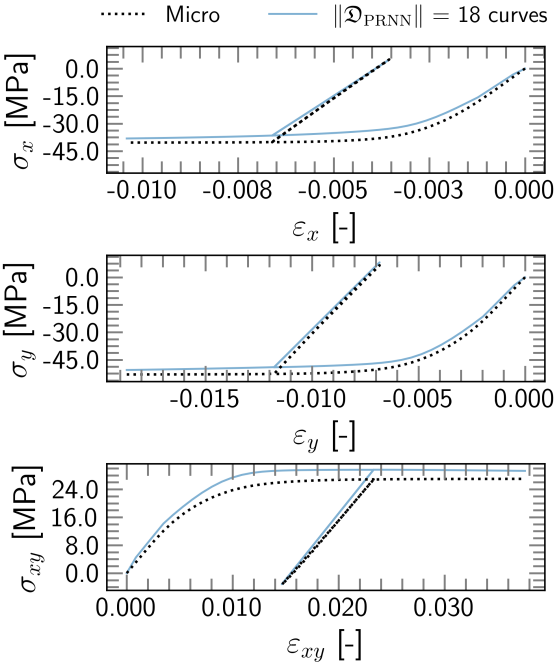
For the representative curve shown in Figure 5.5a, the microscopic stress-strain relation along with equivalent plastic strain in the bulk points, and the microscopic traction versus displacement jumps with damage in the cohesive points are shown in Figure 5.6. The weights connecting the cohesive points to the output are negative to the magnitude where the softplus activation function applied on the weights convert these values to a small number close to zero (a maximum value of weight of around 0.02). This means that the contribution from the cohesive points in the network is negligible.

This phenomenon is more apparent when looking at extreme cases. Figures 5.5b and 5.5c depict the prediction of the network on curves from the same test set with low and high errors, respectively. Figure 5.5b shows a case when the micro-structure is loaded mostly in compression, in which case debonding is not critical, therefore the network with little to no contribution from the cohesive points performs well. On the other hand, when micro-scale damage is triggered and the micro-model is loaded in tension, as shown in Figure 5.5c, the lack of ability of the network to predict this loss of stiffness in the unloading branch is more evident.

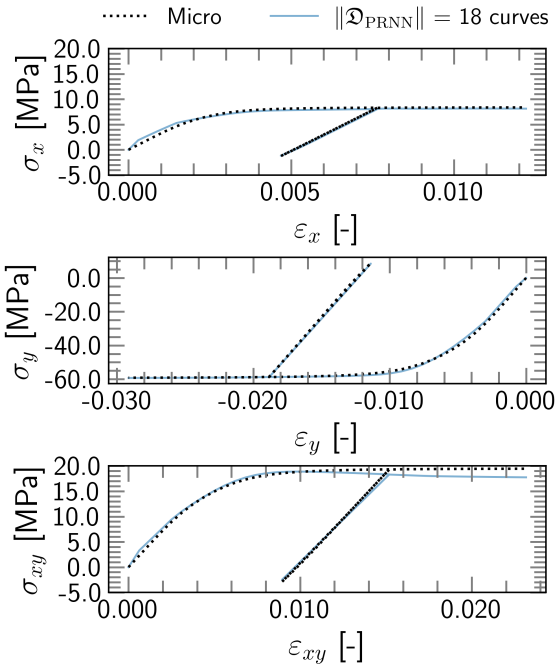
Since the best architecture seems to be one where the contribution from the cohesive zone model is close to zero (i.e. in the extreme case, we would obtain the architecture with bulk points only), there is no extensive study on whether the network is improved by adding more monotonic curves to the training data. Moreover, the accuracy of the network when predicting within the monotonic region and during unloading in the compressive region is remarkable, therefore the improvement in this accuracy by adding more monotonic curves to the training set would be marginal and thus unnecessary. The weights associated with the cohesive points in the decoder layer still remain negative enough to make their contribution to the macroscopic response of the network negligible. These conclusions suggest

that even though there are cohesive points added to the network, their impact remains insignificant unless they are actively engaged during training.

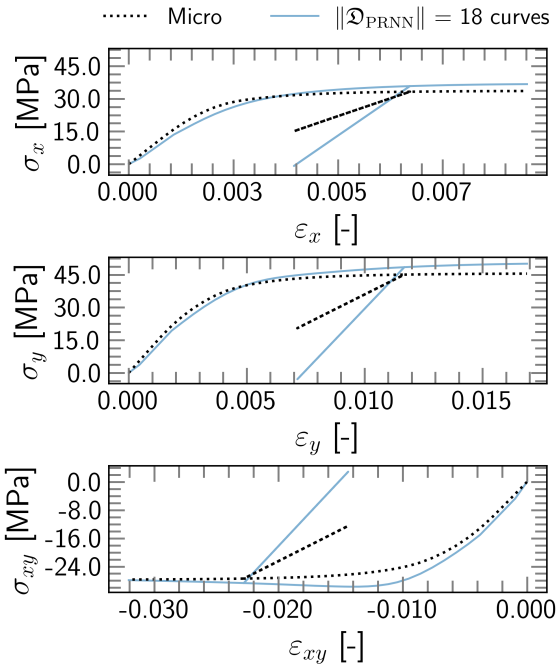
From the findings of training with monotonic curves, the conclusions drawn in Section 4.1 are sustained; the network has to be trained in a way that it can learn all nonlinearities present in the full-order micro-model. Hence, for the remainder of this work, the training and validation datasets comprise only non-monotonic data.



(a) Representative curve



(b) Curve with low error



(c) Curve with high error

Figure 5.5: Prediction on random non-monotonic curves with one cycle of unloading using networks trained with 18 canonical curves

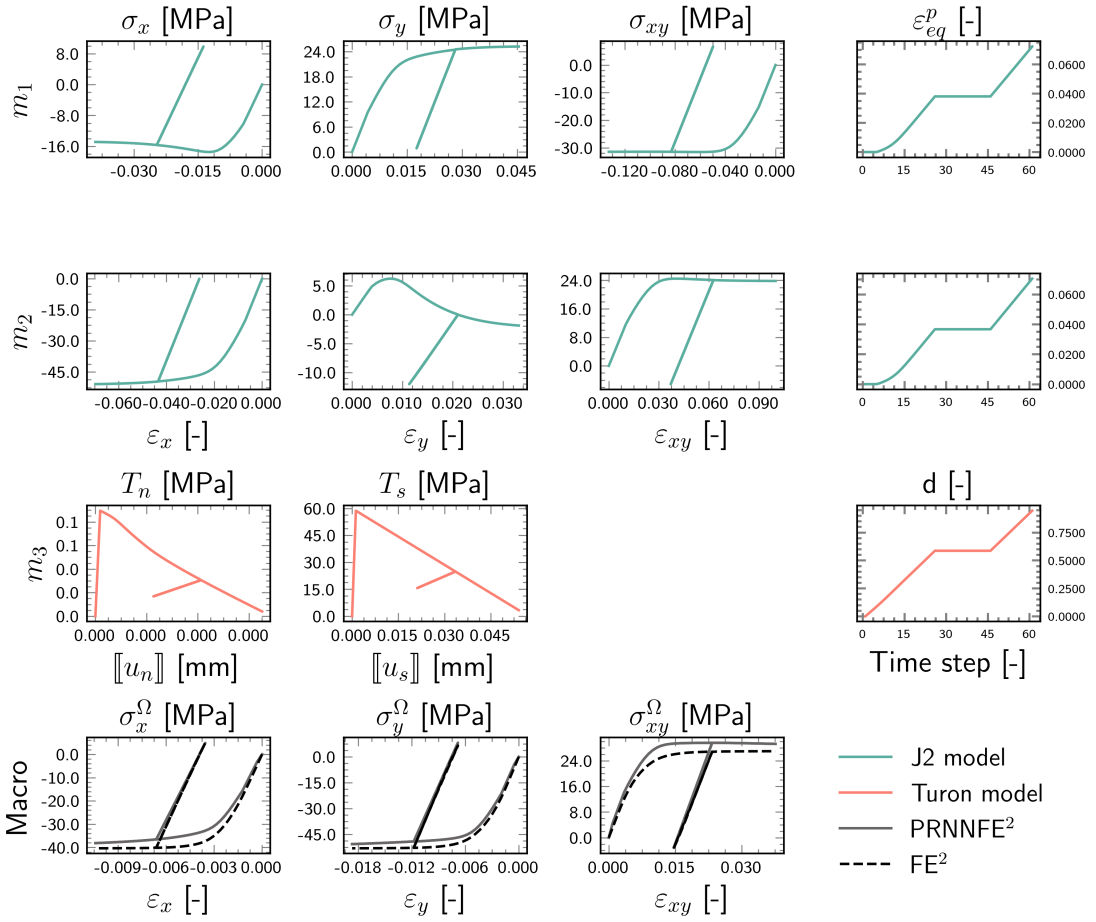


Figure 5.6: Micro and macro response of PRNN on representative curve

5.2. Training with non-monotonic and non-proportional data

For the PRNN to be able to predict unloading behavior, the cohesive points in the material layer have to be active. This can be achieved by training with curves that undergo unloading. In this section, non-monotonic and non-proportional data was used to train and validate the network.

The goal is then to utilize these activated cohesive points to extrapolate to loading scenarios that it has not seen during training. First, the training set size was determined while keeping the validation set at 54 curves of the same type. After selecting the optimal training set size, networks with different combinations of bulk and cohesive points were considered to obtain the configuration of the material layer that results in the best performance.

5.2.1. Training size selection

When investigating how the training set size affects the performance of the network, different numbers of bulk and cohesive points were considered. The networks were then evaluated by comparing their validation errors across different training sizes. The ratio of $n_{bulk}/n_{coh} = 1$, where n_{bulk} and n_{coh} correspond to the number of bulk and cohesive points in the material layer, respectively, ranging from a minimum configuration of one bulk and cohesive point to seven bulk and cohesive points. The training dataset consisted of 18, 36, 54, 72, and 90 non-monotonic and non-proportional curves.

Figure 5.7 displays the lowest validation MSE across the various training data sizes for the different material layer configurations. A training data size of 72 was selected, as the average value of the lowest MSE for different material layer sizes reaches minimum here. It is important to note the increase in overall validation error of the networks trained on non-monotonic and non-proportional curves compared to the ones trained on monotonic and proportional data. This can be explained by the large

randomness in the curves, due to which the network is much less prone to overfitting the data.

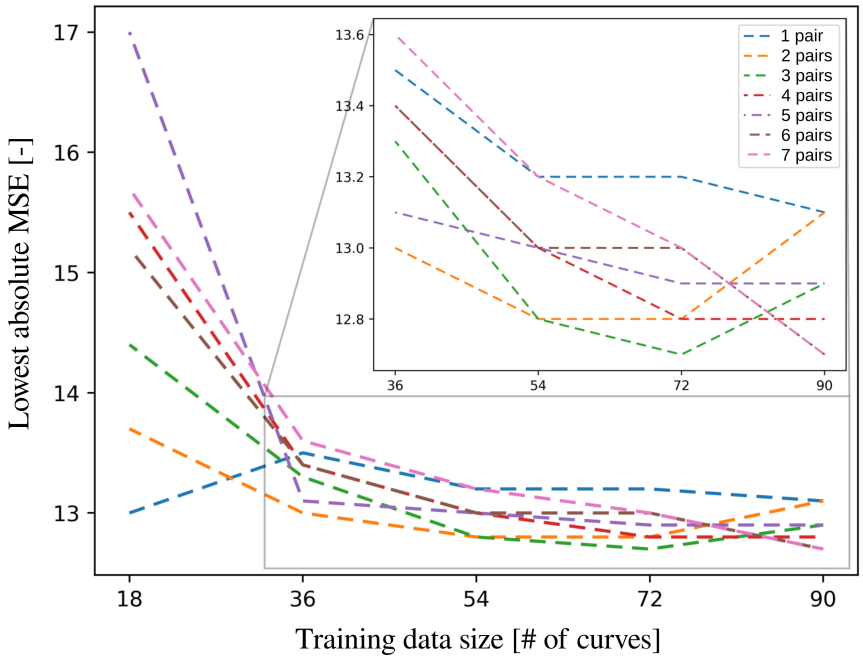


Figure 5.7: Lowest validation MSE for different material layer sizes across various training data sizes

5.2.2. Varying the material layer configuration

In this section, networks were trained and validated on 72 and 54 curves of the same type, respectively. The selection of the material layer size was based on the mean squared error (MSE) values obtained on the validation set. To determine the number of bulk and cohesive points in the network, several combinations of bulk and cohesive points without the fixed proportion of 1 were tried. This allowed for a broader exploration of bulk-cohesive point combinations.

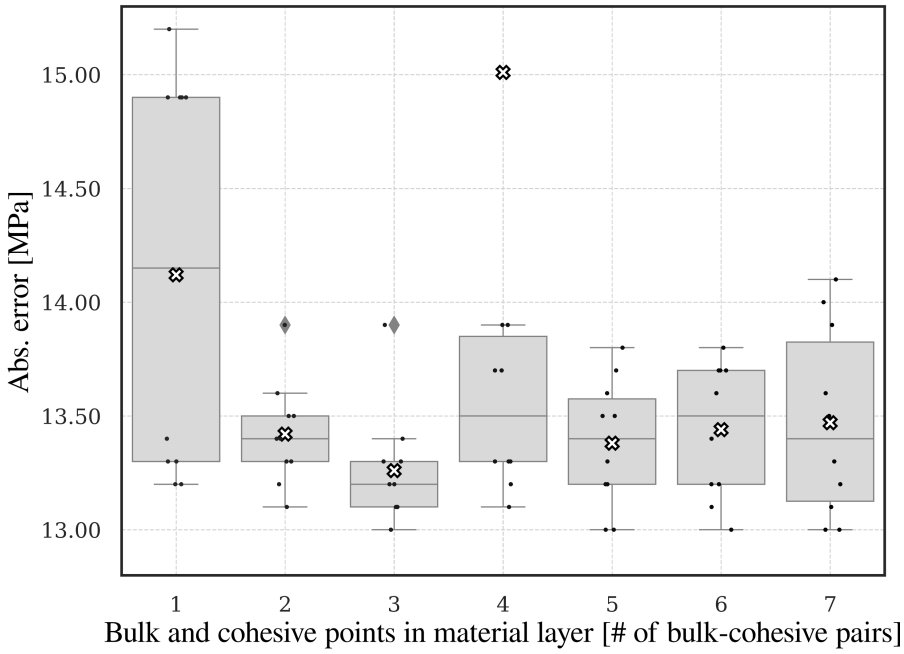
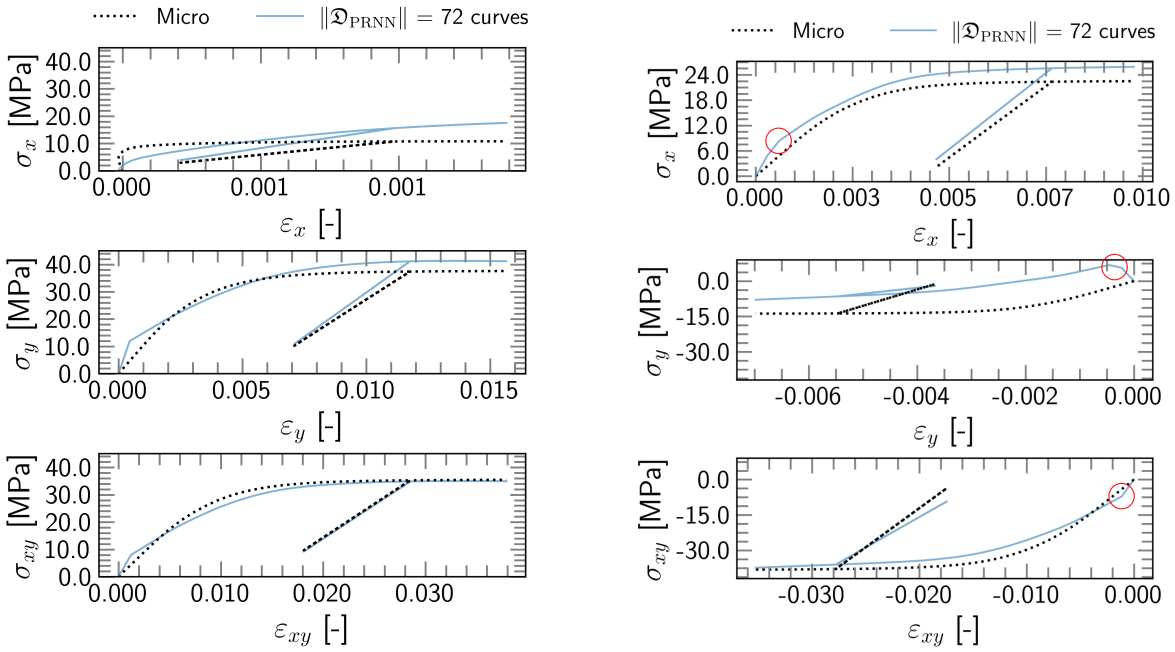


Figure 5.8: Validation error for the PRNN with an equal number of bulk and cohesive points trained on 72 GP curves

Figure 5.8 illustrates the result of the method described above for the particular case where the number of bulk and cohesive points are equal. In that case, the difference between the lowest MSE obtained by networks with different material layer sizes is not significant, and there is no clear trend of increase or decrease in MSE after a certain size. After fine-tuning the material layer size and configuration, the best network, a network with a combination of 2 bulk and 3 cohesive points was selected to illustrate results in this section with a validation error of 12.8 MPa.

For the test set, 54 curves from the non-monotonic, proportional dataset with one cycle of unloading considered in the previous sections were kept. Figure 5.9a shows that the selected network is able to capture the unloading phenomena in some cases. The stiffness of the unloading of the shear component of the prediction of the network is captured accurately. The monotonic region of these curves is also captured quite accurately.

Another interesting property of this network can be seen in this figure; since the network is minimizing the mean squared error on the data, it prioritizes fitting data with larger stress values. Therefore, the prediction of the network in Figure 5.9a in x direction seems largely inaccurate when zoomed in, however, relative to the other two directions this error is small.



(a) Prediction of network on a curve with the lowest error

(b) Prediction of network on a representative curve

Figure 5.9: Prediction of network with 2 bulk and 3 cohesive points trained on 72 GP curves on test curves with one cycle of unloading

Even though the considered networks are able to perform well in some cases, their prediction lacks consistent accuracy. Figure 5.9b shows the prediction of the PRNN with 2 bulk and 3 cohesive points on a representative curve of the test set with an average test MSE of 9.27 MPa. The network not only misses capturing the decrease of stiffness during unloading but also loses accuracy on the monotonic part.

There is a large kink along the initial monotonic part of the curves shown in Figure 5.9b, which is highlighted by red circles. This new phenomenon prompted an investigation into the factors causing the sudden change in slope. Recall that the approximation of the network on macro-scale stresses is a result of a linear combination of the local stresses and tractions derived from the material model nested in the material layer. Figure 5.10 shows the microscopic stresses and tractions in the material points in the PRNN with 2 bulk and 3 cohesive points. For example, cohesive points m_4 and m_5 undergo tension, which is equivalent to what happens at the interface points where the fiber and matrix are debonding, while cohesive point m_3 undergoes compression.

To account for the loss in stiffness due to microscale damage, the weights associated with the cohesive points are increased in magnitude so that the unloading branch of these curves can decrease

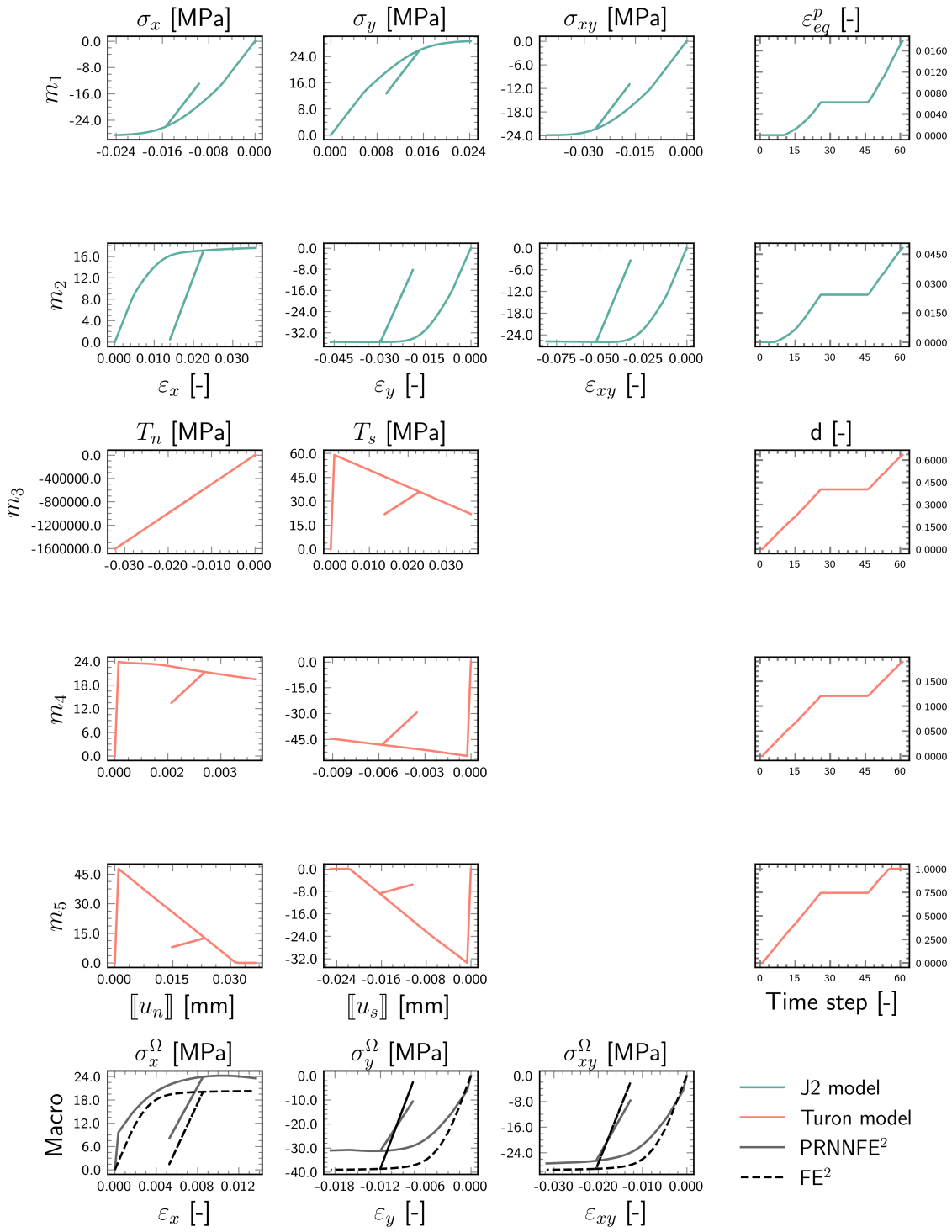


Figure 5.10: Micro and macro response of network with 2 bulk and 3 cohesive points on representative curve

the stiffness that would result from the plasticity model. However, as the weights associated with the cohesive points are increased, the contribution of the initial high dummy stiffness defined in the cohesive law is also increasing. Consequently, a large initial stiffness is observed in the macro response for small timesteps, which then causes the kink to appear on the curve.

It is therefore interesting to think about whether the fit on the monotonic part of the curve can be improved if the dummy stiffness of the cohesive model in the network is decreased. This means a

smaller initial stiffness of the cohesive model, and also a reduced stiffness on the unloading branch for the same traction force. The effect of reducing the dummy stiffness from K_1 to K_2 on the cohesive model is illustrated in Figure 5.11. Damage initiation is at a larger value of displacement jump for the case with reduced dummy stiffness. Note, however, that changing the dummy stiffness of the cohesive law in the network while keeping it the same in the full-order micromodel creates an inconsistency between the PRNN and the FE^2 method. It also contradicts one of the fundamental ideas of the network, according to which all nonlinear material models in the RVE have to be included in the network, with the same material properties as observed in the micro-model. Therefore, the following section dedicated to observing the effect of reducing the dummy stiffness in the network is an additional study to investigate whether deviating from the core idea of the network could potentially resolve its current issues.

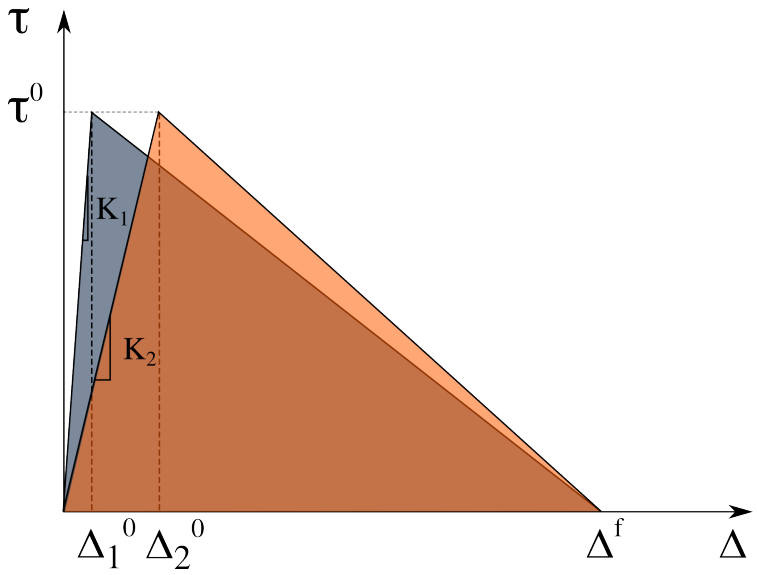


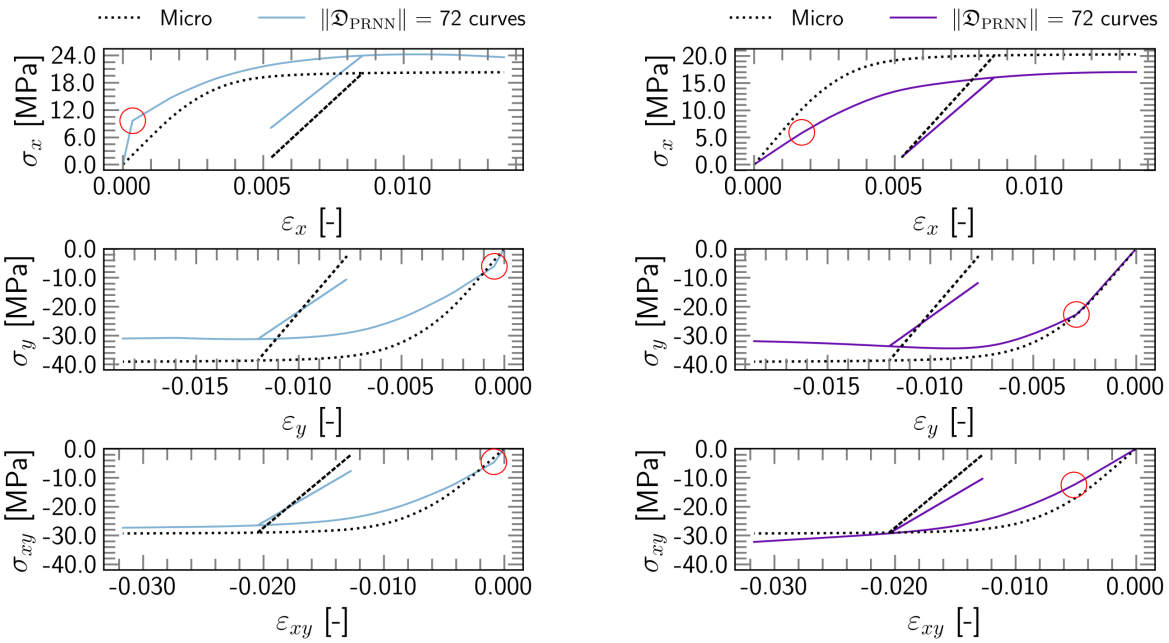
Figure 5.11: The effect of changing the dummy stiffness on the constitutive relation

5.2.3. Modified model with lower K

To investigate the effect of reducing the dummy stiffness used in the cohesive zone model in the network, the PRNN with 2 bulk and 3 cohesive points was selected. The dummy stiffness was reduced by two orders of magnitude, while other parameters were kept the same. The lowest validation MSE was found to be 12.3 MPa, which is not a significant reduction compared to the one resulting from the network with the original dummy stiffness (12.8 MPa). The network's prediction was evaluated on the same test set as in the foregoing chapters. Figure 5.12 shows the two PRNN's predictions on the same test curve using the original and the reduced dummy stiffness.

Two highlights can be drawn from Figure 5.12b. First, the kink becomes less prominent, resulting in a significantly smoother initial monotonic region. Second, the appearance of the kink is delayed, occurring at a later timestep compared to Figure 5.12a. The impact of reducing the dummy stiffness used in the network is clearly visible, the smoothness of the curves is indeed largely influenced by the dummy stiffness of the cohesive zone model. However, this reduction in dummy stiffness does not lead to an improvement in the network's overall accuracy. The average MSE on the test set remains relatively similar to the case with the original dummy stiffness, with a slight decrease from 9.27 MPa to 9.25 MPa. Thus, reducing the dummy stiffness not only creates a gap between the physics behind the network and FE^2, it also technically does not resolve the issue of kink. It only mitigates its appearance and delays it.

The fundamental problem causing the kink phenomenon remains unresolved, which exposes an important inconsistency between the architecture of the network and the underlying physics in the full-order solution. In the FE^2 method, introduced in Section 2.1, the micro-scale tractions computed by interface elements interact with the bulk points in the equilibrium equations of the RVE. However, these micro-scale tractions do not contribute to the homogenized macro-scale stresses in the RVE, only the micro-scale stresses from the bulk points.



(a) Network with original dummy stiffness in cohesive model

(b) Network with reduced dummy stiffness in cohesive model

Figure 5.12: Prediction of networks with 2 bulk and 3 cohesive points with different dummy stiffness trained on 72 GP curves on test curves with one cycle of unloading

In the PRNN discussed thus far, all types of nonlinearities were incorporated in the same material layer with the micro-scale tractions connected to the output with a dense layer. This arrangement does not allow for direct interaction between the micro-scale tractions and the bulk points, and as supported earlier by the kink phenomenon, the dense connection to the output layer poses some challenges. The current network architecture does not adequately represent the link between the cohesive and bulk points, nor does it capture the homogenization procedure.

This conclusion leads to another important realization: while incorporating all sources of nonlinearity of the micro-model is necessary, it is not sufficient. The architecture of the PRNN also must be carefully adjusted to better represent the underlying physics of FE^2 modeling. This conclusion leads to raise the question of whether there is an architecture of the PRNN with the plasticity model and cohesive zone model that is able to capture microscale damage. The following chapter focuses on designing different architectures of the PRNN to better represent the computation in the full-order micro-model, and on studying whether the proposed architectures are able to accurately predict micro-scale damage.

6

Modified architectures

The architecture proposed in Chapter 5 was not able to effectively capture micro-scale damage. When the cohesive points are incorporated in the material layer in the same way as the bulk points, the network is unable to accurately represent the loss of stiffness that results from micro-scale damage. This section investigates different possible architectures to improve the physical representation of the BVP. The goal is to design an architecture that creates a connection between the cohesive and bulk points so that this link is a better reflection of what happens in the RVE when interface elements are considered. For ease of implementation of the modified architectures, PyTorch with automatic differentiation was used to construct the modified networks explored in this section.

For that purpose, we take a step back and conduct a focused analysis. First, training and testing with only a single curve is considered. This approach allows us to observe the network's capability to learn and represent the details of that specific curve before a richer training set is considered. It is important to note that this analysis leads to highly overfitting the data. However, the purpose of the initial phase of this study is to evaluate whether the specific type of architecture proposed is able to learn the behavior of the microstructure.

In this section, five different architectures were considered, from which one of these consists of the architecture shown in Section 5 to serve as a reference, and the other four are new designs. To conduct the stepping stone for this analysis, a curve was selected from the random, non-monotonic, and proportional dataset with one unloading cycle. Networks with different architectures were trained on this individual curve, and their ability to fit it was observed. Once it is established that the network with the chosen architecture can deliver accurate predictions, the problem can be scaled up to larger datasets.

6.1. Bulk and cohesive points in the same material layer

The performance of the network on a larger amount of data was seen in Chapter 5. It is interesting to investigate how well the network is able to predict one curve only. One bulk and cohesive point is considered for this analysis, which resulted in the lowest MSE of 2.15 MPa. The network's prediction on the curve that it was trained on is shown in Figure 6.1.

The network demonstrates excellent accuracy on the unloading branch, indicating that it is able to learn a part of the micro-structural behavior. However, the accuracy of the monotonic region seems to be compromised, and the kink phenomenon discussed in the previous chapter is also prominent. Additionally, training this network on a richer dataset would lead to significant errors, as demonstrated in Chapter 5.

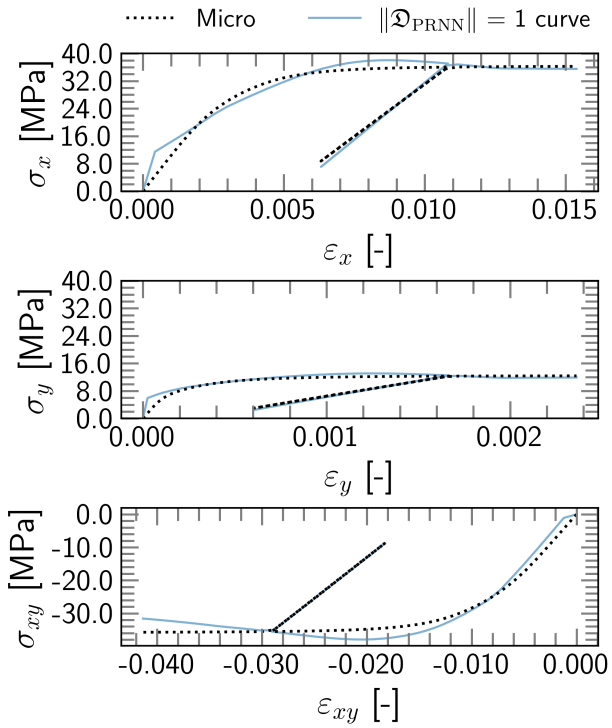


Figure 6.1: Prediction of network with one bulk and cohesive point from the previous densely connected architecture on training curve with one cycle of unloading

6.2. Input to bulk points from cohesive points

To better represent the computation in the FE^2 method, all the architectures proposed in this section involve using the outputs of the cohesive points as input to the bulk points. The input of the network (macroscopic strains) are connected to the cohesive points in different setups. Then, the output of the cohesive points and the macroscopic strains are densely connected to the bulk points. The last layer of the network converts the outputs of the bulk points (micro-scale stresses) to the final output of the network (macroscopic stresses).

By using the outputs (either tractions or the history variable damage) of the cohesive points as an input to the bulk points, the idea is that the bulk points receive a modified local strain value other than the global value, allowing for accounting for a different local stress value to occur. This method also ensures that only the bulk points contribute directly to the stress homogenization procedure, unlike in the architecture from Section 6.1 with the cohesive points connected to the output directly by the decoder layer.

6.2.1. Tractions as input to bulk points

In the first new architecture, the tractions of the cohesive points are input to the bulk points. The layout of the architecture can be seen in Figure 6.2. The idea behind this new design is for the cohesive points to interact with the bulk points without directly contributing to the homogenized stress output.

However, difficulties arise during training of this network. In the initial steps of training, the tractions computed by the cohesive zone model are several orders of magnitude higher than the global strain input of the network. When linearly combined, the values obtained from the cohesive points dominate the input to the bulk points which have the plasticity model nested in them. These points then immediately undergo plastic deformation due to the large initial input values, and the model is often unable to compute the local stresses. Therefore no results are obtained for this architecture.

One solution could be to decrease the initial weights connecting the cohesive points to the bulk points to match the order of magnitude of the input global strains. On one hand, this would lead to comparable output values for the shear component, but on the other hand, it would showcase an issue

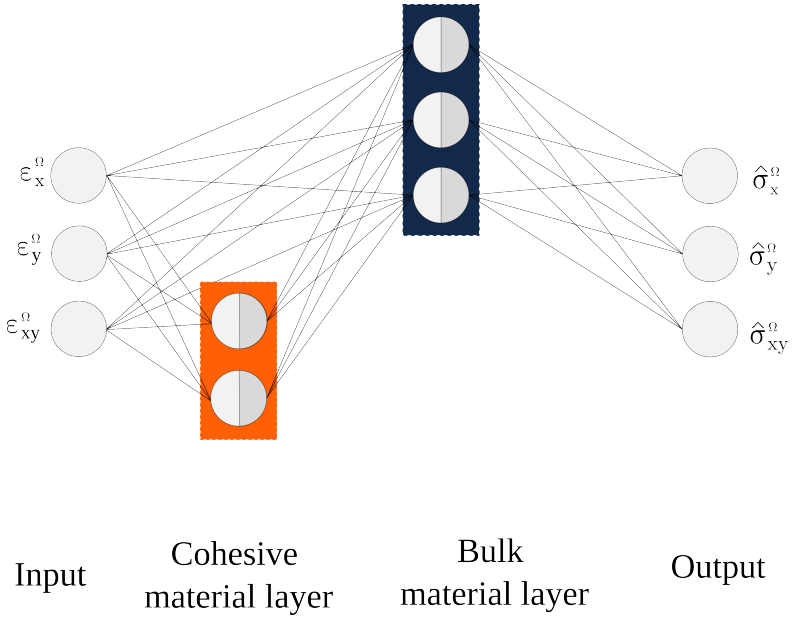


Figure 6.2: Layout of a modified architecture with traction as input to bulk point

related to the non-monotonic nature of the output of the cohesive points. The traction forces initially increase before damage initiation, and once the displacement jump surpasses Δ_0 (Figure 3.3b), the traction values decrease. This creates a discontinuity, which is also present when unloading/reloading takes place. Moreover, a larger weight reduction is necessary if the cohesive point is in compression due to the high negative traction values (Figure 3.3a). Therefore, a more suitable solution is needed.

6.2.2. Damage as input to bulk points

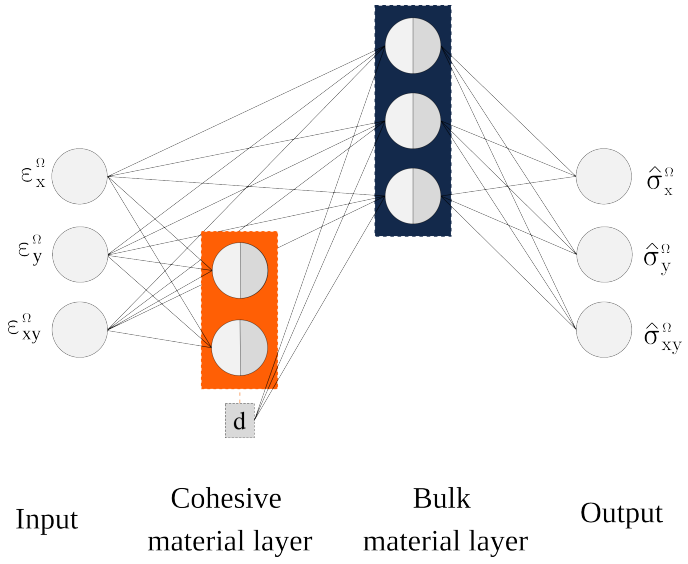


Figure 6.3: Layout of a modified architecture with damage as input to bulk point

To overcome the discontinuity posed by passing tractions to the bulk points, the damage from the cohesive zone model is used instead. Damage is the internal variable stored in the cohesive points, which is either increasing or remains the same in case of unloading. This damage value modifies the local strain value that the bulk points receive, resulting in a modified value of local stresses for the same level of macroscopic strain. This gives rise to a modified tangent stiffness matrix that could capture the decrease in stiffness during unloading. The layout of the architecture can be seen in Figure 6.3, where

in this case all bulk points receive the input damage from all the cohesive points when enlarging the network.

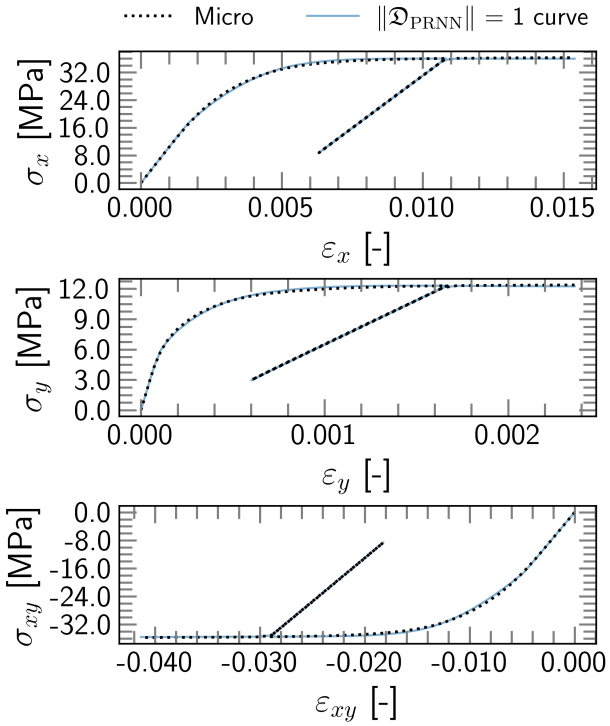
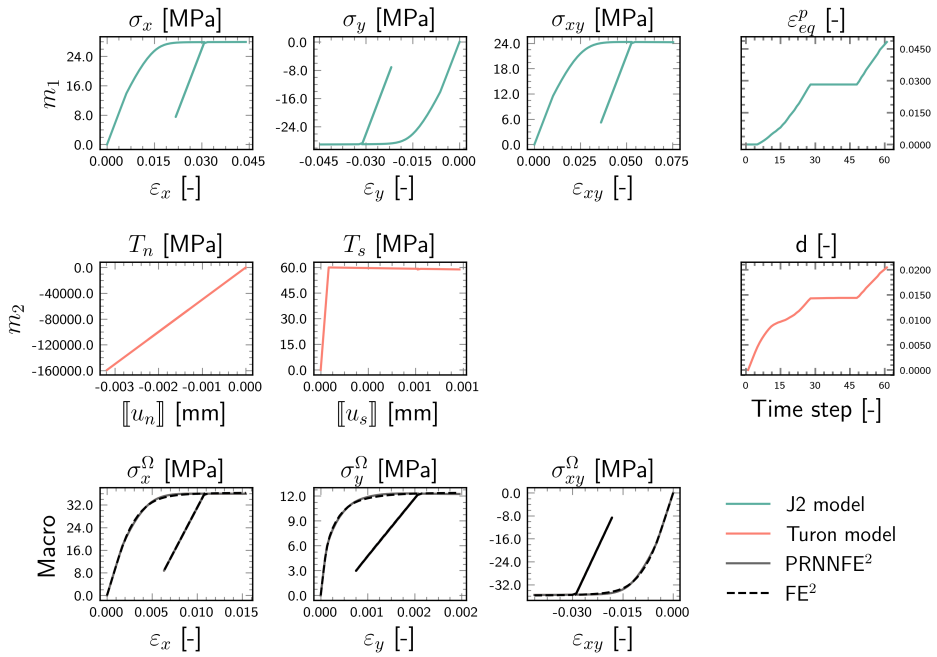


Figure 6.4: Prediction of network that feeds damage to bulk points with one bulk and cohesive point on training curve with one cycle of unloading

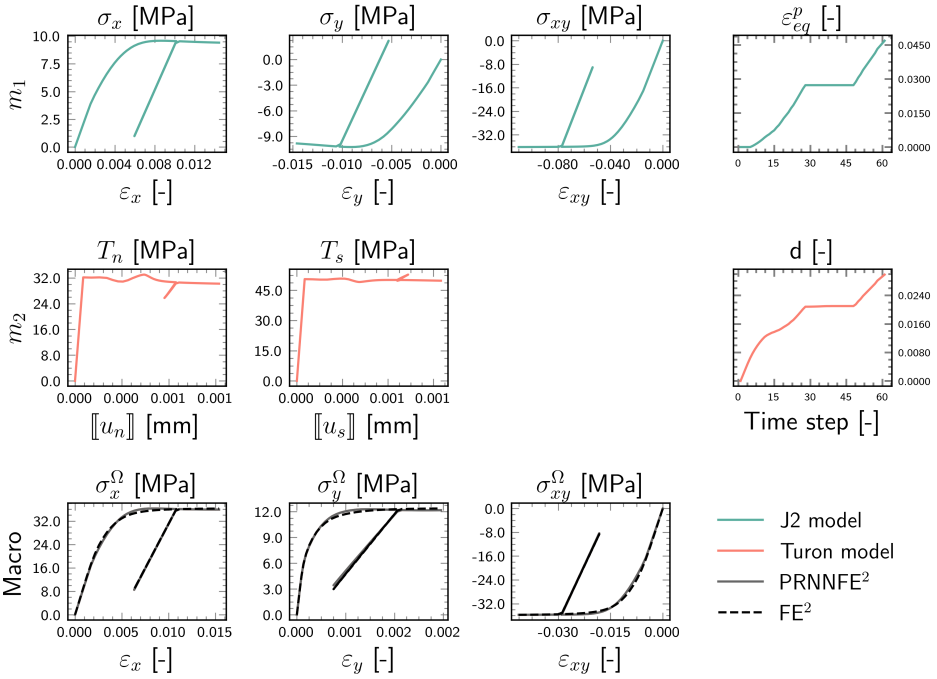
The prediction of this network on the curve it was trained on is shown in Figure 6.4. The accuracy of the prediction is remarkable with an error of 0.23 MPa, which suggests that the network with this architecture can account for the damage of the microstructure when considering one curve only. The response of the material points for two different initializations can be seen in Figure 6.5, where it can be observed that the network can provide accurate predictions with the cohesive point being either in tension or compression.

Before enriching the training dataset with non-monotonic curves, it is interesting to investigate its performance with increased proportional, non-monotonic curves. The training data was first increased to two non-monotonic curves. This led to an increase in training error (1.87 MPa), and there was no significant difference between the error of the two curves it was trained on. The prediction of the network on one of the curves it was trained on is shown in Figure 6.6a, where it can be observed that he accuracy of the network decreased, but the network was still able to predict the unloading behavior. When the training data was increased to four non-monotonic curves, the network with one bulk and one cohesive point was no longer able to provide accurate predictions on the training curves (5.33 MPa). Therefore, the number of material points was increased keeping the proportion of bulk and cohesive points even. The network with 3 bulk and cohesive points was found to be the most optimal in this case, with an error of 2.71 MPa. The network's prediction on the two curves it was trained on had no significant difference regarding accuracy. One of these predictions can be seen in Figure 6.6b, which showcases that the accuracy of the prediction further decreased. This initial study emphasizes the need for a model selection procedure for increased training set sizes.

In order to be able to generalize the network so that it can predict different loading cases, training with curves from the non-proportional dataset is recommended. Following the same methodology as before, the network was first trained with one curve only to evaluate whether it can capture the effect of microscale damage. This time, the network with one bulk and one cohesive point loses accuracy, which prompted to study different material layer sizes to observe its effect on the PRNN's performance. For that, the number of bulk points was kept equal to the number of cohesive points, and the lowest training



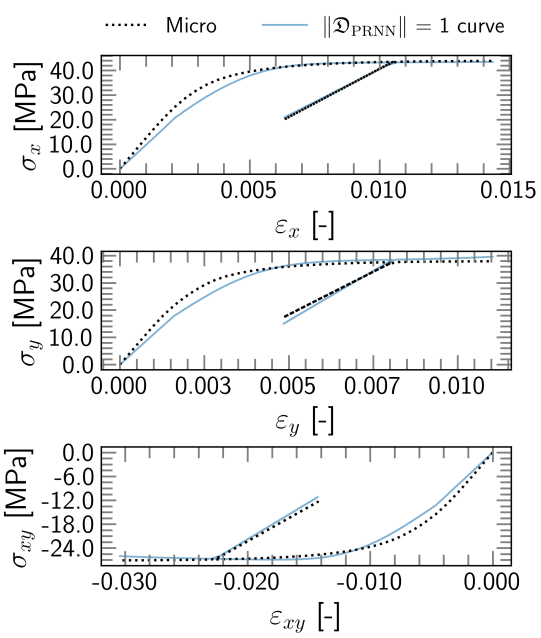
(a) Cohesive point in compression



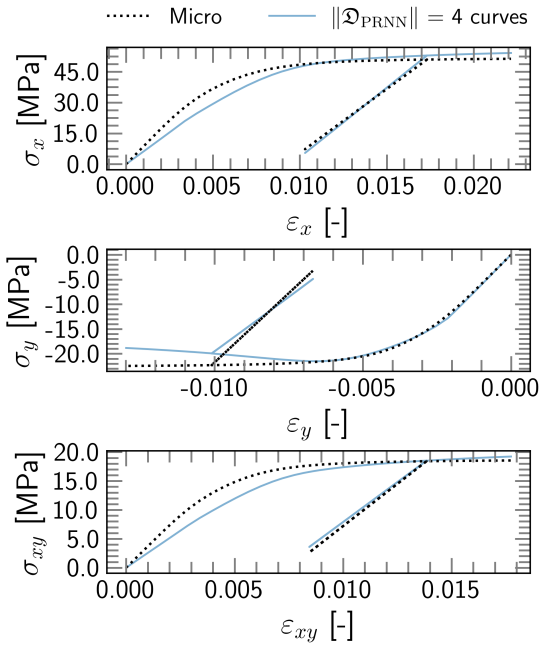
(b) Cohesive point in tension

Figure 6.5: Micro and macro response of networks with different initializations that feeds damage to bulk points with one bulk and cohesive point on training curve with one cycle of unloading

error was found to be 4.36 MPa with a configuration of 3 bulk and cohesive points. The prediction of this network on the curve it was trained on along with the response of the cohesive points can be seen in Figure 6.7. Besides the relatively large error, there is an apparent problem with the smoothness of the curve in the initial region. One of the reasons being the early initiation of damage that can be observed in material points m_4-m_6 in Figure 6.7, where the damage reaches a value of 1 already at the second time step, distorting the solution. Additionally, no significant difference was found among the



(a) Network with 1 bulk and cohesive point trained with 2 curves



(b) Network with 3 bulk and cohesive points trained with 4 curves

Figure 6.6: Prediction of networks trained on non-monotonic curves with one cycle of unloading on training curves

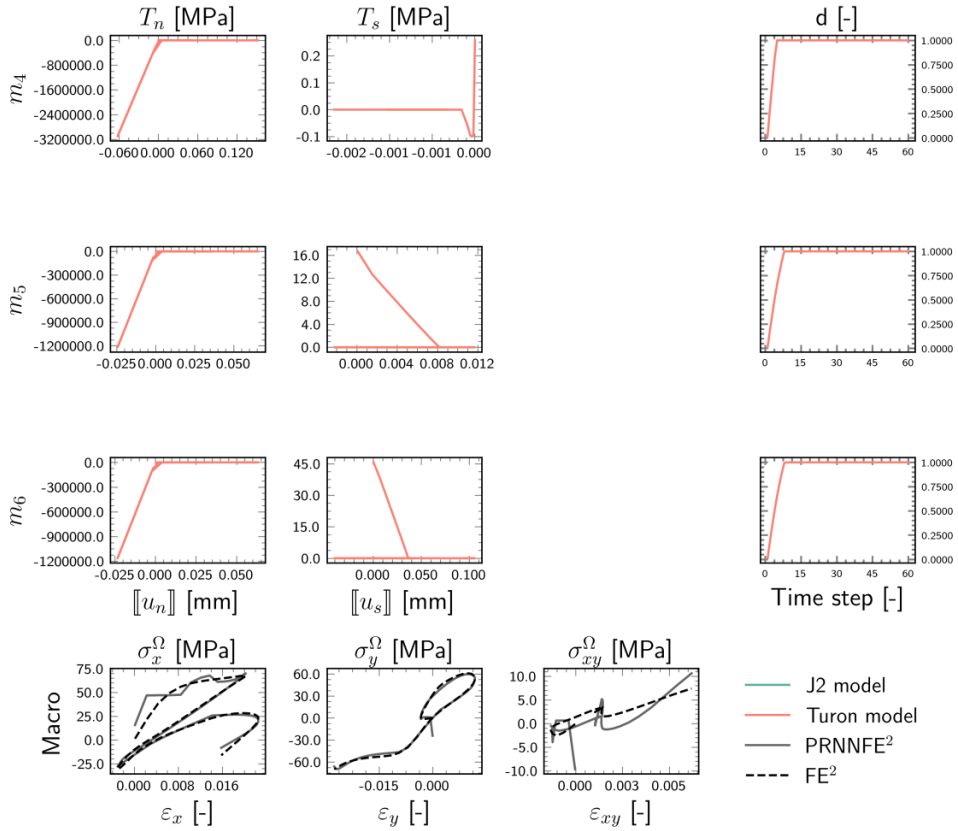


Figure 6.7: Micro and macro response of network that feeds damage to bulk points with three bulk and cohesive points on non-proportional training curve

different sizes of the material layers.

An additional observation can be made: Figure 6.7 illustrates that all three cohesive points are in compression. Once the material point enters the compression phase during training, the point gets stuck in it for the rest of the training. This is due to the fact that in the normal component, the displacement jump resulting from compressive force is set to zero in the Turon cohesive zone model. Therefore, perturbing the weights and biases associated with the cohesive point does not influence the output of the network thus the gradients become zero. This suggests that initialization of the networks plays an important role.

Though challenges appear when training the network with the non-proportional curve, a larger and richer training set was considered to observe whether the issue with the early damage initiation persists. Since the goal of this study is to get an initial idea of how enriching the training dataset influences the behavior of the network, and whether increasing the number of material points has an effect on network accuracy, the training size and the number of material points were kept small (between 2 and 3 points of bulk and between 3 and 6 cohesive points). Training was carried out with 18, and validated on 54 curves from the non-proportional dataset.

From the model selection procedure, a network with a combination of 2 bulk and 3 cohesive points was selected to illustrate results with a validation error of 16.2 MPa. For the test set, 54 curves from the non-monotonic, proportional dataset with one cycle of unloading, which were considered in the previous sections were kept. The prediction of the network on a representative test curve can be seen in Figure 6.8, and as expected, the accuracy of the network is not sufficient. However, interesting observations are made when looking at the figure, where the response of the fictitious cohesive points is illustrated: the damage no longer reaches a value of 1 already at the second time step, and not all cohesive points are in compression, as observed for previous cases.

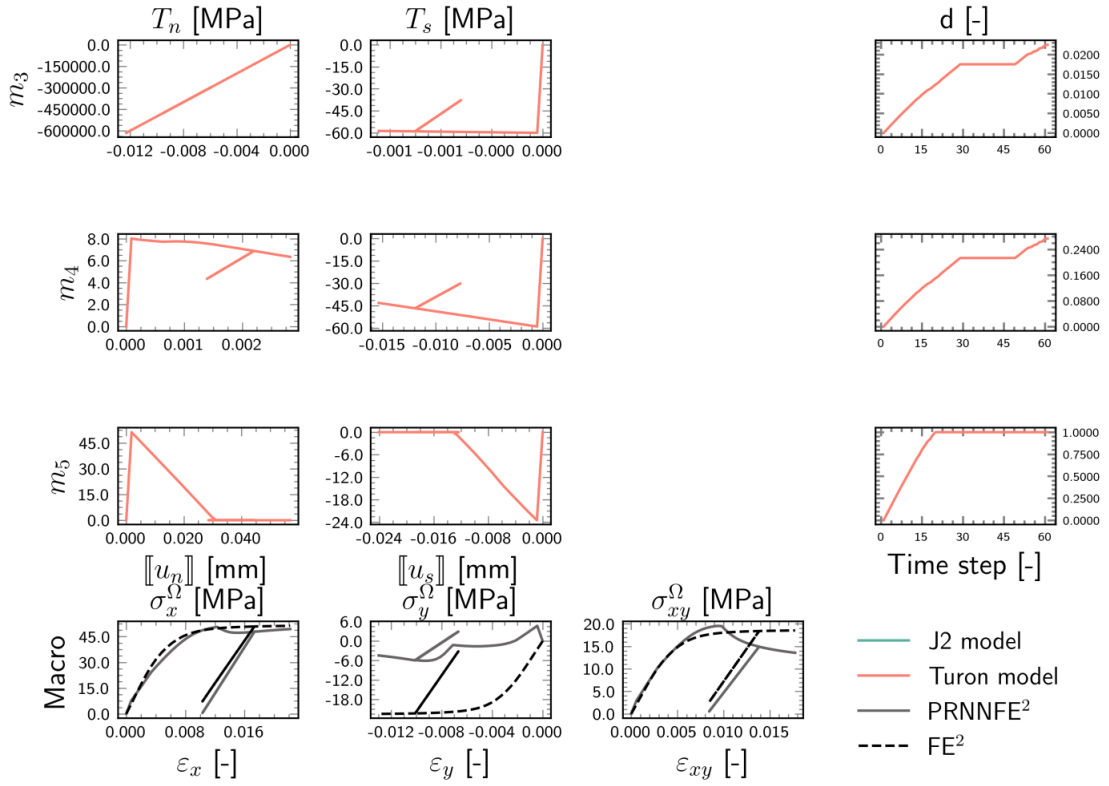


Figure 6.8: Micro and macro response of network that feeds damage to bulk points with two bulk and three cohesive points trained on 18 non-proportional curves, on a representative curve from test set

The issue of the early initiation of damage leading to a large influence on the accuracy of the network prediction needs to be addressed so that the good performance observed previously for smaller datasets can be replicated. Furthermore, there was no significant decrease in validation error due to increasing the number of material points. Recall that in the current architecture, a single linear layer

connects the macroscopic strains to local displacement jumps, while in reality, this relation is highly non-linear. In the micromodel, damage is not likely to initiate at the first time step already, but later as larger strain values develop. In this setup, damage is already developing at the first time step, and quickly escalates. Therefore, adding nonlinearity to account for this delayed damage initiation is explored. This option is investigated in two different ways, by either adding a hidden layer with a nonlinear activation function before the cohesive layer or by applying nonlinearity directly at the cohesive points.

6.2.3. Damage as input to bulk points with a hidden layer before cohesive points

To account for the nonlinear relation between macroscopic strains and local displacement jumps, a hidden layer of size 18 with a hyperbolic tangent activation function is added before the cohesive point. This allows for an arbitrary nonlinear relation but with little physical interpretation behind the chosen activation function. The layout of the architecture can be seen in Figure 6.9.

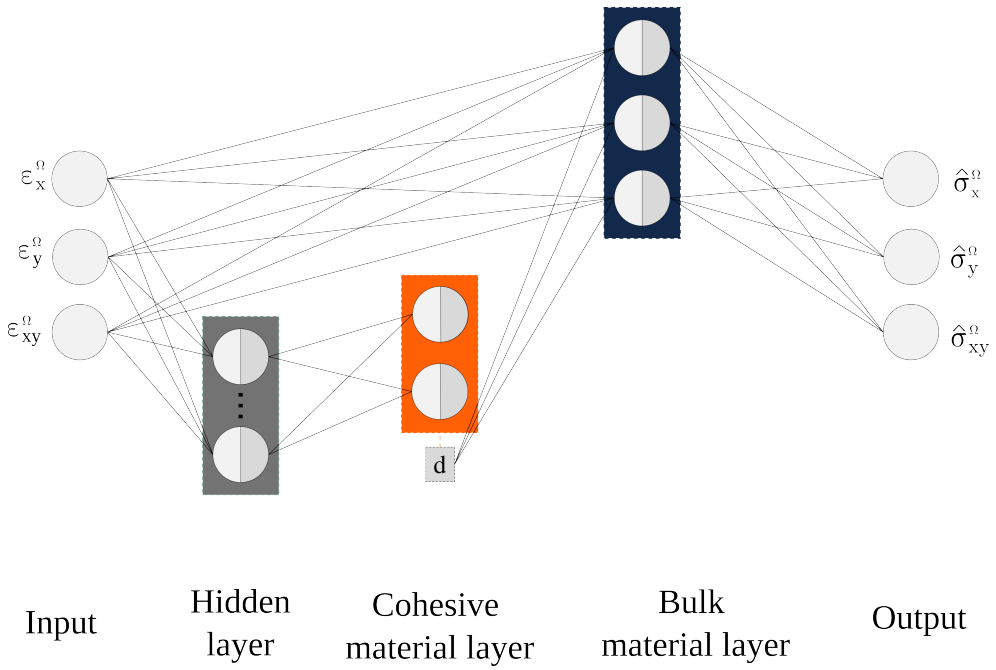


Figure 6.9: Layout of a modified architecture with damage as input to bulk point with a hidden layer before the cohesive point

To compare the response of this architecture to the previous cases in Section 6.2.2, training and testing on one non-proportional curve is again explored. Different material layer sizes were studied by setting the number of bulk points equal to the number of cohesive points. A network with 5 bulk and cohesive points had the prediction with the lowest error (3.47 MPa). The micro-scale response of the cohesive points which are all in compression, and the macro-scale response of the network are shown in Figure 6.10. The same problem seems to persist, the damage evolution is sudden in all cohesive points which leads to large strain values in the bulk point. To solve this problem, a new way of introducing nonlinearity with a more physical meaning behind it is considered so that it can be more easily manipulated to better understand the underlying issue.

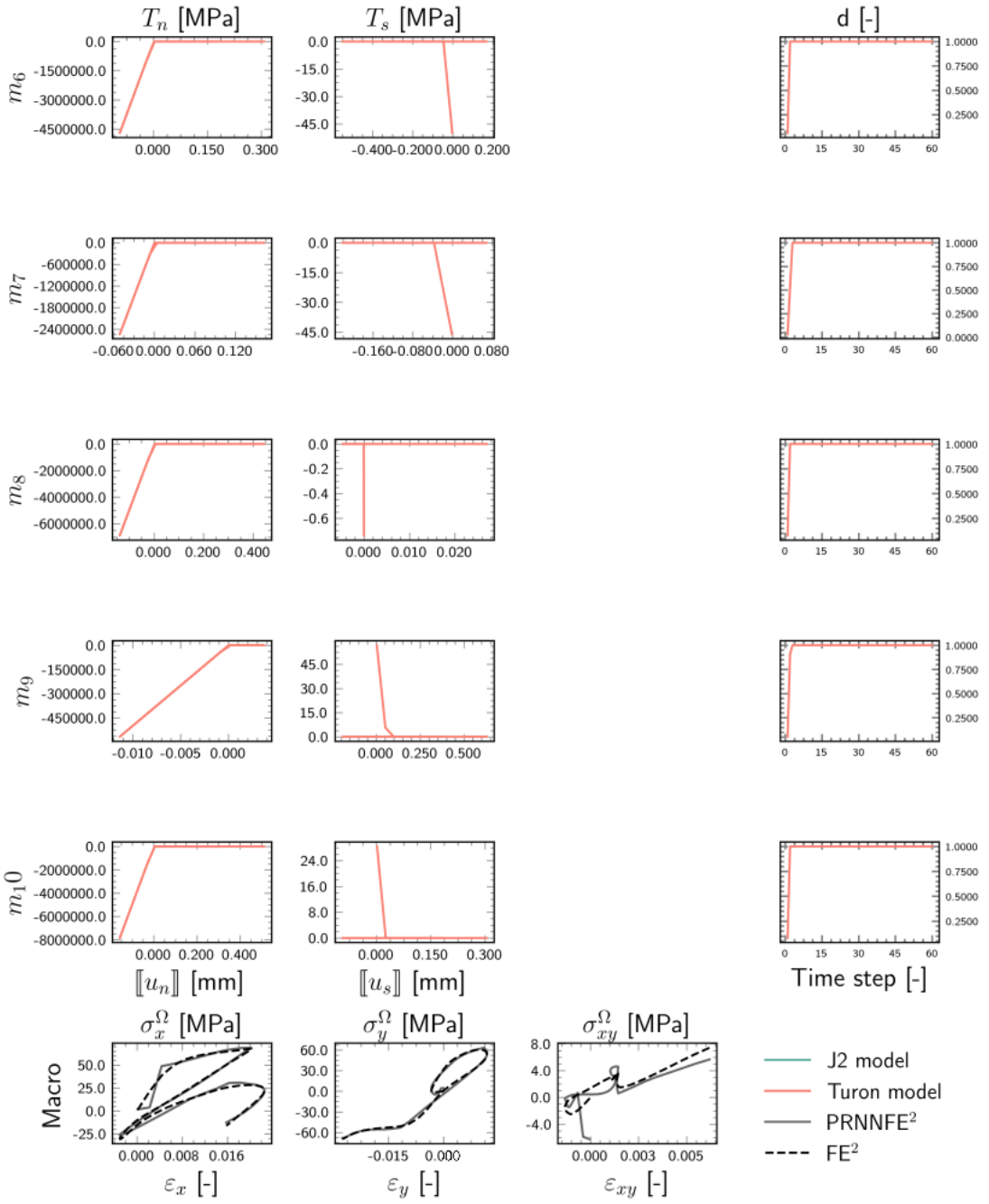


Figure 6.10: Micro and macro response of network that feeds damage to bulk points with one bulk and cohesive point, with a hidden layer before the cohesive point, on non-proportional training curve

6.2.4. Damage as input to bulk points with Leaky ReLU at cohesive points

Next, the non-linearity to represent the relation between global strains and local displacement jumps is applied directly at the cohesive point. The cohesive points are densely connected with the input layer through a linear relation and the resulting values of that are activated with a leaky ReLU function to obtain the final local displacement jumps. In that activation function, a bias term is considered. The idea of using the bias is to replicate the delayed displacement jump compared to the applied strain since the term allows for a shift of the function.

The choice for this specific activation function comes from the fact that the standard ReLU on its own does not allow the gradients to update for negative strain values, since the function equates the negative strain values to zero displacement jumps. For the gradients to update, it is necessary to apply

a small slope for negative strain values, but it is important to keep it to a very small value close to zero so that the model does not allow negative displacement jumps to occur in the normal component. Therefore, Leaky ReLU is used instead of standard ReLU. To work optimally, the slope of the function and its shift in the x direction are implemented in the network as a learnable parameter. In addition to this, a slight modification of the Leaky ReLU activation function is needed. If the function is applied as is at the cohesive points, the displacement jumps that are passed to the cohesive points will be calculated by the following equation:

$$[\![\mathbf{u}]\!] = \text{leaky\_relu}(\mathbf{w_1} * \boldsymbol{\epsilon}^{\omega} + \mathbf{b_1}) \tag{6.1}$$

with

$$\text{leaky\_relu}(x) = \begin{cases} x & \text{if } x \geq 0.0 \\ 0.01 \cdot x & \text{if } x < 0.0 \end{cases} \tag{6.2}$$

applied component-wise. In this method, however, the bias does not act as the value by which the function is shifted in the x direction, which was originally intended. Therefore the function has to be modified so that the activation function is calculated by:

$$\text{leaky\_relu\_mod}(x, w, b) = \begin{cases} w \cdot (x - b) & \text{if } x \geq b \\ 0.01 \cdot w \cdot (x - b) & \text{if } x < b \end{cases} \tag{6.3}$$

applied piece-wise. The difference between the standard Leaky ReLU and its modifications using Eqs. (6.2) and (6.3) with an arbitrary bias of 0.01 and weight of 0.5 is shown in Figure 6.11.

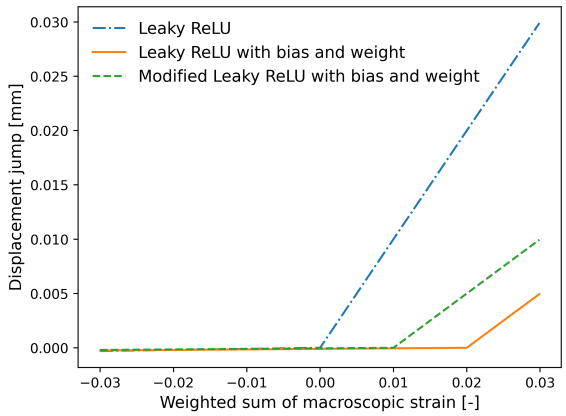


Figure 6.11: Difference between discussed Leaky ReLU functions for arbitrary bias and weight value

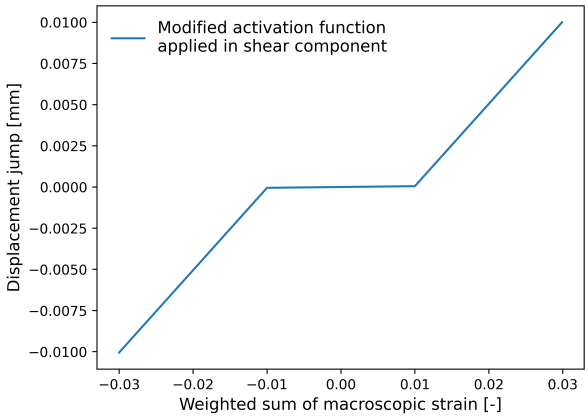


Figure 6.12: Function applied on the shear component of cohesive points for arbitrary bias and weight value

It is important to mention that this method of shifting the Leaky ReLU function is mainly applicable to the normal component of the displacement jump. To account for the delay in damage initiation in the shear component, a different modification is proposed. For small absolute strain values, a small slope is applied to obtain displacement jump values close to zero. For strain absolute values larger than the bias parameter, the slope of the curve is defined by the weight parameter corresponding to the modified function. The function applied in the shear component of the cohesive point is shown in Figure 6.12 and is described by:

$$\text{leaky\_shear}(x, w, b) = \begin{cases} w \cdot (x - b + 0.01 \cdot b) & \text{if } x \geq b \\ w \cdot (x + b - 0.01 \cdot b) & \text{if } x \leq -b \\ 0.01 \cdot w \cdot x & \text{otherwise} \end{cases} \tag{6.4}$$

For physical interpretation of the activation function, the weight and bias values corresponding to the activation function at the cohesive points were forced to obtain positive values by applying a SoftPlus activation on the parameters. This, in theory, allows for a delay in damage initiation which corresponds to a positive bias value, and an increase of local displacement jump for increasing macroscale strain values.

The network with the modified activation function applied at the cohesive points was trained on the non-proportional curve used to train the network in the previous section. When setting the number of bulk points equal to the number of cohesive points, a network with 5 bulk and cohesive points predicted the training curve with the lowest error (4.58 MPa). This error is slightly higher than the error of the prediction of the network with no nonlinearity added to the macroscale strain-local displacement jump relation (4.36 MPa) and of the prediction of the network with the hidden layer implemented before the cohesive layer (3.47 MPa). However, Figure 6.13 shows that due to the applied nonlinear activation function, damage initiation takes place at a later timestep. This result is promising and the network with this architecture requires further investigation on whether increasing the training dataset with more non-proportional curves improves network accuracy.

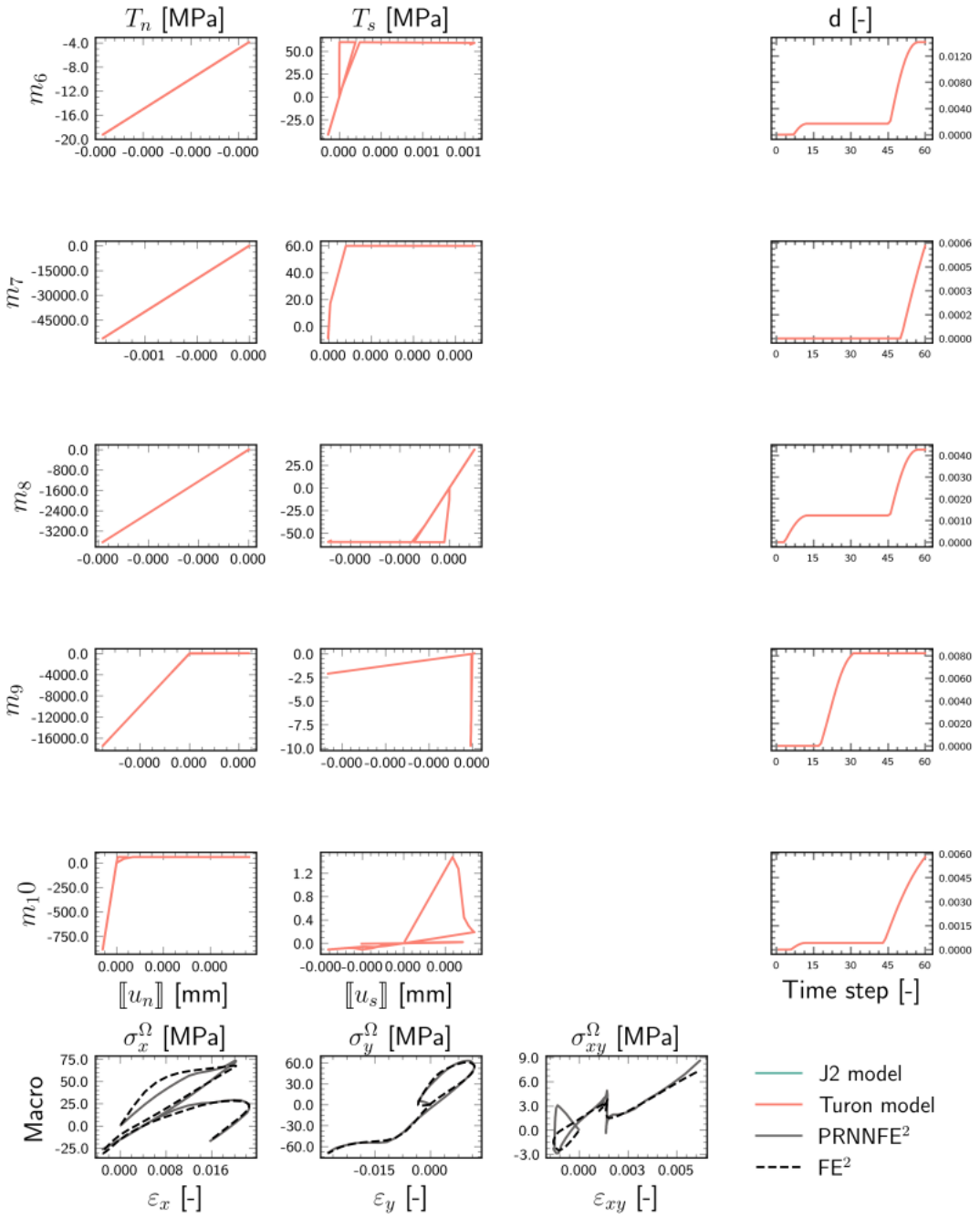


Figure 6.13: Micro and macro response of network that feeds damage to bulk points with 5 bulk and cohesive points, with a custom nonlinear activation function the cohesive point, on non-proportional training curve

7

Conclusions and recommendations

The motivation of this thesis was to extend an existing surrogate model, namely the Physically Recurrent Neural Network (PRNN) [12], to account for microscale damage in composite materials. Investigating the PRNN as a surrogate model was prompted by its exceptional performance when predicting elastoplastic behavior in composite materials. The main idea of the PRNN is that the constitutive relations in the full-order micromodel are directly implemented in the hidden layer of the network which allows for a direct link to the micromodel. Path-dependency naturally arises from the material models in the network, which leads to accurate predictions on a significantly smaller set of training data than what is required for networks without physical interpretation. The hidden layer of the existing PRNN comprises bulk material points with elastoplastic material models embedded in them. The objective of this thesis is to extend the existing PRNN to predict the effect of debonding at the fiber-matrix interface.

For this, the first attempt at using this framework for microscale damage consisted of applying the network with bulk material points only, as it was proposed in [12]. A brief parameter study was conducted to investigate whether the network without all nonlinear models included in the RVE can predict the effect of microscale damage. Following the findings and methodology for testing with bulk models only, the PRNN was then extended with cohesive points in the material layer. The loss of prediction accuracy due to the influence of the dummy stiffness in the cohesive zone model prompted a study on the effect of changing this variable.

Next, new architectures of the PRNN with bulk and cohesive points were proposed. The performance of these networks was evaluated by focusing on one training curve only. In this chapter, a summary of the performance of the proposed PRNNs is presented in the context of their alignment with the primary research objective outlined in this thesis. Additionally, the limitations of the PRNNs are discussed along with recommendations for future work.

7.1. Conclusions

The PRNN without the cohesive zone model introduced in Chapter 4 was not able to capture microscale damage. The network, in its current state, was only able to learn one out of two types of nonlinearities present in the full-order micromodel. The only source of nonlinearity in the network comes from the elastoplastic material model, while the full-order micromodel also accounts for damage. This is reflected when the prediction of the network was compared to the high-fidelity solution. It was observed that the network had a good fit on the monotonic region, but unloading was predicted with the elastic stiffness which comes from the elastoplastic material behavior. Increasing or enriching the training data did not lead to significant improvements. This is in line with the general guideline in [12] that all types of nonlinearities present in the RVE need to be included in the network. Therefore, a new architecture with the cohesive zone model implemented in the PRNN is required.

For the initial design, cohesive points with the CZM was incorporated into the same material layer of the PRNN as the plasticity model. This PRNN was introduced in Chapter 5, and it was not able to capture microscale damage. First, when trained on monotonic data, the network was able to predict accurately only on the monotonic part of the test curves. Since the second type of nonlinearity (damage) was not seen during training, the cohesive points of the network remained inactive. This resulted in a

set of weights that minimized the contribution of the cohesive points to almost zero, which resembles the performance of the network with bulk points only. Therefore an important conclusion was drawn: it is not enough to include cohesive points in the material layer, they also have to be activated during training in order to have a significant contribution to the output of the network. Therefore, training with non-monotonic data is required.

When trained on non-monotonic and non-proportional dataset, the network performed well in some cases. However, the accuracy was not consistent among the test set and the prediction of the PRNN lost accuracy on the monotonic region, where a discontinuity was observed in a form of a kink. This kink resulted from the way the stress homogenization is represented in the network. 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 combination of the outputs of each material point. When the weights connecting the cohesive points to the output increase, the high dummy stiffness of the cohesive zone model becomes significant and causes a large initial stiffness of the global output. This phenomenon prompted a study on the effect of decreasing the dummy stiffness of the CZM in the network. However, reducing the dummy stiffness contradicts the fundamental idea behind the PRNN, since the material models used in the full-order micromodel are now different from the one in the network, and therefore it is undesirable to follow this path further than as a complementary study.

In this study, the value of the dummy stiffness was decreased by two orders of magnitude. The prediction of the network on the monotonic phase is visually smoother, but the accuracy of the network prediction did not improve. The kink became less prominent, however, it did not disappear. This led to an important conclusion: the underlying problem lies in the current architecture of the network. The layout of the PRNN does not resemble the physics of the full-order solution, where only the bulk points contribute to the stress homogenization. In contrast to this, the network predicts the stress output as a linear combination of both stresses coming from the bulk models and tractions coming from cohesive zone models. Therefore, it was concluded that a new architecture of the network that includes cohesive and bulk points is necessary.

New architectures were proposed in Chapter 6 based on the conclusions drawn in the previous chapters. The networks first were trained on a single curve with one cycle of unloading for a more focused study. When the internal variable of the cohesive point (damage) together with macroscale strains were weighted together and used as input input to the bulk points, the PRNN with one bulk and cohesive point was able to accurately capture the loss of stiffness during the unloading cycle on its training curve. The fit of the monotonic region was also remarkable. This promising result indicates that the network, in this configuration, is able to capture microscale damage.

Additional curves were introduced to the training data to observe the PRNN's performance on a more diverse dataset. The training data was first increased to two and four proportional, non-monotonic curves. This initial exploration with proportional curves suggested that increasing the material points could mitigate the loss of accuracy when training size is increased, given accurate prediction on a single curve. Then, the network was trained on a single non-proportional curve to extend the research to a more complex scenario where unloading takes place at different loading steps for different duration of time. An increased number of material points was needed to represent this curve, and a larger training error was observed than for the non-monotonic but proportional case. When training data was increased, changing the material layer size did not have a significant influence on the network's accuracy. It is important to note however that a relatively small number of material points (a maximum of 3 bulk and 5 cohesive points) were considered in this initial phase compared to the number of points used to training with one curve only.

Another important feature was highlighted when training with non-monotonic curves. For the case of training with one curve, the cohesive points indicated fully damaged condition at the second timestep. This was not the case when the training dataset was enlarged, however, damage was still initiated in the second time step. This observation led to investigating the input of the cohesive points more closely. In this architecture, the macroscale strains are connected to the local displacement jumps by a linear transformation, which does not represent the actual damage evolution in the micromodel. In reality, the material is not necessarily damaged right after loading, instead, this relation is highly non-linear. This resulted in exploring solutions to account for the highly nonlinear relation between macroscale strains and local displacement jumps.

A hidden layer with nonlinear activation functions was implemented between the input and the cohesive material layer to better represent their nonlinear relation. The issue with the fully damaged mi-

crostructure at the second timestep persisted, therefore a modified function with more physical meaning was introduced at the cohesive points instead. When trained on a single non-proportional curve, the damage evolves at a later timestep which showcases the effect of the function and shows potential to be applied for an increased training dataset.

Though it cannot be concluded yet that the architecture found is able to capture microscale damage when it is trained on a larger dataset, the results suggest that further investigation of this layout will lead to finding the optimal network that is able to accurately capture the microscale damage with the right mitigation strategy.

7.2. Limitations and recommendations

Understanding the challenges and limitations of employing PRNNs as surrogate models for microscale damage prediction is crucial. One such challenge is associated with the nature of the cohesive zone model. There is a small range of input values to the cohesive model for which the parameters associated with the layer connecting the input and cohesive material layer are being updated during training. As seen in the previous chapter, once the damage reaches a value of 1, the gradients of these parameters vanish (i.e. become zero) and no new information is learned from the cohesive point after this point. To overcome this difficulty, a different traction-separation law could be implemented. For instance, a traction-separation law with exponential decay would ensure that the damage value never reaches 1 but a value close to it which would allow for the model to never enter a fully damaged state.

Another case when gradients vanish is when the fictitious cohesive point is in compression. Then, the parameters associated with the normal component of the cohesive point no longer update as changing this value will not contribute to the damage parameter of the CZM. This leads to utilizing only the shear component of the cohesive point when using damage as an input to the bulk points and makes the model highly sensitive to initialization. A more extensive study on the initialization is therefore recommended.

An additional challenge surfaced when implementing the PRNN in PyTorch compared to C++. The computational time related to training the network significantly increased, which limited the combinations of material layer size and the size of the training set considered in Chapter 6. Therefore, valuable information would be obtained if a study on the effect of increasing the material layer size with more points were conducted. This may show whether the network is able to provide accurate predictions without introducing more complexities to the problem, such as nonlinearities. If that is the case, implementation of the network discussed in Section 6.2.2 in C++ is suggested to be able to conduct a more in-depth analysis with a larger training set if needed.

Additionally, it is interesting to investigate whether the PRNN, in any of its architectures, is able to predict strain localization if either a predefined crack path or distributed cracking is allowed in the full-order micromodel.

Once the proposed physically recurrent neural network is optimized and accurate prediction is established on the microscale, it can be applied in a multiscale setting to evaluate the overall response of an arbitrary structure.

Bibliography

[1] Alan Baker, Stuart Dutton, and Donald Kelly. *Composite Materials for Aircraft Structures*. American Institute of Aeronautics and Astronautics (AIAA), 2004. ISBN: 978-1-56347-540-7.

[2] Evan C Bentz, Thanasis C Triantafillou, and Sidney Mindess. “Fiber-reinforced polymer composites for construction–state-of-the-art review”. In: *Journal of Composites for Construction* 13.3 (2009), pp. 219–236.

[3] Karthikayen Raju, Tong-Earn Tay, and Vincent Beng Chye Tan. “A review of the FE2 method for composites”. In: *Multiscale and Multidisciplinary Modeling, Experiments and Design* 4.1 (Mar. 2021), pp. 1–24. ISSN: 2520-8179. DOI: 10.1007/s41939-020-00087-x. URL: https://doi.org/10.1007/s41939-020-00087-x.

[4] E. Monteiro, J. Yvonnet, and Q.C. He. “Computational homogenization for nonlinear conduction in heterogeneous materials using model reduction”. In: *Computational Materials Science* 42.4 (2008), pp. 704–712. ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2007.11.001. URL: https://www.sciencedirect.com/science/article/pii/S0927025607003242.

[5] F. El Halabi et al. “FE2 multiscale in linear elasticity based on parametrized microscale models using proper generalized decomposition”. In: *Computer Methods in Applied Mechanics and Engineering* 257 (2013), pp. 183–202. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2013.01.011. URL: https://www.sciencedirect.com/science/article/pii/S0045782513000224.

[6] M. Cremonesi et al. “A PGD-based homogenization technique for the resolution of nonlinear multiscale problems”. In: *Computer Methods in Applied Mechanics and Engineering* 267 (2013), pp. 275–292. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2013.08.009. URL: https://www.sciencedirect.com/science/article/pii/S0045782513002120.

[7] F. Ghavamian and A. Simone. “Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network”. In: *Computer Methods in Applied Mechanics and Engineering* 357 (2019), p. 112594. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2019.112594. URL: https://www.sciencedirect.com/science/article/pii/S0045782519304700.

[8] Hernan J. Logarzo, German Capuano, and Julian J. Rimoli. “Smart constitutive laws: Inelastic homogenization through machine learning”. In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), p. 113482. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.113482. URL: https://www.sciencedirect.com/science/article/pii/S0045782520306678.

[9] Ehsan Haghighat et al. “A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics”. In: *Computer Methods in Applied Mechanics and Engineering* 379 (2021), p. 113741. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.113741. URL: https://www.sciencedirect.com/science/article/pii/S0045782521000773.

[10] Arunabha M. Roy and Suman Guha. “A data-driven physics-constrained deep learning computational framework for solving von Mises plasticity”. In: *Engineering Applications of Artificial Intelligence* 122 (2023), p. 106049. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2023.106049. URL: https://www.sciencedirect.com/science/article/pii/S0952197623002336.

[11] Ziqi Li et al. “A mechanics-informed machine learning approach for modeling the elastoplastic behavior of fiber-reinforced composites”. In: *Composite Structures* 323 (2023), p. 117473. ISSN: 0263-8223. DOI: https://doi.org/10.1016/j.compstruct.2023.117473. URL: https://www.sciencedirect.com/science/article/pii/S026382232300819X.

[12] M.A. Maia et al. “Physically recurrent neural networks for path-dependent heterogeneous materials: Embedding constitutive models in a data-driven surrogate”. In: *Computer Methods in Applied Mechanics and Engineering* 407 (2023), p. 115934. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2023.115934. URL: https://www.sciencedirect.com/science/article/pii/S0045782523000579.

[13] Kun Wang and WaiChing Sun. “Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning”. In: *Computer Methods in Applied Mechanics and Engineering* 346 (2019), pp. 216–241. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2018.11.026. URL: https://www.sciencedirect.com/science/article/pii/S0045782518305851.

[14] Zeliang Liu. “Deep material network with cohesive layers: Multi-stage training and interfacial failure analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 363 (2020), p. 112913. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.112913. URL: https://www.sciencedirect.com/science/article/pii/S0045782520300967.

[15] Zeliang Liu. “Cell division in deep material networks applied to multiscale strain localization modeling”. In: *Computer Methods in Applied Mechanics and Engineering* 384 (2021), p. 113914. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.113914. URL: https://www.sciencedirect.com/science/article/pii/S0045782521002516.

[16] I.B.C.M. Rocha, P. Kerfriden, and F.P. van der Meer. “On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning”. In: *Journal of Computational Physics: X* 9 (2021), p. 100083. ISSN: 2590-0552. DOI: https://doi.org/10.1016/j.jcpx.2020.100083. URL: https://www.sciencedirect.com/science/article/pii/S2590055220300354.

[17] J. Oliver et al. “Reduced order modeling strategies for computational multiscale fracture”. In: *Computer Methods in Applied Mechanics and Engineering* 313 (2017), pp. 560–595. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.09.039. URL: https://www.sciencedirect.com/science/article/pii/S0045782516303322.

[18] P. Kerfriden et al. “A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics”. In: *Computer Methods in Applied Mechanics and Engineering* 256 (2013), pp. 169–188. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2012.12.004. URL: https://www.sciencedirect.com/science/article/pii/S004578251200374X.

[19] R. Hill. “Elastic properties of reinforced solids: Some theoretical principles”. In: *Journal of the Mechanics and Physics of Solids* 11.5 (1963), pp. 357–372. ISSN: 0022-5096. DOI: https://doi.org/10.1016/0022-5096(63)90036-X. URL: https://www.sciencedirect.com/science/article/pii/002250966390036X.

[20] Charu C. Aggarwal. *Neural Networks and Deep Learning*. Springer, 2018. ISBN: 978-3-319-94463-0. DOI: https://link.springer.com/book/10.1007/978-3-319-94463-0.

[21] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[22] Christopher Olah. *Understanding LSTM Networks*. 2015. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[23] M. Raissi, P. Perdikaris, and G.E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.10.045. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[24] A. Turon et al. “A damage model for the simulation of delamination in advanced composites under variable-mode loading”. In: *Mechanics of Materials* 38.11 (2006), pp. 1072–1089. ISSN: 0167-6636. DOI: https://doi.org/10.1016/j.mechmat.2005.10.003. URL: https://www.sciencedirect.com/science/article/pii/S0167663605002000.