Physically Recurrent Neural Networks for accelerating multiscale simulations of complex materials

Alves Maia, M.

DOI
10.4233/uuid:2b66e341-af1b-4222-b426-cd1c441ee5a1

Publication date
2025

Document Version
Final published version

Citation (APA)
Alves Maia, M. (2025). *Physically Recurrent Neural Networks for accelerating multiscale simulations of complex materials*. [Dissertation (TU Delft), Delft University of Technology].
https://doi.org/10.4233/uuid:2b66e341-af1b-4222-b426-cd1c441ee5a1

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

PHYSICALLY RECURRENT NEURAL NETWORKS

for accelerating multiscale simulations of complex materials

Marina A. Maia

PHYSICALLY RECURRENT NEURAL NETWORKS FOR ACCELERATING MULTISCALE SIMULATIONS OF COMPLEX MATERIALS

PHYSICALLY RECURRENT NEURAL NETWORKS FOR ACCELERATING MULTISCALE SIMULATIONS OF COMPLEX MATERIALS

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof.dr.ir. H. Bijl,
chair of the Board for Doctorates
to be defended publicly on
monday 19, january 2026 at 10:00 o'clock

by

Marina ALVES MAIA

Master of Science in Civil Engineering
Federal University of Ceará, Brazil
born in Fortaleza, Brazil

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus	chairperson
Dr.ir. F.P. van der Meer	Delft University of Technology, *promotor*
Dr. I.B.C.M. Rocha	Delft University of Technology, *copromotor*
Independent members:	
Prof.dr.ir. L.J. Sluys	Delft University of Technology, the Netherlands
Prof.dr. M. Fagerström	Chalmers University of Technology, Sweden
Prof.dr. L. Noels	Université de Liège, Belgium
Prof.dr.ing. B. Rosic	University of Twente, the Netherlands
Prof.dr. K. Veroy-Grepl	Eindhoven University of Technology, the Netherlands
Prof.dr.ir. E. Schlangen	Delft University of Technology, the Netherlands, reserve member

Keywords: Heterogeneous materials, multiscale analysis, surrogate modeling, neural networks, constitutive model
Printed by: Ipskamp Printing

ISBN 978-94-6518-203-2

An electronic copy of this dissertation is available at
https://repository.tudelft.nl/.

Eu cheguei de muito longe
E a viagem foi tão longa
E na minha caminhada
Obstáculos na estrada
Mas enfim aqui estou.

Erasmo Carlos

CONTENTS

SUMMARY

Modeling the mechanical behavior of high-performance materials often requires accounting for interactions occurring at a lower scale than the one of interest. This scale transition can be addressed in many ways, with different levels of fidelity and computational effort. Naturally, a trade-off exists between these two aspects, and no single method — analytical, numerical or computational — perfectly balances them. Among the high-fidelity options to model complex materials (e.g., composite laminate and concrete) is the concurrent multiscale analysis, or simply FE^2.

In FE^2, two distinct scales, e.g. macro and micro, are solved iteratively. At the microscale, the material geometry is explicitly described by the so-called Representative Volume Element (RVE), where relatively simple constitutive models describe its constituents. At the macroscale, an RVE is coupled to each integration point, and homogenization operators downscale strains and upscale stresses, removing the need for a (macroscopic) constitutive model. However, this generality is associated with high, often prohibitive, computational costs. The limited scalability of FE^2 hinders its adoption in solving real-life engineering problems, driving the need for acceleration strategies that retain the generality of the multiscale framework.

In the last decade, machine learning-based techniques emerged as a popular alternative to reduce computational costs in these simulations. The use of a surrogate model to replace the RVE altogether is arguably the most popular one. Nevertheless, critical issues in data-driven surrogate models remain unsolved and are particularly evident when modelling history-dependent materials. Among them are the data-hungry nature, limited extrapolation capabilities and lack of interpretability.

To address these issues, we introduce a novel class of neural networks (NNs): the Physically Recurrent Neural Networks (PRNNs). The idea is to preserve the knowledge built into constitutive models by embedding them in an encoder-decoder NN architecture with several links to the computational homogenization framework. This hybrid approach, which is non-intrusive and unbound to a specific material model, seeks to combine the benefits of purely data-driven models with those of classical physics-based models.

Starting with a composite micromodel with elastic inclusions and elastoplastic matrix, we demonstrate how training data requirements can be dramatically reduced compared to standard state-of-the-art approaches, with speed-ups over four orders of magnitude compared to FE^2. Then, we illustrate how architectural design choices not only improve interpretability but also push training requirements towards a new lower bound. Next, we incorporate cohesive zone models to model microscopic debonding. In later chapters, we shift to a 3D finite strain setting and adapt the method to handle hyperelasticity and elasto-viscoplasticity. The final chapter focuses on a real-life scientific application, followed by closing remarks, contributions and future research directions.

SAMENVATTING

Het modelleren van het mechanische gedrag van hoogwaardige materialen vereist vaak om een aanpak die interacties van een kleinere schaal meeneemt dan de schaal van interesse. Deze schaalovergang kan op verschillende manieren worden overbrugd, met verschillende niveaus van nauwkeurigheid en rekenkundige inspanning. Er bestaat geen enkele methode — analytisch, numeriek of computationeel — die deze afweging perfect balanceert. Een van de meest nauwkeurige opties om complexe materialen (bijv. composietlaminaat en beton) te modelleren is de gelijktijdige multischaalanalyse, kortweg FE^2.

Bij FE^2 worden twee aparte schalen, bijvoorbeeld macro en micro, iteratief opgelost. Op microschaal wordt de materiaalgeometrie expliciet beschreven door het zogeheten Representatief Volume Element (RVE), waarbij relatief eenvoudige constitutieve modellen de individuele materialen beschrijven. Op macroschaal wordt aan elk integratiepunt een RVE gekoppeld; homogenisatie-operatoren schalen vervormingen naar de microschaal en spanningen naar de macroschaal, ter vervanging van een macroscopisch constitutief model. Deze algemeenheid gaat echter gepaard met hoge, vaak onhaalbare, rekenkosten. De beperkte schaalbaarheid van FE^2 belemmert de toepassing ervan bij het oplossen van reële technische problemen, wat een behoefte geeft aan versnellingsstrategieën die de algemeniteit van het meerschalige kader behouden.

In het laatste decennium zijn op machine learning gebaseerde technieken naar voren gekomen als een populaire manier om de rekenkosten in dergelijke simulaties te verminderen. De meest populaire methode is wellicht het gebruik van een surrogaatmodel ter vervanging van het RVE. Desalniettemin blijven kritieke problemen bij op data gebaseerde surrogaatmodellen onopgelost, die met name duidelijk zijn bij het modelleren van geschiedenisafhankelijke materialen. Tot deze problemen behoren onder andere de benodigdheid van veel trainingsdata, beperkte extrapolatiecapaciteit en gebrek aan interpreteerbaarheid.

Om deze kwesties aan te pakken introduceren we een nieuwe klasse *neural networks* (NNs): de *Physically Recurrent Neural Networks* (PRNNs). Het idee is om de kennis die in constitutieve modellen zit ingebouwd te behouden door deze in een encoder-decoder NN-architectuur op te nemen, met meerdere koppelingen aan computationele homogenisatie. Deze hybride aanpak, die niet-intrusief is en niet gebonden aan een specifiek materieel model, probeert de voordelen van volledig op data gebaseerde modellen te combineren met die van klassieke op fysica gebaseerde modellen.

Beginnend met een composiet micromodel met elastische vezels en een elastoplastische matrix, tonen we aan hoe de benodigde trainingsdata dramatisch kan worden teruggebracht vergeleken met standaard state-of-the-art benaderingen, met snelheidsverbeteringen van meer dan vier ordegroottes ten opzichte van FE^2. Vervolgens verhelderen we hoe ontwerpkeuzes in de architectuur niet alleen de interpreteerbaarheid verbeteren,

maar ook de ondergrens van de trainingsvereisten verlegt. Daarna nemen we cohesieve-zonemodellen mee om microscopische onthechting te modelleren. In latere hoofdstukken schakelen we over naar een 3D-formulering met eindige vervormingen en passen we de methode aan om hyperelasticiteit en elastoviscoplasticiteit aan te kunnen. Het slothoofdstuk richt zich op een praktijkgerichte wetenschappelijke toepassing, gevolgd door afsluitende opmerkingen, bijdragen en toekomstige onderzoeksrichtingen.

ACKNOWLEDGEMENTS

Dear friend, family, colleague, or complete unknown, you should know I have a severe case of skipping-to-the-acknowledgments when I get my hands on books or theses. Not that the story or the science aren't interesting (and I do hope you'll give this one a chance), but because knowing who the authors are thankful for in their lives makes me feel a little bit closer to them. It is usually in this section that you see a glimpse of the untold challenging times, those that science could help with and those it couldn't, and who or what was there to help. It is no different here. This is a compressed tale threaded by *thank-yous* for all those who have somehow been part of this journey or of who I am. I hope that by the end of this section, you either find your own name among these lines or feel a little more familiar with mine.

Looking back, when I moved to the Netherlands amid a lockdown, I remember being torn between two feelings: excitement and fear. Excited to be starting something new, following a career I always aspired to, living in a beautiful country (and equally rainy and windy, I found out soon enough), and getting distance from a political situation that was dragging my country down. On the other hand, there was fear: of not keeping up, of not making friends, of missing my family too much, and so many other struggles those who have moved away from their hometowns probably share.

For a long time, I believed I had to be completely confident about everything before jumping on an opportunity. Lucky for me, I had people in my life to remind me how unrealistic that was. One particular phrase my partner, Luiz, said to me the day I sent the e-mail accepting the PhD offer stuck with me: *"Vai com medo mesmo"* (do it scared). And I did. In time, I realized these conflicts would not go away, and that was not a bad thing. I do miss my friends and family in Brazil, and I occasionally question myself more than the healthy amount. But I have also made amazing friends in my new home, and I have come to understand that uncertainty will forever be part of a scientist's life.

I was incredibly lucky to have had Iuri Rocha and Frans van der Meer as my supervisory team. From day one, you two had your doors open to me. Over these four and a half years, you've balanced guidance and freedom to give me the space and the time I needed to make my own mistakes, and to discover my own strengths and voice. Your diligence, curiosity, enthusiasm, and vision are something to aspire to, and they helped me tailor every part of this thesis.

Iuri, I'll always be grateful to you. You've given me an opportunity of a lifetime. I had the chance to visit so many amazing places, meet so many inspiring people, and learned so much. But mostly, thank you for never holding back on criticism. Because of that, I allowed myself to be quite proud of this book we've built. Thank you for pushing me forward when I doubted myself. Thank you, Cristyna, for welcoming me so well into your home. **Clara** is such a joy, and very lucky to have you and Iuri as her parents.

Frans, I can't thank you enough either. I've learned a lot from your sharp eye for clarity

and your ability to bring structure when I was drowning in details. Many times in our monthly meetings, you have unknowingly helped me see the bigger picture again. Your feedback, light or heavy, always pushed me to do better, and your words of encouragement stay with me.

I am also grateful to my MSc supervisor, **Evandro Parente**, and co-supervisor, **Antônio Macário**, for being such inspirations during those formative years. Evandro was the bridge that led me to Iuri and, in many ways, to this journey. I still cherish the fun times and companionship I shared with everyone at LMCV, and in particular the help from **Elias Saraiva**.

Back in the Netherlands, another piece of this thank you puzzle is my officemates. **Anne Poot**, **Leon Riccius**, **Joep Storm**, and **Winston Lindqwister**, you've made my days at the office lighter and more fun. From our quick chats at the end of the workday to heated lunch debates, to birthday parties and small exchanges throughout the day about bugs in our code, memes, or the best ways to make figures and animations, these are some of the ordinary moments I will carry with me. Thank you for being my company on this ride. I'm rooting for you, and I hope I can be in person in your defenses to watch you thrive. Anne and Joep, *een extra dankjewel* for switching to English anytime any of us non-Dutch speakers were around. You probably don't give it too much thought, but I consider a thoughtful gesture from you to make sure everyone felt included (apart from when it was not our business, of course!).

I also had the pleasure to share the workspace with so many other great researchers who made coffee breaks something to enjoy, and conference trips a time to look forward to. **Pieter Hofman**, **Xinrui Zhang**, **Til Gärtner**, **Dragan Kovačević**, **Suman Battharai**, **Sijmen Zwarts**, **Renan Barros** and **Sergio Cordeiro**, it was a pleasure sharing a bit of our lives outside the grounds of TU Delft. **Zhuojun Nan**, **Fanxiang Xu**, and **Lu Ke**, our time together at the office was shorter, but I'll remember fondly the talks we've shared. Dragan and Lu, thank you for being so patient and helpful with the rhythm I had to offer in our collaborations. I am proud of our work together. I also want to thank **Ehsan Ghane** for initiating a collaboration that, even at a distance and despite the visa challenges, was so fruitful.

Many thanks to **Pierre Kerfriden**, with whom I had the pleasure of collaborating on my first paper. Even though our collaboration was brief, his creativity and insightful questions left a lasting impact on this thesis. I would also like to thank **Bert Sluys** for his approachability and effort to make rookies feel welcome in the department. I especially enjoyed our almost perfectly synchronized coffee breaks discussing F1.

Seeing the work developed for this thesis being used, extended, or improved by others was, and I suspect will always be, such a happy moment. In this spirit, I had a good feeling I would love supervising. But I also hesitated. You might recall something in the first few paragraphs about waiting to be ready. Fortunately, this time, I had Frans and Iuri giving me the push I needed and trusted me to start early with **Nóra Kovács**. Nora, thank you for all the contributions you left in my journey, I could not have had a better experience as a supervisor than having you as my first student. Yet, what I am most grateful for is your friendship. In *almost inaudible* Portuguese: muito obrigada. I'm honored you and Anne are my paranymphs.

In the following years, I supervised two more students, **Paul van IJzendoorn** and **Ruben**

van Gils. Paul, thank you for embracing the uncertainty of a new method and shedding light on it so that other students can build from your findings. Ruben, it was rewarding to see your interest sparkle throughout the project and see your confidence grow in the topic. You are part of this thesis, too.

On the other side of the Atlantic, I've had the unwavering support of my family and friends, who have been so present in my life that it often felt as if they were one train away. Yet, writing this paragraph is by far the hardest part of this section. Not because I can't recall moments when they were there for me, but because any attempt to summarize in a few words how much they mean to me feels inadequate. But let me try anyway.

To my friends **Wendy Quintanilha**, **Carla Marília**, **Gledson Mesquita**, **Renan Maia**, and **Geovanny Moreno**, I hope you know you are my treasure. Whenever I felt overwhelmed, unfit, or neglectful of myself or of you, you were there to welcome me and remind me how much I had been missing. I love and am so proud of what we have. I miss our times at Pici, at each other's homes, sipping coffee, stressing about exams, all while laughing at the silliest things. I miss us at Bull's trying to make the bill add up (and it rarely did), or quietly asking things like *"do we like this person?"* as if the group had a mind of its own. Thank you for easing the hardships and sharing the joys in the most ordinary moments while we figure out life together.

I want to acknowledge my dear friend **Karol Araújo**, who has been in my life since childhood. I am lucky to have friends with whom we may go long without talking, yet everything is the same once we reach out. The same goes for **Augusto Tremarim**. Thank you, Gus, for making me laugh at my lifelong pessimistic ways, and for always being there to continue our conversations, no matter how much time has passed without (too much) judgment and a fresh and expanding catalog of (un)believable stories.

To **my family**, **aunts**, **uncles**, and **cousins**, my deepest gratitude. To my mom, especially: **Mom**, you often downplay your strengths and how many great things you have achieved, but I wish you could see yourself through my eyes just for a moment. A funny, generous, loving, strong woman who has done everything for her daughters and whoever needed a helping hand. You'd also see a bit of a hothead too, but that's your sauce and to paraphrase a certain character from one of my favorite sitcoms: *"I am Latin, so I get to feel whatever I want"*. Mom, you are my cornerstone and my motivation to seek a better life. I wish I could give you the world, but for now, please accept this humble acknowledgment that you're a core part of who I am, and therefore, of this thesis.

Dad, even though we don't see eye-to-eye on politics, I know we share the belief that education is the way to pave a better future. You told me knowledge was something no one could take away from me. Whenever I would approach you with news or questions — from accepting my first job as a CAD drawer to crossing the Atlantic to do a PhD, you were quick to say I could do it. I was always a bit of a planner, so having you, a dreamer, encouraging me to aim higher is a gift. Thank you, **Cerleiyde**, for making the visit to the Netherlands possible, for taking care of my dad, and for bringing **Grazy**, my little sister, into the world. I also want to acknowledge my little brother, **Caio Victor**, who is actually no longer little and is, in fact, taller than me. Although you two are growing fast, you will always be my little ones.

Speaking of dreamers, I want to thank my older sister, **Mariana**. I remember we used to fight about everything when we were little, measuring any food down to the millimeter

to split it into two perfect halves. Over time, it became clear we also fought (together, this time) for our *dreams*. We dream about trips – and we were lucky enough to have some of them fulfilled; concerts – we usually arrive early to secure front-row at our favorite bands and leave the venues exhausted the next day, knees bruised but smiles all over our faces; and so much more. *I know it's hard sometimes*, but I am glad you're with me on my ride.

A heartfelt thank you to **tia Jacinta**, **tia Palma**, and **João Jackson**. Tia Jacinta has always been my cheerleader and has taken care of me as if I were her own since forever. Thank you, Jess. Tia Palma and João have always supported the family and helped hold down the fort at home. I am also incredibly grateful to the family I gained along the way, especially **Dona Raimunda** — my mother-in-law, who, like my own mother, was not particularly thrilled by the idea of us moving so far away, but became nothing but supportive of me throughout the ups and downs of my PhD. Dona Rai, I am in awe of your ever-growing desire to learn and to explore.

My gratitude also goes to those who are less likely to read this thesis, but are just as valued. My high-school teachers, **Emanuel Tiago**, **Viviane Frutuoso**, **Roquelane**, **Kildery Amorim**, **George Barbosa**, and **Jam Silva**. You inspired me greatly to follow the academic path I am now on. You may not remember, but I will never forget the textbooks you gave me when my family couldn't afford them. Thank you to the bands that kept me company while writing this book, particularly **Muse** and **Twenty One Pilots**, for giving me energy, lifting my spirits, and helping me feel understood.

Finally, I want to thank **Luiz**. Writing this thesis would not have been possible without you. You saw me at my best, followed me in conferences, heard me talking about neural networks and composites more than anyone not doing a PhD on this would ever need to, and celebrated with me the tiny and big achievements. We traveled Europe making amazing memories, and I hope we can keep doing that around the world forever. But what I want others to know is that you were there to see me at my worst, too. When I was stressed, lost, exhausted, skipping weekend after weekend to work, or feeling hurt. In these times, you offered me a hug or a shoulder to cry on, and let me feel it all until, suddenly, I found myself laughing again at the silliest things with you. I remember one day in particular when you picked me up at the bus station. I felt like a knot was blocking my throat. When you saw me, you simply took me in your arms and hugged me for what felt like hours. And the knot slowly dissolved into a feeling of warmth. This feeling is something I can only hope to repay you in our time together. *I wish I sang some better words in an order that is new*, but really all I want to say boils down to I love you.

I've spent a lot of time in this section telling you how amazing the people in my life are, and I hope that, whether you are a friend, family, or a stranger, you are also lucky to have a support network. I chose to focus on the people, rather than the setbacks, because I think the *people* deserve the recognition. But a long journey such as this is, of course, filled with hardships. There were also many days of self-doubt, loneliness, or dealing with someone unkind. In those days, having my people around me was the key to seeing things differently. And for that, no acknowledgments section will ever be enough.

Marina Alves Maia
Rijswijk, Fall 2025

1

INTRODUCTION

On a given day, a given circumstance...
you think you have a limit.
And you then go for this limit
and you touch this limit,
and you think, “Okay, this is the limit."
As soon as you touch this limit,
something happens
and you suddenly can go a little bit further.

Ayrton Senna

In many engineering applications, understanding how phenomena at smaller scales affect the overall structural behaviour at larger scales is essential. Central to this is the faithful characterization of the material employed in structural components. This stage is vital to ensure safe and reliable designs — as prescribed in regulations — and represents a doorway to more efficient designs. With advances in the manufacturing of highly tailorable materials, such as fibre-reinforced polymer (FRP) composites, metamaterials and engineered cementitious composites, detailed knowledge of the material and its microstructure is key to unlocking their full potential. For that, experimental characterization alone is not enough. Computational models are crucial for deepening our understanding of material behavior and for enabling predictive simulations in complex applications.

In the aerospace industry, for example, FRP composites are commonly employed in manufacturing aircraft and launch vehicles due to their exceptional strength-to-weight ratio [1]. In civil engineering, the most widely used material in the world, concrete, is another example of a composite [2]. In both domains, the materials can be described at different levels of observation [3, 4]. Going from macro to micro (and beyond) allows complex phenomena, such as fibre-matrix debonding, matrix cracking, delamination, and aggregate bridging that inherently occur at different length scales, to be captured more accurately without resorting to empirical relations that rely on oversimplified assumptions.

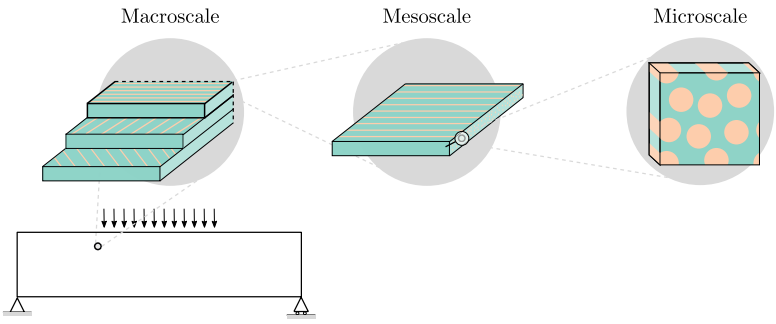


Figure 1.1: Fiber-reinforced composite models across the scales.

To illustrate the different scales of observation in heterogeneous materials, consider the example illustrated in Fig. 1.1. Accurately modeling these interactions is a challenge in itself, one that has been in demand for decades now. In that regard, some methods trade flexibility for efficiency, while others offer generality at the cost of increased computational effort. Examples of highly efficient approaches to predict the average - or *homogenized* - response of heterogeneous materials are mean-field homogenization schemes (e.g. Mori-Tanaka [5] and Self-Consistent method) based on Eshelby's solution [6]. The limitations in this case regard geometric features of inclusions, volume fraction, stress localization and material nonlinearity [7].

Another alternative lies with models based on numerical homogenization [8–11], where the homogenized constitutive behavior is defined through a set of parameters that need to be calibrated based on experiments or numerical simulations at smaller scales. While

fast, the balance between the number of parameters, data availability, and ability to predict general loading conditions is not trivial. Some of these models can be seriously limited if certain assumptions and interactions are not considered. An example of that can be found in [9], where a study on a recently developed homogenized orthotropic plasticity model for FRP composites quantified the loss of accuracy due to necessary simplifications such as ignoring the influence of stress in the fibre direction on the plasticity and use of a constant plastic Poisson ratio.

At the opposite end in terms of efficiency are multiscale methods, particularly FE^2 [12]. In this case, two scales (macro and micro) that exchange information are considered and solved iteratively. The main difference from regular Finite Element (FE) is that no explicit constitutive model relating strains to stresses at the macroscale exists. Instead, a micromodel is embedded at each integration point of the macroscale mesh, and its homogenized response is used to compute the macroscopic behavior. The micromodel, in turn, consists of another FE mesh that describes the geometry of the heterogeneous material. At the lower scale, phenomena such as orthotropic behaviour, (visco-)plasticity, ageing, damage, and strain/stress localization can be more easily incorporated, making FE^2 a compelling choice for modelling complex materials.

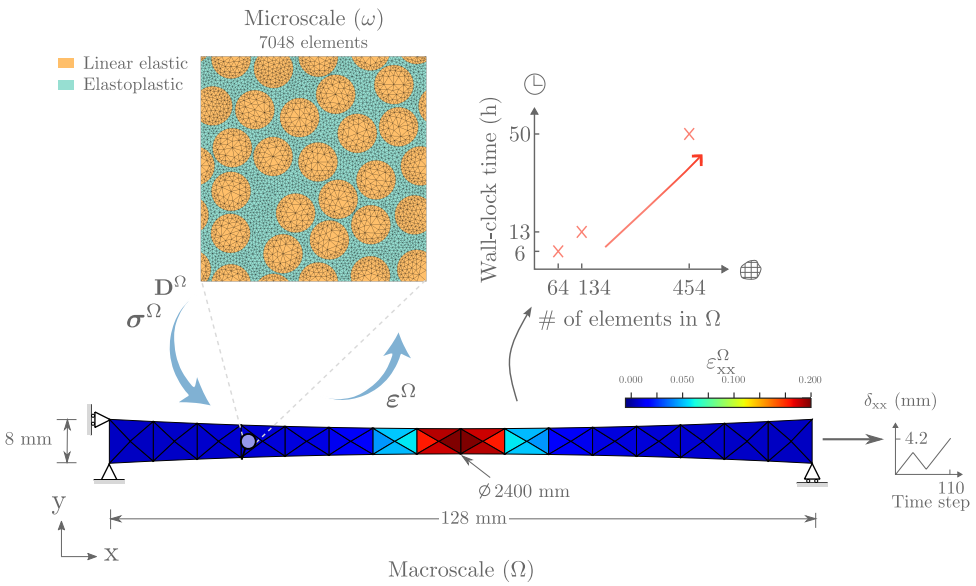


Figure 1.2: Example of FE^2 computational cost for a 2D dogbone problem.

This generality, however, is associated with extreme computational cost, even for simple academic examples. A full-order FE^2 simulation requires solving numerous local FE problems at every time increment, quickly becoming intractable for large-scale and time-dependent analyses. To give the reader a sense of time, consider the 2D dogbone problem illustrated in Fig. 1.2. Details on the implementation and CPU settings are discussed later in Chapter 2; here we focus on the wall-clock time vs number of elements comparison depicted on the right-hand side of Fig. 1.2. Even for coarse meshes and

a relatively small macroscopic domain, the multiscale simulation already spans hours, between 6 and 13 in this example, and can quickly reach days as the macromesh is refined. Such limited scalability has created an urgency to develop acceleration strategies that preserve the generality of the multiscale framework with improved computational efficiency.

Several methods have been proposed to tackle this computational bottleneck, with varying degrees of success in terms of accuracy, generality, and efficiency [13]. Examples include parallel computing [14, 15], Fast Fourier Transform methods [16, 17], clustering-based methods [18–20] and Reduced Order Modeling (ROM) [21–26]. In addition to these, a dominant strategy with rapid adoption in the material modelling community is the use of *surrogate models*. In this approach, the micromodel is replaced altogether with a surrogate model, which at a significantly lower computational cost plays the role of the homogenized constitutive law. Thus eliminating the main bottleneck of the framework.

The surrogate model usually consists of a data-driven model trained on snapshots of the micromodel being replaced. Or, in the case of history-dependent materials, sequences of snapshots. In such cases, Recurrent Neural Networks (RNN) and their more complex variations (e.g. Long-Short Term (LSTM) and Gated-Recurrent Unit (GRU)) are the standard approaches [27–34]. The versatility of these models in adapting to a wide variety of constitutive behaviours with little to no changes in their architecture, in combination with the streamlined strategy of replacing the micromodel, has granted these models the leading choice in the field.

Despite their extensive use in the literature, three vital issues remain unresolved in the context of material modelling:

- Firstly, although the hidden state in RNNs and the internal variables in a physics-based constitutive model play a similar role, the **network mechanism is** still regarded as **a black-box**. Insights into any latent physical patterns learned by the network are thus far limited to simple settings (e.g., homogeneous material in 1D problems [35, 36]);

- Secondly, RNNs have a **limited ability to extrapolate**. This issue is usually tackled with ever larger training sets and intricate design of experiments;

- However, even in 1D or 2D problems, a large variety of loading/unloading cases is required to cover similar paths and patterns encountered in actual microscale simulations, which exposes the third issue: the **data-hungry nature** of RNNs.

Many surrogate models have been proposed to tackle some of these issues. Examples include Gaussian Processes (GPs) [37–39], Deep Neural Networks [40–42], Graph Neural Networks [43], Transformers [44, 45], Convolution Neural Networks [46], and frameworks that incorporate some of these methods with MOR techniques [23, 47–52]. While these models demonstrate powerful approximation capabilities and account for a variety of aspects in their formulations (e.g. uncertainty quantification and different micromodel geometries), as data-driven approaches, they usually rely on extensive training datasets and struggle to generalize to unseen loading cases. Amid these developments,

a particular trend gained traction in recent years: *introducing physics knowledge into machine learning-based models*.

Propelled by the success of Physics-Informed Neural Networks (PINNs) as partial differential equation solvers [53], the idea of enriching the loss function with extra terms to enforce physics constraints has quickly found its way into the material modelling community. First, in applications dealing with homogeneous and elastoplastic materials [54, 55], then in more general formulations for strain-rate independent inelastic materials [56]. So-called Thermodynamics-based Artificial Neural Networks (TANNs) [56], for example, incorporate thermodynamic constraints in their architecture through the computation of numerical derivatives of the network with respect to its inputs. Specifically, the derivatives of the free-energy and their relation with stress, dissipation rate and internal state variables (e.g. displacement field and internal force). A follow-up work extended TANNs to deal with heterogeneous materials [57], where the key contribution is related to the automatic identification of a reduced set of internal variables at the level of the micromodel through the use of autoencoders.

Although the PINN-inspired approaches offer improved extrapolation properties and reduced training set size requirements compared to black-box NNs, softly enforcing constraints through the loss function cannot guarantee their fulfillment in unseen loading scenarios in the online phase. This is further discussed in [58], where a comparative study is carried out considering three basic classes of NNs: black-box NNs, NNs enforcing physics in a weak form and NNs enforcing physics in a strong form. The latter class of NNs hard-code physics via custom NN architectures to ensure that the constraint is fulfilled in any loading condition. In this new paradigm, Input Convex Neural Networks (ICNNs) and automatic differentiation are key tools employed in energy-based formulations (instead of the usual strain-stress direct mapping) from which stresses and other quantities can be derived [59–61].

Another aspect explored in this new wave of physics-enhanced networks is symmetry properties, particularly material frame indifference and material symmetries [62, 63]. For example, the special structure proposed in [62], named Symmetric Positive Definite Neural Networks (SPD-NNs), embeds symmetry and positive definiteness directly into the architecture used to predict the tangent stiffness matrix. This improves stability and robustness, two vital features when integrating the model within numerical solvers. However, as with any NN-based model, SPD-NNs only work on test data that does not deviate much from the training data.

More recently, so-called Physics-Augmented Neural Networks (PANNs) [64–67] and Constitutive Artificial Neural Networks (CANNs) [68, 69] have also gained traction in the literature. These approaches differ in their philosophy and levels of interpretability, but both are designed to automatically fulfil common kinematical, thermodynamical and material objectivity constraints, with special concern for polyconvexity to improve numerical stability. CANNs, in particular, aim at interpretable networks for automated model discovery. This is addressed through the combination of sparsity and specialized activation functions that mimic known constitutive models, in contrast to the general (convex) functions employed in PANNs. A downside of both of these highly-tailored architectures is the difficulty of extending them to history-dependent materials. Up to now, PANNs and CANNs remain specialized to a **narrow range of constitutive behaviour**

(e.g. hyperelastic or viscoelastic materials).

In parallel with CANNs, the authors in [70] introduced a new approach, known as EUCLID, for the automated discovery of isotropic hyperelastic constitutive laws. The novel approach does not need stress data and builds interpretable models by promoting regularization techniques that induce the selection of a reduced number of models from a "library" of material models. Since then, EUCLID has been extended to deal with a generalized library of models [71], including inelasticity, and has been incorporated into a Bayesian framework for uncertainty quantification [72]. Beyond these milestones, the applicability and scalability of the method to handle heterogeneous materials remains an open challenge.

Two other noteworthy and unique strategies to accelerate multiscale simulations are the hyper-reduction via empirical cubature method (ECM) [73] and Deep Material Networks (DMNs) [74]. A common thread in these works is the incorporation of constitutive models directly into their formulation. In the ECM, two stages of reduction are performed. In the first stage, the number of degrees of freedom of the micromodel is reduced using Proper Orthogonal Decomposition (POD). In the second, a reduced subset of integration points in this micromodel is selected to integrate the internal force vector and tangent stiffness matrix with modified (and positive) weights as accurately as possible. This set of points is obtained by solving a series of combinatorial optimization problems. In combination with the reduction in the solution space brought by POD, the ECM yields significant speed-ups in the online phase [49, 75]. The method has also been successfully extended to deal with geometrically parametrized domains in computational homogenization applications [76, 77].

On the other hand, ECM relies on a heuristic selection of the integration points, which can result in a suboptimal selection. Furthermore, the increase in POD modes is typically followed by an increase in the number of points, straining the offline training phase and progressively compromising the gains in the online phase. This is partially addressed in [78], where a two-stage strategy further reduces the initial subset found by ECM by enforcing sparsification. Essentially, the sparsification algorithm drives small weights to zero and readjusts the position and weights of the remaining points accordingly. Though increased accuracy and efficiency are achieved with fewer integration points, the consideration of a further reduction stage escalates the complexity of an already intricate model, especially given its intrusive nature (i.e. changes in the FE solver are needed) [76, 79].

In DMNs, a binary-tree network architecture leverages analytical homogenization to combine building blocks made of constitutive models to learn an equivalent topology of the micromodel [74]. They excel in extrapolating from linear elastic data to nonlinear and history-dependent behaviour, but training and online evaluation are not straightforward. These two stages involve different input spaces, and an iterative Newton-Raphson scheme is required for the prediction stage. Nevertheless, DMNs represent a powerful framework for multiscale material modeling. Since their introduction, several extensions have been proposed [80, 81], with a recent review providing an overview of present developments and future directions in improving the method for broader applications in multiscale modeling [82]. To put DMNs into perspective with some of the methods discussed so far, consider the spectrum in Fig. 1.3, where a variety of models are

roughly ranked in terms of physics embedding, ranging from black-box models to the full-order micromodel. The strengths and limitations of these methods are further discussed throughout the chapters where they are most relevant.

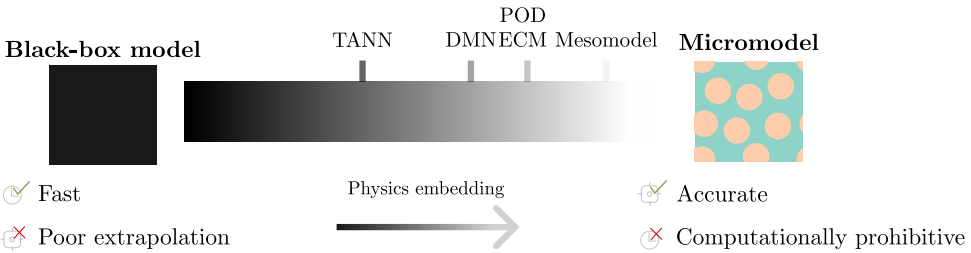


Figure 1.3: Spectrum of methods to model the constitutive behavior of heterogeneous materials.

Despite this broad spectrum of available models, as of yet, no single strategy has emerged as a definite solution. While each model shows promise in specific applications, their shortcomings hinder their applicability in multiscale settings to solve real-world problems. Some models struggle with the training complexity of millions of parameters and the critical scaling of computational memory space required for offline and online phases as the sequence length increases [44, 45]. Others are unsuitable for multiscale simulations, as they have either been tailored to a specific constitutive behaviour or tested in a rather restricted design space. Several lack robustness under extrapolation or require intrusive integration within numerical solvers. Even physics-aware architectures, though more consistent, interpretable and less data-hungry than conventional NNs, remain tailored to specific classes of materials, relying on hand-crafted features.

Based on this background, we delimit the scope and the aim of this thesis in Section 1.1, followed by the outline in Section 1.2, where a brief description of the key aspects and contributions of each chapter is summarized.

1.1. SCOPE AND AIM

This thesis presents a novel class of neural networks, the Physically Recurrent Neural Networks (PRNNs), for accelerating the multiscale simulation of complex materials. The core idea of PRNNs is to preserve the knowledge built-in on constitutive models by embedding them in an encoder-decoder architecture. This hybrid approach aims to reconcile the benefits of purely data-driven models with decades-old classical physics models to address the main challenges discussed in the previous section.

The method is tested in a variety of micromechanical and FE^2 scenarios for a wide range of constitutive behaviours and has been proven to be a robust alternative of practical utility in a real-life scientific application with experimental data for validation. Although we have limited the numerical examples to composite micromodels, PRNNs are envisioned to be general, and their application to other complex materials is expected to be both promising and within reach. This positions the present work as the cornerstone towards a broader range of applications.

As we explore the foundations, capabilities and limitations of this network in several applications involving path and time-dependent heterogeneous materials, the following research question is posed:

To what extent embedding classical material models in data-driven models can benefit the building of more robust, accurate and interpretable surrogate models for accelerating the multiscale simulations of complex materials?

1.2. THESIS OUTLINE

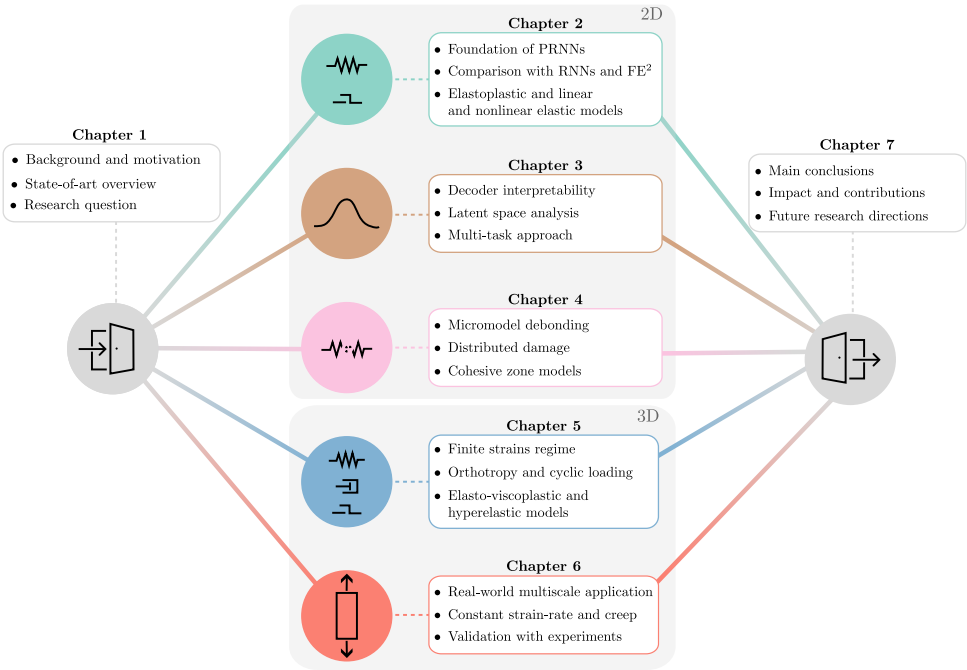


Figure 1.4: Chapters organization.

The remainder of this thesis is organized as follows:

- In Chapter 2, the foundations of PRNNs are laid, including two multiscale applications to demonstrate the efficiency, accuracy and robustness of the network. Three different constitutive models are explored in this chapter: linear and nonlinear elastic, and elastoplastic models;
- In Chapter 3, based on the same micromodel and constitutive models as Chapter 2, we shift our focus to the impact of changes in the architecture of the network on aspects such as interpretability and accuracy. We demonstrate how the new physically motivated constraints improve generalization and help push training requirements to a new lower bound;

- In Chapter 4, the first key extension to PRNNs is discussed: the inclusion of damage models to capture microscale debonding of composite materials. For that purpose, we present the necessary modifications to the network's architecture to accommodate the new inputs (displacement jumps) and outputs (damage variable and tractions) and to integrate them with the remaining (bulk) constitutive models studied so far;

- In Chapter 5, the second key extension of PRNNs is presented. This time, we expand the model applicability to deal with rate and path-dependent 3D problems in a finite strain framework. In the numerical applications, the Eindhoven Glassy Polymer (EGP) model - an advanced elasto-viscoplastic material model for polymers - is assigned to one of the constituents of the micromodel.

- In Chapter 6, the PRNN presented in Chapter 5 is employed in a real-life scientific application. The goal is to reproduce a set of experiments through a surrogate-based multiscale approach to model constant strain-rate and creep on unidirectional thermoplastic composites under off-axis loading.

Finally, the main remarks of this work and its impact, including ongoing collaborations, are presented in Chapter 7, along with promising future research directions.

REFERENCES

[1] A. K. Hamzat, M. S. Murad, I. A. Adediran, E. Asmatulu, and R. Asmatulu. "Fiber-reinforced composites for aerospace, energy, and marine applications: an insight into failure mechanisms under chemical, thermal, oxidative, and mechanical load conditions". *Advanced Composites and Hybrid Materials* 8.1 (2025). ISSN: 2522-0136. DOI: 10.1007/s42114-024-01192-y.

[2] H. Van Damme. "Concrete material science: Past, present, and future innovations". *Cement and Concrete Research* 112 (2018). SI : Digital concrete 2018, 5–24. ISSN: 0008-8846. DOI: https://doi.org/10.1016/j.cemconres.2018.05.002.

[3] A. Elmasry, W. Azoti, S. A. El-Safty, and A. Elmarakbi. "A comparative review of multiscale models for effective properties of nano- and micro-composites". *Progress in Materials Science* 132 (2023), 101022. ISSN: 0079-6425. DOI: https://doi.org/10.1016/j.pmatsci.2022.101022.

[4] J. F. Unger and S. Eckardt. "Multiscale Modeling of Concrete: From Mesoscale to Macroscale". *Archives of Computational Methods in Engineering* 18.3 (2011), 341–393. ISSN: 1886-1784. DOI: 10.1007/s11831-011-9063-8.

[5] T. Mori and K. Tanaka. "Average stress in matrix and average elastic energy of materials with misfitting inclusions". *Acta Metallurgica* 21.5 (1973), 571–574. ISSN: 0001-6160. DOI: https://doi.org/10.1016/0001-6160(73)90064-3.

[6] J. D. Eshelby and R. E. Peierls. "The determination of the elastic field of an ellipsoidal inclusion, and related problems". *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 241.1226 (1957), 376–396. DOI: 10.1098/rspa.1957.0133.

[7] A. Jain. “Micro and mesomechanics of fibre reinforced composites using mean field homogenization formulations: A review”. *Materials Today Communications* 21 (2019), 100552. ISSN: 2352-4928. DOI: https://doi.org/10.1016/j.mtcomm.2019.100552.

[8] T. Vaughan and C. McCarthy. “A combined experimental–numerical approach for generating statistically equivalent fibre distributions for high strength laminated composite materials”. *Composites Science and Technology* 70.2 (2010), 291–297. ISSN: 0266-3538. DOI: https://doi.org/10.1016/j.compscitech.2009.10.020.

[9] F. P. van der Meer. “Micromechanical validation of a mesomodel for plasticity in composites”. *European Journal of Mechanics - A/Solids* 60 (2016), 58–69. ISSN: 0997-7538. DOI: https://doi.org/10.1016/j.euromechsol.2016.06.008.

[10] F. Naya, C. González, C. Lopes, S. Van der Veen, and F. Pons. “Computational micromechanics of the transverse and shear behavior of unidirectional fiber reinforced polymers including environmental effects”. *Composites Part A: Applied Science and Manufacturing* 92 (2017), 146–157. ISSN: 1359-835X. DOI: https://doi.org/10.1016/j.compositesa.2016.06.018.

[11] A. Tessarin, M. Zaccariotto, U. Galvanetto, and D. Stocchi. “A multiscale numerical homogenization-based method for the prediction of elastic properties of components produced with the fused deposition modelling process”. *Results in Engineering* 14 (2022), 100409. ISSN: 2590-1230. DOI: https://doi.org/10.1016/j.rineng.2022.100409.

[12] F. Feyel and J.-L. Chaboche. “FE2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials”. *Computer Methods in Applied Mechanics and Engineering* 183.3 (2000), 309–330. ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(99)00224-8.

[13] K. Matouš, M. G. Geers, V. G. Kouznetsova, and A. Gillman. “A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials”. *Journal of Computational Physics* 330 (2017), 192–220. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2016.10.070.

[14] S. Kim, E. Kissel, and K. Matouš. “Adaptive and parallel multiscale framework for modeling cohesive failure in engineering scale systems”. *Computer Methods in Applied Mechanics and Engineering* 429 (2024), 117191. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2024.117191.

[15] N. Kovachki, B. Liu, X. Sun, H. Zhou, K. Bhattacharya, M. Ortiz, and A. Stuart. “Multiscale modeling of materials: Computing, data science, uncertainty and goal-oriented optimization”. *Mechanics of Materials* 165 (2022), 104156. ISSN: 0167-6636. DOI: https://doi.org/10.1016/j.mechmat.2021.104156.

[16] T. de Geus, J. Vondřejc, J. Zeman, R. Peerlings, and M. Geers. “Finite strain FFT-based non-linear solvers made simple”. *Computer Methods in Applied Mechanics and Engineering* 318 (2017), 412–430. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.12.032.

[17] P. A. Hessman, F. Welschinger, K. Hornberger, and T. Böhlke. "On mean field homogenization schemes for short fiber reinforced composites: Unified formulation, application and benchmark". *International Journal of Solids and Structures* 230-231 (2021), 111141. ISSN: 0020-7683. DOI: https://doi.org/10.1016/j.ijsolstr.2021.111141.

[18] B. P. Ferreira, F. Andrade Pires, and M. Bessa. "Adaptivity for clustering-based reduced-order modeling of localized history-dependent phenomena". *Computer Methods in Applied Mechanics and Engineering* 393 (2022), 114726. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.114726.

[19] S. Deng, C. Soderhjelm, D. Apelian, and R. Bostanabad. "Reduced-order multiscale modeling of plastic deformations in 3D alloys with spatially varying porosity by deflated clustering analysis". *Computational Mechanics* 70.3 (Sept. 2022), 517–548. ISSN: 1432-0924. DOI: 10.1007/s00466-022-02177-8.

[20] S. Chaouch and J. Yvonnet. "An unsupervised machine learning approach to reduce nonlinear FE2 multiscale calculations using macro clustering". *Finite Elements in Analysis and Design* 229 (2024), 104069. ISSN: 0168-874X. DOI: https://doi.org/10.1016/j.finel.2023.104069.

[21] J. Yvonnet and Q.-C. He. "The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains". *Journal of Computational Physics* 223.1 (2007), 341–368. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2006.09.019.

[22] J. Hernández, J. Oliver, A. E. Huespe, M. Caicedo, and J. Cante. *Computational homogenization of inelastic materials using model order reduction*. CIMNE, 2014.

[23] J. Oliver, M. Caicedo, A. E. Huespe, J. A. Hernández, and E. Roubin. "Reduced order modeling strategies for computational multiscale fracture". *Computer Methods in Applied Mechanics and Engineering* 313 (2017), 560–595. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.09.039.

[24] M. Raschi, O. Lloberas-Valls, A. Huespe, and J. Oliver. "High performance reduction technique for multiscale finite element modeling (HPR-FE2): Towards industrial multiscale FE software". *Computer Methods in Applied Mechanics and Engineering* 375 (2021), 113580. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.113580.

[25] D. R. Brandyberry, X. Zhang, and P. H. Geubelle. "Multiscale design of nonlinear materials using reduced-order modeling". *Computer Methods in Applied Mechanics and Engineering* 399 (2022), 115388. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.115388.

[26] P. Diercks, K. Veroy, A. Robens-Radermacher, and J. F. Unger. "Multiscale modeling of linear elastic heterogeneous structures via localized model order reduction". *International Journal for Numerical Methods in Engineering* 124.20 (2023), 4580–4602. DOI: https://doi.org/10.1002/nme.7326.

[27] Y. Heider, K. Wang, and W. Sun. "SO(3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials". *Computer Methods in Applied Mechanics and Engineering* 363 (2020), 112875. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.112875.

[28] B. Li and X. Zhuang. "Multiscale computation on feedforward neural network and recurrent neural network". *Frontiers of Structural and Civil Engineering* 14.6 (2020), 1285–1298. ISSN: 2095-2449. DOI: 10.1007/s11709-020-0691-7.

[29] L. Wu and L. Noels. "Recurrent Neural Networks (RNNs) with dimensionality reduction and break down in computational mechanics; application to multi-scale localization step". *Computer Methods in Applied Mechanics and Engineering* 390 (2022), 114476. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.114476.

[30] H. J. Logarzo, G. Capuano, and J. J. Rimoli. "Smart constitutive laws: Inelastic homogenization through machine learning". *Computer Methods in Applied Mechanics and Engineering* 373 (2021), 113482. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.113482.

[31] M. B. Gorji, M. Mozaffar, J. N. Heidenreich, J. Cao, and D. Mohr. "On the potential of recurrent neural networks for modeling path dependent plasticity". *Journal of the Mechanics and Physics of Solids* 143 (2020), 103972. ISSN: 0022-5096. DOI: https://doi.org/10.1016/j.jmps.2020.103972.

[32] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, and M. A. Bessa. "Deep learning predicts path-dependent plasticity". *Proceedings of the National Academy of Sciences* 116.52 (2019), 26414–26420. ISSN: 0027-8424. DOI: 10.1073/pnas.1911815116.

[33] E. Ghane, M. Fagerström, and M. Mirkhalaf. "Recurrent neural networks and transfer learning for predicting elasto-plasticity in woven composites". *European Journal of Mechanics - A/Solids* 107 (2024), 105378. ISSN: 0997-7538. DOI: https://doi.org/10.1016/j.euromechsol.2024.105378.

[34] H. L. Cheung and M. Mirkhalaf. "A multi-fidelity data-driven model for highly accurate and computationally efficient modeling of short fiber composites". *Composites Science and Technology* 246 (2024), 110359. ISSN: 0266-3538. DOI: https://doi.org/10.1016/j.compscitech.2023.110359.

[35] A. Koeppe, F. Bamer, M. Selzer, B. Nestler, and B. Markert. "Explainable artificial intelligence for mechanics: Physics-Explaining neural networks for constitutive models". *Frontiers in Materials* 8 (2022). DOI: 10.3389/fmats.2021.824958.

[36] B. Liu, E. Ocegueda, M. Trautner, A. M. Stuart, and K. Bhattacharya. "Learning macroscopic internal variables and history dependence from microscopic models". *Journal of the Mechanics and Physics of Solids* 178 (2023), 105329. ISSN: 0022-5096. DOI: https://doi.org/10.1016/j.jmps.2023.105329.

[37] A. L. Frankel, R. E. Jones, and L. P. Swiler. "Tensor basis Gaussian Process models of hyperelastic materials". *Journal of Machine Learning for Modeling and Computing* 1.1 (2020), 1–17. ISSN: 2689-3967. DOI: 10.1615/jmachlearnmodelcomput.2020033325.

[38] 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". *Journal of Computational Physics: X* 9 (2021), 100083. ISSN: 2590-0552. DOI: https://doi.org/10.1016/j.jcpx.2020.100083.

[39] J. N. Fuhg, M. Marino, and N. Bouklas. "Local approximate Gaussian process regression for data-driven constitutive models: development and comparison with neural networks". *Computer Methods in Applied Mechanics and Engineering* 388 (2022), 114217. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.114217.

[40] H. Eivazi, J.-A. Tröger, S. Wittek, S. Hartmann, and A. Rausch. "FE2 Computations with Deep Neural Networks: Algorithmic Structure, Data Generation, and Implementation". *Mathematical and Computational Applications* 28.4 (2023). ISSN: 2297-8747. DOI: 10.3390/mca28040091.

[41] N. Feng, G. Zhang, and K. Khandelwal. "Finite strain FE2 analysis with data-driven homogenization using deep neural networks". *Computers & Structures* 263 (2022), 106742. ISSN: 0045-7949. DOI: https://doi.org/10.1016/j.compstruc.2022.106742.

[42] F. Aldakheel, E. S. Elsayed, T. I. Zohdi, and P. Wriggers. "Efficient multiscale modeling of heterogeneous materials using deep neural networks". *Computational Mechanics* 72.1 (July 2023), 155–171. ISSN: 1432-0924. DOI: 10.1007/s00466-023-02324-9.

[43] N. N. Vlassis and W. Sun. "Geometric learning for computational mechanics Part II: Graph embedding for interpretable multiscale plasticity". *Computer Methods in Applied Mechanics and Engineering* 404 (2023), 115768. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.115768.

[44] Y. Zhongbo and P. L. Hien. "Pre-trained transformer model as a surrogate in multiscale computational homogenization framework for elastoplastic composite materials subjected to generic loading paths". *Computer Methods in Applied Mechanics and Engineering* 421 (2024), 116745. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2024.116745.

[45] E. Pitz and K. Pochiraju. "A neural network transformer model for composite microstructure homogenization". *Engineering Applications of Artificial Intelligence* 134 (2024), 108622. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2024.108622.

[46] D. W. Abueidda, S. Koric, N. A. Sobh, and H. Sehitoglu. "Deep learning for plasticity and thermo-viscoplasticity". *International Journal of Plasticity* 136 (2021), 102852. ISSN: 0749-6419. DOI: https://doi.org/10.1016/j.ijplas.2020.102852.

[47] "A PGD-based homogenization technique for the resolution of nonlinear multiscale problems". *Computer Methods in Applied Mechanics and Engineering* 267 (2013), 275–292. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2013.08.009.

[48] F. Ghavamian, P. Tiso, and A. Simone. "POD–DEIM model order reduction for strain-softening viscoplasticity". *Computer Methods in Applied Mechanics and Engineering* 317 (2017), 458–479. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.11.025.

[49] I. B. C. M. Rocha, F. P. van der Meer, and L. J. Sluys. "Efficient micromechanical analysis of fiber-reinforced composites subjected to cyclic loading through time homogenization and reduced-order modeling". *Computer Methods in Applied Mechanics and Engineering* 345 (2019), 644–670. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2018.11.014.

[50] M. Guo and J. S. Hesthaven. "Reduced order modeling for nonlinear structural analysis using Gaussian process regression". *Computer Methods in Applied Mechanics and Engineering* 341 (2018), 807–826. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2018.07.017.

[51] M. Guo and J. S. Hesthaven. "Data-driven reduced order modeling for time-dependent problems". *Computer Methods in Applied Mechanics and Engineering* 345 (2019), 75–99. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2018.10.029.

[52] F. Casenave, N. Akkari, F. Bordeu, C. Rey, and D. Ryckelynck. "A nonintrusive distributed reduced-order modeling framework for nonlinear structural mechanics — Application to elastoviscoplastic computations". *International Journal for Numerical Methods in Engineering* 121.1 (2020), 32–53. DOI: https://doi.org/10.1002/nme.6187.

[53] M. Raissi, P. Perdikaris, and G. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". *Journal of Computational Physics* 378 (2019), 686–707. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.10.045.

[54] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes. "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics". *Computer Methods in Applied Mechanics and Engineering* 379 (2021), 113741. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.113741.

[55] R. Arora, P. Kakkar, B. Dey, and A. Chakraborty. *Physics-informed neural networks for modeling rate- and temperature-dependent plasticity*. 2022. DOI: 10.48550/arxiv.2201.08363.

[56] F. Masi, I. Stefanou, P. Vannucci, and V. Maffi-Berthier. "Thermodynamics-based Artificial Neural Networks for constitutive modeling". *Journal of the Mechanics and Physics of Solids* 147 (2021). ISSN: 00225096. arXiv: 2005.12183.

[57] F. Masi and I. Stefanou. "Multiscale modeling of inelastic materials with Thermodynamics-based Artificial Neural Networks (TANN)". *Computer Methods in Applied Mechanics and Engineering* 398 (2022), 115190. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.115190.

[58] M. Rosenkranz, K. A. Kalina, J. Brummund, and M. Kästner. "A comparative study on different neural network architectures to model inelasticity". *International Journal for Numerical Methods in Engineering* 124.21 (2023), 4802–4840. DOI: https://doi.org/10.1002/nme.7319.

[59] N. N. Vlassis and W. Sun. "Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening". *Computer Methods in Applied Mechanics and Engineering* 377 (2021), 113695. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.113695.

[60] X. He and J.-S. Chen. "Thermodynamically consistent machine-learned internal state variable approach for data-driven modeling of path-dependent materials". *Computer Methods in Applied Mechanics and Engineering* 402 (2022). A Special Issue in Honor of the Lifetime Achievements of J. Tinsley Oden, 115348. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.115348.

[61] M. Eghbalian, M. Pouragha, and R. Wan. "A physics-informed deep neural network for surrogate modeling in classical elasto-plasticity". *Computers and Geotechnics* 159 (2023), 105472. ISSN: 0266-352X. DOI: https://doi.org/10.1016/j.compgeo.2023.105472.

[62] K. Xu, D. Z. Huang, and E. Darve. "Learning constitutive relations using symmetric positive definite neural networks". *Journal of Computational Physics* 428 (2021), 110072. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2020.110072.

[63] K. Garanger, J. Kraus, and J. J. Rimoli. "Symmetry-enforcing neural networks with applications to constitutive modeling". *Extreme Mechanics Letters* 71 (2024), 102188. ISSN: 2352-4316. DOI: https://doi.org/10.1016/j.eml.2024.102188.

[64] D. K. Klein, M. Fernández, R. J. Martin, P. Neff, and O. Weeger. "Polyconvex anisotropic hyperelasticity with neural networks". *Journal of the Mechanics and Physics of Solids* 159 (2022), 104703. ISSN: 0022-5096. DOI: https://doi.org/10.1016/j.jmps.2021.104703.

[65] D. K. Klein, F. J. Roth, I. Valizadeh, and O. Weeger. "Parametrized polyconvex hyperelasticity with physics-augmented neural networks". *Data-Centric Engineering* 4 (2023), e25. DOI: 10.1017/dce.2023.21.

[66] L. Linden, D. K. Klein, K. A. Kalina, J. Brummund, O. Weeger, and M. Kästner. "Neural networks meet hyperelasticity: A guide to enforcing physics". *Journal of the Mechanics and Physics of Solids* 179 (2023), 105363. ISSN: 0022-5096. DOI: https://doi.org/10.1016/j.jmps.2023.105363.

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

[68] K. Linka, M. Hillgärtner, K. P. Abdolazizi, R. C. Aydin, M. Itskov, and C. J. Cyron. "Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning". *Journal of Computational Physics* 429 (2021), 110010. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2020.110010.

[69] K. Linka and E. Kuhl. “A new family of Constitutive Artificial Neural Networks towards automated model discovery”. *Computer Methods in Applied Mechanics and Engineering* 403 (2023), 115731. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.115731.

[70] M. Flaschel, S. Kumar, and L. De Lorenzis. “Unsupervised discovery of interpretable hyperelastic constitutive laws”. *Computer Methods in Applied Mechanics and Engineering* 381 (2021), 113852. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.113852.

[71] M. Flaschel, S. Kumar, and L. De Lorenzis. “Automated discovery of generalized standard material models with EUCLID”. *Computer Methods in Applied Mechanics and Engineering* 405 (2023), 115867. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.115867.

[72] A. Joshi, P. Thakolkaran, Y. Zheng, M. Escande, M. Flaschel, L. De Lorenzis, and S. Kumar. “Bayesian-EUCLID: Discovering hyperelastic material laws with uncertainties”. *Computer Methods in Applied Mechanics and Engineering* 398 (2022), 115225. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.115225.

[73] J. A. Hernández, M. A. Caicedo, and A. Ferrer. “Dimensional hyper-reduction of nonlinear finite element models via empirical cubature”. *Computer Methods in Applied Mechanics and Engineering* 313 (2017), 687–722. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.10.022.

[74] Z. Liu, C. T. Wu, and M. Koishi. “A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials”. *Computer Methods in Applied Mechanics and Engineering* 345 (2019), 1138–1168. ISSN: 00457825. DOI: 10.1016/j.cma.2018.09.020. arXiv: 1807.09829.

[75] R. A. van Tuijl, J. J. C. Remmers, and M. G. D. Geers. “Integration efficiency for model reduction in micro-mechanical analyses”. *Computational Mechanics* 62.2 (Nov. 2017), 151–169. ISSN: 1432-0924. DOI: 10.1007/s00466-017-1490-4.

[76] T. Guo, O. Rokoš, and K. Veroy. “A reduced order model for geometrically parameterized two-scale simulations of elasto-plastic microstructures under large deformations”. *Computer Methods in Applied Mechanics and Engineering* 418 (2024), 116467. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2023.116467.

[77] T. Guo, V. G. Kouznetsova, M. G. D. Geers, K. Veroy, and O. Rokoš. “Reduced-Order Modeling for Second-Order Computational Homogenization With Applications to Geometrically Parameterized Elastomeric Metamaterials”. *International Journal for Numerical Methods in Engineering* 126.1 (2025), e7604. DOI: https://doi.org/10.1002/nme.7604.

[78] J. A. Hernández, J. R. Bravo, and S. A. Parga. “CECM: A continuous empirical cubature method with application to the dimensional hyperreduction of parameterized finite element models”. *Computer Methods in Applied Mechanics and Engineering* 418 (2024), 116552. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2023.116552.

[79] C. Czech, M. Lesjak, C. Bach, and F. Duddeck. “Data-driven models for crashworthiness optimisation: intrusive and non-intrusive model order reduction techniques”. *Structural and Multidisciplinary Optimization* 65.7 (2022). ISSN: 1615-1488. DOI: 10.1007/s00158-022-03282-1.

[80] Z. Liu. “Deep material network with cohesive layers: Multi-stage training and interfacial failure analysis”. *Computer Methods in Applied Mechanics and Engineering* 363 (2020), 112913. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.112913.

[81] Z. Liu. “Cell division in deep material networks applied to multiscale strain localization modeling”. *Computer Methods in Applied Mechanics and Engineering* 384 (2021), 113914. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.113914.

[82] T.-J. Wei, W.-N. Wan, and C.-S. Chen. *Deep Material Network: Overview, applications and current directions*. 2025. arXiv: 2504.12159 [cs.CE].

2

EMBEDDING CONSTITUTIVE MODELS IN DATA-DRIVEN MODELS

In this chapter, Physically Recurrent Neural Networks (PRNNs) are presented for the first time. PRNNs build on the flexibility and high-tailorability of NNs to integrate constitutive models into an encoder-decoder network architecture with several links with the computational homogenization procedure. The proposed approach is unbound to a specific class of constitutive model, is non-intrusive, and can be readily incorporated into multiscale frameworks.

These features are demonstrated in a set of numerical examples based on a composite micromodel made of elastic fibers embedded in an elastoplastic matrix. The network's performance in predicting the homogenized response of this micromodel is evaluated in two stages. The first stage dives into a series of challenging scenarios for conventional surrogate models and a thorough comparison with a state-of-the-art Recurrent Neural Network (RNN). In the second stage, the robustness of the PRNN is tested on two multiscale problems, where aspects such as speed-up and robustness are also assessed.

For coherence with the remaining parts of this thesis, some figures were updated, the introduction shortened and two new equations were introduced in Section 2.2.2 with respect to the published source material:

M. A. Maia, I. B. C. M. Rocha, P. Kerfriden, and F. P. van der Meer. "Physically recurrent neural networks for path-dependent heterogeneous materials: Embedding constitutive models in a data-driven surrogate". *Computer Methods in Applied Mechanics and Engineering* 407 (2023), 115934. DOI: https://doi.org/10.1016/j.cma.2023.115934

2.1. INTRODUCTION

As discussed in Chapter 1, while surrogate models gained popularity in accelerating multiscale simulations of complex materials, four key challenges continue to hinder their reliable and widespread adoption in real-world applications. First is the black-box nature of purely data-driven models, which restricts interpretability and explainability, creating opaqueness in understanding and enforcing physically consistent behavior. Second, the limited ability to extrapolate undermines the robustness of surrogate models in practical applications, where unseen loading scenarios are common. This is addressed by ever-larger training data sets, highlighting the third issue: the data-hungry nature of many data-driven models. Among them are Recurrent Neural Networks (RNNs), the most popular approach for dealing with path-dependent behavior as they can naturally handle sequential data. Lastly, although a recent effort has been made to carefully incorporate physics laws in NNs through high-tailorable architectures that fulfill several physics-related requirements by construction (e.g., CANNs and PANNs), extending and generalizing these models to complex material behaviors remains a big challenge.

In this chapter, we introduce a new class of neural networks, the Physically Recurrent Neural Networks (PRNNs), to address these issues. In this first contribution, the applications are focused on path-dependent materials and accelerating concurrent finite element simulations. In Section 2.2 the FE^2 method is presented, followed by a brief discussion on how RNNs work in Section 2.3, while in Section 2.4, the main features of the novel neural network are described. In Section 2.5, the Design of Experiments and methodology adopted for the comparative study shown in Section 2.6 is described. In this study, the performance of the proposed network is compared to an RNN for a single-scale problem. In Section 2.7, the novel approach is integrated into an FE^2 framework and tested in two applications for robustness and accuracy. In Section 2.8, the network is tested for other combinations of material models to illustrate its flexibility. Finally, conclusions are presented in Section 2.9.

2.2. CONCURRENT MULTISCALE ANALYSIS

Let Ω define the macroscopic domain being modeled. To find the internal stresses and displacement field of such body in absence of body forces, a boundary value problem that satisfies the following equilibrium equations is defined as

$$\operatorname{div}(\boldsymbol{\sigma}^{\Omega}) = \mathbf{0} \tag{2.1}$$

where $\operatorname{div}(\cdot)$ is the divergence operator and $\boldsymbol{\sigma}^{\Omega}$ is the macroscopic stress, which depends on the macroscopic displacement field $\mathbf{u}^{\Omega}$ (for simplicity, this dependence is omitted). The governing equations are subjected to the boundary conditions

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

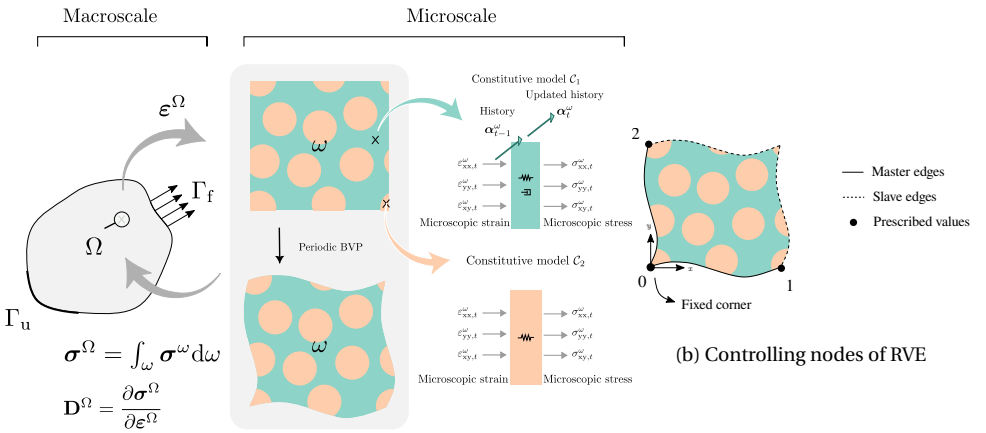
where $\mathbf{n}$ is the normal to the surface Γ_f and $\mathbf{u}^{\Gamma_u}$ and $\mathbf{t}^{\Gamma_f}$ represent a set of Dirichlet and Neumann boundary conditions acting on the body surface such that $\Gamma_u \cap \Gamma_f = \emptyset$ as illustrated in Fig. 2.1a. To relate strains and stresses, a constitutive model $\mathcal{D}^{\Omega}$ is required:

$$\boldsymbol{\sigma}^{\Omega} = \mathcal{D}^{\Omega}\left(\boldsymbol{\varepsilon}^{\Omega}, \boldsymbol{\alpha}^{\Omega}\right) \tag{2.3}$$

where $\boldsymbol{\alpha}^{\Omega}$ are history variables that account for path-dependency and $\boldsymbol{\varepsilon}^{\Omega}$ is the macroscopic strain defined under small displacement assumptions as

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

In the concurrent multiscale approach, the model $\mathcal{D}^{\Omega}$ is not directly formulated but is instead obtained by nesting a lower scale finite element model to each integration point. In that scale, the microscopic structure of complex materials can be explicitly modeled using simpler constitutive models for each of the components. Further discussion on how to solve the microscopic problem and link both scales is shown in Sections 2.2.1 and 2.2.2.



(a) Schematic representation of FE^2

(b) Controlling nodes of RVE

Figure 2.1: Scheme of FE^2 framework and definition of the boundary value problem on RVE.

To solve the boundary value problem at the macroscale, the Finite Element (FE) method is employed to discretize the domain Ω into a number of elements connected by nodes with N degrees of freedom. The global equilibrium is solved iteratively in its discretised weak form

$$\mathbf{r} = \mathbf{f}^{\Gamma} - \mathbf{f}^{\Omega}(\mathbf{u}^{\Omega}) = \mathbf{0} \tag{2.5}$$

where $\mathbf{r} \in \mathbb{R}^N$ is a residual vector that goes to zero when equilibrium is reached, $\mathbf{f}^{\Gamma} \in \mathbb{R}^N$ is the global external vector that represents the Neumann boundary conditions and $\mathbf{f}^{\Omega} \in \mathbb{R}^N$ is the global internal force vector given by a volume integral

$$\mathbf{f}^{\Omega} = \mathop{\mathrm{A}}_{e=1}^{n_e} \int_{\Omega_e} \mathbf{B}_e^{\mathrm{T}} \boldsymbol{\sigma}^{\Omega}(\mathbf{u}_e^{\Omega}) \, \mathrm{d}\Omega \tag{2.6}$$

where A is an assembly operator that takes into account the connectivities between the elements and the global system and $\mathbf{B}_e$ is a matrix with the spatial derivatives of the

shape functions used to interpolate nodal displacements of element e. Finally, an iterative procedure is adopted to solve Eq. (2.5) for the macroscopic displacement field

$$\Delta \mathbf{u}^{\Omega} = \mathbf{u}_n^{\Omega} - \mathbf{u}_o^{\Omega} = -\mathbf{K}_o^{-1}\mathbf{r}_o \tag{2.7}$$

where the subscripts o and n refer to old and new analysis increments, respectively and $\mathbf{K} \in \mathbb{R}^{N\times N}$ is the global tangent stiffness matrix given by

$$\mathbf{K} = \bigwedge_{e=1}^{n_e} \int_{\Omega_e} \mathbf{B}_e^{\mathrm{T}}\, \mathbf{D}_e^{\Omega}(\mathbf{u}_e^{\Omega})\, \mathbf{B}_e \, \mathrm{d}\Omega, \tag{2.8}$$

and $\mathbf{D}^{\Omega}$ is the constitutive tangent matrix, discussed in Section 2.2.2.

The key difference to a classical FE simulation lies in the embedding of another FE model in the macroscopic integration points. Here, to obtain the internal forces in Eq. (2.6) and the tangent stiffness matrix in Eq. (2.8) for a single integration point of the macroscale, one needs to run an entire FE model instead of a single evaluation of a homogeneous material model. This is the most computationally expensive part of the framework and is where the proposed network aims to tackle. The approach here is to replace the solution to the microscopic problem (discussed in Section 2.2.1) with a surrogate model, specifically a neural network. The homogenization procedure required to upscale the responses to the macroscale is discussed in Section 2.2.2.

2.2.1. MICROSCOPIC SCALE

Let ω be a Representative Volume Element (RVE) of the microscopic material features whose behavior is to be upscaled. Assuming that the principle of separation of scales (*i.e.* $\Omega \gg \omega$) holds, the two scales can be linked by enforcing

$$\mathbf{u}^{\omega} = \boldsymbol{\varepsilon}^{\Omega}\mathbf{x}^{\omega} + \tilde{\mathbf{u}} \tag{2.9}$$

where the linear displacement field is the result of the imposed macroscopic strains $\boldsymbol{\varepsilon}^{\Omega}$ and the fluctuation field $\tilde{\mathbf{u}}$ is the result of microscopic inhomogeneities. The principle of separation of scales implies the strain averaging theorem that states that the macroscopic strains are considered uniform over the RVE domain

$$\boldsymbol{\varepsilon}^{\Omega}(\mathbf{x}^{\Omega}) = \frac{1}{|\omega|}\int_{\omega} \boldsymbol{\varepsilon}^{\omega}(\mathbf{x}^{\omega})\, \mathrm{d}\omega \tag{2.10}$$

where $\boldsymbol{\varepsilon}^{\omega}$ is the microscopic strain tensor. Therefore, the microscopic displacement field in Eq. (2.9) can only satisfy Eq. (2.10) if the fluctuation displacement field vanishes at the RVE boundary when upscaling quantities. An additional requirement on the fluctation field having zero resultant work at the boundaries arises from the Hill-Mandel principle. Both requirements are met using Periodic Boundary Conditions (PBC) to represent the behavior of a macroscopic bulk material point. Fig. 2.1b illustrates the node groups and boundary edges needed to implement the PBC. In Section 2.5, the generation of $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ paths for the training of the surrogate models is obtained by setting a user-defined function to set the prescribed displacements $\mathbf{u}_1$ and $\mathbf{u}_2$, corresponding to the controlling nodes shown in Fig. 2.1b.

Finally, keeping the hypothesis of small strains, the stress equilibrium problem is described as

$$\nabla \boldsymbol{\sigma}^{\omega} = \mathbf{0} \qquad \boldsymbol{\sigma}^{\omega} = \mathcal{D}^{\omega}\left(\boldsymbol{\varepsilon}^{\omega}, \boldsymbol{\alpha}^{\omega}\right) \qquad \boldsymbol{\varepsilon}^{\omega} = \frac{1}{2}\left(\nabla \mathbf{u}^{\omega} + (\nabla \mathbf{u}^{\omega})^{\mathrm{T}}\right) \tag{2.11}$$

where $\mathbf{u}^{\omega}$ is the microscopic displacement field and $\boldsymbol{\sigma}^{\omega}$ and $\boldsymbol{\varepsilon}^{\omega}$ are the microscopic stress and strain tensors, respectively. An analogous procedure to the one detailed in Section 2.2 is used to find the microscopic displacement field (subjected to the periodic boundary conditions). Note that at this scale, regular physics-based material models $\mathcal{D}^{\omega}$ (*e.g.* elastoplasticity, viscoelasticity, etc.) are employed to represent the constitutive behavior of the homogeneous material of the discretized elements.

2.2.2. HOMOGENIZATION PROCEDURE

After convergence of the microscopic displacement field $\mathbf{u}^{\omega}$, the upscaling procedure is performed based on the Hill-Mandel principle. The principle postulates that the variation of the macroscopic stress power must equal the variation of volume average of the microscopic power over the RVE. Formulated in terms of virtual work, it reads

$$\frac{1}{|\omega|}\int_{\omega} \boldsymbol{\sigma}^{\omega} : \delta\boldsymbol{\varepsilon}^{\omega} \mathrm{d}\omega = \boldsymbol{\sigma}^{\Omega} : \delta\boldsymbol{\varepsilon}^{\Omega}. \tag{2.12}$$

Considering the definition in Eq. (2.9) and the microscopic strain expression in Eq. (2.11), as well as the use of PBC, the left-hand side of Eq. (2.12) can be rewritten as

$$\begin{aligned}
&\frac{1}{|\omega|}\int_{\omega} \boldsymbol{\sigma}^{\omega} : \delta\left(\boldsymbol{\varepsilon}^{\Omega} + \tilde{\boldsymbol{\varepsilon}}\right) \mathrm{d}\omega \\
&= \frac{1}{|\omega|}\int_{\omega} \boldsymbol{\sigma}^{\omega} : \underbrace{\delta\boldsymbol{\varepsilon}^{\Omega}}_{\text{Constant over RVE}} \mathrm{d}\omega + \underbrace{\frac{1}{|\omega|}\int_{\omega} \boldsymbol{\sigma}^{\omega} : \tilde{\boldsymbol{\varepsilon}} \mathrm{d}\omega}_{\text{Vanishes with PBC}} \\
&= \left(\frac{1}{|\omega|}\int_{\omega} \boldsymbol{\sigma}^{\omega} \mathrm{d}\omega\right) : \delta\boldsymbol{\varepsilon}^{\Omega}
\end{aligned} \tag{2.13}$$

where $\tilde{\boldsymbol{\varepsilon}}$ corresponds to the microscopic fluctuation strain field. Comparing the last expression in Eq. (2.13) to the right-hand side of Eq. (2.12), we recognize the homogenized stress as

$$\boldsymbol{\sigma}^{\Omega} = \frac{1}{|\omega|}\int_{\omega} \boldsymbol{\sigma}^{\omega} \mathrm{d}\omega. \tag{2.14}$$

In practice, we use the divergence theorem to transform the volume integral to a surface integral over the RVE transverse boundaries. As for the macroscopic constitutive tangent stiffnes matrix $\mathbf{D}^{\Omega}$, a probing operator $\mathcal{P}$ is applied on the global microscopic tangent stiffness matrix $\mathbf{K}^{\omega}$ without the need to invert it as proposed by Nguyen *et al.* [1].

2.3. RECURRENT NEURAL NETWORKS

In this section, a brief overview of the working mechanisms of Recurrent Neural Networks is presented. Although part of this chapter is dedicated to comparing them with

the novel approach, here the idea is to use well-known concepts from ANNs and RNNs to illustrate features of the proposed network in the following sections.

As a starting point, consider a conventional feed-forward neural network to surrogate the nonlinear constitutive relationship of a path-independent material given by the following parametric regression model:

$$\widehat{\boldsymbol{\sigma}}^{\Omega} = \mathcal{F}\left(\boldsymbol{\varepsilon}^{\Omega}, \mathbf{W}, \mathbf{b}\right) \tag{2.15}$$

where $\mathbf{W}$ and $\mathbf{b}$ are weights and biases calibrated through a fitting procedure based on observations of the actual microscopic model. During training, the strains are fed to the first neural layer (input layer) and values are propagated until the final layer (output layer) to give the predicted stresses $\widehat{\boldsymbol{\sigma}}^{\Omega}$. These are in turn compared to the ground truth value according to a loss function. Based on that, the model parameters are adjusted so that the error between the predicted stresses and the actual stresses is minimized

$$\mathbf{W}, \mathbf{b} = \operatorname{argmin}_{\overline{\mathbf{W}}\,\overline{\mathbf{b}}} \sum_{i \in \mathbf{X}} \parallel \boldsymbol{\sigma}_i^{\Omega}(\boldsymbol{\varepsilon}_i^{\Omega}) - \widehat{\boldsymbol{\sigma}}_i^{\Omega}(\boldsymbol{\varepsilon}_i^{\Omega}, \overline{\mathbf{W}}, \overline{\mathbf{b}}) \parallel^2 \tag{2.16}$$

where $\mathbf{X} \in \mathbb{R}^{n_\varepsilon \times N}$ is a snapshot matrix with N pairs of $\boldsymbol{\varepsilon}^{\Omega}$-$\boldsymbol{\sigma}^{\Omega}$ obtained from microscopic simulations. This setting is the most straight-forward way to map pairs of macroscopic strains and stresses but does not offer good generalization properties once path-dependency is introduced. In that case, one way to overcome the lack of history information is to extend their feature space with *e.g.* previous (incremental) strains and/or stresses [2, 3].

As an alternative, RNNs offer additional parameters (*i.e.* the hidden state) and mechanisms (*i.e.* the gates that control the flow of information being propagated) to learn history information from sequential data in an implicit way. These parameters describe the evolution of the so-called hidden state and can encapsulate information from previous iterations without the need to introduce history variables in the feature space. In a regular RNN, the outputs and hidden state are given by

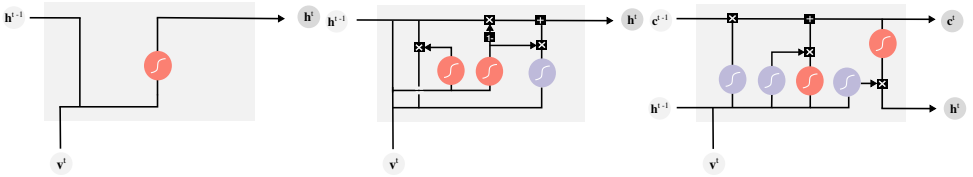
$$\begin{aligned} \mathbf{h}^t &= \phi(\mathbf{W}_1 \mathbf{v}^t + \mathbf{W}_s \mathbf{h}^{t-1} + \mathbf{b}_s) \\ \widehat{\boldsymbol{\sigma}}^t &= \phi(\mathbf{W}_2 \mathbf{h}^t + \mathbf{b}_2) \end{aligned} \tag{2.17}$$

where $\phi(\cdot)$ is an activation function, $\mathbf{W}_s$ and $\mathbf{b}_s$ are the additional model parameters (compared to conventional feed-forward neural networks), $\mathbf{v}^t$ are the current neuron values coming from the last layer and $\mathbf{h}^t$ and $\mathbf{h}^{t-1}$ are current and previous states, respectively. This arrangement allows the network to learn how stress evolves for a sequence of strains instead of building a regression model from independent stress-strain pairs and is illustrated in Fig. 2.2a. However, in practice, the efficiency of RNNs are impeded by vanishing gradient problems and are not suitable for long-term history dependent problems.

To overcome that, more sophisticated architectures (more popularly known as cells) have been proposed. Among the most popular ones are the Gated Recurrent Unit (GRU) and the Long-Short Term Memory (LSTM), illustrated in Figs. 2.2b and 2.2c, respectively. The internal mechanisms, also known as gates, used to control the flow of information passing from one state to another are represented by the colored circles. For each gate,

additional parameters need to be learned by the network in a way that the element-wise application of the sigmoid (red circles) or tanh (purple circles) functions can wisely retain what should be preserved and what can be forgotten in a long sequence.

Despite best efforts, these architectures are still vulnerable to overfitting, compromising their ability to generalize well to new data. Potential solutions to prevent this phenomenon include regularization techniques such as L2 penalty, early stopping and dropout. In this chapter, a special type of dropout proposed by Kingma, Salimans, and Welling [4] is used in combination with a GRU architecture is considered to perform the comparison with the proposed network. In this Bayesian GRU, the regular dropout with continuous noise (*i.e.* Gaussian dropout) is reinterpreted as a variational method that allows optimal dropout rates to be inferred from the data as opposed to it being fixed and defined in advance as usual. This circumvents the need for a validation set during model selection. For more details, the interested reader is referred to [4].



(a) Classical Recurrent cell (b) Gated Recurrent Unit (GRU) cell (c) Long-Short Term Memory (LSTM) cell

Figure 2.2: Different architectures for recurrent cells: red circles correspond to the element-wise application of the sigmoid function, while purple circles correspond to the tanh function.

2.4. PHYSICALLY RECURRENT NEURAL NETWORKS

This section presents the neural network proposed to capture path-dependent behavior of heterogeneous microscopic models. The core task of the network is to learn how the macroscopic strain $\boldsymbol{\varepsilon}^{\Omega}$ can be dehomogenized into a small set of representative material points and how their responses can be combined to obtain the homogenized macroscopic stresses $\boldsymbol{\sigma}^{\Omega}$. For this, the parametric regression model $\mathcal{R}$ illustrated in Fig. 2.3 is proposed: a combination of a data-driven encoder, a material layer with embedded physics-based material models and a data-driven decoder. Each of these components are discussed in detail in Sections 2.4.1 to 2.4.3, respectively. Finally, the training process is described in Section 2.4.4 and the use of this network as constitutive model in FE^2 frameworks is discussed in Section 2.4.5.

2.4.1. ENCODER

The encoder consists of all parameters that convert the macroscopic strain from the input layer to the values used as input of the material layer, which corresponds to the grey lines in Fig. 2.3. Since these values are the inputs of actual material models that are

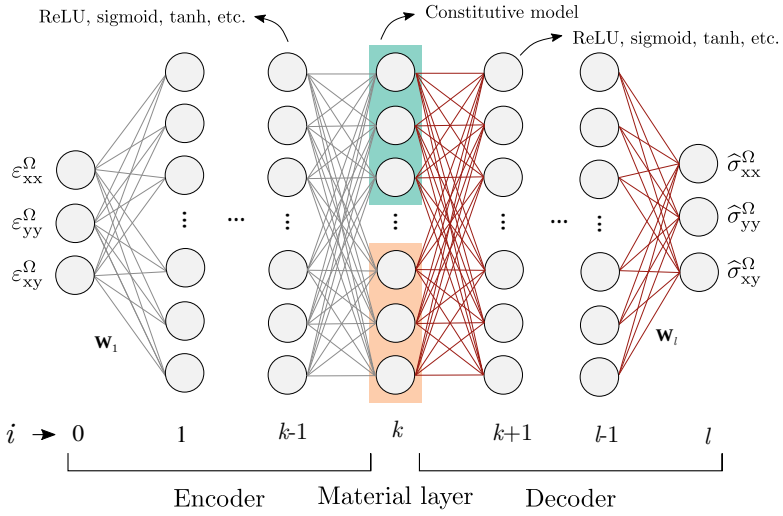


Figure 2.3: The proposed architecture.

later combined by the decoder into the prediction of the macroscopic stresses, we interpret the role of the encoder as being the microscopic periodic boundary-value problem (BVP) solved with FE, only at a much lower computational cost. On the other hand, no information on the displacement field of the micromodel is retrieved by the network as the encoder is learned based exclusively on snapshots of macroscopic stresses. This understanding is depicted in Fig. 2.4 by the grey curved line linking the strains from the macroscopic scale and the fictitious strains seen by the material points in the network.

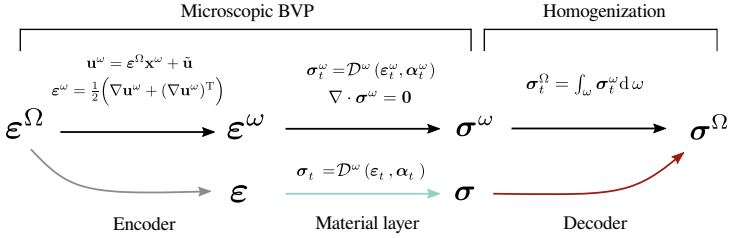


Figure 2.4: Interpretation of the proposed network with respect to a full-order solution.

As for the architecture, an arbitrary number of layers and units (with conventional activation functions as illustrated in Fig. 2.3) can be used. In case regular dense layers are employed, the neuron states ($\mathbf{a}_{i-1}$) from the previous layer i-1 are propagated to the following layer i according to

$$\mathbf{v}_i = \mathbf{W}_i \mathbf{a}_{i-1} + \mathbf{b}_i \qquad \mathbf{a}_i = \phi(\mathbf{v}_i), \tag{2.18}$$

where $\mathbf{W}_i \in \mathbb{R}^{n_i \times n_{i-1}}$ is a weight matrix and $\mathbf{b}$ is a bias term with n_i being the number of neurons of layer i, and ϕ is an activation function applied in an element-wise manner

to the neuron values of i to introduce nonlinearity into the network. In the particular case where the dense layer is either the input or the output layers, no activation function is applied and $\mathbf{v}_0$ is set to $\boldsymbol{\varepsilon}^\Omega$. This results in $\mathbf{a}_0 = \mathbf{v}_0 = \boldsymbol{\varepsilon}^\Omega$. Popular activation functions include the sigmoid, tanh and ReLU.

In the present investigation, the main network architecture considered for the numerical applications consists of three layers: input, material and output layers. This results in an encoder with linear relationship between macroscopic strains and local strains, as no activation function is applied to the input layer. It is worth stressing that this architecture does not yield path-dependent local strain paths, in contrast to the actual RVE where the strain distribution will generally be path-dependent. However, the homogenized response is path-dependent, through the history variables in the material layer. The effect of introducing path-dependency to the encoder is discussed in Appendix.

2.4.2. MATERIAL LAYER

The material layer is responsible for introducing explicitly the same physics-based material models used in the RVE that the network will be a surrogate for. To properly incorporate them and take full advantage of its outputs, important changes on how neurons are evaluated compared to regular dense layers are proposed. First, instead of introducing nonlinearity in a element-wise manner with a scalar-to-scalar activation function, neurons are grouped in m sets of the size of the input layer (see colored boxes in Fig. 2.3) and then evaluated as a subgroup. Each subgroup is referred to as a *fictitious material point* and its size is equal to the length of the strain vector (*i.e.* length 3 for the present investigation in two dimensions). In this arrangement, each neuron of the subgroup j represents one component of the strain vector $\boldsymbol{\varepsilon}_j$, as illustrated in Fig. 2.5a.

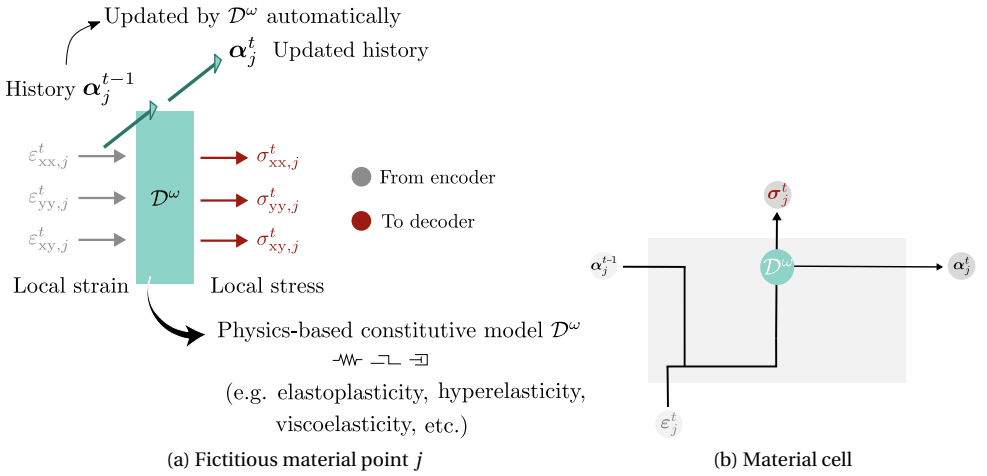


Figure 2.5: Schemes of (a) fictitious material point and its view as (b) as a cell.

Supposing the micromodel contains n material models $\mathcal{D}_1^\omega$, ... $\mathcal{D}_n^\omega$, several combinations of them can be employed in the material layer. The choice on which material

models should be used to evaluate the fictitious material points depends on the types of non-linearity embedded in each of these models. For simplicity, we choose to illustrate the network with a general material model $\mathcal{D}^{\omega}$ for all fictitious material points. Such model can take the form of any of the n material models $\mathcal{D}_i^{\omega}$ with known material properties used in the micromodel. Here, $\mathcal{D}^{\omega}$ takes as input the current strain $\boldsymbol{\varepsilon}^t \in \mathbb{R}^{n_\varepsilon}$ and the internal variables from previous time step $\boldsymbol{\alpha}^{t-1} \in \mathbb{R}^{n_{\text{IntVar}}}$, where n_{IntVar} is the number of internal variables of the material model. With that, the model is used to evaluate current stress state $\boldsymbol{\sigma}^t \in \mathbb{R}^{n_\varepsilon}$ and updated internal variables $\boldsymbol{\alpha}^t$. These quantities motivated the tailor-made architecture of the proposed layer.

To store the internal variables used as input/output of the material model, an auxiliary vector $\mathbf{h}_j \in \mathbb{R}^{n_{\text{IntVar}}}$ referred as history vector is defined. In the particular case of a subgroup with a material model with no internal variables (*e.g.* linear elastic model), $\mathbf{h}_j$ does not exist since $n_{\text{IntVar}} = 0$. For the first time step, the history vector is initialized as zero for all m subgroups. As information reaches the material layer and the material model is called, three outputs are made available: the stresses $\boldsymbol{\sigma}_j^t$, the updated internal variables $\boldsymbol{\alpha}_j^t$ and the tangent stiffness matrix $\mathbf{D}_j^t \in \mathbb{R}^{n_\varepsilon \times \varepsilon}$. In this layer, only the stresses are propagated forward. To do this, each stress component is associated to a unit of the subgroup, as illustrated in Fig. 2.5a. Then, the updated internal variables $\boldsymbol{\alpha}^t$ are stored in $\mathbf{h}_j^t$ so that when new strains $\boldsymbol{\varepsilon}_j^{t+1}$ are fed to the fictitious material point, the material model is aware of its own history so far, making the $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ path of each subgroup unique. This architecture is illustrated in Fig. 2.5b. For the sake of notation clarity, from now on we omit the time index t when referring to current values.

Note that $\mathbf{h}_j$ is not learned through a set of parameters, but obtained as an automatic output of the (path-dependent) material model employed in subgroup j. This works as the physical memory of the network as it stores the history variables that describe a given fictitious material point. Using internal variables obtained directly from the material models is where the proposed approach crucially differs from purely data-driven RNNs. This is further discussed in Section 2.4.6, where we assess how this network compares to other methods in the literature. It is also worth mentioning that since no data from the microscale has been collected and imposed in the network, the paths seen by the fictitious material points do not need to hold any similarity with actual integration points of the microscopic model.

Using standard machine learning notation, the material layer propagates previous neuron states ($\mathbf{a}_{k-1}$) and applies the material model $\mathcal{D}^{\omega}$ as follows

$$\mathbf{v}_k = \mathbf{W}_k \mathbf{a}_{k-1} + \mathbf{b}_k \Rightarrow \mathbf{a}_k, \mathbf{h} = \mathcal{D}^{\omega}(\mathbf{v}_k, \mathbf{h}^{t-1}) \tag{2.19}$$

where $\mathbf{W}_k \in \mathbb{R}^{n_k \times n_{k-1}}$ is the weight matrix connecting layers k-1 and k, $\mathbf{b}_k \in \mathbb{R}^{n_k}$ is a bias term. In addition, $\mathbf{v}_k$ are the neuron values (correspond to the concatenated vector of all microscopic strains $\boldsymbol{\varepsilon}_j$), $\mathbf{h}^{t-1}$ and $\mathbf{h}$ are history-related term (correspond to concatenated vector of all internal variables $\boldsymbol{\alpha}_j^{t-1}$) resulting from the material models with path-dependent behavior from past and current time step and $\mathbf{a}_k$ are the current neuron states (correspond to the concatenated vector of all microscopic stresses $\boldsymbol{\sigma}_j$).

CHOICE OF CONSTITUTIVE MODEL

A general guideline on how to select the constitutive models used for evaluating the fictitious material points is to employ all different sources of nonlinearity with their respective known material properties in the material layer. To illustrate that, consider the micromodel used in Sections Sections 2.6 and 2.7, a composite microstructure with two material models: a linear elastic model $\mathcal{D}_2^\omega$ to describe the fibers and an elastoplastic model with isotropic hardening $\mathcal{D}_1^\omega$ to describe the matrix. The latter starts as linear-elastic and evolves into the plastic regime once the yield stress is reached. Motivated by that, only model $\mathcal{D}_1^\omega$ is employed in all fictitious material points since the network still can make any of the subgroups to behave linear elastically by passing small strains to the material model and subsequently scaling the stresses to give a significant elastic contribution through the decoder. This is illustrated in Section 2.7.1 in a numerical example, where the response of all fictitious material points are shown for a single macroscopic point.

However, if instead of a linear elastic model, a nonlinear elastic model was used to describe the fibers, the network would not perform optimally. In that case, although the elastoplastic model does introduce nonlinearity to the network, the (nonlinear) contribution from the fibers is no longer embedded in that model. Furthermore, if the nonlinear elastic model was the one chosen to evaluate all subgroups, the network would essentially become a feed-forward one with no history information taken into account (explicitly or implicitly), losing the ability to predict elastic unloading. For such a micromodel, both material models would need to be considered.

Another interesting case is that of a micromodel with two elastoplastic phases with different material properties. This time, depending on the contrast of the material properties, a single material model with a fixed set of properties coming from one of the two phases might be enough to reproduce the homogenized response of the micromodel. While both cases are illustrated in Section 2.8, naturally, far more complex arrangements than the ones discussed here are found in practice. This is also true for the potential extensions to the current approach. In the last scenario, for instance, making the material properties of each fictitious material point a trainable feature might be advantageous. This could also be a valuable feature when dealing with experimental data or with a micromodel with continuously varying material properties. Addressing these extensions is object of ongoing research.

2.4.3. DECODER

The decoder consists of all parameters that convert the outputs from the material layer to the predicted macroscopic stress $\widehat{\boldsymbol{\sigma}}^\Omega$ in the output layer, which corresponds to the brown lines in Fig. 2.3. Similar to the encoder, an arbitrary number of conventional layers and units can be employed. In the full-order solution, after convergence of the microscopic BVP, the macroscopic stresses are obtained by the volume average of microscopic stresses over the entire RVE. In the network, since the solution of the microscopic BVP is replaced by the encoder and the microscopic material points in the RVE are replaced by the few fictitious material points, the decoder is then analogous to the homogenization operator that transforms local stresses to macroscopic stresses, as illustrated by the brown curved line in Fig. 2.4.

In the present work, we use a single dense layer (output) with linear activation and physics-motivated modifications to perform the task. With this, all the nonlinearity of the network arises from the models in the material layer. As discussed previously, the decoder can be understood as the averaging operator in a multiscale approach and with the chosen architecture (dense-material-dense), the weights of the output layer can be seen as the relative contribution of each fictitious material point to the macroscopic stress. Based on that, a constraint on the positivity of the weights of the output layer is considered. For that, a softplus function $\rho(\cdot)$ is applied element-wise on the weights matrix before computing the neuron values of the last layer

$$\mathbf{v}_l = \rho(\mathbf{W}_l)\, \mathbf{a}_{l-1} + \mathbf{b}_l \tag{2.20}$$

where $\mathbf{b}_l$ is set to zero and $\mathbf{a}_{l-1}$ corresponds to the stresses coming from the material layer. This procedure guarantees that, after the transformation, weights will always be positive.

2.4.4. TRAINING

The goal of the training phase is to minimize a loss function given by

$$\ell_{\text{avg.}} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \parallel \boldsymbol{\sigma}^{\Omega}(\boldsymbol{\varepsilon}_i^{\Omega}) - \widehat{\boldsymbol{\sigma}}^{\Omega}(\boldsymbol{\varepsilon}_i^{\Omega}) \parallel^2 \tag{2.21}$$

where N is the number of snapshots. Based on it, a Stochastic Gradient Descent (SGD) optimization algorithm is used to update the trainable parameters $\mathbf{W}$ and $\mathbf{b}$

$$\begin{aligned} \mathbf{W}^n &= \mathbf{W}^o - \mathcal{A}\Big(\frac{1}{B}\sum_{i=1}^{B} \frac{\partial L_i}{\partial \mathbf{W}}\Big) \\ \mathbf{b}^n &= \mathbf{b}^o - \mathcal{A}\Big(\frac{1}{B}\sum_{i=1}^{B} \frac{\partial L_i}{\partial \mathbf{b}}\Big) \end{aligned} \tag{2.22}$$

where L_i is the loss of the i-th sample, o indicates current values, n indicates updated values and B is the size of the sample mini-batch used in the update. Finally the $\mathcal{A}$ operator depends on the solver. In this case, the Adam optimizer [5] is used.

To compute the gradients appearing in Eq. (2.22), backpropagation in time is employed in a similar fashion as done to RNNs: based on the network state ($\mathbf{v}$ and $\mathbf{a}$) after computing each training curve with n pairs of $\boldsymbol{\sigma} - \boldsymbol{\varepsilon}$, the chain rule is used to propagate the derivative of the loss functions starting from the output layer and progressively moving back through the network and through time. Commonly, this process is dealt with by automatic differentiation, but we present the expressions to allow for integrating the network directly into existing FE software. For this, two auxiliary quantities are defined. The first is defined for each layer and helps propagating the error through the network $\mathbf{d}_i \in \mathbb{R}^{n_i}$. Starting from the output layer l, it is defined as

$$\mathbf{d}_l = \frac{\partial L}{\partial \mathbf{a}_l} = \widehat{\boldsymbol{\sigma}}^{\Omega} - \boldsymbol{\sigma}^{\Omega}. \tag{2.23}$$

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

$$\bar{\mathbf{d}}_i = \mathbf{d}_i \odot \frac{\partial \phi(\mathbf{v}_i)}{\partial \mathbf{v}} \tag{2.24}$$

where $\odot$ represents the Hadamard product. After that, it is possible to compute the gradients of the trainable parameters:

$$\frac{\partial L}{\partial \mathbf{W}_i} = \bar{\mathbf{d}}_i \mathbf{a}_{i-1}^T \qquad \frac{\partial L}{\partial \mathbf{b}_i} = \bar{\mathbf{d}}_i. \tag{2.25}$$

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

$$\mathbf{d}_{i-1} = \mathbf{W}_i^T \bar{\mathbf{d}}_i, \tag{2.26}$$

and the algorithm moves to Eq. (2.24) for layer i-1.

When reaching the material layer, recall that the internal variables are stored in $\mathbf{h}$ and used for keeping track of the evolution of the internal variable through time. For that reason, a second auxiliary quantity is introduced and Eq. (2.24) is replaced by

$$\bar{\mathbf{d}}_i = \mathbf{d}_i \odot \frac{\partial \mathbf{a}_i}{\partial \mathbf{v}_i} + \mathbf{d}_h^{t+1} \odot \frac{\partial \mathbf{h}}{\partial \mathbf{v}_i} \tag{2.27}$$

where the first term concerns the derivatives of stresses with respect to strains, the second term concerns the derivatives of the current internal variables with respect to strains and $\mathbf{d}_h \in \mathbb{R}^{n_i}$ is given by

$$\mathbf{d}_h = \mathbf{d}_i \odot \frac{\partial \mathbf{a}_i}{\partial \mathbf{h}} + \mathbf{d}_h^{t+1} \odot \frac{\partial \mathbf{h}}{\partial \mathbf{h}^{t-1}}. \tag{2.28}$$

Note that the derivatives of the stresses with respect to the strains of material point j are an output of the material model: the tangent stiffness matrix $\mathbf{D}_j$. The remaining derivatives in Eqs. (2.27) and (2.28) are evaluated using central finite differences. Naturally, computing gradients with other methods would also be possible. For instance, if the material model used in the network supports automatic differentation, storing the internal variables in $\mathbf{h}$ for backpropagation can be bypassed as the derivatives are automatically obtained in this approach.

Finally, to obtain the gradients of the trainable parameters including the history-dependence coming from the material layer and compute the values $\mathbf{d}$ of the previous layer, we consider Eq. (2.27) instead of Eq. (2.24) in the expressions shown in Eq. (2.25) and Eq. (2.26), respectively.

2.4.5. USE AS CONSTITUTIVE MODEL

To make new stress predictions, the macroscopic strain $\boldsymbol{\varepsilon}^\Omega$ is fed to the input layer and a complete forward pass is performed. The final activated neuron values of the output layer give the predicted stress. To obtain the macroscopic consistent tangent stiffness matrix $\mathbf{D}^\Omega$, a complete backward pass is required:

$$\mathbf{D}^\Omega = \frac{\partial \widehat{\boldsymbol{\sigma}}^\Omega}{\partial \boldsymbol{\varepsilon}^\Omega} = \frac{\partial \mathbf{a}_l}{\partial \mathbf{v}_0} = \mathbf{J}, \tag{2.29}$$

which is obtained with a backward pass through the network

$$\mathbf{J}_i = \mathbf{J}_{i+1}\mathbf{I}_i^{\phi}\mathbf{W}_i \quad \text{with} \quad \mathbf{J}_{l+1} = \mathbf{I} \tag{2.30}$$

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

$$\mathbf{I}_i^{\phi} = \text{diag}(\frac{\partial \phi(\mathbf{v}_i)}{\partial v}), \tag{2.31}$$

except for the material layer. In that case, such matrix is full and consists of the concatenation of the tangent stiffness matrix of all fictitious material points. It is worth mentioning that despite the linear dependency on the tangent stiffness matrices of the material models, the Jacobian matrix of the network does not inherit their spectral properties.

2.4.6. ANALOGIES TO OTHER METHODS

In this section, the parallels between features of the proposed network and related works in the literature are briefly discussed. One possible analogy comes from hyper-reduced-order models [6]. With the architecture chosen for the present investigation, both methods work on a reduced number of material points with modified (integration) weights. However, in the network, these points are only fictitious and learned by the encoder based on snapshots of the homogenized stresses. Moreover, each stress component is associated with a different weight. By contrast, the material points in the hyper-reduction approach exist in the microscopic model and a single modified integration weight of each material point selected is used to compute all its stress/internal force components.

Following the discussion on the encoder, it is worth highlighting how this feature would be framed with respect to asymptotic homogenization schemes such as Mori-Tanaka [7]. In this type of solution, the microscopic problem is also not solved explicitly and only average fields are calculated. Relying on the equivalent inclusion idea and on Eshelby's solution [8], the strain concentration tensor is obtained analytically and yields the full solution of the microcopisc model as it correlates the average field of the phases in the micromodel with its average field. In our network, although the macroscopic stresses are also obtained by relating macroscopic and (fictitious) microscopic strains through an encoder, here no average field is calculated for each of the phases. Indeed, not every phase needs to be included in the material layer and multiple strain paths for the same phase are considered. Furthermore, while Mori-Tanaka is accurate for moderate volume fractions of the inclusions, such restriction is not present in our method.

Compared to PINNs, in which physical constraints are explicitly included in the loss function, here, most physical constraints are naturally taken care of by the physics-based material models directly embedded in the material layer. The proposed approach is also more general as it can be directly used for arbitrary material models and is not particularly tailored to a single type of model (*e.g.* elastoplastic behavior [9, 10]). As an added benefit, our model selection procedure makes physical sense: we add more material points or material models to the network.

Another noteworthy strategy with relevant analogies to our method is the DMN [11]. In this approach, the contribution of a few material points evaluated using the classical constitutive models in the RVE is also employed to make predictions in the online

phase. On the same reasoning as discussed in Section 2.4.2, since the inputs come directly from actual material models, path-dependency is captured naturally. However, the main concept and architecture of DMNs are different from the ones explored here. In the offline phase, the goal of the DMN is to find a topological representation of the RVE with fewer degrees of freedom (*i.e.* material points) based only on the elastic stiffness matrices of the different material phases that compose the original micromodel. For the online phase, the feature space is increased to include residual stresses of the micromodel components, and an iterative procedure is implemented. The authors compare the incremental strains of the material points at the beginning of the iteration with the one obtained by a de-homogenization process that backpropagates the macroscopic incremental strain from the output layer to the bottom layer (*i.e.* input layer). Upon convergence, the set of internal variables of each material point at the bottom layer is therefore updated.

In the present work, the feature space is the same in both phases and no iterative procedure is employed in the forward pass, which simplifies implementation and reduces even further the number of material model calls. Here, the strain path each fictitious material point follows is simply described by the encoder and not all phases need to be included in the network. The homogenized stresses and tangent stiffness are obtained in a single forward and backward pass, respectively. Furthermore, the backpropagation in our approach is considerably simpler than the DMN. Although the use of homogenization (and de-homogenization) operations in the DMN assigns physical interpretation to the model, it also makes training a rather intricate process.

Finally, to draw a parallel with LSTMs, one might understand $\mathbf{h}$ as the cell state $\mathbf{c}$, but instead of using bijective and smooth functions such as the sigmoid and tanh functions to describe the evolution of the material response, the material model itself is directly employed. This bypasses the need to learn new parameters to regulate the flow of information kept or forgotten throughout time (see Fig. 2.2b) and has important implications for the training process. The most important one is the ability to mirror physical behaviors such as elastic unloading/reloading without ever seeing the pattern during training, a stark contrast with LSTMs and GRUs that usually require extensive training sets with multiple cycles of loading and reloading at different strain levels with different step sizes. The physical interpretation of the nonlinearity is directly embedded in the network. In the numerical examples of this chapter, the nonlinearity is due to plasticity, but other effects such as hyperelasticity, visco-plasticity, stiffness degradation, or any combination thereof, could be embedded by adapting the constitutive model that is used in the material layer.

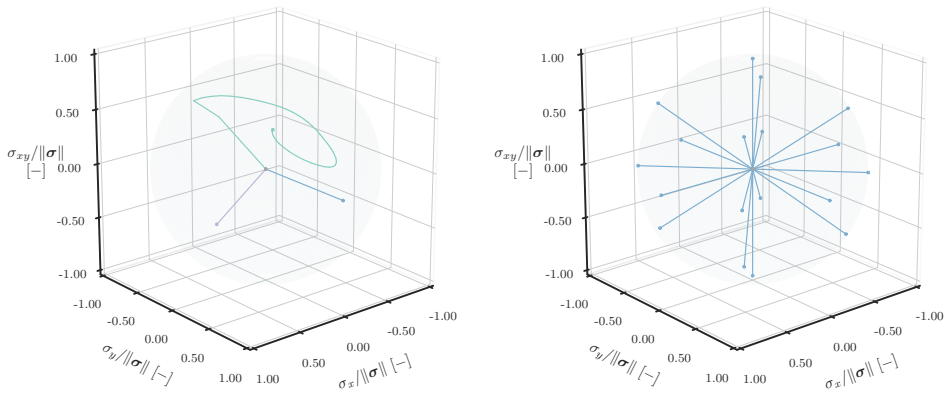
2.5. DESIGN OF EXPERIMENTS

One critical aspect of the training and testing of surrogate models is the formulation of a sampling plan. Typically, a uniform distribution of the sampling points is desirable, but that task becomes more complex when path-dependent behavior is present. In this case, pairs of strains and stresses are collected and processed as sequences, which leads to potentially infinite-dimensional parameter spaces.

Two strategies are considered for generating the loading paths for training, validat-

ing and testing. In the first approach, proportional loading paths are generated, which means that the stress ratio between the components is constant. Here, the sequence of strains is created based on two features: the loading function $\lambda(\Delta\varepsilon, t)$ and the loading direction given by the unit vector $\mathbf{n}$, where $\Delta\varepsilon$ is the step size and t is the current time step. For each time step, $\mathbf{n}$ is multiplied by the scalar-valued loading function λ creating a new set of strains, which is in turn applied at the controlling nodes of the microscopic model.

For monotonic loading, the loading function is as depicted in Fig. 2.7a. The values in the unit vector can come from prior knowledge of the material as illustrated in Fig. 2.6b, in which only fundamental cases such as uniaxial strain, pure shear, and biaxial cases are considered, or from random distributions as represented by the purple line Fig. 2.6a. In the present work, the random directions are obtained by sampling values from n_ε independent Gaussian distributions ($X \sim \mathcal{N}(0,1)$) and subsequently normalizing the vectors.



(a) Proportional loading path with *a priori* known direction (blue) and random direction (purple), and non-proportional loading path in random direction (green)

(b) Proportional loading with *a priori* known directions

Figure 2.6: 3D stress-time view of different types of loading paths investigated in this study.

Despite the simplicity in creating such paths, RNNs trained exclusively on monotonic cannot predict cyclic responses. Thus, to create non-monotonic sequences, a linear piecewise function as the one depicted in Fig. 2.7b is used. Note that even though the loading function is changed, the unit vector is kept constant for the entire strain sequence, yielding proportional loading. However, to cover the entire space of possible cyclic responses, a large (and *a priori* unknown) number of curves comprehending different unloading points with different duration of unloading/reloading and step sizes is necessary. For this study, that matter is first handled in a simplified way by only sampling two different cycles of unloading/reloading.

Finally, in a more general approach, a second strategy to create the $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ paths is considered: the *random walks*. These are typically defined by sampling random strain increments with random loading directions for each time step, resulting in non-proportional loading. In this chapter, random walks are created by associating the prescribed strains to independent Gaussian Processes (GPs) with $X \sim \mathcal{N}(\mu, \sigma^2)$ and covariance function given by

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x}_p - \mathbf{x}_q\|^2\right) \tag{2.32}$$

where $\mathbf{x}_p$ and $\mathbf{x}_q$ are the time step indices of the strain sequence being sampled, σ_f^2 is the variance and ℓ is a length scale. In this setting, the lengthscale controls the smoothness of the strain path and the variance controls how large the step size can be for each prescribed degree of freedom of the controlling nodes. Similar approaches were employed by Mozaffar *et al.* [12] and Logarzo, Capuano, and Rimoli [13].

Algorithm 1: Generation of random loading path using GPs

Input : lengthscale ℓ, variance σ_f^2, number of strain components $N_{\text{components}}$, number of time steps N_{steps}
Output: macroscopic strains $\mathfrak{D}_\varepsilon$ and macroscopic stresses $\mathfrak{D}_\sigma$

initialize datasets: $\mathfrak{D}_\varepsilon \leftarrow \varnothing$, $\mathfrak{D}_\sigma \leftarrow \varnothing$
for $i \in [1, 2, ..., N_{components}]$ **do**
  initialize input and output datasets for GP_i: $\mathbf{X}_{\text{GP}_i} \leftarrow 0$, $\mathbf{Y}_{\text{GP}_i} \leftarrow 0$
  initialize GP_i: $\text{GP}_i \leftarrow$ `initGP(` $\mathbf{X}_{\text{GP}_i}, \mathbf{Y}_{\text{GP}_i}, \ell, \sigma_f^2$)
for $t \in [1, 2, ..., N_{steps}]$ **do**
  initialize current macroscopic strain: $\boldsymbol{\varepsilon}_{\text{current}} \leftarrow \varnothing$
  for $i \in [1, 2, ..., N_{components}]$ **do**
    sample from posterior distribution: $\varepsilon_i \leftarrow \text{GP}_i$`::samplePosterior (` t `)`
    add value to strain vector: $\boldsymbol{\varepsilon}_{\text{current}} \leftarrow \boldsymbol{\varepsilon}_{\text{current}} \cup \varepsilon_i$
  solve micromechanical BVP: $\boldsymbol{\sigma}_{\text{current}} \leftarrow$ `fullModel::materialUpdate (` $\boldsymbol{\varepsilon}_{\text{current}}$ `)`
  if *convergence* **then**
    store equilibrium solution of micromodel: `fullModel::storeSolution( )`
    add macroscopic strains and stresses to dataset: $\mathfrak{D}_\varepsilon \leftarrow \mathfrak{D}_\varepsilon \cup \boldsymbol{\varepsilon}_{\text{current}}$, $\mathfrak{D}_\sigma \leftarrow \mathfrak{D}_\sigma \cup \boldsymbol{\sigma}_{\text{current}}$
    for $i \in [1, 2, ..., N_{components}]$ **do**
      add time step and current strain to dataset of GP_i: $\mathbf{X}_{\text{GP}_i} \leftarrow \mathbf{X}_{\text{GP}_i} \cup t$, $\mathbf{Y}_{\text{GP}_i} \leftarrow \mathbf{Y}_{\text{GP}_i} \cup \varepsilon_i$
      update GP_i with new data: GP_i`::update (` $\mathbf{X}_{\text{GP}_i}, \mathbf{Y}_{\text{GP}_i}$ `)`
return $(\mathfrak{D}_\varepsilon, \mathfrak{D}_\sigma)$

The details of the present implementation are given in Algorithm 1. Note that instead of drawing the entire strain sequence for a given component, we sample it step by step and update the GP dataset before sampling again. This strategy results in the same strain sequence given a fixed random seed throughout the steps, but in this way the GPs can

also be used in applications where the number of loading steps is changed on-the-fly. Following the work of Logarzo, Capuano, and Rimoli [13], we define the mean of all GPs to be zero and include $t = 0$ and $\varepsilon_i = 0$ as a prior. In addition to that, references to *fullModel* (*i.e.* the full-order microscopic model) in Algorithm 1 are kept as minimal and general as possible. One example of loading path resulting from Algorithm 1 is illustrated in Fig. 2.6a.

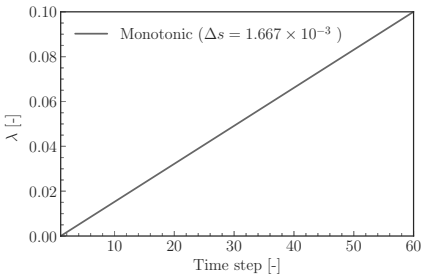
Both strategies generate $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ curves containing 60 time steps, unless stated otherwise. To summarize the types of loading studied in the following sections:

- Type I: monotononic and proportional loading paths with *a priori* known directions. The 18 directions used to train the proposed network are illustrated in Fig. 2.6b and include uniaxial strains, pure shear, biaxial cases and biaxial with shear cases.
- Type II: monotonic and proportional loading paths randomly spread across the design space. The loading directions are generated randomly and the loading function is as shown in Fig. 2.7a.
- Type III: non-monotonic and proportional loading paths randomly spread across the design space. Again, the loading directions are random, but the loading function is now given by Fig. 2.7b and includes one cycle of unloading;
- Type IV: Variations to Type III:
  - Type IVa: same loading directions as the test set of Type III, but unloading/reloading takes place at a different point in time as shown in Fig. 2.7c;
  - Type IVb: same loading directions as the test set of Type III, but time step is 10 × smaller. Thus, to reach the same norm as the original curve in Type II, 600 time steps are evaluated, as depicted in Fig. 2.7d;
- Type V: non-monotonic and non-proportional loading paths randomly spread across the design space. A GP-based path described by Eq. (2.32) is illustrated in Fig. 2.6a. Fig. 2.7e ilustrates the strain paths of each component using this approach with lengthscale $\ell = 20$ and $\sigma_f = 1.0 \times 10^{-3}$.

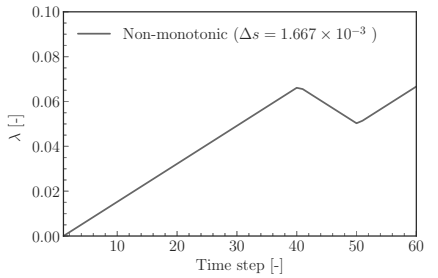
2.6. ASSESSING THE NETWORK PERFORMANCE

In this section, the performance of the proposed network is compared to a state-of-the-art RNN trained on different training dataset sizes and methods to sample the design space. The comparison is done for a single micromodel. Specifically, four scenarios are investigated: (i) predicting unloading/reloading behavior from monotonic data, (ii) predicting unloading/reloading behavior from non-monotonic data, (iii) predicting unseen patterns from non-monotonic data, and (iv) training with non-monotonic and non-proportional loading paths. In the first three scenarios, our network is trained exclusively on the fundamental loading cases of Type I (18 curves), while the training of the RNN is an open question to be addressed in the following sections.

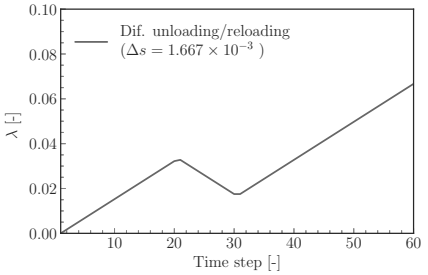
From now on, the network proposed in this chapter will be referred to as Physically Recurrent Neural Network, or simply PRNN. The PRNN was trained for 80 000 epochs,



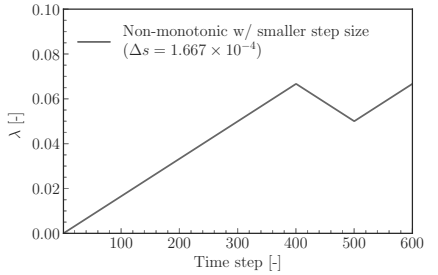
(a) Types I and II: monotonic



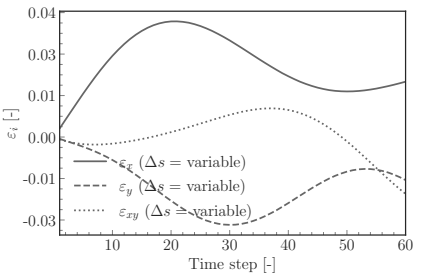
(b) Type III: non-monotonic



(c) Type IVa: different unloading/reloading



(d) Type IVb: non-monotonic with smaller step size



(e) Type V: random non-proportional and non-monotonic with $\ell = 20$

Figure 2.7: Proportional and non-proportional loading functions

while the RNN was trained for 60 000 epochs with an early stopping criterion, which consists of interrupting training if the best validation loss so far is not improved over 5000 epochs. The Adam optimizer is used in all cases with batch size of 9 and default parameters suggested by Kingma and Ba [5], with the exception of the learning rate of 0.01 for the RNNs. The layer sizes are chosen through model selection to provide optimal results and fair comparison with our approach to the best of our knowledge. The methodology adopted is briefly described in Section 2.6.1.

The microscopic model consists of an RVE with 36 elastic fibers (volume fraction = 0.6) with properties E = 74 000 MPa and $\nu = 0.2$ embedded in an elastoplastic matrix with isotropic hardening. The geometry and the mesh with 7048 elements are depicted in Fig. 2.8. The elastoplastic matrix is modeled using the von Mises yield criterion with properties E = 3130 MPa, $\nu = 0.3$ and yield stress given by

$$\sigma_y = 64.8 - 33.6 \exp(-\varepsilon_{eq}^p / 0.003407) \tag{2.33}$$

where ε_{eq}^p is the equivalent plastic strain defined as

$$\varepsilon_{eq}^p = \sqrt{\frac{2}{3}\,\boldsymbol{\varepsilon}^p : \boldsymbol{\varepsilon}^p}, \tag{2.34}$$

and $\boldsymbol{\varepsilon}^p$ is the plastic strain. Plane stress conditions are assumed.

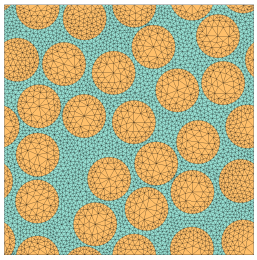


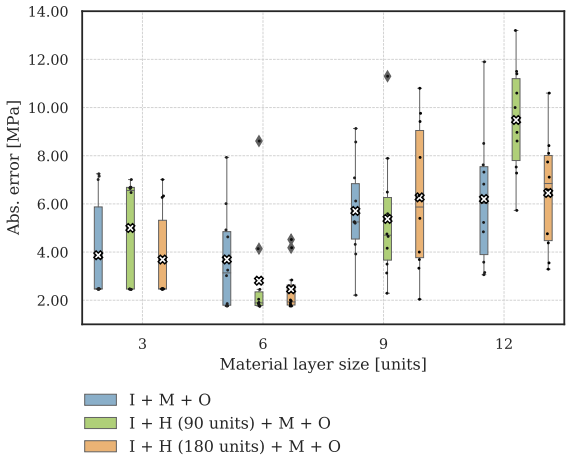
Figure 2.8: Geometry and mesh discretization of microscopic model adopted in this chapter.

In the following section, a grid-search strategy is employed to choose the best architecture for the networks. The goal is to find the optimum architecture before heading to the testing sections. For that, different architectures and weight initializations are considered to mitigate the effect of randomness.

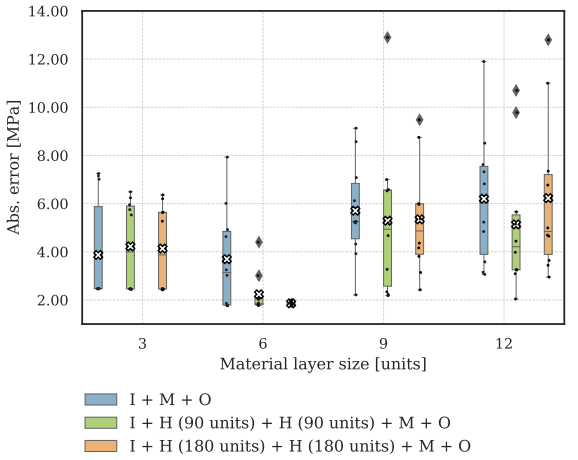
2.6.1. MODEL SELECTION

For the PRNN, three types of architectures are considered. First, networks with input, material and output layers are considered. Then, models with a hidden layer activated with the tanh function before the material layer are investigated. Finally, networks with two hidden layers in the encoder are considered. Note that this is not an exhaustive search as many other architectures are possible and suitable for both the encoder and decoder. In this study, the size of the hidden layers in the encoder are pre-defined (either 90 or 180 units) and the size of the material layer is variable.

Another important remark is that, in this case, only the elastoplastic model is used in the network so that the size of the material layer is the only variable in the model selection. Recall that the network still can make a subgroup to behave elastically by passing small strains to the material model. The training and the validation sets, $\mathfrak{D}_{\text{PRNN}}$ and $\mathcal{V}_{\text{PRNN}}$, consist of 18 Type I curves and 54 Type II curves, respectively. Fig. 2.9 shows the boxplots with the average validation error of each run alongside the mean error value over different architectures.



(a) Effect of including one hidden layer in the encoder and material layer of variable size



(b) Effect of including two hidden layers in the encoder and material layer of variable size

Figure 2.9: Absolute error of PRNNs trained on $\mathfrak{D}_{\text{PRNN}} = \{18 \text{ Type I curves}\}$ over validation set $\mathcal{V}_{\text{PRNN}} = \{54 \text{ Type II curves}\}$. Letters I, H, M and O refer to the input, hidden, material and output layers, respectively.

In all cases, the networks with 6 units (*i.e.* two fictitious material points) performed best. Furthermore, despite the larger variance in models without hidden layers other than the material layer itself, the best networks of this type are as accurate as the ones with one or two hidden layers with significantly fewer parameters (see lower bounds

in the boxplots). Another concern in using overly complex models in this particular case is the difficulty to assess the accuracy of the tangent stiffness matrix for use in FE^2applications without probing the model in numerical applications or explicitly including it in the formulation (*e.g.* through the loss function). In that light, we opt for the most parsimonious architecture - the one with a single material layer between the input and output layers - for the rest of the chapter as it led to more robust stresses and tangent stiffness matrix predictions in both single-scale tests and multiscale applications.

For simplicity, the architecture of the Bayesian RNN is composed of an input layer, a single GRU cell and an output (dense) layer. Again, weights and biases are randomized in each initialization, the training set ($\mathfrak{D}_{\text{RNN}}$) consists of the fixed set of 18 Type I curves, 90 Type II curves randomly chosen from a pool of 1800 curves, and 90 Type III curves also randomly chosen from a pool of 1800 curves, amounting to 198 loading paths. That way, all types of curves used for training in the following sections are covered. Fig. 2.10 shows the boxplot with the average training error of each run alongside the mean error value of all runs represented by the x marker. In this study, it is found that the GRU with 128 hidden variables performs best. In this case, no validation set is needed to determine the best dropout rate as the type of RNN used in this investigation infers it from the training data by default (see Section 2.3).

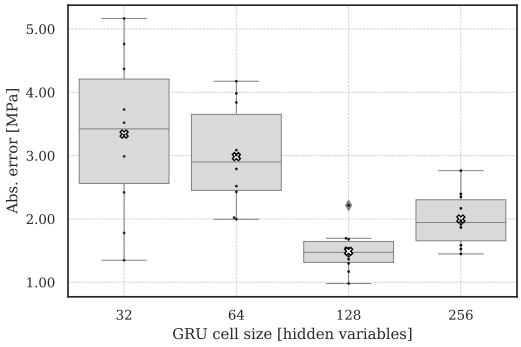


Figure 2.10: Training error for RNNs trained on $\mathfrak{D}_{\text{RNN}}$ = {18 Type I curves, 90 Type II and 90 Type III curves}.

2.6.2. PREDICTING UNLOADING/RELOADING FROM MONOTONIC DATA

In this section, the training process of the RNNs on Type II curves (*i.e.* without unloading) is reported. The first test set consists of 100 Type II curves. Fig. 2.11 shows the average error of the RNNs over the test set compared to the best (blue triangle), worst (upside-down blue triangle), and average error (blue circle) found by the PRNNs. Note that the secondary axis starts with 18 curves, this is because the known directions used for training the PRNNs are also a fixed set in the training of the RNNs. The training of the RNNs is stopped with 288 curves. At that stage, a similar level of accuracy between the PRNNs and the RNNs is obtained (although with a training set 16 times larger) and the addition of new curves only yields a marginal gain in accuracy.

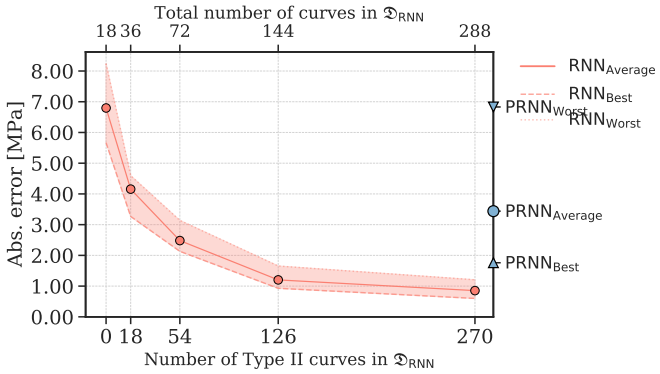


Figure 2.11: Absolute error over monotonic test set $\mathcal{T}_{II}$ = {100 Type II curves} for RNNs trained on Types I and II and PRNNs on 18 Type I curves.

Next, a new test set with 100 Type III curves is evaluated by the same networks trained on the 288 monotonic curves. This time, the RNN fails to capture unloading and the addition of more monotonic data is ineffective, as shown in Fig. 2.12a. This outcome is not new to the literature and it is not surprising that RNNs need to see unloading behavior during training in order to be able to describe it. However, in contrast to the RNNs, the PRNNs provide the same level of accuracy for the test sets with and without unloading, even when not exposed to unloading data during training. In Fig. 2.12b a single representative case from test set $\mathcal{T}_{III}$ is plotted using the best RNN and PRNN. Both networks show good agreement with the reference solution until unloading starts (a feature not covered during training), but only the PRNN is capable of capturing the elastic unloading/reloading.

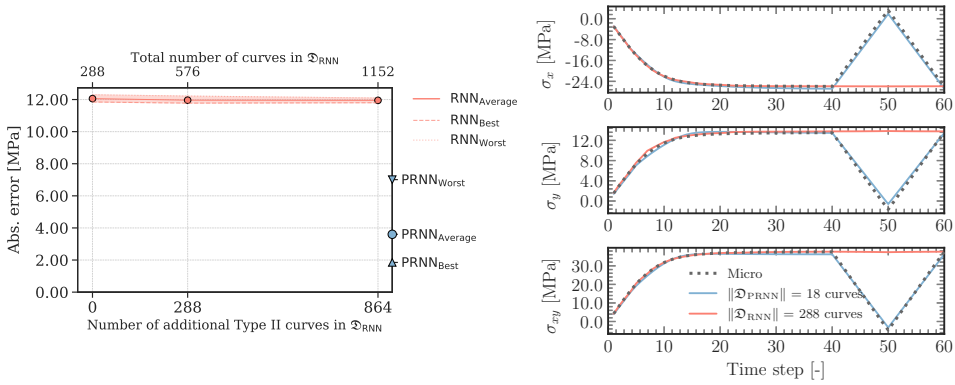
2.6.3. PREDICTING UNLOADING/RELOADING BEHAVIOR FROM NON-MONOTONIC DATA

Following the conclusions of Section 2.6.2, the training set of the RNN is expanded to include curves with the same unloading behavior as the one observed in the test set $\mathcal{T}_{III}$. The 288 monotonic curves of Types I and II from the previous section are combined with an increasing number of non-monotonic curves of Type III. This time, with the right features included in the training set, Fig. 2.13 shows a monotonic decrease of the average error for the RNN on $\mathcal{T}_{III}$ curves. However, the performance of the RNN only meets the one obtained by the PRNN with around 32 times more data.

2.6.4. PREDICTING UNSEEN PATTERNS FROM NON-MONOTONIC DATA

In this section, three additional test sets are considered for the RNN trained on Types I, II and III and the PRNN trained on Type I only. The goal is to test the ability of the networks to predict the macroscopic stress with patterns different from those seen during training.

First, we consider a test set with 100 unseen curves of Type IVa, which consists of proportional curves in random directions with a different predefined unloading/reloading



(a) Absolute error over test set $\mathcal{T}_{III}$ = {100 Type III curves} for RNNs trained on Types I and II and PRNNs on Type I only

(b) Stress-time view of representative case from test set $\mathcal{T}_{III}$ using the best RNN and PRNN

Figure 2.12: Absolute error over non-monotonic test set $\mathcal{T}_{III}$ for RNNs and PRNNs trained only on 18 Type I curves and representative case.

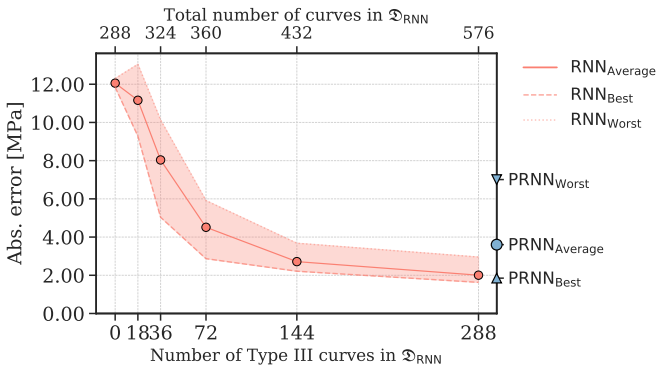
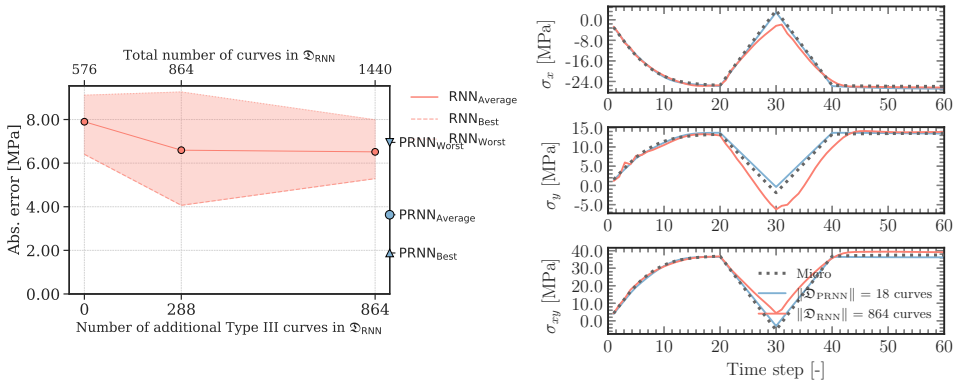


Figure 2.13: Absolute error over non-monotonic test set $\mathcal{T}_{III}$ = {100 Type III curves} for RNNs trained on Types I, II and II and PRNNs on 18 Type I curves.

behavior than that of Type III. The average error for that set is shown in Fig. 2.14a. Here, the 576 curves from the previous section are no longer enough to provide good accuracy when predicting a different unloading/reloading. By adding more Type III curves to the training set of the RNN, the average error decreases from 6.4 MPa to around 4.0 MPa but no significant gain in the accuracy is observed when the total number of curves is larger than 864 curves. Based on that, a representative case from test set $\mathcal{T}_{IVa}$ is shown in Fig. 2.14b. Despite the relative low error from both networks, note that the RNN looses performance once unloading starts while the PRNN continues to show good agreement throughout the entire loading path.

For the next scenario, a test set with 100 Type IVb curves are considered. These curves



(a) Absolute error over test set $\mathcal{T}_{IVa}$ = {100 Type IVa curves} for RNNs trained on Types I, II and III and PRNN on Type I only

(b) Stress-time view of representative case from test set $\mathcal{T}_{IVa}$ using the best RNN and PRNN

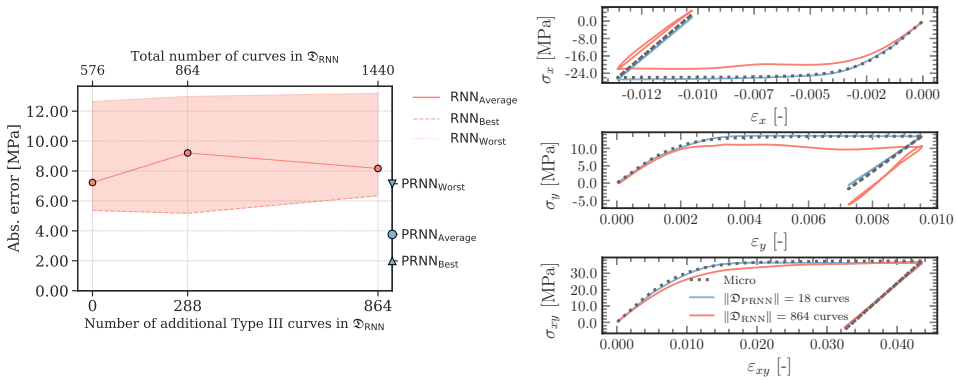
Figure 2.14: Absolute error over non-monotonic test set $\mathcal{T}_{IVa}$ = {100 Type IVa curves} for RNNs trained on Types I, II and II and PRNNs on 18 Type I curves and representative case.

have the same unloading/reloading behavior as Type III, but with a 10× smaller time step. Fig. 2.15a illustrates the average error of 10 networks over that test set and again, it is clear that the addition of new curves with patterns different from the exact one being tested is not beneficial to the RNN. Again, the PRNN provides good accuracy. Essentially, the PRNN is only as sensitive to step size as the material models embedded in it. Fig. 2.15b illustrates the networks' predictions for a curve in the test set $\mathcal{T}_{IVb}$.

As a final test, a set of 100 Type V curves, which corresponds to non-proportional and non-monotonic paths, is considered. This type of curve combines the two previous features: different step sizes and different unloading/reloading locations. Fig. 2.16a shows the average error for additional non-monotonic curves in the training of the RNNs. It is clear that the RNN completely fails to capture non-proportional paths (lowest error around 32 MPa) and that the addition of more data with features different than those being tested is a waste of resources. Although in different levels, a similar trend of loss of accuracy is also observed in the PRNNs, where the best, average and worse performances result in errors around 8.9 MPa, 17.0 MPa and 11.2 MPa respectively. Fig. 2.16b illustrates the different order of error between the PRNN and the RNN on a representative case from test set $\mathcal{T}_{V}$.

2.6.5. TRAINING ON NON-MONOTONIC AND NON-PROPORTIONAL LOADING

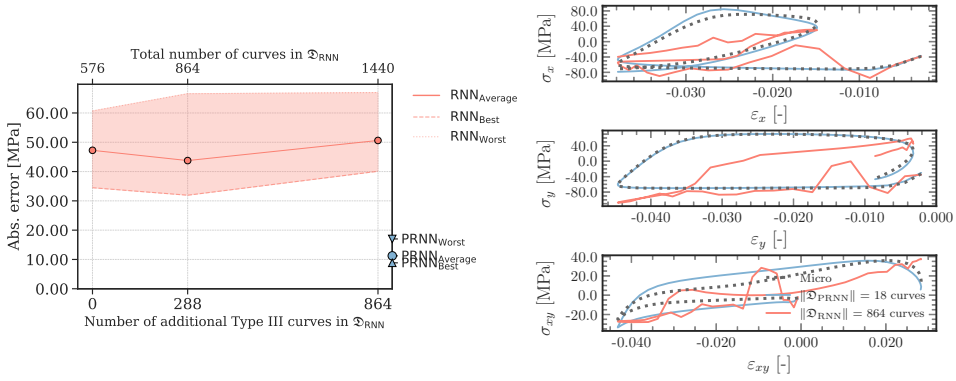
In this section, both networks are trained on the most generic set of curves, *i.e.* random non-monotonic and non-proportional curves of Type V. In addition to that, we trained the PRNNs on the known and proportional loading cases for comparison purposes. In



(a) Absolute error over test set $\mathcal{T}_{\text{IVb}}$ = {100 Type IVb curves} for RNNs trained on Types I, II and III and PRNNs on Type I only

(b) Representative $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ curve from test set $\mathcal{T}_{\text{IVb}}$ using the best RNN and PRNN

Figure 2.15: Absolute error over different step size test set $\mathcal{T}_{\text{IVb}}$ = {100 Type IVb curves} for RNNs trained on Types I, II and III and PRNNs on 18 Type I curves and representative case.



(a) Absolute error over test set $\mathcal{T}_{\text{V}}$ = {100 Type V curves} for RNN trained on Types I, II and III and PRNN on Type I only

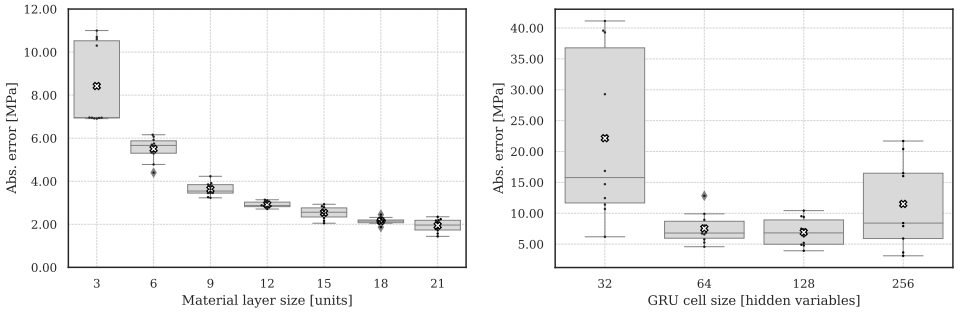
(b) Representative $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ curve from test set $\mathcal{T}_{\text{V}}$ using the best RNN and PRNN

Figure 2.16: Absolute error over non-proportional and non-monotonic test set $\mathcal{T}_{\text{V}}$ = {100 Type V curves} for RNNs trained on Types I, II and III and PRNNs on 18 Type I curves and representative case.

that case, the size of the material layer is kept at 6 units and three training dataset sizes are considered. First, only the pure uniaxial cases are included, which yields 6 loading cases. Then, the 4 biaxial cases are added to the previous training dataset, resulting in

10 loading cases. And finally, we add the 8 cases with biaxial and shear loading, which amounts to the 18 fundamental paths shown in Fig. 2.6b.

When training both networks on non-proportional paths a new model selection procedure was carried out to determine the optimum size of the material layer and the GRU cell, respectively. In this preliminary study, 10 different weights initialization are considered again. For training the RNNs and the PRNNs, 2304 and 198 Type V curves are used, respectively. Figs. 2.17a and 2.17b show the boxplot with the average error of each run alongside the mean error value. In this case, the networks with 18 units (which corresponds to six fictitious material points) performed better. For the Bayesian RNN, the GRUs with 128 hidden variables continue to provide the best performance. Therefore, this architecture is the one used in the comparison presented below.



(a) Validation error for PRNNs trained on $\mathfrak{D}_{\text{PRNN}}$ = {198 Type V curves} and $\mathcal{V}_{\text{PRNN}}$ = {54 Type V curves}

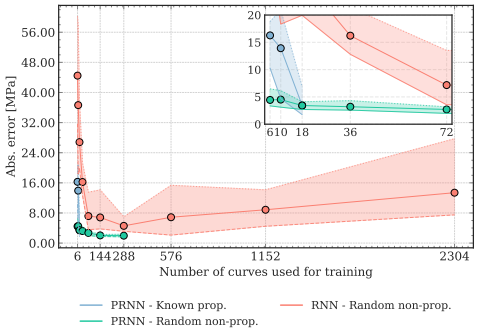
(b) Training error for RNNs trained on $\mathfrak{D}_{\text{RNN}}$ = {2304 Type V curves}

Figure 2.17: Model selection of PRNN and RNN for non-monotonic and non-proportional loading.

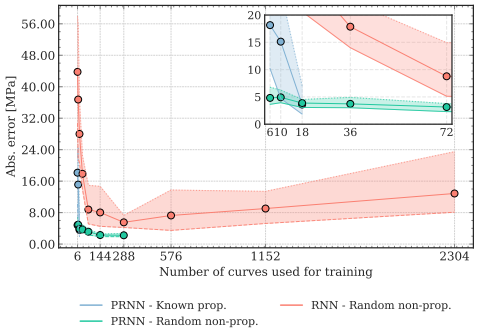
This time, all test sets discussed in Sections 2.6.2 to 2.6.4 are used again to assess the accuracy of the networks with the new sampling strategy. Figs. 2.18a-2.18e show the best, worst and average error for the cases studied so far in order. Based on this study, a few important insights are worth mentioning: after a certain point (around 576 curves), the RNNs reach an optimum level of accuracy and the addition of new curves no longer boosts predictions for proportional loading cases ($\mathcal{T}_{\text{II}}$, $\mathcal{T}_{\text{III}}$, $\mathcal{T}_{\text{IVa}}$ and $\mathcal{T}_{\text{IVb}}$). This is in line with the behavior observed in the previous sections, in which the RNNs would only perform well when trained with the same features as in the test set. And more importantly, changing the sampling strategy also showed to have limited effect on improving their performance. Granted, increasing even further the number of curves used for training as well as the complexity of the RNN might help in that task. However, the point stands that with the PRNN, this is not necessary. Note that for the same training set sizes, the PRNN with either the known or the random curves performs better than using the RNNs.

Finally, when choosing between known and random curves for training the PRNN, the latter shows comparable errors with the first when predicting proportional loading, but is significantly more accurate (see detail in Fig. 2.18e) for non-proportional loading. For that reason, the PRNN trained on Type V is chosen to illustrate the network's capacity

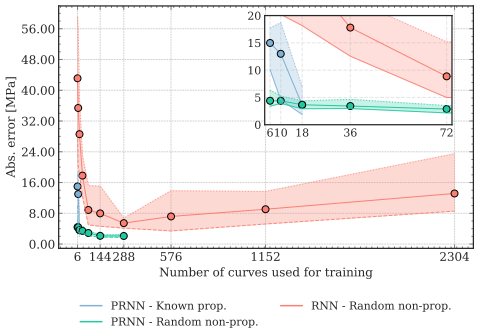
in the following FE^2 examples. On average, the accuracy of the PRNN reaches a plateau around 36 curves. From that point on, the benefit of adding new data is limited.



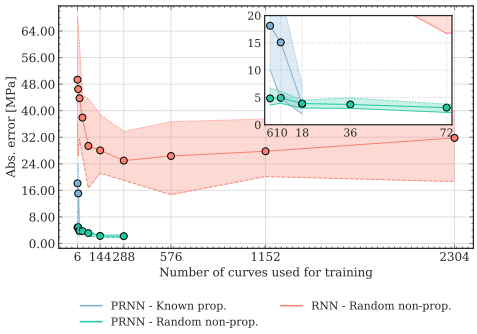
(a) Test set $\mathcal{T}_{\text{II}}$ = {100 Type II curves}



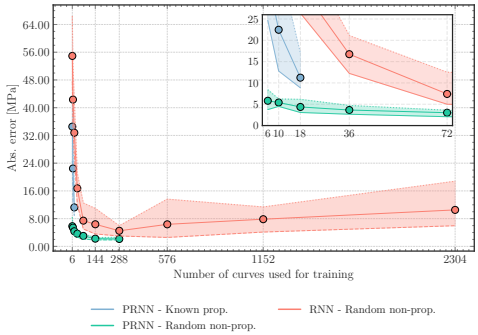
(b) Test set $\mathcal{T}_{\text{III}}$ = {100 Type III curves}



(c) Test set $\mathcal{T}_{\text{IV}_\text{a}}$ = {100 Type IVa curves}



(d) Test set $\mathcal{T}_{\text{IV}_\text{b}}$ = {100 Type IVb curves}



(e) Test set $\mathcal{T}_{\text{V}}$ = {100 Type V curves}

Figure 2.18: Absolute error of networks trained on different sampling strategies and different test sets.

2.7. FE^2 APPLICATIONS

In this section, the PRNN trained with 36 non-proportional and non-monotonic Type V curves and lowest test error for test set $\mathcal{T}_V$ in Section 2.6.5 is employed as the constitutive model in two numerical examples. Results obtained with the PRNN as constitutive model are compared against results obtained with full FE^2 with the same micromodel that the network was trained to be a surrogate for. Both types of analysis are performed with an in-house Finite Element code using the open-source Jem/Jive C++ numerical analysis library [14].

At the macroscale, an arc-length method with an adaptive-stepping scheme [15] is adopted to tackle potential convergence issues. This way, if a loading step does not converge with the given (full) step size, a reduction factor is applied to it until the loading step converges or until a maximum number of reductions in the initial step has been reached, terminating the analysis.

All simulations, including the PRNN training, were executed on a single core of a Xeon E5-2630V4 processor on a cluster node with 128 GB RAM running CentOS 7.

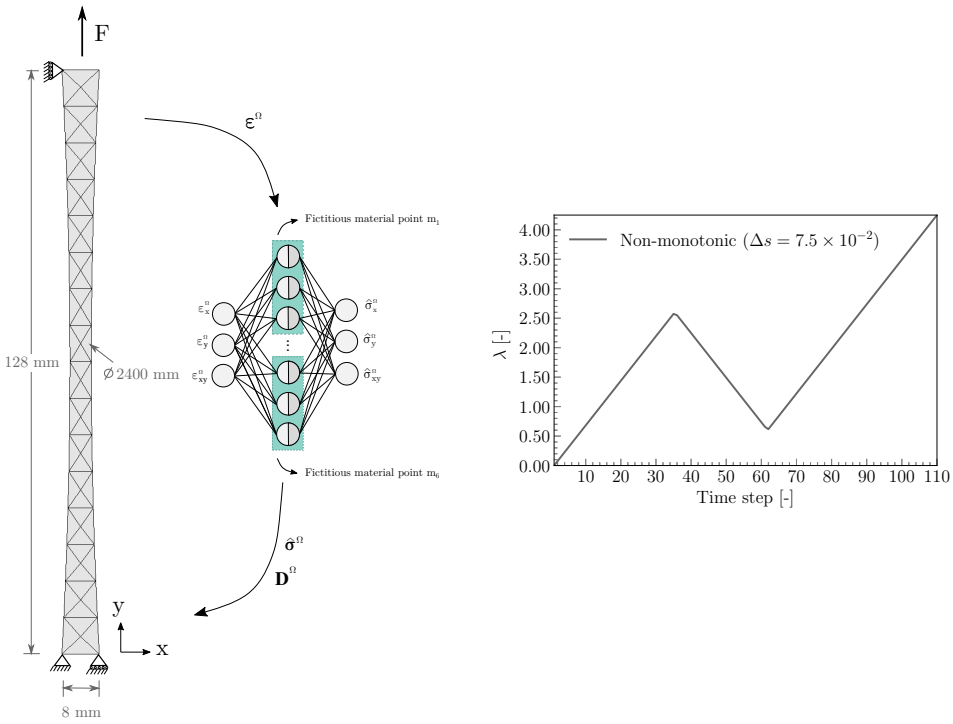
2.7.1. TAPERED BAR

The first example consists in a tapered composite specimen with length of 128 mm and height of 8 mm loaded in transverse tension. In this setting, the 36-fiber RVE model used to train the networks in the previous sections is embedded at each integration point of the macroscale. The geometry and the boundary conditions are shown in Fig. 2.19a. The FE^2 problem is solved for 110 load steps with unloading according to the function shown in Fig. 2.19b. At this point, the macroscopic response is already in the plastic regime.

The strain field at the end of the analysis is shown in Fig. 2.20a along with the location of one macroscopic integration point. This point is used to illustrate the state of the PRNN throughout the time steps. Recall that the network used in this section consists of 18 units (*i.e.* 6 fictitious material points). Thus, each row in Fig. 2.20b corresponds to a fictitious material point, each with its own stress path and internal variables (even though only one of them is plotted).

In this case, each component of the macroscopic response (*i.e.* homogenized stresses) is simply the linear combination of the local stresses of the 6 material models. It can be observed that the two macroscopic responses are in excellent agreement, with minor deviations in stress components with low magnitude. This is only visualized for a single point, but it is emphasized that in this multiscale problem, agreement in the evolution of the stresses in a single integration point indicates that the whole problem is solved accurately. Moreover, the equivalent plastic deformation of each material point is plotted in the last column. Note that despite the plastic response of the RVE after time step 20, three of the fictitious material points of the network (m_1, m_2 and m_6) remain in the elastic regime.

The accuracy of the PRNN is further assessed by inspecting the load-displacement curve at the top edge of the bar. Fig. 2.21a shows the load-displacement curve using the full-order solution and the network's response for different macroscopic mesh discretizations. For the refinement studied so far ($\Delta^{\Omega}_{\text{elem}} = 8\,\text{mm}$), it is clear that the proposed network can capture accurately the entire nonlinear response, albeit with minor



(a) Geometry and boundary conditions

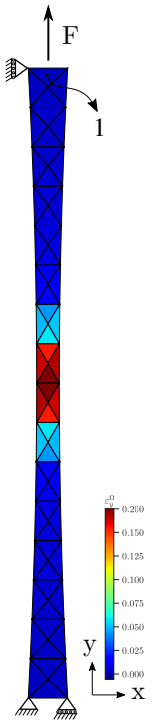
(b) Loading function

Figure 2.19: Tapered bar FE^2 example with (a) geometry and boundary conditions and (b) loading function.

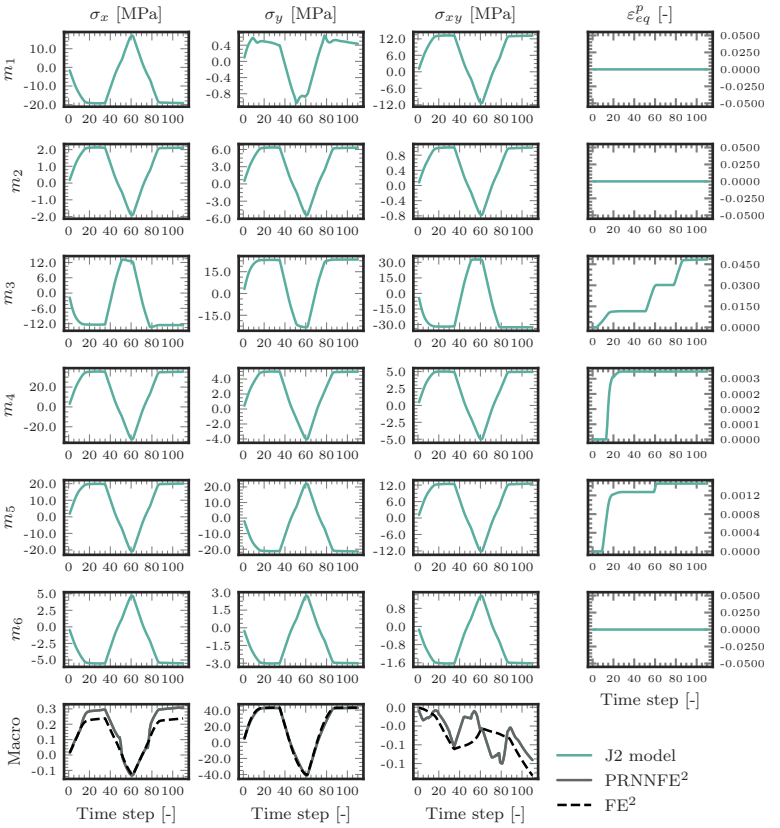
deviations when the tapered bar changes from tension to compression and then back again to tension.

Next, the global accuracy of the method is verified. Since the surrogate model is not as accurate as the full-order model, a different equilibrium solution at a certain time step affects the equilibrium in the following loading steps, leading to accumulated error and diverging $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ paths. In this case, since no reduction in the step size was observed at any moment, the simple average error between the PRNN prediction and the full-order solution is calculated for each time step averaged over all integrations points at the macroscale, as illustrated in Fig. 2.21b. For most of the simulation, the average absolute error in the predictions remains below 1 MPa with two peaks around 4 MPa and 6 MPa when the loading is reversed, following the trends observed in Fig. 2.21a. For reference, the lowest error of the network for test set $\mathcal{T}_{\mathrm{V}}$ is plotted.

For the purpose of assessing the efficiency of the network in accelerating the FE^2 simulations, three different levels of mesh refinement of the tapered bar are taken into account, being the coarses the one shown in Fig. 2.19. Table 2.1 summarizes the wall-clock time spent in the analysis of the different discretizations, as well as the speed-up



(a) Strain field at the end of the analysis

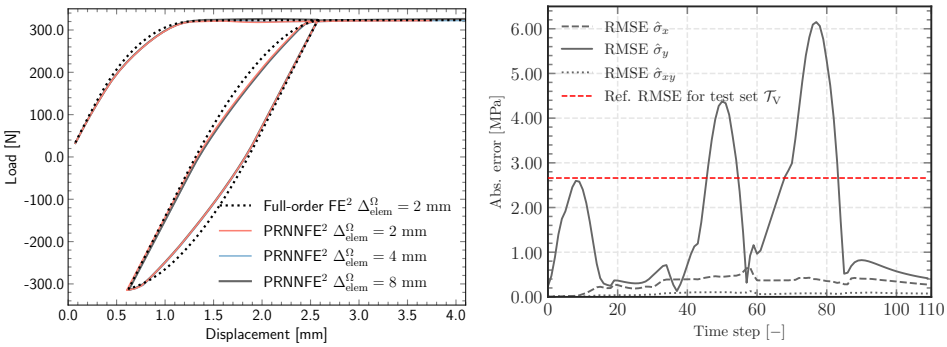


(b) Latent space view for integration point 1

Figure 2.20: Strain field using the PRNN on the left and detailed view of latent space of PRNN for a single macroscopic integration point on the right.

in comparison to the full-order solution. For the mesh used to illustrate this section ($\Delta^{\Omega}_{\text{elem}} = 8$ mm), replacing the solution of the BVP of the micromodel with the network led to a speed-up over 26 000 with the accuracy reported in Fig. 2.21b. Considering the offline costs, the training time is still lower than that of using the full-order solution with 134 macroscopic elements, which is a very modest number of elements for a multiscale problem.

Since the network is trained to replace the solution of the microscopic model, no additional training is required for the analysis of more complex cases where the macroscale problems require more elements and time steps. Hence, in general, higher speed-ups should be achieved with denser meshes. However, in this particular problem, this is not always the case. An increase from the coarsest to the intermediary mesh is observed, but no gain is achieved when refining even further. In that case, the reduction in per-



(a) Load-displacement curve obtained with the full-order FE^2 approach and with PRNN for different macroscopic mesh discretizations

(b) Average RMSE at each time step of analysis with $\Delta^{\Omega}_{\text{elem}} = 8$ mm

Figure 2.21: Tapered bar FE^2 example with (a) load-displacement curve with the full-order solution and best PRNN and (b) average error of PRNN's predictions at each time step of the analysis with $\Delta^{\Omega}_{\text{elem}} = 8$ mm.

Table 2.1: Computational cost for different mesh discretizations and efficiency of network in FE^2 approach.

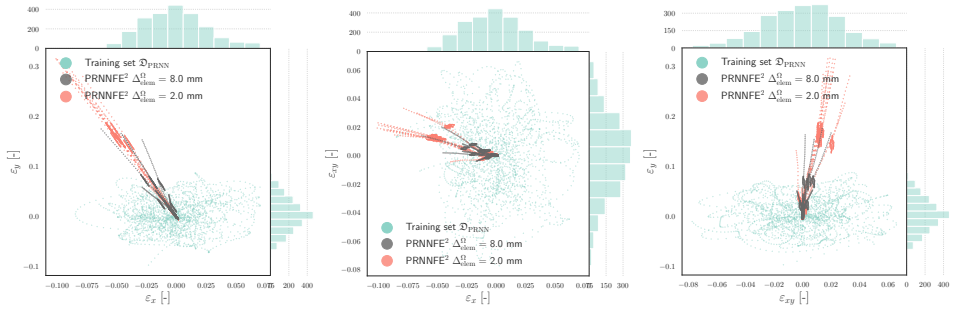
	Macroscale element size ($\Delta^{\Omega}_{\text{elem}}$) [mm]	8	4	2
	Number of elements at the macroscale	64	134	454
Online	FE^2 wall-clock time [s]	21574	47644	178800
	PRNNFE^2 wall-clock time [s]	0.81	1.55	8.41
	Speed-up[a] [-]	26560	30746	21526
Offline	Av. wall-clock time per curve (dataset gen.) [s]	265[b]	N/A	N/A
	Av. training time (excl. dataset gen.) [s]	38045[b]	N/A	N/A

[a]Evaluated as FE^2 wall-clock time/PRNNFE^2 wall-clock time and averaged over 5 runs

[b]One-off cost regardless of macroscopic mesh discretization

formance due to the higher number of iterations caused by the necessity of adaptively reducing the step size in order to ensure convergence. In contrast, the full-order solution was more numerically stable for this mesh density and the adaptive-stepping scheme was not triggered.

As the mesh is refined and strain localization takes place (see the red region in Fig. 2.20a), even higher strain levels are achieved, pushing the network to make predictions in unexplored regions during training, as illustrated in Fig. 2.22. Note that the network is already making far-reaching predictions in the coarsest discretization, although in a less extensive way. In the mesh with $\Delta^{\Omega}_{\text{elem}} = 8$ mm, the maximum strain does not exceed 0.2, while $\Delta^{\Omega}_{\text{elem}} = 2$ mm leads to strains higher than 0.3. In spite of these complicating as-



(a) Macroscopic strains in x and y (b) Macroscopic strains in x and xy (c) Macroscopic strains in xy and y

Figure 2.22: Joint distribution of strains from the training set of the best PRNN and strain distribution obtained by PRNNFE^2 for the tapered bar problem with different macroscopic mesh discretizations.

pects, it is worth mentioning this is still a significant speed-up. Moreover, a far less severe effect on the global accuracy is observed, as illustrated by the almost overlapping load-displacement curves in Fig. 2.21a. In that sense, the adaptive-stepping scheme plays an important role to help overcome convergence issues.

2.7.2. PLATE WITH MULTIPLE HOLES

As a final example, a composite plate with multiple cutouts with geometry and boundary and loading conditions as illustrated in Fig. 2.23 is studied. Again, an FE^2 approach is employed to solve the problem for the same microscopic model with which all the networks in Section 2.6 were trained for. This time, no unloading is imposed and 150 load steps with $\Delta s = 5.0 \times 10^{-3}$ are considered. The load-displacement curve at the right edge of the plate is plotted in Fig. 2.24a using both the full-order solution and the network. Again, good agreement is observed between the macroscopic responses. The slight inaccuracy between those are quantified in Fig. 2.24b, in which the average absolute error of the component with the highest magnitudes (σ_x) is around 1 MPa for almost the entire simulation.

The displacement field at the end of the analysis is shown in Fig. 2.25a, where the location of five macroscopic integration points are marked for further inspection. The stress paths for each of these points are illustrated in Fig. 2.25b, where the full-order solution and the network prediction are plotted in black and gray, respectively. Note how the stress paths are non-proportional even for the relatively simple loading condition observed in the macroscale. The integration point on the edge of one of the cutouts, namely point 4, is also the one with the highest stress magnitude and the closest to a uniaxial state in the x direction while the other points experience multiaxial loading more strongly.

In terms of efficiency, the solution using the network is around 26 705 times faster than the full-order solution, which took approximately 258 780 s (around 72 h). The order of

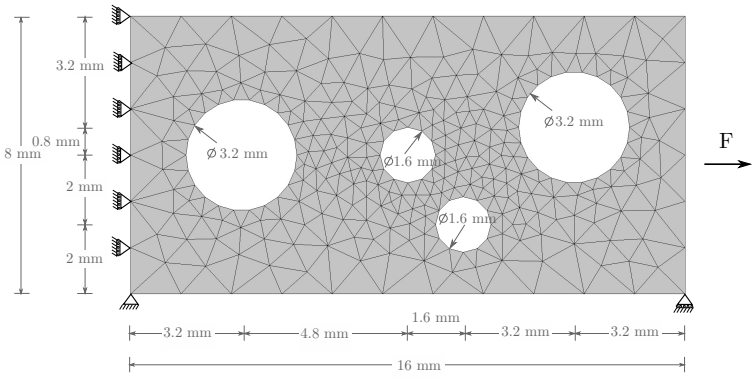
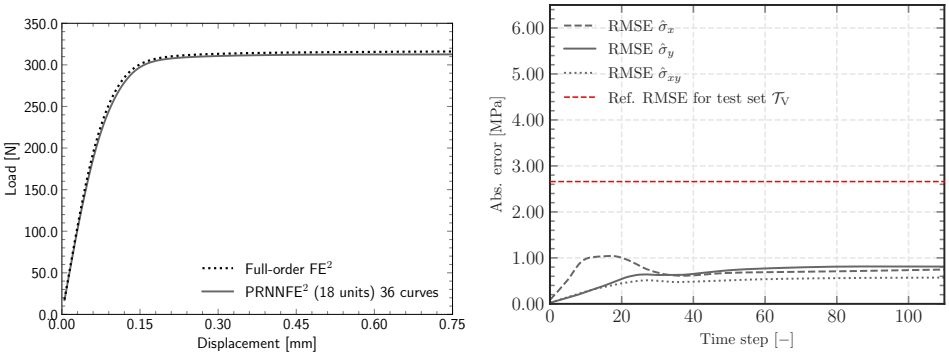


Figure 2.23: Plate with cutouts: geometry and boundary and loading conditions.



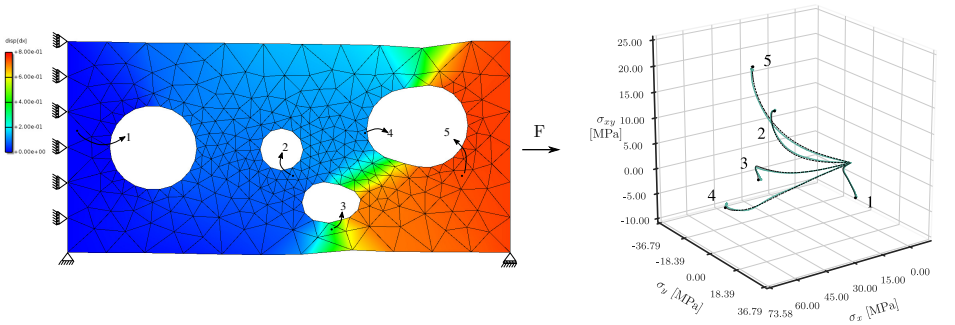
(a) Load-displacement curve

(b) Average RMSE

Figure 2.24: Plate with cutouts: (a) load-displacement curve and (b) average error of PRNN's predictions at each time step of the analysis.

magnitude in the speed-up is similar to that obtained in the tapered bar problem. Although no additional offline costs are incurred because the network has been trained before for the same microscopic model and it does not depend on the macroscopic problem at hand, it is worth stressing that the runtime of the full-order solution exceeds the sum of the online and offline costs of the PRNN.

Finally, this example shows that the network can capture multiaxial stress states and non-proportional loading as obtained in FE^2 simulations accurately. No convergence issues were encountered in the PRNNFE^2 simulation which points to the smoothness of the predictions that is not always guaranteed with surrogate models (see *e.g.* RNN curves in Fig. 2.16b).



(a) Displacement field at the end of the analysis with PRNN and selection of integration points for inspection

(b) Stress-time view of macroscopic integration points. Full-order solution in black and network's prediction in green

Figure 2.25: Plate with cutouts: (a) displacement field at the end of the analysis and (b) selected integration points shown in stress-time view.

2.8. EXTENDED EXPERIMENTS

In this section, two additional studies are carried out to demonstrate the flexibility of the proposed approach to handle various types of material models with different levels of complexity, as well as to help identify potential pitfalls when choosing the architecture and the design of experiments. In both scenarios, the same RVE geometry presented in Sections 2.6 and 2.7 is used.

2.8.1. TWO ELASTOPLASTIC PHASES WITH DIFFERENT MATERIAL PROPERTIES

In this study, plane stress conditions are kept, but both matrix and fibers are now described by the elastoplastic model with the von Mises yield criterion and isotropic hardening with the equivalent plastic strain given by Eq. (2.36) and material properties as described in Table 2.2.

Table 2.2: Material properties of RVE modeled by two elastoplastic models.

	E [MPa]	ν [-]	σ_y [MPa]
Constitutive model of matrix $\mathcal{D}_1^\omega$	3130	0.37	$64.8 - 33.6\exp^{-\varepsilon_{eq}^p/0.003407}$
Constitutive model of fibers $\mathcal{D}_2^\omega$	2130	0.25	$77.76 - 33.6\exp^{-\varepsilon_{eq}^p/0.003407}$

For training the networks, we follow the steps discussed in Section 2.6, in which the initial training set comprises 18 Type I curves and the validation set consists of 54 Type II curves. The architecture of the networks consists of an input layer, a material layer with all fictitious material points evaluated by the constitutive model $\mathcal{D}_1^\omega$ and an output layer.

Again, 10 initializations and four different layer sizes are considered. Fig. 2.26 shows the boxplots with the average validation error of each run. We select the architecture with three units (or one fictitious material point) and assess its performance on test sets $\mathcal{T}_{\text{III}}$ and $\mathcal{T}_{\text{V}}$ with 100 curves of Types III and V each respectively. The lowest average error for both test sets are around 0.63 MPa and 3.13 MPa, respectively. Fig. 2.27 illustrates the accuracy of the network on a representative case from each of the test sets. Note that despite the increase in the average error over non-monotonic and non-proportional curves, the network is still capable of capturing relatively well its global trend.

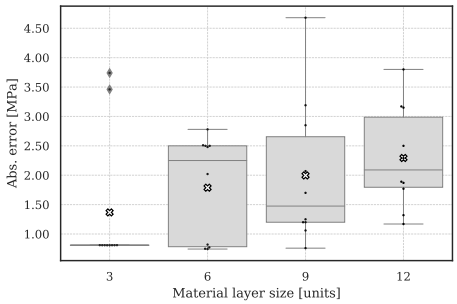
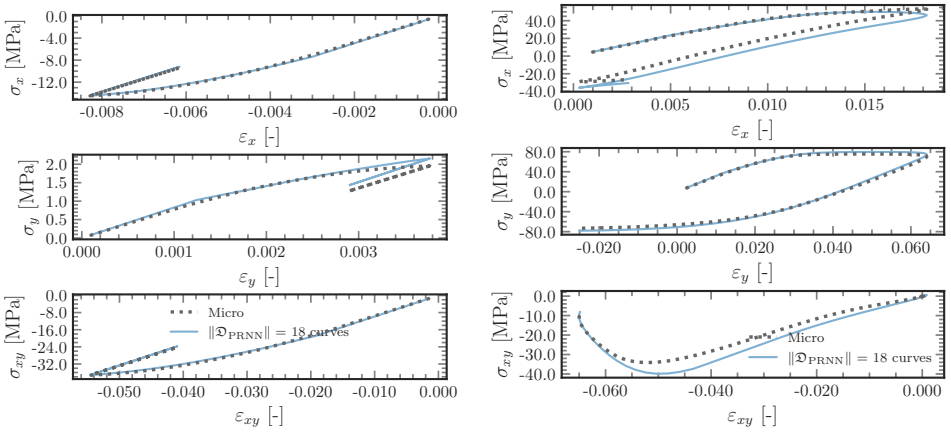


Figure 2.26: Absolute error for PRNNs trained on $\mathfrak{D}_{\text{PRNN}}$ = {18 Type I curves} over validation set $\mathcal{V}_{\text{PRNN}}$ = {54 Type II curves} for RVE with two elastoplastic phases.



(a) Representative $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ curve from non-monotonic test set $\mathcal{T}_{\text{III}}$

(b) Representative $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ curve from non-proportional and non-monotonic test set $\mathcal{T}_{\text{V}}$

Figure 2.27: Performance of the best PRNNs trained on 18 Type I curves over different test sets for RVE with two elastoplastic phases.

2.8.2. ELASTOPLASTIC AND NONLINEAR ELASTIC PHASES WITH THE SAME MATERIAL PROPERTIES

In this study, a more complex elastoplastic model is considered: the material model proposed by Melro *et al.* [16]. For the sake of brevity, the details of the implementation are spared and the reader is referred to [16, 17] for further clarification. This model uses a pressure-dependent yield criterion:

$$f(\boldsymbol{\sigma}, \sigma_c, \sigma_t) = 6J_2 + 2I_1(\sigma_c - \sigma_t) - 2\sigma_c\sigma_t \tag{2.35}$$

where I_1 and J_2 are stress invariants and σ_c and σ_t are compressive and tensile yield stress, both defined as general hardening functions of the equivalent plastic strain ε^p_{eq} with increment given by

$$\Delta\varepsilon^p_{eq} = \sqrt{\frac{1}{1+2\nu_p^2}\,\Delta\boldsymbol{\varepsilon}^p : \Delta\boldsymbol{\varepsilon}^p} \tag{2.36}$$

where ν_p is the plastic Poisson's ratio, related to the non-associative flow rule. Plane strain conditions are assumed.

Based on this constitutive model, we concoct a modified version for which the updated internal variables calculated within the return mapping scheme are not stored at the end of every time step. This way, although dictated by a strictly identical hardening law and yielding criterion, no history-dependence is carried from one loading step to another, resulting in a material that behaves elastically in the sense that the loading and unloading follow the same path. This artificial material model allows us to illustrate unique scenarios that challenge the applicability of the proposed network.

For this study, the elastoplastic model by Melro *et al.* [16] $\mathcal{D}_1^\omega$ is used to describe the matrix and our modified nonlinear elastic version $\mathcal{D}_2^\omega$ to describe the fibers. The same material properties are adopted for both constitutive models, with E = 3130 MPa, ν = 0.37, ν_p = 0.32 and the two hardening laws in Eq. (2.35) given by

$$\begin{aligned} \sigma_t &= 64.8 - 33.6\exp(-\varepsilon^p_{eq}/0.003407) \\ \sigma_c &= 1.2\left(64.8 - 33.6\exp(-\varepsilon^p_{eq}/0.003407)\right) \end{aligned}. \tag{2.37}$$

Three different PRNN models are considered. First, the training and validation sets consist of 18 Type I curves and 54 Type II curves, respectively. The architecture consists of an input layer, a material layer containing two fictitious material points evaluated using the constitutive model $\mathcal{D}_1^\omega$ and an output layer. In the second model, the training set consists of 18 Type V curves and 54 Type V curves are used for validation. Recall that these curves are non-proportional and non-monotonic loading paths generated by GPs. The architecture in the first model is kept. Finally, in the third option, we train and validate with the same amount and type of data as in the second model, but the architecture is changed. This time, we evaluate one fictitious material point with $\mathcal{D}_1^\omega$ and the other with $\mathcal{D}_2^\omega$. Five different initializations are considered for each model. Fig. 2.28 shows the performance of the best PRNNs using the different strategies on a representative case from test set $\mathcal{T}_{III}$ which comprises 100 Type III curves.

The model with only $\mathcal{D}_1^\omega$ trained on Type I curves does very well in describing the monotonic behavior. Because both materials present in the micromodel have the same

monotonic response, a single material model in the network is sufficient for describing the monotonic behavior. However, upon unloading, the network is only able to reproduce the secant unloading behavior embedded in $\mathcal{D}_1^\omega$ and fails to capture the contribution from the nonlinear elastic unloading of $\mathcal{D}_2^\omega$. Furthermore, in this particular combination of constitutive models and under mononotonic loading, the response of the micromodel is essentially the same as that of a homogenenous material. This emphasizes the point that the success we have had so far in capturing unloading accurately after training only on monotonic data came from the fact that the material points in the network included representative assumptions on the unloading behavior.

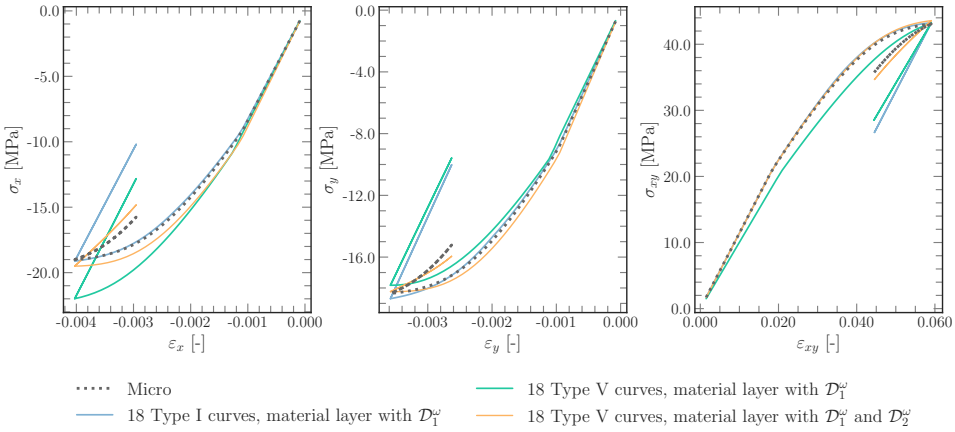


Figure 2.28: Representative $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ curves from test set $\mathcal{T}_{\text{III}}$ using the best PRNN trained on different training sets and constitutive models.

Training the PRNN with Type V curves does not improve the performance. On the contrary, the model loses accuracy even for the monotonic part. By minimizing the error of the entire loading path, which now includes multiple unloading cycles, the fitting ends up as a compromise between the loading and unloading behavior. Finally, it is observed that the network with both materials included in the material layer captures the mixed unloading behavior much better, although this does come at a cost concerning how well the monotonic part is described. Results can probably be improved by including more material points of one or both types and increasing the amount of training data.

2.9. CONCLUSION

In this chapter, a novel network with embedded physics-based constitutive models is proposed as surrogate model for the behavior of path-dependent materials in FE^2 simulations. The central idea is to address common problems in modeling path-dependent materials using black-box models (*e.g.* unique mapping between input and output and limited extrapolation abilities) by taking a step back and reintroducing physics into the network in a way that requires very little extra coding effort with respect to existing FE^2

frameworks. This is done by employing the same material models used for the microscopic level as part of one of the layers of the network.

To accommodate this non-standard neural layer the following changes with respect to standard neural network architectures are proposed. First, neurons are assembled in groups of the size of the number of strain/stress components of the problem. These are referred to as fictitious material points. Secondly, to take advantage of all the information coming from the physics-based material model, we store the updated internal variables used to fully describe the state of the fictitious material point in an auxiliary vector. With that, when new strain values are fed, the material point will start from the last stored internal variables. As a consequence, each subgroup follows a unique path without the need to increase the feature space with extra history variables.

The properties and assumptions made by the physics-based constitutive model are inherited by the network and play a major role in reducing the amount of data required to mirror physical and complex behaviors such as elastic unloading/reloading. Here, the decomposition of the strain in elastic and plastic parts is an assumption built in the material model used to describe the nonlinear microscopic material phase and is also observed in the network when the local stresses of the fictitious material points are evaluated. This simple but highly-flexible arrangement allows the network to capture arbitrary unloading behaviors with only monotonic data, a stark contrast with other popular models such as RNNs. The PRNN inherits from FE^2 the idea that complex behavior of heterogeneous materials can be accurately described by letting simpler constitutive models that represent the microscopic constituents interact. The difference is that the interaction between the constituents is not based on micromechanics directly but learned from data obtained from micromodel simulations.

Based on that, an extensive numerical comparison involving a state-of-the-art black-box model, namely a Bayesian Recurrent Neural Network (referred to in this chapter as RNN), was carried out in order to elucidate the abilities of the proposed network (referred to as PRNN). First, we trained both networks only on 18 monotonic curves with known directions and proportional loading in a similar fashion as done to calibrate classical mesomodels. Such strategy led to poor performance when trying to predict other random directions from the RNN, but good accuracy from our method (Fig. 2.11). Following that conclusion, the size of the RNN's training dataset was sequentially increased until the addition of more data did not result in significant gains in accuracy. At that stage, the PRNN performed with the same level of accuracy but with a factor of 16 times less data.

Next, both networks were used to predict non-monotonic loading. For that scenario, the PRNN showed the same level of accuracy as before with the same minimal training dataset (Fig. 2.12a). Such outstanding result is not observed in the RNNs, which again required a larger training set. This time, non-monotonic loading curves were added until the RNN's accuracy could no longer be significantly improved. As a result, a 32 times larger training dataset in comparison to the one used to train our network was necessary. Furthermore, while our network continues to perform well in all the scenarios tested so far, two other situations exposed the pitfalls of RNNs: (i) when trying to predict unloading in a different location than the one seen in training (Fig. 2.14a) and (ii) when the step size was modified (Fig. 2.15a). This is typically tackled by sampling different unloading

behaviors with different step sizes, leading to the choice of arbitrarily long sequences. However, we showed that this is a trivial scenario for the PRNN. The network is only as sensitive to step size as the material models it includes.

In the last test, both networks were used to predict non-proportional and non-monotonic paths and neither succeeded, although they failed at very different levels. While the lowest error of the RNN was around 32 MPa, the best PRNN led to an error around 9 MPa error (Fig. 2.16a). Based on that, a second approach to generate the dataset was considered. Random strain paths were generated from Gaussian Processes priors, which produces non-proportional and non-monotonic loading as opposed to the proportional loading previously considered for training. This time, the size of the training dataset and the type of loading was also a variable for the PRNN. It was found that training the PRNN on random non-proportional and non-monotonic curves yields higher accuracy than training with known, proportional, and monotonic curves for all loading scenarios (Fig. 2.18). Although training with known directions is appealing, having a network that provides lower errors and consistent performance with random directions is also interesting. Ultimately, the PRNN consistently outperformed the RNN with 64 times less data.

After ensuring the PRNN capacity in several challenging scenarios for black-box models, one of the networks trained on non-proportional and non-monotonic curves was chosen to surrogate the microscopic model in a set of two FE^2 examples. The first example concerned a tapered bar in transverse tension and was used to illustrate how the different material models in the PRNN behave for a single macroscopic integration point (Fig. 2.19). For different discretizations, speed-ups between 21 000 and 31 000 were obtained for the online phase (Table 2.1). Such substantial efficiency gain is explained from the dramatically reduced number of material model calls and the bypassing of solving the nonlinear microscopic system of equations for macroscopic stress evaluation for a single load step of each macroscopic integration point.

In the last example, a similar order of magnitude of speed-up was observed and the accuracy of the PRNN was illustrated by comparing the $\boldsymbol{\varepsilon}$-$\boldsymbol{\sigma}$ paths of different macroscopic integration points of a plate with multiple cutouts subjected to tension (Fig. 2.25b). For the analyzed cases, the time needed for a single FE^2 analysis exceeded the total offline and online time for the PRNN analysis, even though the selected problems had a very modest number of macroscopic elements. Moreover, performing subsequent macroscale analysis with the same material would not require any additional offline training and would therefore leverage the complete speed-up of four orders of magnitude.

Finally, two additional studies were carried out to further demonstrate the flexibility of the proposed approach in handling various types of material models with different levels of complexity. In the first study, an elastoplastic model with different material properties was used to describe the two phases of a micromodel, while in the second a more complex elastoplastic model and a nonlinear elastic phase were considered to describe the constituents. For the first part, results followed the trend in which an accurate model can be obtained by training with monotonic data only and a single model in the network. In the second study, results illustrate potential pitfalls in not including both sources of nonlinearity and how to address the issue to obtain an accurate and robust surrogate model.

APPENDIX. ENCODER WITH EXPLICIT PATH-DEPENDENCY

In this section, a different architecture for the encoder is presented. Here, the strains in each of the subgroups $\boldsymbol{\varepsilon}_j$ in the material layer are calculated based on the macroscopic strain $\boldsymbol{\varepsilon}^{\Omega}$ and on the internal variables stored in the previous time step $\boldsymbol{\alpha}_j^{t-1}$. For that purpose, a new set of weights is introduced in the network to learn how $\boldsymbol{\alpha}_j^{t-1}$ relates with the local strains $\boldsymbol{\varepsilon}_j$. This formulation is depicted in Fig. 2.29, where an extended version of Fig. 2.5b is used to illustrate the new connections. The purple arrow represents the new set of weights, while the blue refers to the parameters that take into account the macroscopic strain (as originally proposed in Section 2.4) and the black lines represent the flow of inputs and outputs of the constitutive model.

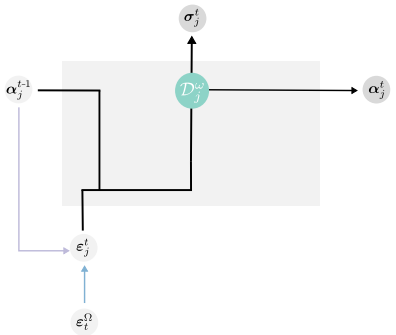


Figure 2.29: PRNN cell with local strains in the fictitious material point j now depend on the last internal variables state.

In the particular case where the architecture consists of input, material and output layers and no biases are considered, the current values used as strains input in the subgroup j can be defined as

$$\boldsymbol{\varepsilon}_j = \mathbf{W}_1^j \, \boldsymbol{\varepsilon}_t^{\Omega} + \mathbf{H}^j \, \boldsymbol{\alpha}_j^{t-1} \tag{2.38}$$

where $\mathbf{W}_1 \in \mathbb{R}^{n_1 \times n_0}$ is the weight matrix connecting the material layer of of size n_1 to the input layer of size n_0, $\mathbf{H} \in \mathbb{R}^{n_1 \times \text{ IntVar}}$ is the additional weight matrix connecting the material layer to the internal variables from the previous time step and $\boldsymbol{\alpha}^{t-1}$ is the concatenation of all internal variables in the material layer. In Eq. (2.38), j is used to refer to the part of the matrix (or vector) concerning the connections in the j-th subgroup. A more general approach is to make the local strains dependent on the internal variables of all subgroups. In that case, the additional weight matrix $\mathbf{H}$ has size $\mathbb{R}^{n_1 \times \, m \cdot \text{IntVar}}$ and Eq. (2.38) simplifies to

$$\boldsymbol{\varepsilon}_j = \mathbf{W}_1 \, \boldsymbol{\varepsilon}_t^{\Omega} + \mathbf{H} \, \boldsymbol{\alpha}^{t-1}. \tag{2.39}$$

To compute the gradients of the weights $\mathbf{H}$, we follow the procedure and notation described in Section 2.4.4, but instead of multiplying the values $\bar{\mathbf{d}}_i$ by the activations of the previous layer (or next layer to be backpropagated), which correspond to the strains input, one must multiply these values by the history vector from the previous time step (which corresponds to $\boldsymbol{\alpha}^{t-1}$):

$$\frac{\partial L}{\partial \mathbf{H}} = \bar{\mathbf{d}}_i \, \mathbf{h}_{t-1}^T \tag{2.40}$$

where $\bar{\mathbf{d}}_i$ is defined in Eq. (2.27). In addition to that, an extra term must be included in Eq. (2.28), $\mathbf{H}\,\bar{\mathbf{d}}_i$, to account for the new connections illustrated by the purple arrow in Fig. 2.29.

In both alternatives, the explicit introduction of the internal variables to the encoder makes the distribution of the macroscopic strain path-dependent. At this point, it is worth stressing that the proposal in Section 2.4 does now show this feature, but does take path-dependency into account in an implicit way by storing the internal variables updated by the material models. The contribution from the path-dependent internal variables also has a role in the tuning of the parameters in the encoder through the back-propagation process (see Eq. (2.27) and Eq. (2.28)).

To assess the effect of introducing this feature in the network, we use the same RVE geometry, material models, material properties, and plane stress conditions as in Section 2.6 and Section 2.7. Following the study in Section 2.6.4 where the network is trained with Type V curves and the architecture consists of an input layer, a single material layer with six fictitious material points and an output layer. Fig. 2.30 shows the performance of the PRNNs on different test sets with non-monotonic loading paths. In both scenarios, the uncertainty bounds over the size of the training set are wider for the case in which the internal variables of all fictitious material points are taken into account to evaluate the strains in each of the subgroups. This is an expected behavior since more parameters need to be tuned by the network in comparison to the methodology proposed in Section 2.4. This difference is reduced if subgroup j takes into account only its own internal variables to evaluate the local strains (see orange curves).

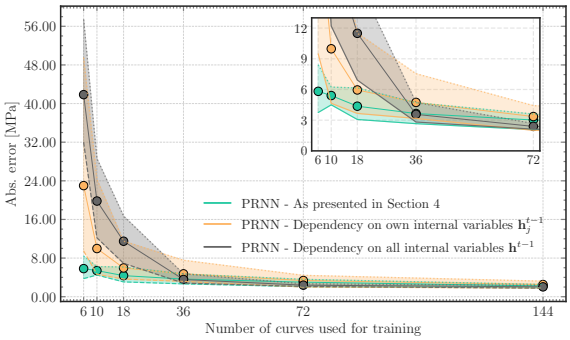
However, as the size of the training set increases, the uncertainty bounds become narrower and the new architectures outperform the simpler case by a small margin before all three methods converge to a similar error level. On that note, it is still unclear whether this gain is worth the increased amount of data or how these networks perform in FE^2 problems. Both topics are open for discussion in future research work. For the present work, results indicate that the absence of an explicit path-dependent encoder is not impeditive to the performance of the network.

REFERENCES

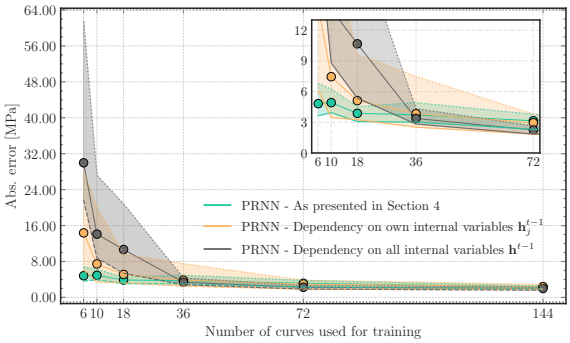
[1] V. P. Nguyen, O. Lloberas-Valls, M. Stroeven, and L. J. Sluys. “Computational homogenization for multiscale crack modeling. Implementational and computational aspects”. *International Journal for Numerical Methods in Engineering* 89.2 (2012), 192–226. ISSN: 00295981. DOI: 10.1002/nme.3237.

[2] M. Lefik, D. P. Boso, and B. A. Schrefler. “Artificial Neural Networks in numerical modelling of composites”. *Computer Methods in Applied Mechanics and Engineering* 198.21 (2009). Advances in Simulation-Based Engineering Sciences – Honoring J. Tinsley Oden, 1785–1804. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2008.12.036.

[3] D. Huang, J. N. Fuhg, C. Weißenfels, and P. Wriggers. “A machine learning based plasticity model using proper orthogonal decomposition”. *Computer Methods in Applied Mechanics and Engineering* 365 (2020), 113008. ISSN: 00457825. DOI: 10.1016/j.cma.2020.113008. arXiv: 2001.03438.



(a) Non-proportional and non-monotonic test set $\mathcal{T}_{\text{IVb}}$ = {100 Type V curves}



(b) Non-monotonic test set $\mathcal{T}_{\text{III}}$ = {100 Type III curves}

Figure 2.30: Absolute error over different test sets of the three different architectures trained on variable number of Type V curves and validated on 54 Type V curves.

[4] D. P. Kingma, T. Salimans, and M. Welling. *Variational Dropout and the Local Reparameterization Trick*. 2015. DOI: 10.48550/arxiv.1506.02557.

[5] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/arxiv.1412.6980.

[6] J. A. Hernández, M. A. Caicedo, and A. Ferrer. "Dimensional hyper-reduction of nonlinear finite element models via empirical cubature". *Computer Methods in Applied Mechanics and Engineering* 313 (2017), 687–722. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.10.022.

[7] T. Mori and K. Tanaka. "Average stress in matrix and average elastic energy of materials with misfitting inclusions". *Acta Metallurgica* 21.5 (1973), 571–574. ISSN: 0001-6160. DOI: https://doi.org/10.1016/0001-6160(73)90064-3.

[8] J. D. Eshelby and R. E. Peierls. “The determination of the elastic field of an ellipsoidal inclusion, and related problems”. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 241.1226 (1957), 376–396. DOI: 10.1098/rspa.1957.0133.

[9] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes. “A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics”. *Computer Methods in Applied Mechanics and Engineering* 379 (2021), 113741. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.113741.

[10] M. Eghbalian, M. Pouragha, and R. Wan. *A physics-informed deep neural network for surrogate modeling in classical elasto-plasticity*. 2022. DOI: 10.48550/arxiv.2204.12088.

[11] Z. Liu, C. T. Wu, and M. Koishi. “A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials”. *Computer Methods in Applied Mechanics and Engineering* 345 (2019), 1138–1168. ISSN: 00457825. DOI: 10.1016/j.cma.2018.09.020. arXiv: 1807.09829.

[12] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, and M. A. Bessa. “Deep learning predicts path-dependent plasticity”. *Proceedings of the National Academy of Sciences* 116.52 (2019), 26414–26420. ISSN: 0027-8424. DOI: 10.1073/pnas.1911815116.

[13] H. J. Logarzo, G. Capuano, and J. J. Rimoli. “Smart constitutive laws: Inelastic homogenization through machine learning”. *Computer Methods in Applied Mechanics and Engineering* 373 (2021), 113482. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.113482.

[14] C. Nguyen-Thanh, V. P. Nguyen, A. de Vaucorbeil, T. Kanti Mandal, and J.-Y. Wu. “Jive: An open source, research-oriented C++ library for solving partial differential equations”. *Advances in Engineering Software* 150 (2020), 102925. ISSN: 0965-9978. DOI: https://doi.org/10.1016/j.advengsoft.2020.102925.

[15] F. P. van der Meer. “Mesolevel Modeling of Failure in Composite Laminates: Constitutive, Kinematic and Algorithmic Aspects”. *Archives of Computational Methods in Engineering* 19.3 (2012), 381–425. DOI: 10.1007/s11831-012-9076-y.

[16] A. R. Melro, P. P. Camanho, F. M. Andrade Pires, and S. T. Pinho. “Micromechanical analysis of polymer composites reinforced by unidirectional fibres: Part I – Constitutive modelling”. *International Journal of Solids and Structures* 50.11 (2013), 1897–1905. ISSN: 0020-7683. DOI: https://doi.org/10.1016/j.ijsolstr.2013.02.009.

[17] F. P. van der Meer. “Micromechanical validation of a mesomodel for plasticity in composites”. *European Journal of Mechanics - A/Solids* 60 (2016), 58–69. ISSN: 0997-7538. DOI: https://doi.org/10.1016/j.euromechsol.2016.06.008.

3

EXPLORING THE LATENT SPACE IN THE LOW-DATA REGIME

In this chapter, we shift focus to the interpretability of PRNNs and the effect of different decoder architectures on the latent space. Particular emphasis is given to a new weight normalization constraint, which acts as a regularization technique and enables robust training in the low-data regime. A brief visual exploration illustrates how these changes impact the latent space, and how the fictitious stress can align with the true state of the RVE without explicit training. Reaping the benefits of a meaningful latent space, a case study illustrates how information from the microscopic level can be retrieved and incorporated into a multi-task approach that does not require extra parameters or larger training sets.

3.1. INTRODUCTION

Building on the developments presented in Chapter 2, this chapter further discusses the interpretability of a specific component of PRNNs, the decoder, and how it can be tailored to build accurate models with limited training data. While these networks can employ an arbitrary multilayer perceptron for both the encoder and decoder, here we consider a linear relationship. We also adopt the same Representative Volume Element (RVE), constitutive models and material properties as presented in Chapter 2, and only embed elastoplastic models in the material layer. This choice simplifies the model selection study and allows us to visualize and interpret more easily how variations on the decoder impact aspects such as interpretability, accuracy, and training requirements.

In this chapter, we are particularly focused on the low-data regime. This scenario reflects a practical constraint and consists of a stronger test of model generalization. Generating large datasets for training purely data-driven models, especially recurrent architectures, can quickly become a computational bottleneck when factoring in the cost of high-fidelity simulations. Strategies such as data augmentation [1] and combining multi-fidelity data [2] alleviate the issue, but the need remains for developing models inherently robust with low training requirements. While PRNNs already show strong performance in this setting, they can benefit from further improvements. In Chapter 2, we showed how 18 monotonic paths in basic loading directions (e.g. uniaxial strain, pure shear, and biaxial cases) were enough to capture unseen unloading/reloading behavior accurately. Here, we investigate how architectural design choices can help push this number to a new lower bound without compromising accuracy.

Robustness in the low-data regime is especially valuable when training on experimental data, helping avoid (additional) time-consuming and costly testing campaigns. Data-efficient models can also be more easily integrated into frameworks where the surrogate is trained on-the-fly as more data/information on the micromodel becomes available. Beyond accuracy, the changes in the decoder are evaluated on their impact on the latent space, specifically on how fictitious quantities relate to the true microscopic ones. We then select a relevant microscopic measure to compare with its fictitious counterpart without explicitly training to approximate it. This sets the stage for incorporating the microscopic quantity of interest into a multi-task approach.

The different decoder connectivities are presented in Section 3.2, followed by an accuracy assessment in the low-data regime in Section 3.3. Then, a brief illustration of the link between the latent space of the PRNN and the microscopic quantities is presented in Section 3.4. In Section 3.5, a case study illustrates to what extent macroscopic and microscopic quantities can be learned simultaneously without compromising performance. Finally, the main conclusions of the chapter are summarized in Section 3.6.

3.2. TOWARDS SPARSITY AND INTERPRETABILITY

In this section, different variations of the decoder architecture are investigated. We start with the basic dense layer design from Chapter 2, in which all stress components from a material point connect to all homogenized stress components. Next, we remove the cross-component connections, allowing only matching components to contribute to the homogenized stress. Finally, we impose that all stress components from a material point

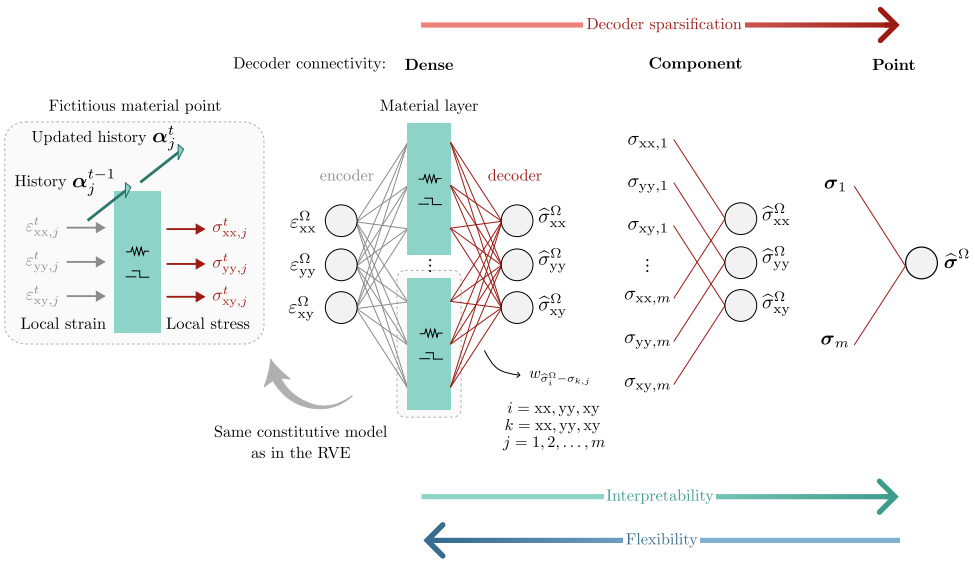


Figure 3.1: Connectivities of decoder investigated in this chapter.

share the same weight. These three options are hereafter referred to as "Dense", "Component" and "Point", respectively, and are illustrated in Fig. 3.1. Their predictions can be summarized as follows

$$\begin{aligned}
&\text{Dense:} && \widehat{\sigma}_i^{\Omega} = \sum_{j=1}^{m} \sum_{k} w_{\widehat{\sigma}_i^{\Omega} - \sigma_{k,j}} \sigma_{k,j} && i = \text{xx, yy, xy} \quad k = \text{xx, yy, xy} \\
&\text{Component:} && \widehat{\sigma}_i^{\Omega} = \sum_{j=1}^{m} w_{\widehat{\sigma}_i^{\Omega} - \sigma_{i,j}} \sigma_{i,j} && i = \text{xx, yy, xy} \\
&\text{Point:} && \widehat{\sigma}_i^{\Omega} = \sum_{j=1}^{m} w_{\widehat{\sigma}^{\Omega} - \sigma_j} \sigma_{i,j} && i = \text{xx, yy, xy}
\end{aligned} \tag{3.1}$$

where the absolute function is applied to the weights $w_{\widehat{\sigma}_i^{\Omega} - \sigma_{(\cdot)}}$ of all architectures to ensure positivity.

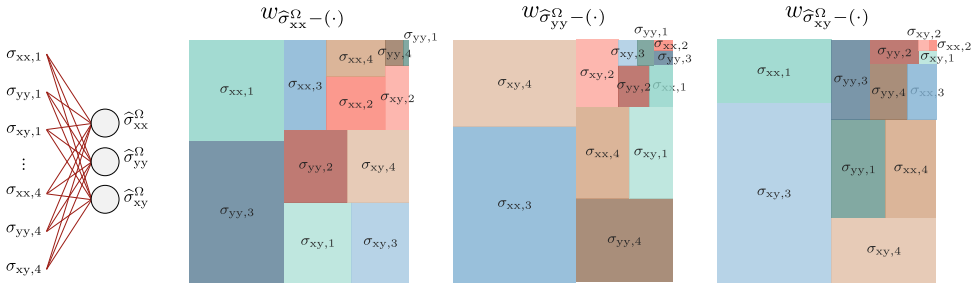
In addition to these changes in the connectivity, we also investigate the addition of a normalization on the sum of the weights connecting to each component i of the homogenized stress:

$$\begin{cases}
\text{Dense: } \sum_{j=1}^{m} \sum_{k} w_{\widehat{\sigma}_i^{\Omega} - \sigma_{k,j}} = 1.0 \\
\text{Component: } \sum_{j=1}^{m} w_{\widehat{\sigma}_i^{\Omega} - \sigma_{i,j}} = 1.0 \\
\text{Point: } \sum_{j=1}^{m} w_{\widehat{\sigma}^{\Omega} - \sigma_j} = 1.0.
\end{cases} \tag{3.2}$$

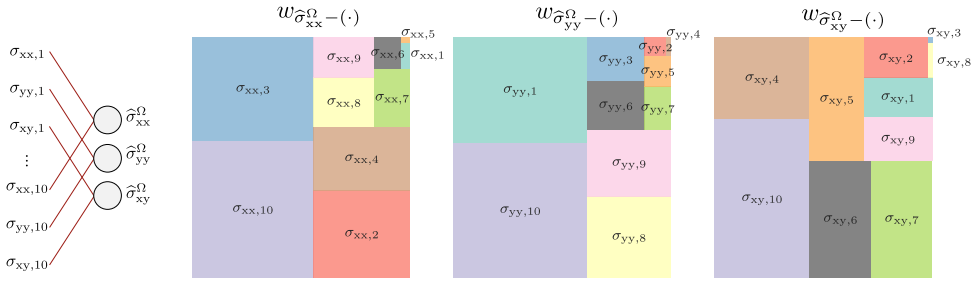
The idea is to reinforce the parallel with Eq. (2.14), where the weights used for the numerical integration are strictly positive, with their sum totaling the volume of the RVE. For the same number of material points, we thus move gradually from a flexible (Dense) to

a more rigid (Point) architecture. In the following sections, we illustrate how these variations affect the network in terms of training requirements, accuracy, and interpretability.

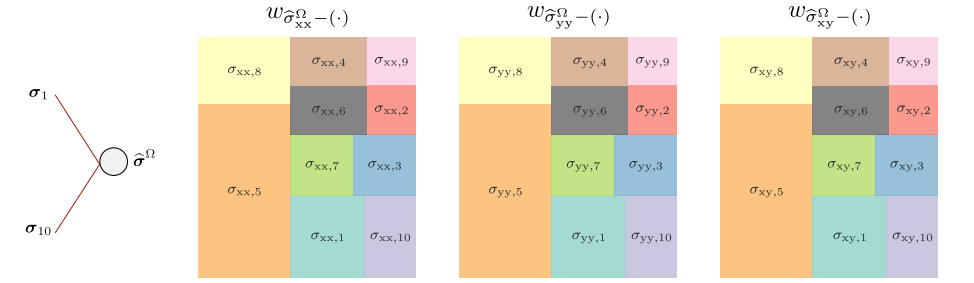
Before assesing their performance, we visualize the decoder architectures discussed previously through the lens of squarified treemaps. Details on model selection are omitted for now but can be found in Section 3.3, where these variations are assessed in terms of accuracy. Here we consider networks trained on 5 loading paths randomly selected from a pool of 1000 non-proportional and non-monotonic paths, referred to as GP-based paths.



(a) Dense decoder with 4 material points: full connectivity between output and material layer allows high expressivity



(b) Component decoder with 10 material points: sparse connectivity removes cross-component contributions



(c) Point decoder with 10 material points: to reflect the computational homogenization of the micromodel full-field solution, shared weights per material point are imposed to compute $\widehat{\boldsymbol{\sigma}}^{\Omega}$

Figure 3.2: Representation of weights from Dense, Component and Point decoders.

In the treemaps, each rectangle represents a connection used to compute a given com-

ponent of the homogenized stress $\widehat{\sigma}_i^{\Omega}$. Specifically, the area of each rectangle is proportional to the magnitude of the corresponding weight, not to be confused with the magnitude of the stress computed by the constitutive model. In Fig. 3.2a, we illustrate the decoder weights of a PRNN with Dense decoder and four material points. Due to the full connectivity between the output and material layer, all stress components from the four material points are used when computing $\widehat{\sigma}_i^{\Omega}$, resulting in a treemap with 12 rectangles. In each treemap, different contributions from the same material point can be seen, highlighting the high flexibility of this option. This flexibility, however, deviates from the true homogenization process. While the size of the rectangles related to cross-components do not necessarily imply they have the highest contribution in the approximate homogenized stress, in the RVE, cross-component stress responses do not contribute at all.

The second alternative, shown in Fig. 3.2b, removes the cross-contribution from fictitious stress components that do not match the predicted homogenized stress component, resulting in a design with less flexibility, but with better interpretability and a closer link to the underlying physical pattern being learned. Note that in this case, because weights are still defined per component and per material point, in some cases, a material point that has close to no contribution in one component can have a significant contribution in another (e.g., see the weights associated with material point 3 in Fig. 3.2b).

Lastly, in Fig. 3.2c, we plot the treemaps for a PRNN with 10 material points using a decoder with connections defined per point. As a result, the representation of the weights considered for each component of the homogenized stress is precisely the same. This layout is closest to the homogenization procedure, where integration weights are assigned per material point, not per component. This is the least flexible architecture but the most interpretable one. In all cases, if the sum of the areas in each treemap amounts to 1, we have the normalized version of the decoder.

3.3. ASSESSING ACCURACY

In this study, several PRNNs are trained over different material layer sizes ranging from 2 to 30 fictitious material points and training set sizes ranging from 1 to 5 curves. For each combination, 10 initializations are considered, and in each one, the training set is randomly drawn from a pool of 1000 curves. The data generation procedure follows the approach discussed in Section 2.5, with GP-based paths being the preferred choice due to their higher complexity. Fixed validation $\mathcal{V}_{\text{GP}}$ and test $\mathcal{T}_{\text{GP}}$ sets consisting of 150 GP-based paths each are considered for training and assessing the network in terms of accuracy.

The Adam algorithm is used to optimize the network's parameters with two stopping criteria. The first is a maximum number of epochs, set to 8000, and the second is an early stopping criterion which interrupts the training if there is no improvement on the validation loss for 400 consecutive epochs. The batch size is set to 1 path, while the remaining optimization parameters are Pytorch's defaults. The training and validation losses are calculated using the Mean Squared Error (MSE), while model performance on the test set is assessed using the Mean Absolute Error (MAE) and its relative counterpart (RMAE) for more interpretable measures of accuracy.

Finally, following the idea that the decoder weights represent a homogenization operator, we modify the initialization of this set of parameters to:

1) draw weights from an uniform distribution between 0 and 1;

2) normalizing weights connecting to $\widehat{\sigma}_i^{\Omega}$, $w_{\widehat{\sigma}_i^{\Omega}-(\cdot)}$, to sum to 1.

Note that this procedure normalizes the decoder weights only at the start of the training procedure. For unnormalized decoders, these weights are free to change and to sum to arbitrary (positive) values as epochs evolve, while for normalized decoders, the sum-to-one instruction is a constraint fulfilled at all times. In the following, we start with the unnormalized and then move to the normalized decoders to disentangle potential accuracy gains stemming from two distinct aspects: sparsity and weight normalization.

3.3.1. UNNORMALIZED DECODERS

In Fig. 3.3, we plot the average errors of the trained networks over the test set $\mathcal{T}_{\text{GP}}$ for all combinations of material layer and training set sizes. Each colored surface corresponds to a different decoder architecture, and each vertice of a surface corresponds to the average error of the different initializations. One observation from this plot concerns the marked difficulty to generalize from limited data, illustrated by the steep growth of errors as the number of training curves decreases.

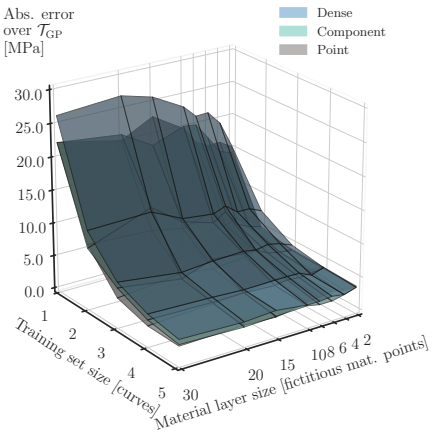
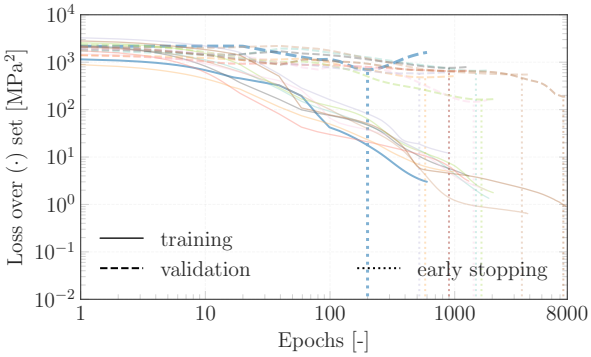
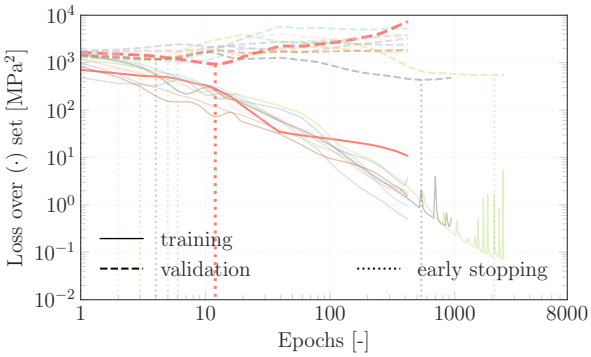


Figure 3.3: Average errors of PRNN trained on different decoders, training set sizes and material layer sizes over test set $\mathcal{T}_{\text{GP}}$.

For one training curve in particular, Fig. 3.4 shows that given enough epochs, the networks eventually fine-tune their predictions over training, which often compromises their generalization on unseen data. To avoid this (potential) overfitting, we employed an early stopping strategy by halting training and selecting the model at the historical lowest validation set, as highlighted by the blue and red curves in Figs. 3.4a and 3.4b, respectively. Still, the amount of data is insufficient for the network to reliably extract any pattern, resulting in consistently high validation errors.



(a) 2 fictitious mat. points (36 weights)



(b) 30 fictitious mat. points (540 weights)

Figure 3.4: Training and validation losses from different initializations of PRNNs with Dense decoder trained on 1 curve and two material layer sizes.

As the decoder connectivity is reduced - from Dense to Component to Point — the network incorporates stronger inductive biases, more closely mirroring the RVE homogenization. This comes at the cost of flexibility, but it also leads to improved generalization. In Fig. 3.3, the surfaces associated to the sparse decoders, Component and Point, are virtually always below the one associated to the Dense architecture. For more detailed comparison, we plot in Fig. 3.5 the relative errors of the three architectures over the training set sizes considered. The networks trained on a single curve show large variance for all decoder variations, ranging from 13% to errors beyond 100%. With smaller networks (2-6 fictitious mat. points), however, the addition of more curves to the training set result in a significant accuracy gain and large variance reduction, consistently reaching the 10% mark with 4 training curves. Finally, with 5 curves, the accuracy of the three decoders converge towards the same level.

To illustrate the impact of the different connectivities on the latent space and prediction, we plot in Fig. 3.6 the local stresses and their decoded contributions to the ho-

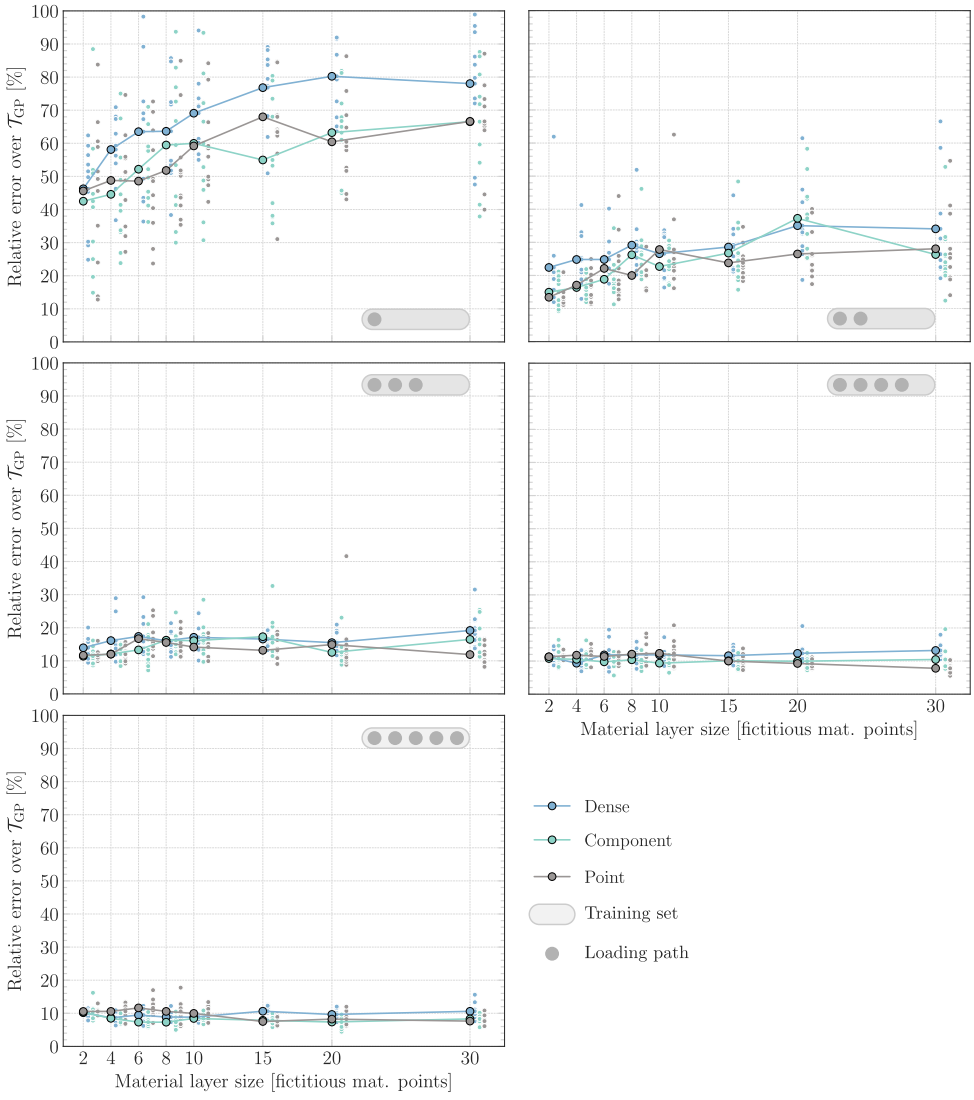
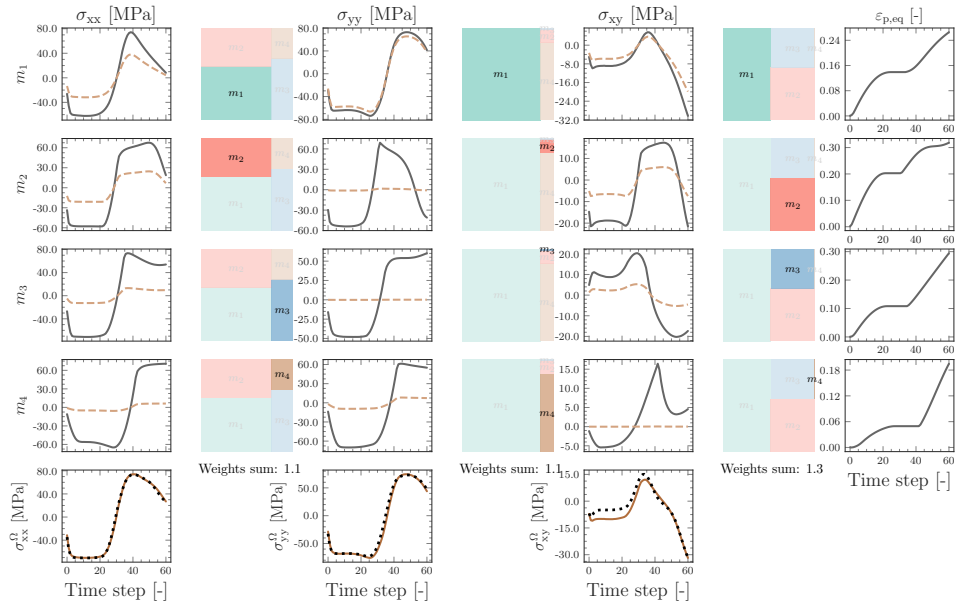
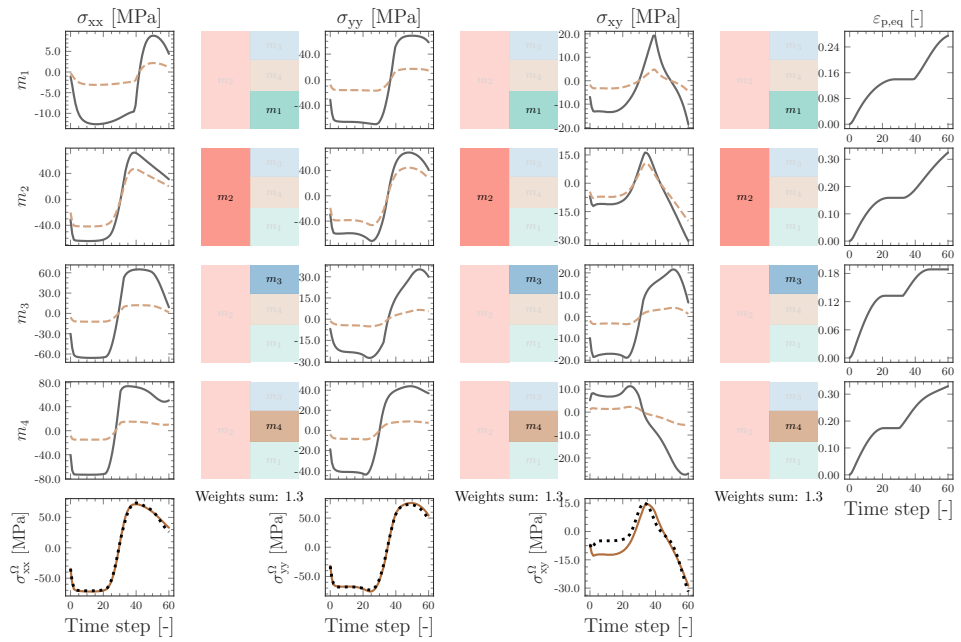


Figure 3.5: Relative error over $\mathcal{T}_{GP}$ for different material layer sizes and training set sizes, with solid lines referring to the average performances.

mogenized stresses for two networks on a representative test path from $\mathcal{T}_{GP}$. Each row corresponds to a fictitious material point, while the last row corresponds to the homogenized stresses. To the right of the local stresses of each point, we illustrate the squarified treemaps representing the decoder weights per component. Although both models have virtually the same accuracy, these plots help elucidate how local stresses contribute differently to the homogenized stress. For example, in Fig. 3.6a, point m_3 contributes to



(a) Component, absolute error 2.7 MPa, relative error 6.4%



(b) Point, absolute error 2.5 MPa, relative error 5.9%

Figure 3.6: Prediction and latent space of PRNNs trained on 5 curves with different decoder types and 4 material points on representative curve from $\mathcal{T}_{GP}$.

the x and xy components but is almost absent to y. With the Point decoder, this kind of adjustment is not possible since the weight is shared across all components of a point.

Overall, although sparse architectures improve performance in the low-data regime, training on limited data, defined here as 1-2 curves, remains a challenge. To address that, we explore in the following section the role of the weight normalization, presented in Section 3.2, in improving generalization and reducing even further training requirements.

3.3.2. NORMALIZED DECODERS

Given the difficulty of generalizing from limited datasets, we now explore how a weight constraint can improve the performance of the network. This constraint (see Eq. (3.2)) is inspired by the interpretation of decoder weights as volume contributions in the homogenization, and we begin by comparing the PRNNs with Dense and Component decoders. Fig. 3.7 shows the relative errors over the test set $\mathcal{T}_{\text{GP}}$ for two training set sizes. A general remark is that the normalized decoders generally show superior performance across all combinations of model complexity and training set size. Most remarkably, they benefit from the inclusion of more material points while generalizing well (i.e. error below 10%) with training sets as small as two curves, a stark contrast to the unnormalized decoders.

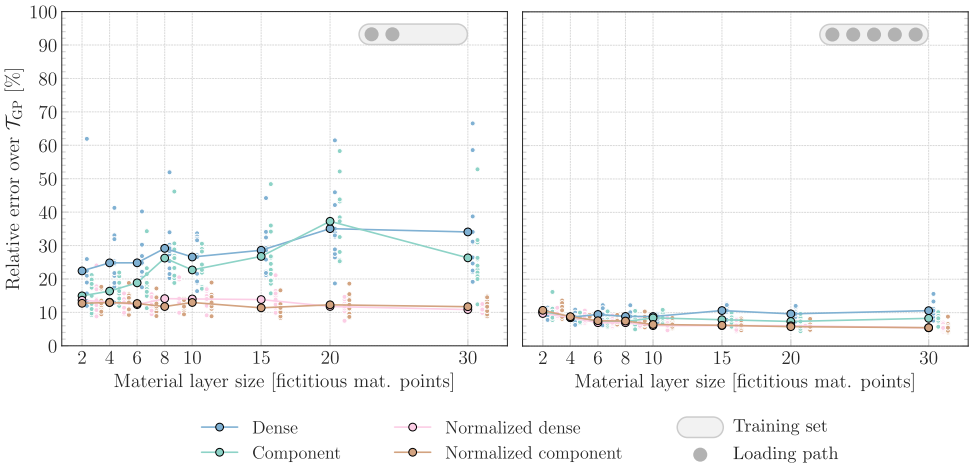
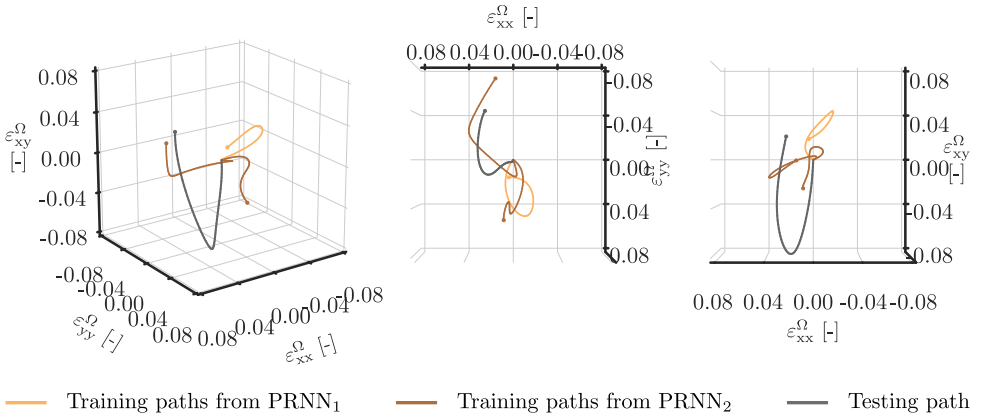


Figure 3.7: Effect of normalization on Dense and Component decoders in terms of relative errors over test set $\mathcal{T}_{\text{GP}}$ for different training set and material layer sizes.

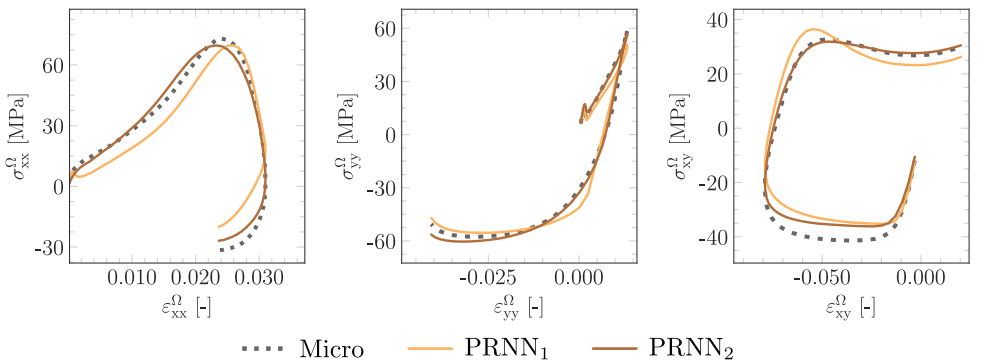
Another positive outcome brought by the normalization is the reduced variance. When training with small sets, the randomly selected training paths might cover distinct parts of the design space (compression and tension) or wander around a narrow portion of it. With large decoder weights, the model becomes overly sensitive to these variations. Normalization mitigates this effect by limiting the influence of any single material point, promoting more stable and consistent outputs across different training sets.

To illustrate the matter, we consider the best PRNN trained on one and two paths with 30 material points from Fig. 3.7. The training and testing paths of each are shown in

Fig. 3.8a. Despite covering distinct regions of the design space, the model trained on a single curve already produces smooth homogenized stresses, though with high bias, as illustrated in Fig. 3.8b. As the training set increases to two curves, the error drops from 20% to 12%. A final remark from Fig. 3.7 is that no clear distinction can be made between the two normalized decoders. While their unnormalized counterparts showed discernible differences in performance under limited data and/or larger material layers, the regularized versions show improved and similar accuracy levels.



(a) Strain view of paths used to train and test networks PRNN$_1$ and PRNN$_2$ with 30 material points



(b) Stress-strain curve of testing path in dark grey shown in Fig. 3.8a using PRNN$_1$ and PRNN$_2$

Figure 3.8: Training and test paths for 2 PRNNs with Normalized component decoder and 30 material points trained on 1 and 2 curves.

Next, we move to the Point architecture. In this case, Fig. 3.9 shows a more nuanced conclusion. With a very strong inductive bias, the Normalized point architecture reaches errors as low as 13% with a training set consisting of a single curve, proving to be, on average, the best option in this extreme scenario. However, as more training data is considered, the gain in accuracy is small and reaches a plateau around 4-5 curves with 12%, while the remaining options surpass it, including its unnormalized counterpart, with errors below 5%.

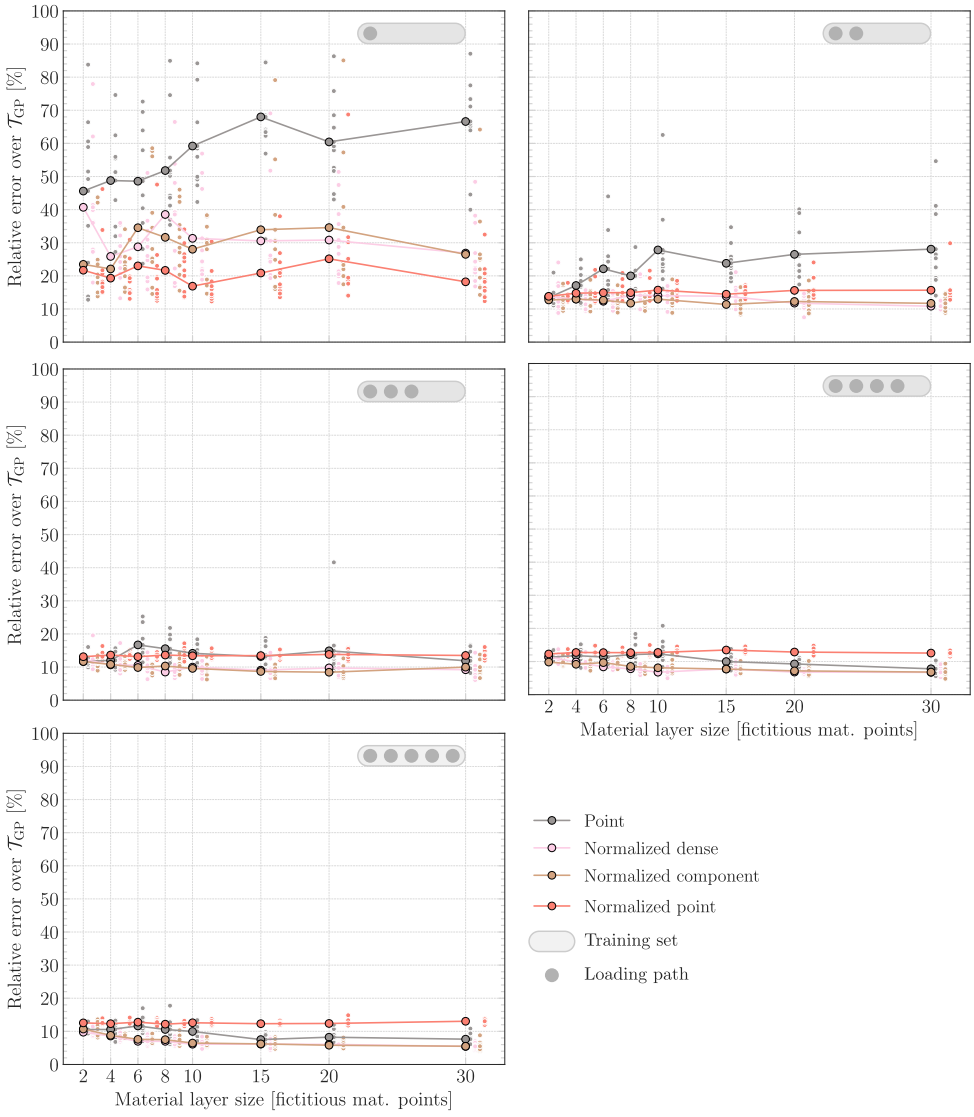


Figure 3.9: Relative error over $\mathcal{T}_{GP}$ for different combinations of material layer sizes and training set sizes using Point and normalized decoders, with solid lines referring to the average performances.

To better understand why normalization is less effective with the Point decoder, we plot in Fig. 3.10 the relative training and validation errors for the smallest and largest training set sizes considered so far for both the normalized and unnormalized versions. As the number of training curves increases, with the Point decoder, both training and

validation errors decrease. In the normalized counterpart, while the validation errors decrease with more data, the training error increases slightly, with a plateau around 10%, indicating underfitting. However, adding more material points has minimal effect, suggesting that despite the relatively strong performance with only one training curve, weight normalization ends up overconstraining the architecture. In short, the same inductive bias that enables robust training on limited data also causes learning to saturate as more data becomes available.

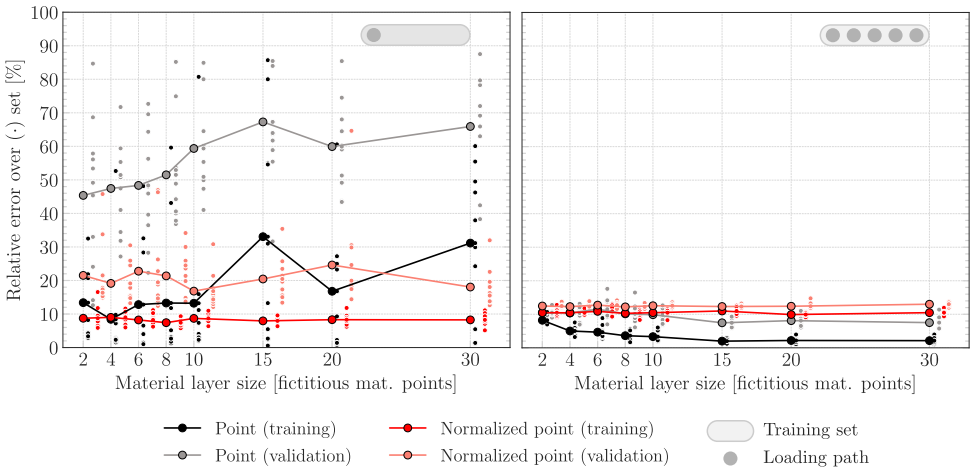


Figure 3.10: Relative error over training and validation sets for PRNNs using Point decoder with and without normalization.

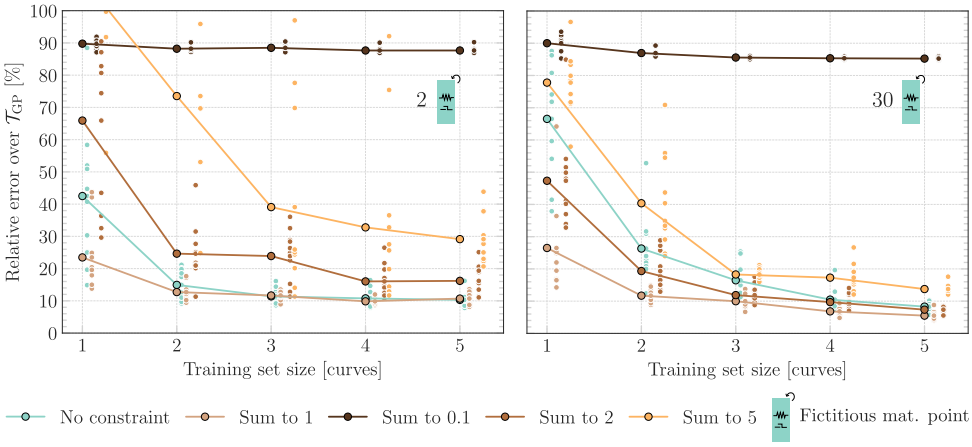
Finally, we briefly illustrate how the choice of enforcing the decoder weights sum to 1.0 compares with other alternatives for the Normalized component and Normalized point. As the performance of the Normalized component was very similar to the Normalized dense one, we expect the same conclusions to hold. For this study, we consider three additional scenarios where the weights per component are normalized to sum to 0.1, 2.0 and 5.0. Fig. 3.11 shows the accuracy of the network in each case. In the Normalized component case, none of the alternatives match the performance of the one explored in this section previously where weights sum to 1. Our choice naturally aligns with the physical interpretation of the decoder as a volume-weighted homogenization. Furthermore, unlike common regularization strategies based on penalty parameters, such as L1 or L2, no additional hyper-parameter is introduced.

For the Normalized point decoder, however, loosening the sum of the weights to higher values shows a positive effect for large networks (30 points) and training sets (4-5 curves), as shown in Fig. 3.11b. Nevertheless, the gain in accuracy is only as good as the version without constraint, while the original sum to 1 version remains the best option with limited data.

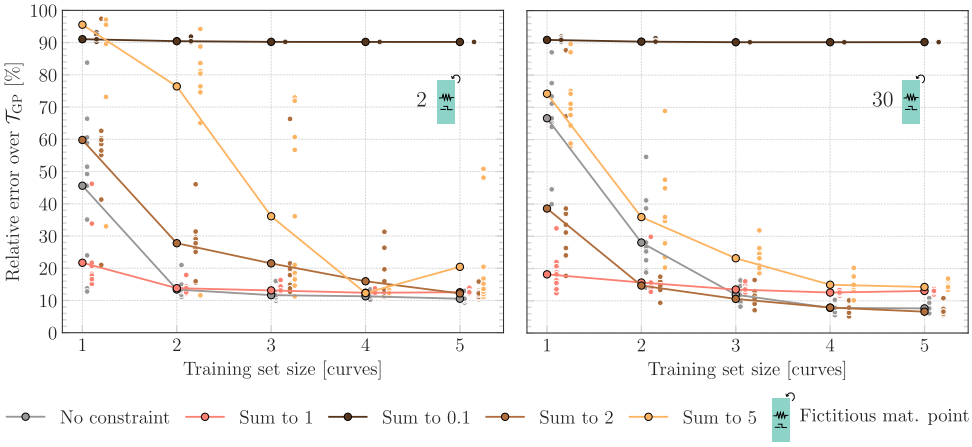
3.4. A VISUAL EXPLORATION OF THE LATENT SPACE

Previously, the effect of the different decoders was illustrated in terms of accuracy. In this section, we illustrate their impact in the latent space variables, shedding light on how the weight normalization constraint and sparsity work through a series of visualizations on representative (test) loading paths.

Recall that our latent space is composed of strains, learned by the encoder, and stresses and internal variables, obtained from the embedded constitutive models. All networks used to illustrate this section were trained on 5 curves and have 30 fictitious material points. Specifically, the networks with the lowest relative test error among the different initializations were selected.



(a) Normalized component



(b) Normalized point

Figure 3.11: Relative error for PRNNs using Normalized component and Normalized point decoders with different sum constraints for fixed material layer sizes.

3.4.1. FICTITIOUS VS MICROSCOPIC STRESSES

Firstly, we compare the distribution of the fictitious stresses obtained in the material layer with the true full-field stress distribution of the micromodel for the different decoder architectures explored in the previous section. Fig. 3.12 shows the histogram comparison for an unseen GP-based curve. To allow a side-by-side comparison, we plot 4 time steps out of the 60 considered for the component with the highest magnitude only. Each row corresponds to the snapshots of a specific type of decoder and is color-coded accordingly.

In Fig. 3.12, only the distribution from the Normalized component shows a visually good match with the true stress distribution in the matrix, while the unnormalized alternatives show a more uniform distribution centered around zero across all time steps. This is reflected in Fig. 3.13, where we plot the mean matrix stress at each time step for the three stress components. Note that the Normalized point also captures the mean matrix stress well, but its accuracy in predicting the homogenized stress, our training target, is somewhat limited, as discussed in Section 3.3.2.

Other metrics could be used to estimate the distance between the two distributions (true and fictitious). The Wasserstein distance, for example, provides more nuanced insights into the distribution differences, going beyond pointwise comparisons or aggregate errors. For simplicity and practicality, however, we focus on the following two measures: the mean and the maximum stress. For this application, these metrics already capture key aspects of the mismatch without introducing additional computational and interpretability burdens. Fig. 3.14 shows the relative errors for the mean and maximum stresses over the test set $\mathcal{T}_{\text{GP}}$ for the different decoders.

Without normalization (left), all decoders struggle to capture either the mean or the maximum, stabilizing at high errors. With normalization (right), errors in the mean reach lower values, with best performances as low as 10%. For the maximum stress, however, only the Normalized component shows a marked improvement, with errors below 20% in the best-case scenario, while the other alternatives remain comparable. These results indicate that normalization is crucial for aligning the fictitious distribution more closely with the true one, but it is not sufficient on its own. The best match between the distributions comes from the Normalized component, an architecture that introduces some structure regarding cross-component stress contributions, unlike the Normalized dense, but still has enough flexibility to allow weights to vary per component, unlike the Normalized point.

Finally, it is worth mentioning that although we started our comparison directly with the matrix phase only, we included in Appendix a comparison on the similarity of the fictitious stresses with the full micromodel response. Based on the lowest error, the fictitious distributions are indeed closer to the matrix phase than to the full micromodel. This agreement is expected, given the dominance of the matrix in the current setup. However, for more intricate micromodels (e.g., a richer mix of constitutive behaviors and/or several phases), such clear alignment should be carefully examined.

3.4.2. FICTITIOUS VS MICROSCOPIC INTERNAL VARIABLES

Next, we investigate the similarity between the fictitious and the true internal variables of the micromodel. This time, we narrowed our study to four decoder types: Dense, Nor-

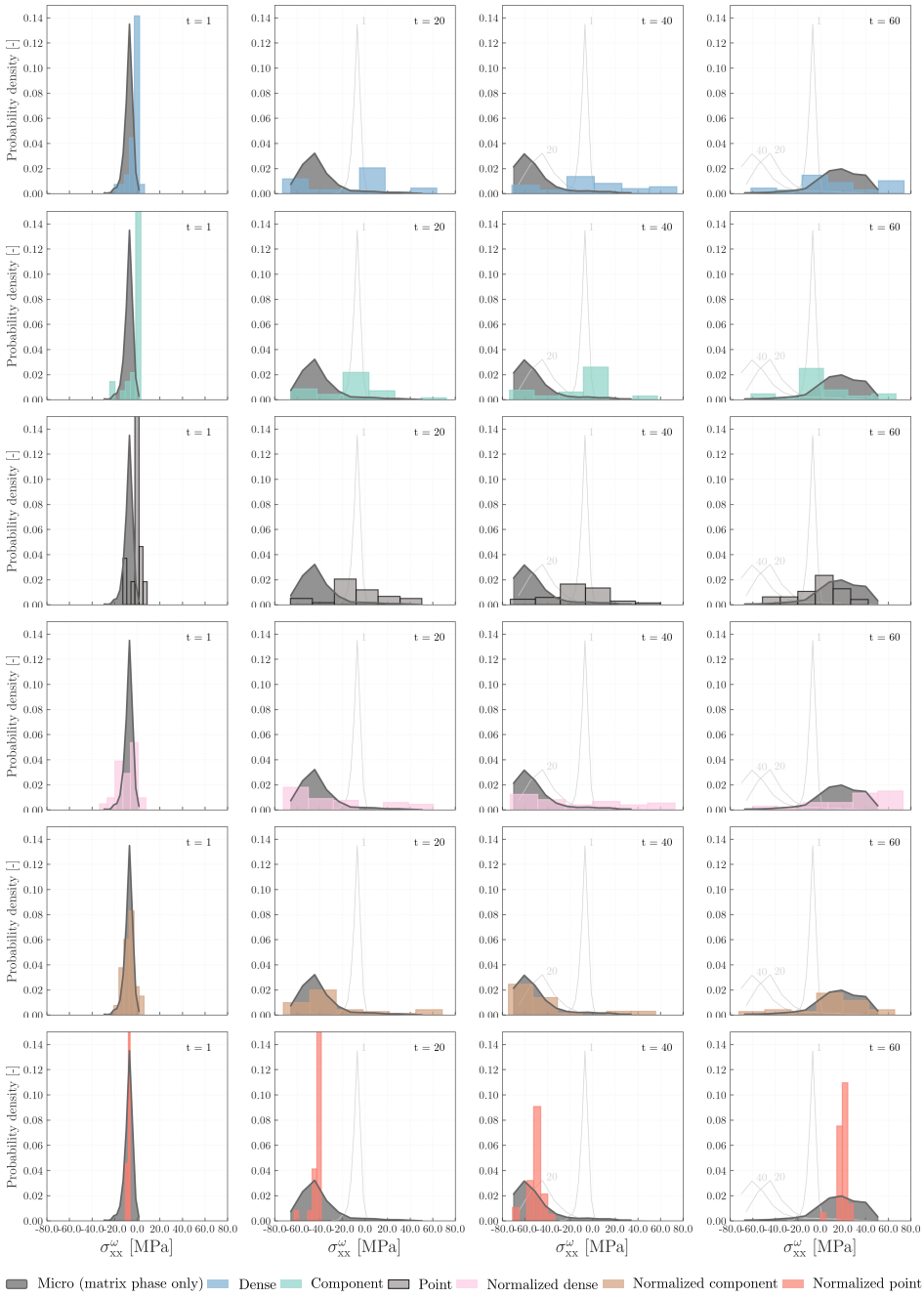


Figure 3.12: Histogram distributions from the stresses computed in the matrix phase of the micromodel vs the stresses obtained from the fictitious material points of PRNNs with different decoder architectures (color-coded) at several time steps of an unseen GP-based loading path.

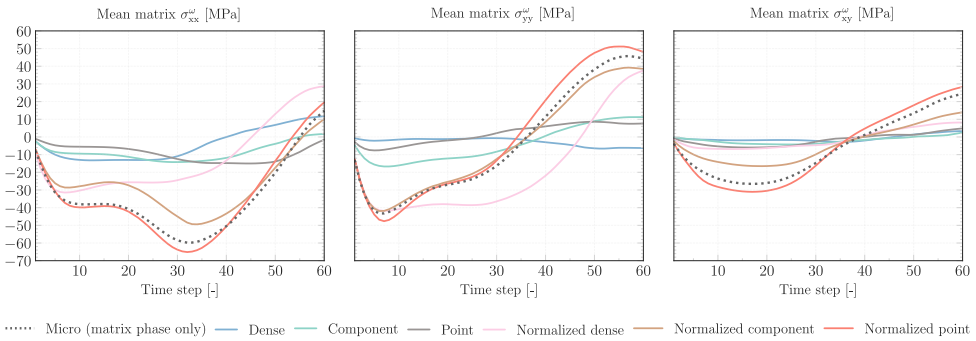


Figure 3.13: Mean matrix stress using different decoder architectures for unseen and representative GP-based path from Fig. 3.12.

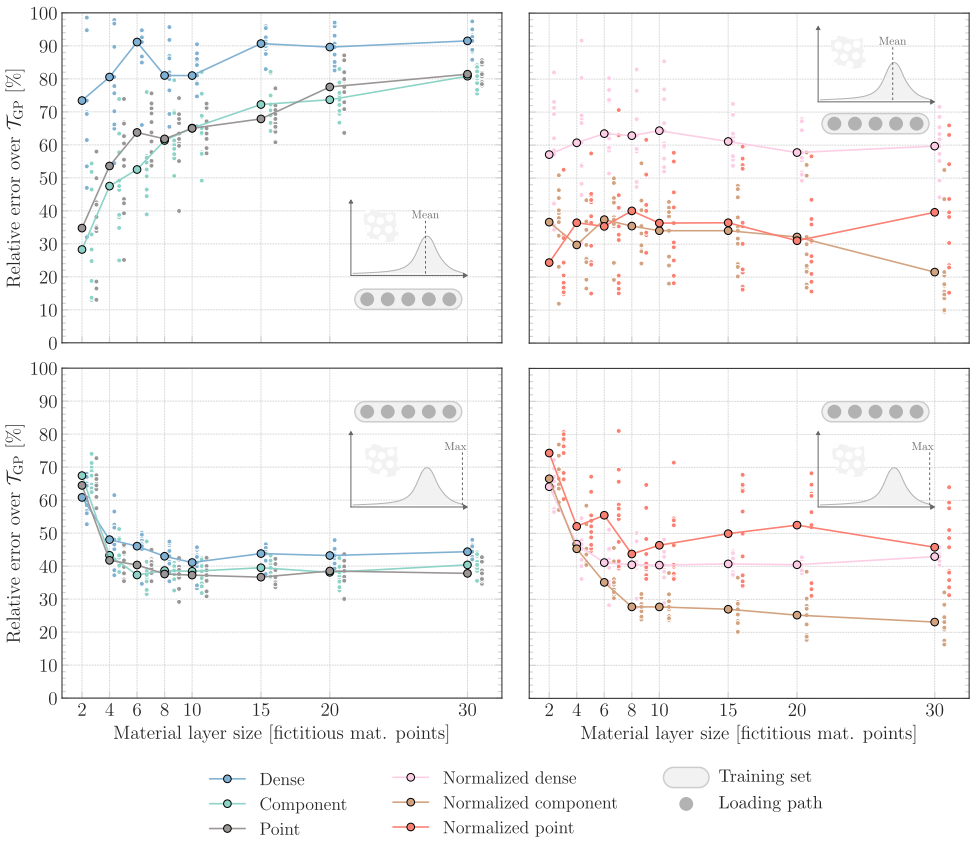


Figure 3.14: Relative error over $\mathcal{T}_{GP}$ of fictitious stresses with different decoders compared to true microscopic stress distribution over matrix phase.

malized Dense, Normalized component, and Normalized point. The first one is used as a baseline reference (no sparsifying structure or weights sum constraint). The testing loading path used for illustration purposes consists of a proportional loading path, specifically uniform stretch in the x-direction with free deformation of the y-direction.

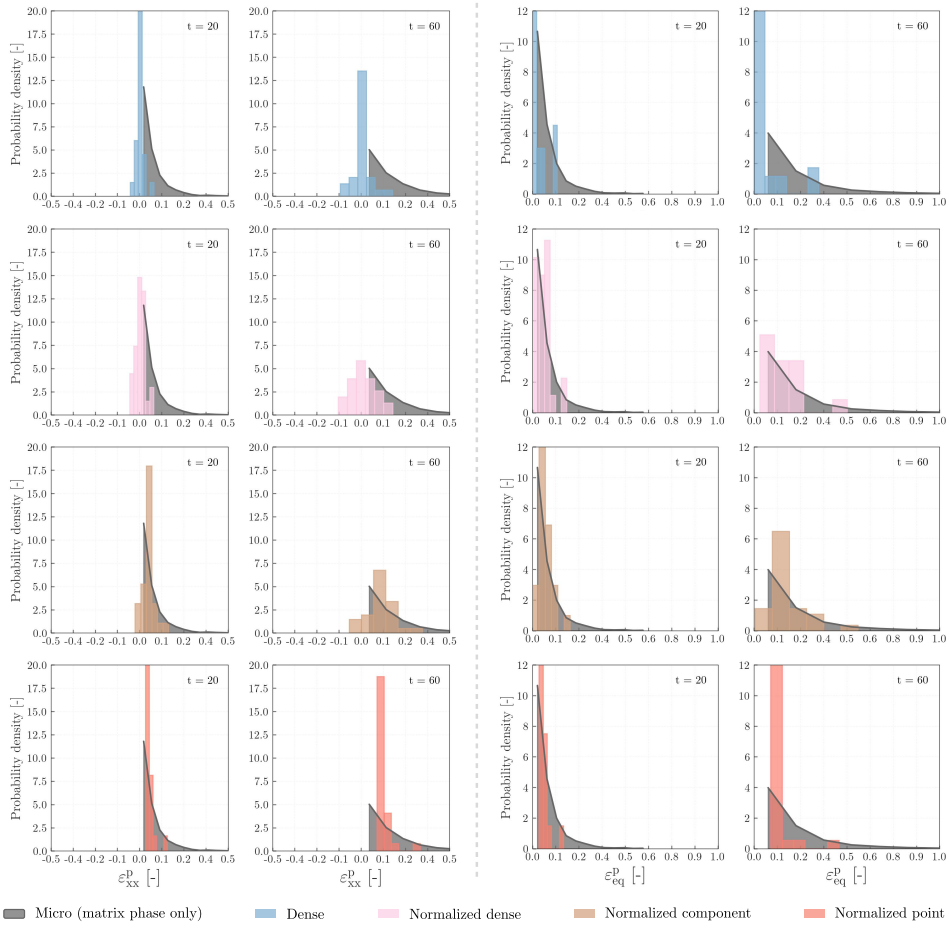


Figure 3.15: Distribution of plastic strains and equivalent plastic strain for PRNNs with different decoders on proportional loading path.

Fig. 3.15 shows two types of histograms: on the left, the distribution of one of the components of the plastic strain tensor, and on the right, the distribution of equivalent plastic strain. Again, the pattern from the network with the Dense decoder produces a distribution roughly centered around zero, while the architectures with sparsity and normalization follow the one-sided true distributions, though with limited accuracy. When it comes to the equivalent plastic strains, however, all alternatives show only positive values, a consequence of using constitutive models grounded in physics laws.

While the tailored changes in the decoder help align the fictitious and true distribu-

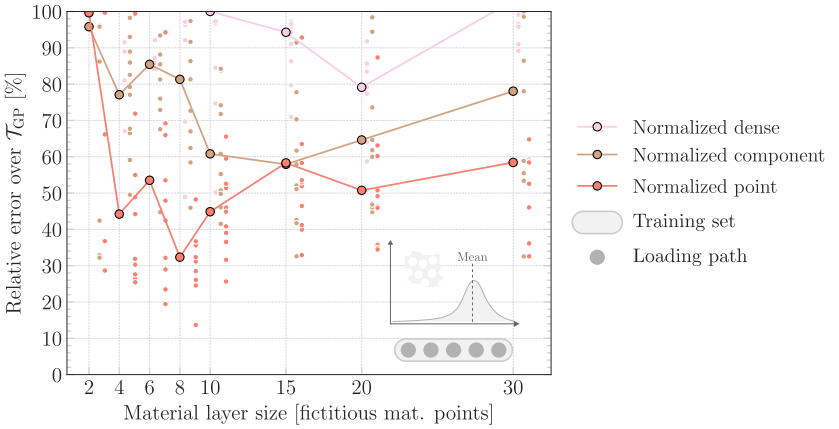


Figure 3.16: Relative error between mean fictitious plastic strains and mean microscopic plastic strains (matrix phase only) for PRNNs with different decoders on proportional loading path.

tions of plastic strain, Fig. 3.16 shows that the overall match remains significantly poorer than the stress comparison in Section 3.4.1. This is expected since the networks are trained only on homogenized stress snapshots and only embed the matrix constitutive model, lacking the richer microstructural details present in the full RVE.

3.5. FROM SINGLE TO DOUBLE-TASK BASED ON THE LATENT SPACE

Building on the findings in Section 3.4 regarding the similarity between the fictitious and the microscopic distributions, we now explore how this knowledge can be leveraged to predict not only homogenized quantities, but also relevant microscale information. In particular, we are interested in predicting the maximum hydrostatic stresses in the matrix, as these have been recently linked to macroscopic failure [3, 4].

While the PRNN cannot predict failure directly, by establishing a threshold maximum stress one could, for instance, transition to a failure state (e.g., by inserting cohesive segments), switch to another surrogate model, or revert back to a full-order micromodel at the corresponding macroscopic integration point. To obtain the maximum hydrostatic stress in the matrix without modifying the current architecture, we add the following steps to the forward pass at each time step:

1) with the stresses computed by material model, compute the hydrostatic stress for each fictitious material point;

2) from the pool of hydrostatic stresses, select the maximum.

Our first test scenario employs the same networks considered in the previous sections - trained to predict $\widehat{\boldsymbol{\sigma}}^{\Omega}$ - to now predict our new quantity of interest $\sigma^{\omega}_{\text{hydro,max}}$. For this,

we consider the test set $\mathcal{T}_{\text{GP}}$ with 150 curves, and refer to the maximum hydrostatic stress values as $\mathcal{T}_{\text{GP}}^{\text{max.}}$ when predicting the "Maximum" task (short for the maximum hydrostatic stress in the matrix) and $\mathcal{T}_{\text{GP}}^{\text{avg.}}$ when predicting the "Average" task (short for the homogenized stresses).

We focus on one specific decoder architecture, the Normalized component, which showed the lowest error between the maximum fictitious stress and the maximum microscopic stress over the matrix phase in Section 3.4.1. The performance over the average task was discussed in detail in Section 3.3, where we showed that with as few as two curves, relative errors in predicting the homogenized stresses remained consistently around 10% or below across all material layer sizes tested.

For the unseen Maximum task, Fig. 3.17 shows that relative errors plateau around 30% for training sets with two curves. With more training data (4-5 curves), the average error decreases to around 24%, with the best performance at approximately 16%. While these results show some correlation between the fictitious and true maximum hydrostatic stresses without training, the accuracy remains insufficient. Therefore, in the next section, we explicitly add the maximum hydrostatic matrix stress as a supervised target.

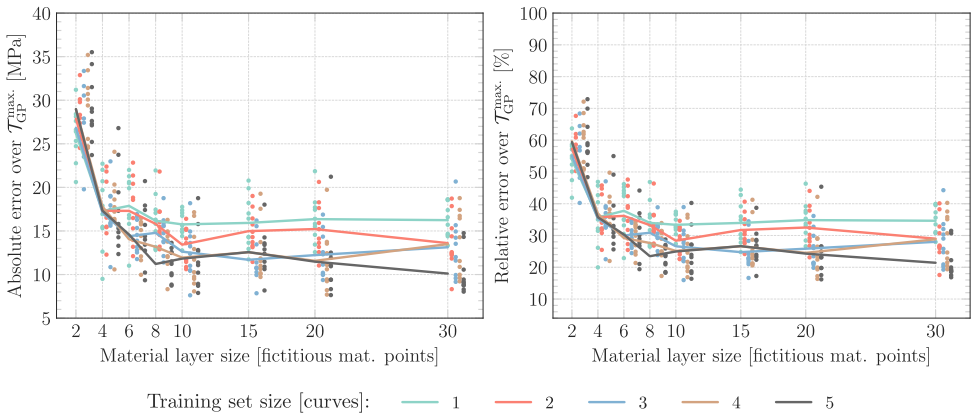


Figure 3.17: Test set errors of PRNN with Normalized component decoder over the unseen task of predicting the maximum hydrostatic matrix stress.

3.5.1. INCORPORATING MAXIMUM HYDROSTATIC STRESS DATA

In this section, we investigate a double-task approach to predict the homogenized stress tensor and the maximum hydrostatic stress over the matrix phase. For that, two alternatives were investigated: a joint and a sequential learning approach. In the joint learning approach, we compute an additional loss term related to the maximum task and add it to the term related to the average task as follows

$$\ell_{\text{double}} = \ell_{\text{avg.}} + \underbrace{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\parallel \sigma^{\omega}_{\text{hydro,max}}(\boldsymbol{\varepsilon}_i^{\Omega}) - \widehat{\sigma}^{\omega}_{\text{hydro,max}}(\boldsymbol{\varepsilon}_i^{\Omega})\parallel^2}_{\ell_{\text{max.}}} \tag{3.3}$$

where $\ell_{\text{avg.}}$ is the loss function shown in Eq. (2.21). In this case, both encoder and decoder are trained with information from both tasks. In the sequential approach, encoder and decoder are instead trained separately. First, we train the encoder of the PRNN to learn the Maximum task. Since no trainable decoder is needed for that, in the second stage, the decoder is trained to learn the Average task while keeping the learned encoder from the Maximum task intact. The two learning schemes are illustrated in Fig. 3.18. In the sequential approach, switching the order of the tasks is not possible, since training for the Average task encompasses both encoder and decoder.

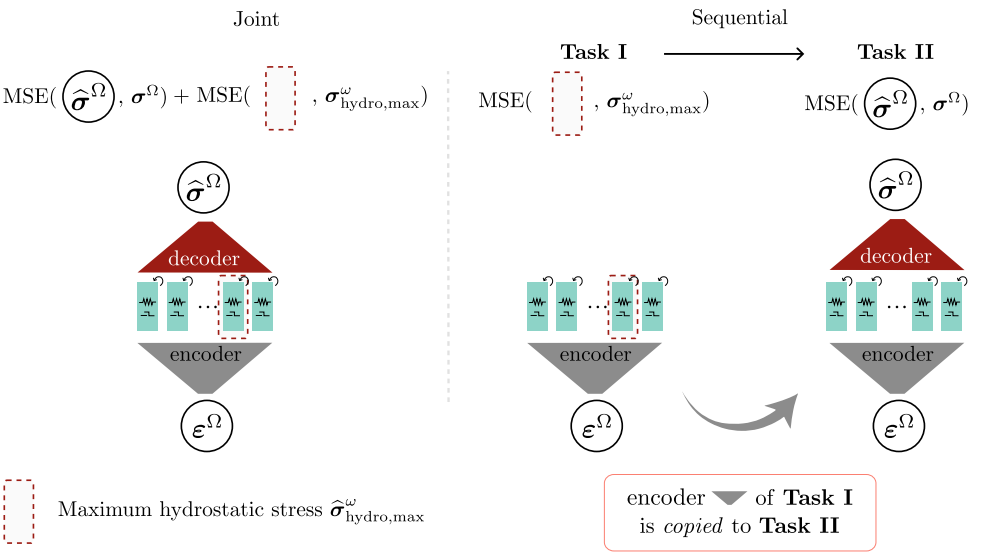


Figure 3.18: Joint vs sequential learning for double-task model.

Fig. 3.19a shows the errors of the two approaches on both tasks. On the Maximum task, there is only a slight gain in jointly learning the two tasks. However, in the second stage of the sequential approach, where the Average task is evaluated, the networks trained jointly show significantly lower errors. Recall that, in the second stage, only the decoder is trainable, limiting the expressivity of the network - especially considering the constraints introduced by the sparsification and weight normalization. The joint learning approach is therefore the preferred approach considering both tasks. To further assess the effectiveness of this strategy, we compare its performance to that of training network independently for each task.

3.5.2. DOUBLE-TASK VS SINGLE-TASKS

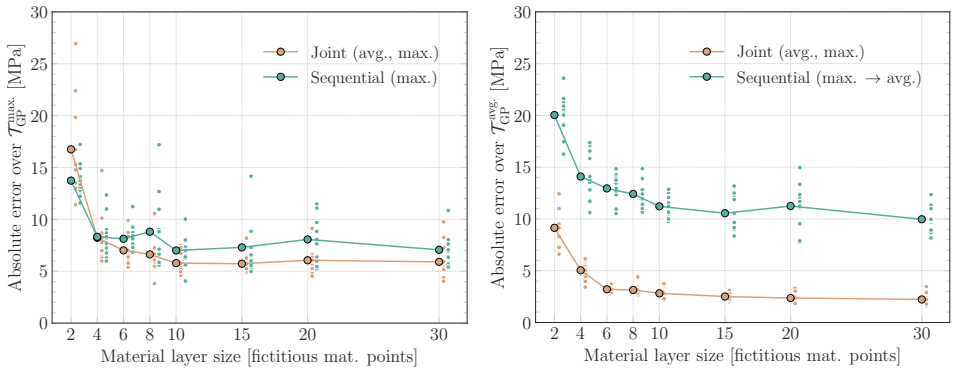
In this section, we refer to the joint learning approach discussed in the previous section as "Double" and compare its performance with the two single-task counterparts ($\ell_{\text{max.}}$ and $\ell_{\text{avg.}}$) to assess the benefits and shortcoming of that strategy. For that purpose, we use the networks trained to learn the Maximum task at the first stage of the sequential

approach shown in Section 3.5.1, here referred to as Single (max.), while the networks trained to learn the homogenized stresses shown in Section 3.3 are referred to as Single (avg.).

In Fig. 3.20a, we plot the 3D landscape of the errors for different training sets and material layer sizes on the Average task. For smaller networks (e.g., 2 and 4 material points), the difference between the single-task and double-task approaches is substantial across all training set sizes. For larger networks, this difference is minimal, and the two surfaces are virtually overlapping. Overall, jointly trained networks tend to trade off their accuracy on the Average task to capture the Maximum task better. Once the network is large enough to capture the complexity of both tasks, this trade-off vanishes, and the performance across both objectives becomes comparable to or better than that of the single-task counterparts.

On the Maximum task, we observe the opposite trend in Fig. 3.20b. This time, the double-task approach yields a more accurate prediction, whereas the networks explicitly trained to predict the maximum hydrostatic stress exhibit slightly higher errors. This suggests that the Maximum task benefits from the additional information on the homogenized stresses, which is particularly useful for bringing the stresses from the latent space closer to the RVE stress distribution. The total error from the two single tasks is summed and shown in Fig. 3.20c for two training set sizes, 2 and 5 curves. With 10 material points and 5 training curves, the total error for both tasks can go as low as 6 MPa. In terms of relative errors, this translates to a 7-10% error in each task, as illustrated in Fig. 3.21.

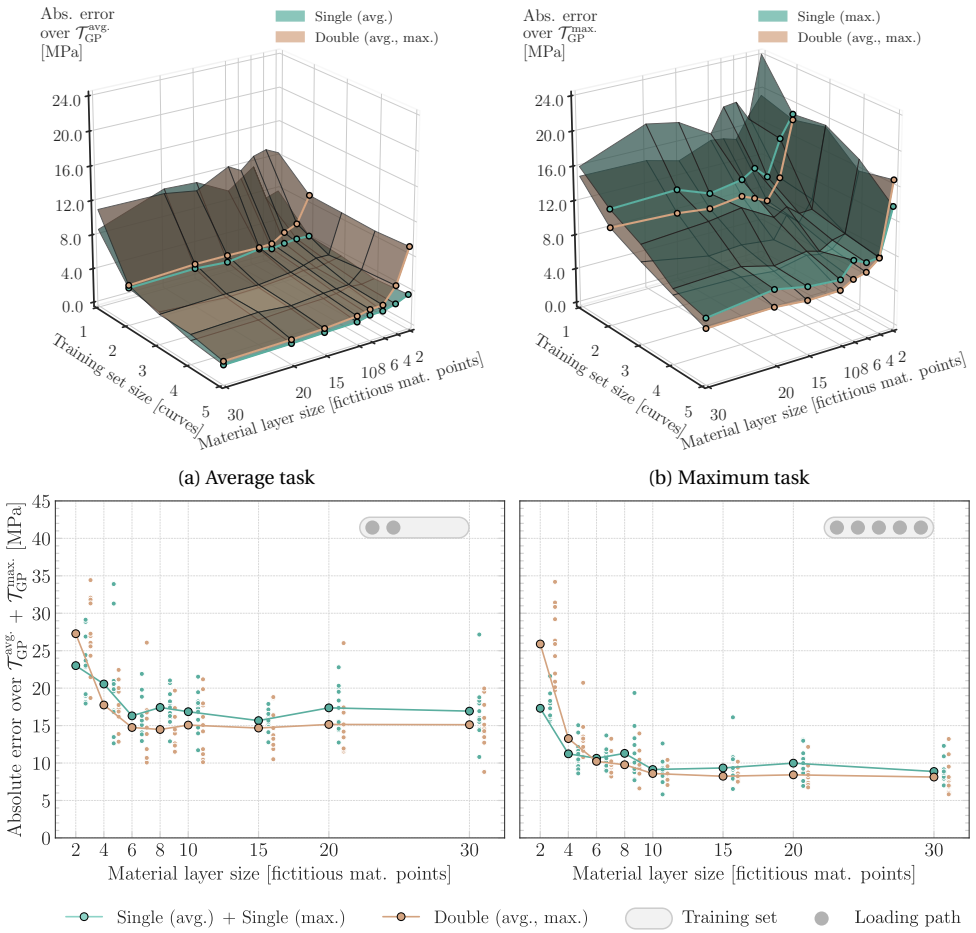
Next, we select the PRNN with the lowest error from the double-task approach to visualize the latent space for one loading path from the test set $\mathcal{T}_{\text{GP}}$. In Fig. 3.22a, we show the hydrostatic stress evolution throughout 60 time steps for each one of the 8 fictitious material points and highlight only the parts where they are maximum. The combination of these parts is plotted in Fig. 3.22b against the true maximum hydrostatic stress from the micromodel.



(a) Errors over Maximum task

(b) Errors over Average task

Figure 3.19: Comparison of PRNNs trained on 5 curves adopting joint vs sequential learning on Maximum and Average tasks over test set $\mathcal{T}_{\text{GP}}$.



(a) Average task

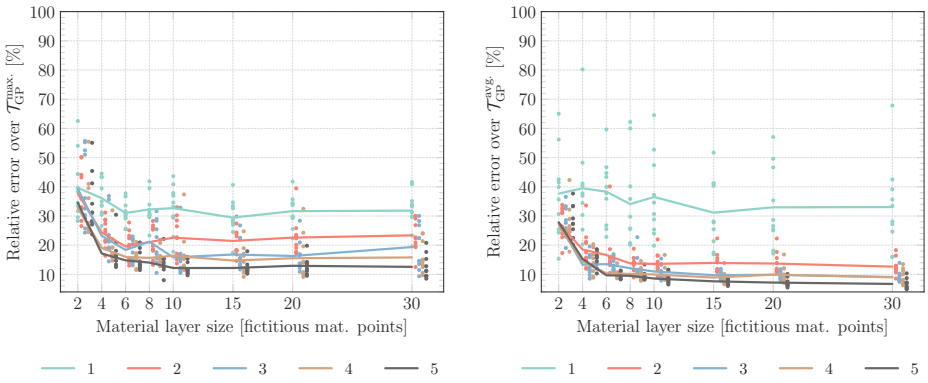
(b) Maximum task

(c) Total error (Maximum + Average) of PRNNs trained on 2 and 5 GP-based curves

Figure 3.20: Test set errors of PRNNs trained for single and double-task and joint learning with limited data sets over the homogenized stress and maximum microscopic hydrostatic stress.

3.6. CONCLUSIONS

In this chapter, different decoder variations for PRNNs with linear encoder and decoder were analyzed, focusing on the impact of weight normalization and sparsity on accuracy and interpretability. We illustrated how the latent space of these networks closely mirrors the RVE state without explicit supervision. Drawing on the physical analogy to RVE volumes, we introduced a constraint that interprets decoder weights as relative volume contributions from fictitious material points. This normalization not only improves interpretability but also enhances robustness and accuracy and is particularly important when training with limited data.



(a) Maximum task

(b) Average task

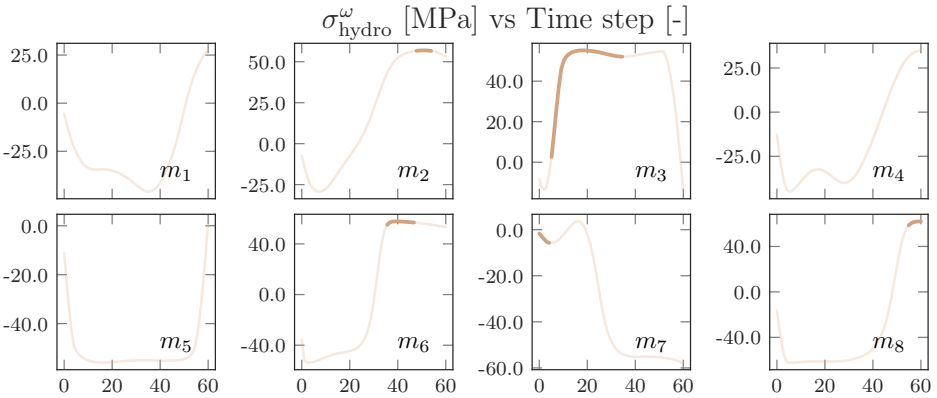
Figure 3.21: Relative error for Maximum and Average tasks of PRNN trained on both homogenized and maximum stress data.

We showed that as few as two training curves are sufficient to consistently achieve homogenized stress prediction errors of around or below 10%, depending on the decoder architecture. While effective in most cases, this constraint can become overly restrictive in configurations where weights are shared per point, thereby limiting flexibility. Beyond the accuracy and robustness gains, normalization serves as a regularizer, eliminating the need for hyperparameter tuning. It also helped align the fictitious stress distribution with the distribution from the RVE, especially given the sparsification. In contrast to other architectures, sparser architectures enable us to disentangle and visualize effects more easily.

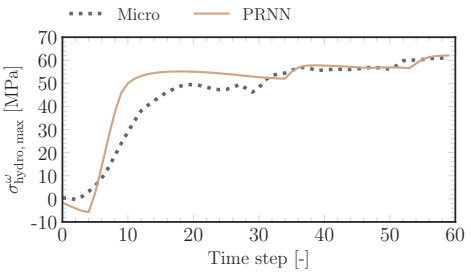
Finally, we demonstrated how a meaningful latent space enables the retrieval of relevant microscopic quantities. In the study case discussed in this chapter, leveraging quantities from the latent space and incorporating them into a joint learning strategy to predict both homogenized and microscopic quantities introduced no additional parameters or architectural complexity. The findings in this chapter emphasize how decoder constraints, sparsity, and physically inspired design choices can improve performance in terms of accuracy and interpretability.

APPENDIX. MATRIX VS FULL MICROMODEL

Fig. 3.23 shows the relative error between the mean and maximum stresses from the fictitious material points of PRNNs with different decoders trained on 5 curves and the true microscopic stresses over two domains: matrix phase only and full micromodel. For both metrics, the difference between the fictitious and the true stresses grows as the networks gains complexity and plateaus around 10%, with the match with the matrix phase quantities being the best overall.



(a) Hydrostatic stresses from the fictitious material points of a PRNN



(b) Unseen GP-based loading path from $\mathcal{T}^{\text{max.}}_{\text{GP}}$

Figure 3.22: Hydrostatic stress from PRNN with 15 material points and trained on 5 GP-based curves on unseen loading paths.

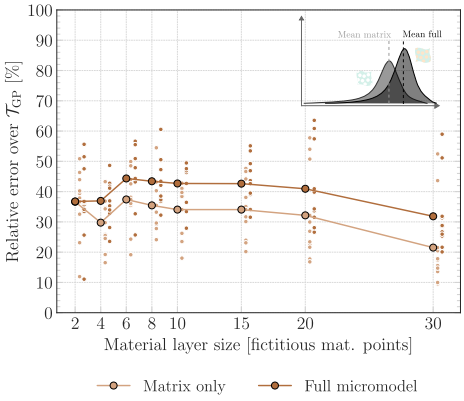
REFERENCES

[1] H. L. Cheung, P. Uvdal, and M. Mirkhalaf. "Augmentation of scarce data — A new approach for deep-learning modeling of composites". *Composites Science and Technology* 249 (2024), 110491. ISSN: 0266-3538. DOI: https://doi.org/10.1016/j.compscitech.2024.110491.

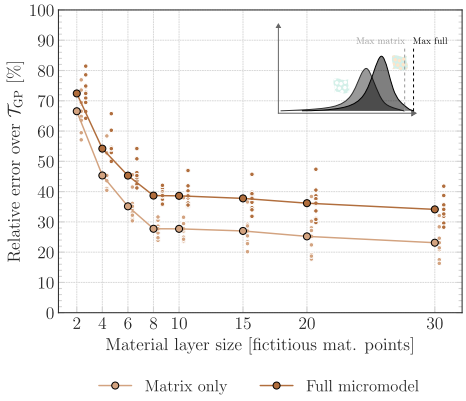
[2] E. Ghane, M. Fagerström, and M. Mirkhalaf. "Multi-fidelity data fusion for inelastic woven composites: Combining recurrent neural networks with transfer learning". *Composites Science and Technology* 267 (2025), 111163. ISSN: 0266-3538. DOI: https://doi.org/10.1016/j.compscitech.2025.111163.

[3] C. Clarijs. "Mechanical performance of glassy polymers: influence of physical ageing and molecular architecture". PhD thesis. Mechanical Engineering, May 2019. ISBN: 978-90-386-4741-8.

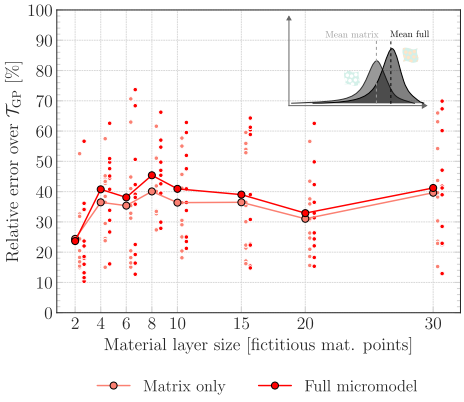
[4] M. Wismans, S. J. J. van den Broek, L. C. A. van Breemen, L. E. Govaert, and T. A. P. Engels. "Micromechanical analysis of age-induced strength reduction in a mul-



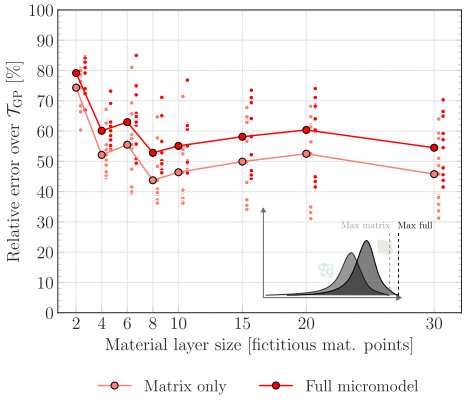
(a) Normalized component - Mean stresses



(b) Normalized component - Maximum stresses



(c) Normalized point - Mean stresses



(d) Normalized point - Maximum stresses

Figure 3.23: Relative error of fictitious stresses from normalized and sparse decoders trained on 5 curves compared to true microscopic stresses over matrix phase and full micromodel.

tiaxially loaded short-fiber reinforced thermoplastic". *Journal of Applied Polymer Science* 140.41 (2023), e54512. DOI: https://doi.org/10.1002/app.54512.

4

DAMAGE MODELS TO CAPTURE DEBONDING AT MICROSCALE

In the previous chapters, PRNNs have been studied in the context of bulk homogenization. This chapter marks our first step towards a formulation that includes damage models into the micromodel. With composite materials in mind, we simulate fiber-matrix interface debonding using interface elements. As a result of the diffuse damage, stiffness degradation is present without macroscopic softening. In the network, this increased complexity brings up new challenges that prompt a series of changes to the original architecture in Chapter 2 to better integrate the new cohesive zone models used to describe the constitutive relations at the interface elements with the remaining (bulk) constitutive models studied thus far.

For coherence with the remaining parts of this thesis, some figures were updated and the introduction was shortened with respect to the published source material:

N. Kovács, M.A. Maia, I. B. C. M. Rocha, C. Furtado, P. P. Camanho, and F. P. van der Meer. "Physically Recurrent Neural Networks for computational homogenization of composite materials with microscale debonding". *European Journal of Mechanics - A/Solids* 112 (2025), 105668. ISSN: 0997-7538. DOI: https://doi.org/10.1016/j.euromechsol.2025.105668

4.1. INTRODUCTION

Much of the recent literature on surrogate modeling focuses on predicting (hyper)elastic or elastoplastic behavior. Surrogate modeling for damage and fracture mechanics applications is a much less explored field. For such cases, some of the critical limitations in conventional surrogate models have only recently started to be unveiled and addressed. For example, Wang and Sun [1] used deep reinforcement learning techniques to create traction-separation laws, but did not apply the model in a multiscale setting. In the works of Liu [2, 3], a deep material network (DMN) was developed, which describes the RVE with a network built up from physics-based building blocks.

In [2], debonding effects in the RVE were captured using adaptable cohesive building blocks within the network. In a multi-stage training strategy, one of the phases from the DMN trained for the bulk material is enriched with the interface interaction learned from a second DMN built on top of cohesive building blocks. As a result, the accuracy of the final network is somewhat limited by the original bulk DMN. In [3], this method was extended to localization problems using a cell-division scheme, which overcame the difficulties of selecting the proper RVE size. While these networks excel in extrapolating from linear elastic data to nonlinear and path-dependent behavior, training and online evaluation are not straightforward. These two stages involve different input spaces, and an iterative Newton-Raphson scheme is required for the online stage [2, 3]. Further, a probabilistic machine learning approach using Bayesian regression was proposed in [4] and also applied to active learning of traction-separation relations in a multiscale setting, but this approach was not made suitable for capturing unloading behavior.

In fracture mechanics problems, the computational cost involved with FE (without a multiscale framework) can likewise be prohibitive. A noteworthy approach to deal with that is proposed in [5], where a domain separation strategy is employed to focus the computational power on the fracture region, which requires most of the attention. The domain separation strategy can also be used in multiscale settings, as proposed by Oliver *et al.* [6]. To further reduce the number of sampling points, it is combined with another key technique based on model order reduction, specifically the Reduced Optimal Quadrature (ROQ). However, these methods are highly dependent on snapshots, and the complexity of the problem increases with nonlinear and path-dependent materials.

In this chapter, we build on the PRNN developments in Chapter 2, where we incorporate knowledge of classical constitutive modeling into a neural network for the bulk homogenization of path-dependent heterogeneous materials. Here, we present a key extension to that framework to account for microscale damage. The study is restricted to diffuse damage in the form of microscale debonding in composite materials. The aim is to describe the stiffness degradation resulting from diffuse damage without the occurrence of macroscopic softening.

A brief discussion on FE^2 is presented in Section 4.2, this time focusing on the considerations needed to account for interface elements, followed by a short introduction to PRNNs. In Section 4.3, the details on the data generation for training and testing the networks are illustrated. In Section 4.4, the performance of the existing PRNN on cases with stiffness degradation is tested, motivating the development of new architectures introduced in Section 4.5 and assessed in Section 4.6. Finally, Section 4.7 presents the main conclusions of this study.

4.2. THEORETICAL BACKGROUND

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

4.2.1. THE FE^2 METHOD

Computational homogenization with the FE^2 method allows for capturing the response of composite materials, for cases where it is difficult to do that on the macroscale due to the complex interaction between nonlinear constituents and microstructural geometry. In this approach, the structure is discretized as a homogeneous macrostructure, where a heterogeneous micromodel is nested into each macroscopic integration point of it [7–9]. The micromodel is assumed to be a representative volume element (RVE). The macroscopic strain values are downscaled as boundary conditions for the micromodel, where the microscopic boundary value problem (BVP) is solved. The microscopic stress values obtained from the BVP are then upscaled back to the macromodel after a homogenization operation. This bypasses the need for any assumptions on the constitutive relation at the macroscale.

The schematics of the FE^2 method is shown in Fig. 4.1. 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. Here, we simulate damage at the fiber-matrix interface, which precludes global softening of the micromodel and avoids the need for inserting a discontinuity on the macroscale. The discontinuity in the microscopic domain is denoted by Γ_d^ω. At the fracture surface, the two opposite sides of the crack are differentiated by a + and a − sign.

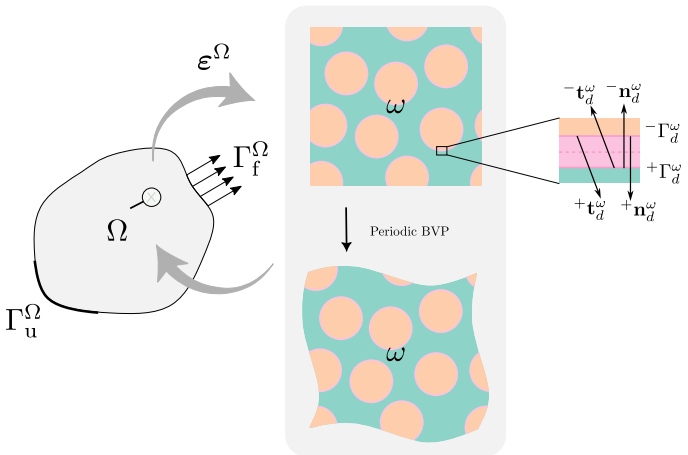


Figure 4.1: FE^2 framework with detailed zoom on interface element.

The displacement field in the micromodel with domain denoted as ω is approximated for the boundary conditions imposed from the macroscale with a finite element discretization of the RVE geometry. The macroscale strain $\boldsymbol{\varepsilon}^\Omega$ is considered to be constant

over the volume aside from a periodic microscale fluctuation field due to the assumption of separation of scales. Nonlinearity in the microscale problem comes from the constitutive models $\mathcal{D}^\omega$ and $\mathcal{T}^\omega$:

$$\boldsymbol{\sigma}^\omega, \boldsymbol{\alpha} = \mathcal{D}^\omega\left(\boldsymbol{\varepsilon}^\omega, \boldsymbol{\alpha}^{t-1}\right) \tag{4.1}$$

$$\mathbf{t}_d^\omega, d = \mathcal{T}^\omega\left([\![\mathbf{u}^\omega]\!], d^{t-1}\right), \tag{4.2}$$

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

After the computation of the full-order solution at the microscale, the resulting stress field is homogenized and returned to the macroscale model. For the accurate coupling between the two scales, the energy between them must be consistent. This micro-to-macro scale transition is usually derived from the Hill-Mandel condition [10], which postulates that the volume average of the variation of work performed on the RVE must be equal to the variation of local work on the macroscale. Based on this condition and on the principle of separation of scales, one can use the superimposition of a microscopic fluctuation field to the homogeneous strain at the microscale, along with the use of periodic boundary conditions so that the average work of the microfluctuations vanish, to obtain the following expression of the homogenized stress

$$\boldsymbol{\sigma}^\Omega = \frac{1}{|\omega|}\int_\omega \boldsymbol{\sigma}^\omega \, d\omega. \tag{4.3}$$

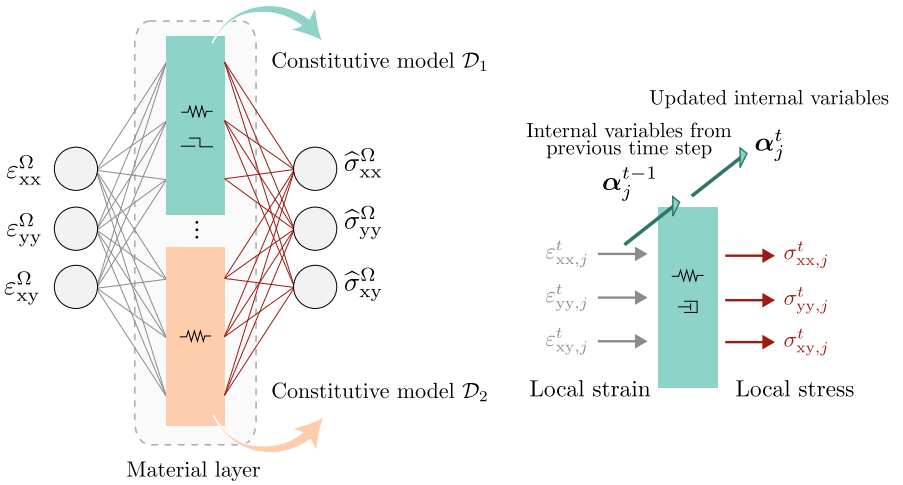
To solve the macroscale problem stress update, a nonlinear finite element solution procedure (e.g. based on the Newton-Raphson method) is needed, making the cost of solving the microscale BVP at each integration point and every macroscale iteration prohibitive for practical applications.

4.2.2. PHYSICALLY RECURRENT NEURAL NETWORK

To tackle the issues related to the black-box nature of neural networks, Physically Recurrent Neural Networks (PRNNs) introduce a new way of embedding knowledge on the physics of a system in a surrogate model. Unlike in PINNs, where the physical constraints of the problem are incorporated in the loss function [11], in PRNNs the actual material models used in the full-order microscopic BVP are implemented in the hidden layer of the network such that their state variables introduce a physics-based recurrency. Fig. 4.2a displays the PRNN in general terms for the case with two constitutive models on the microscale, $\mathcal{D}_1$ and $\mathcal{D}_2$.

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

$$\boldsymbol{\varepsilon} = \mathbf{W}_1\boldsymbol{\varepsilon}^\Omega + \mathbf{b}_1 \tag{4.4}$$



(a) PRNN architecture

(b) Fictitious material point in material layer

Figure 4.2: Architecture of PRNN with only bulk models.

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, 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 $\mathbf{b_1} = \mathbf{0}$.

These local strains are passed through the material layer, which provides the essence of the physically recurrent neural network. The material layer consists of cells, each with three inputs and three outputs, and possibly internal variables, representing a fictitious integration point. In these points, a classical constitutive model $\mathcal{D}^\omega$ converts the local strains $\boldsymbol{\varepsilon}$ to local stresses $\boldsymbol{\sigma}$. This constitutive model is the exact same model that is used to compute the stress in the integration points of the full-order micromodel. The number of fictitious material points in the layer is a model hyperparameter to be tweaked via model selection.

History dependence is naturally included in the PRNN by storing the internal variables $\boldsymbol{\alpha}$ of each material point, which for example in plasticity can be plastic deformation, as they are computed in the assigned constitutive model. Therefore, path-dependency does not need to be learned from data. This stands in contrast with regular recurrent neural networks, where the evolution of history variables is also learned through additional learnable parameters and standard activation functions (e.g. with LSTM or GRU cells). The operation in the fictitious material point j at time t is shown in Fig. 4.2b and can be described by

$$\boldsymbol{\sigma}_j^t, \boldsymbol{\alpha}_j^t = \mathcal{D}^\omega(\boldsymbol{\varepsilon}_j^t, \boldsymbol{\alpha}_j^{t-1}). \tag{4.5}$$

After the local stresses are computed in the material layer, a decoder is applied to these values. In the particular architecture shown in Fig. 4.2a, the decoder consists of a dense

layer with a SoftPlus activation function applied on the weights.This is done to represent the homogenization process through numerical integration, in which weights are strictly positive. Therefore, the macroscale stress output of the network is obtained by

$$\widehat{\boldsymbol{\sigma}}^{\Omega} = \phi_{\text{sp}}(\mathbf{W}_2)\boldsymbol{\sigma} + \mathbf{b}_2 \tag{4.6}$$

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

During training, the following loss function is minimized:

$$\text{L} = \frac{1}{N}\sum_{t=1}^{N} ||\boldsymbol{\sigma}^{\Omega}(\boldsymbol{\varepsilon}_t^{\Omega}) - \widehat{\boldsymbol{\sigma}}^{\Omega}(\boldsymbol{\varepsilon}_t^{\Omega})||^2 \tag{4.7}$$

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

Because of the history variables $\boldsymbol{\alpha}$, back-propagation in time becomes necessary and stress/strain pairs are grouped in paths (time series). Data and gradient handling is therefore analogous to when training RNNs, with the key difference being that memory in PRNNs is physical and interpretable. It is worth emphasizing that if the material model is implemented with automatic differentiation support, gradients are handled automatically through general-use packages such as pytorch and tensorflow. Otherwise, a detailed implementation of how to incorporate these using finite differences is given in Chapter 2.

In the full-order micromodel used in Chapter 2, J_2 plasticity was used for the matrix and a linear elasticity model for the fibers. In the previous study, the PRNN was able to find a solution with only the elastoplastic model (i.e. the constitutive model used to describe the matrix) in the material layer with no loss of accuracy. The expected linear elastic behavior in the fibers is reproduced in elastoplastic material points when small enough strain values are passed by the encoder and stresses are amplified in the decoder, making one or more matrix material points effectively work as if they were linear elastic fiber points.

4.3. DATA GENERATION

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

4.3.1. FULL-ORDER MICROMODEL

The FE model of the microscale is shown in Fig. 4.3 and consists of 25 periodically arranged fibers with diameter of 5 µm embedded in a matrix to result in a fiber volume fraction of 0.6. This single RVE is ussed to generate all datasets in this study. There are two bulk constitutive models: a plasticity model for the matrix, and a linear elastic model for the fibers. The geometry, mesh, and bulk material properties in the RVE are kept as in Chapter 2 and Chapter 3 except that zero-thickness interface elements are positioned at the fiber-matrix interfaces. Limiting damage to the fiber-matrix interface means that no global failure can take place. Tractions at the interface elements are computed from displacement jumps with the bilinear cohesive zone model (CZM) by Turon *et al.* [12].

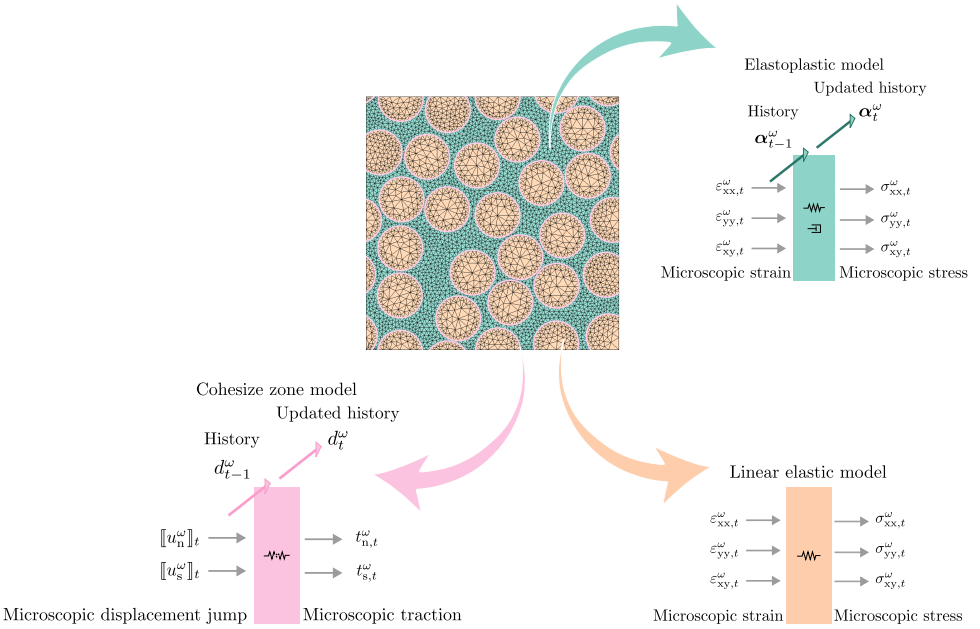


Figure 4.3: Full-order micromodel adopted in this chapter.

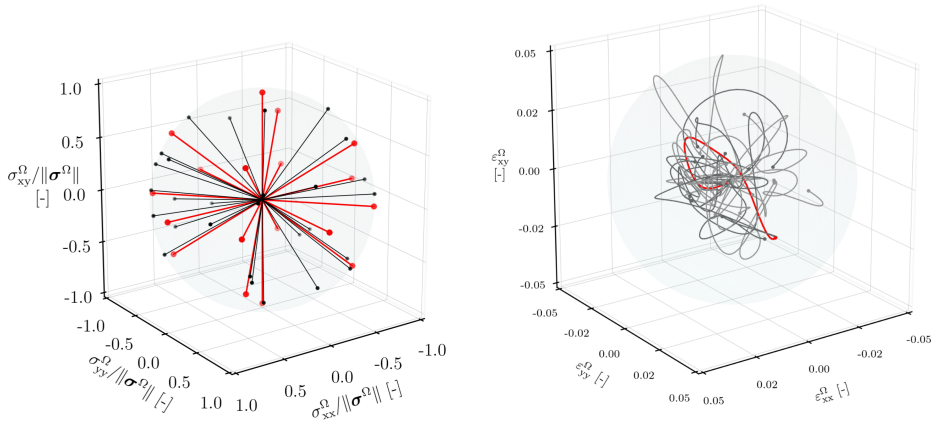
For the CZM properties, we use equal normal and shear strengths $\tau_n^0 = \tau_s^0 = 60$ MPa, mode I and mode II fracture energy $G_{\mathrm{Ic}} = 0.874\,\mathrm{kJ\,m^{-2}}$, $G_{\mathrm{IIc}} = 1.717\,\mathrm{kJ\,m^{-2}}$, mode interaction parameter $\eta = 1$, and penalty stiffness $K = 5 \cdot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

4.3.2. LOAD PATH GENERATION

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

PROPORTIONAL LOADING

For proportional loading, we use a modified arc-length algorithm to enforce the directions of the applied stress, as described in [13]. In this method, the proportionality of the stress response is enforced by considering a constant load vector with increments defined in terms of displacement magnitude, specifically the sum of the unsigned applied displacements. For this study, the increments are fixed at $\Delta s = 1.67 \times 10^{-3}$ mm. The loading directions are categorized as either fundamental or random. The fundamental directions contain 18 common loading cases often used for traditional material model calibration, shown in red in Fig. 4.4a, and include pure tension, compression, shear, biaxial tension, and combinations thereof. On the other hand, the random directions are obtained by sampling three values, each corresponding to a component of the load vector, from a normal distribution $\mathcal{N}(0,1)$ and normalizing them to a unit vector, with examples shown in black in Fig. 4.4a.



(a) Proportional loading path directions

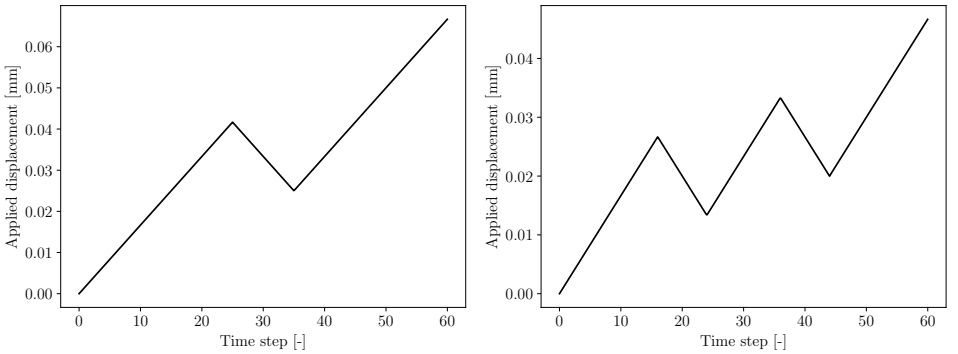
(b) Non-proportional loading paths

Figure 4.4: Types of loading paths considered in this chapter.

In this study, only non-monotonic loading is considered. During non-monotonic loading, the direction in which the step size is kept fixed, but unloading takes place at different loading steps for a predefined amount of time. The loading functions that define the relation between t and the magnitude of loading for non-monotonic cases considered in this chapter are shown in Fig. 4.5. In the arc-length formulation, this corresponds to the imposed value for the unsigned sum of the displacements at the controlling nodes.

NON-PROPORTIONAL LOADING

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



(a) Function with one unloading cycle

(b) Function with two unloading cycles

Figure 4.5: Loading functions used to generate proportional non-monotonic loading curves.

by

$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{4.8}$$

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

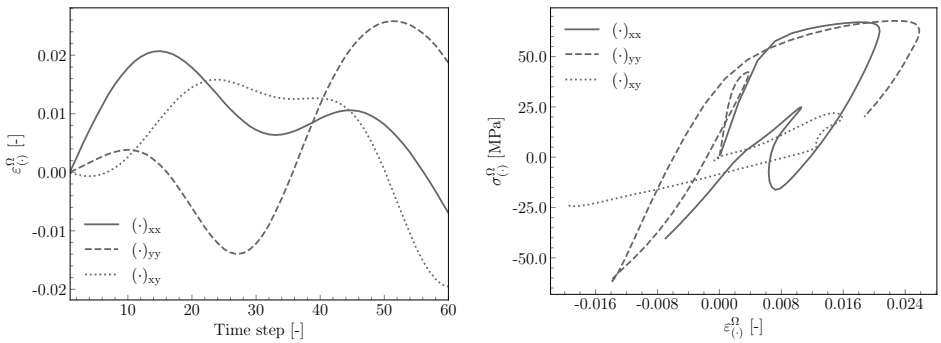
$$\Sigma_{ij} = k(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}||x_i - x_j||^2\right), \tag{4.9}$$

with σ_f^2 being the variance that determines the step size and ℓ being the length scale that controls the smoothness of the generated path. With increased variance σ_f^2 the strains are able to attain larger values, and with increased length scale the path becomes smoother. Values $\sigma_f^2 = 0.0001667$ and $\ell = 200$ are adopted. A subset of the load paths generated by GPs is shown in Fig. 4.4b, with one path highlighted in red for clarity. We also show in Fig. 4.6a the corresponding strain paths for the highlighted loading path and the corresponding stress-strain curves obtained from the full-order micromodel in Fig. 4.6b.

4.4. PERFORMANCE OF PRNN WITH BULK MODEL ONLY

This section investigates whether the PRNN as proposed in Chapter 2 is able to capture stiffness degradation due to microscale damage. The architecture consists of one input layer, one material layer with bulk integration points only, and one output layer, as depicted in Fig. 4.7. All bulk material points embed a J_2 plasticity model to convert 2D local strains to 2D local stresses.

In Chapter 2, the network could find a way to make elastoplastic material points reproduce linear elasticity by appropriately scaling encoder and decoder weights. In the



(a) Strain paths

(b) Stress-strain curves

Figure 4.6: Example of non-proportional GP-based loading path.

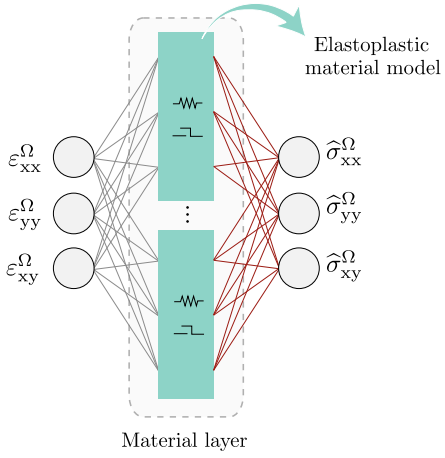


Figure 4.7: PRNN with elastoplastic model only.

following, we demonstrate how such an approach does not work for distributed damage. To highlight this inability to learn as clearly as possible, the networks here are trained and tested on the same curves. Specifically, the non-monotonic, proportional dataset with one cycle of unloading in the 18 fundamental directions is used. Networks with different material sizes are trained by adding bulk points to the network until the mean value of the Mean Squared Errors (MSEs) no longer decreases with additional points. The training MSE across the different material layer sizes is shown in Fig. 4.8, with 10 networks with different initializations per size plotted as blue dots and the purple line representing the mean value for each material layer size. The best performing network with 7 fictitious material points and a training MSE of 4.61 MPa is selected for further examination.

The prediction of the network on two fundamental loading scenarios is shown in Fig. 4.9:

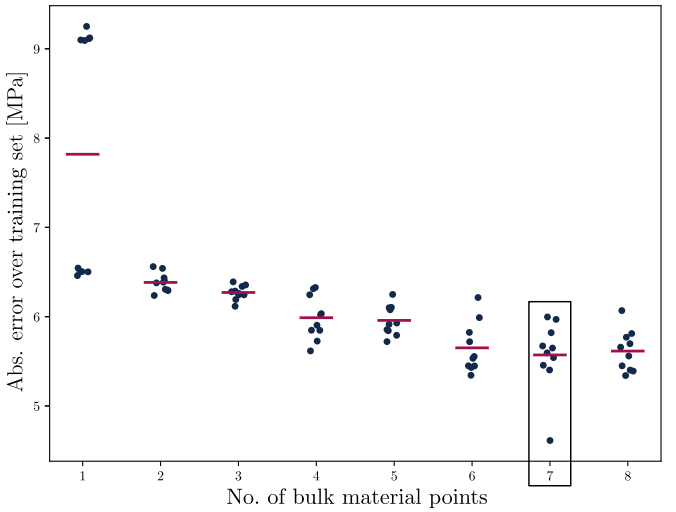


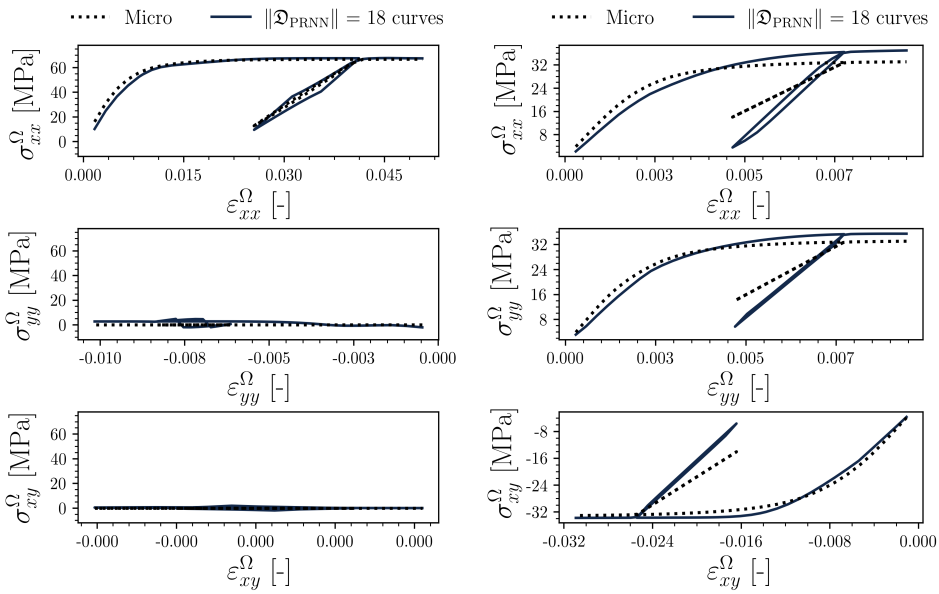
Figure 4.8: Training error for PRNN trained on 18 fundamental curves with one cycle of unloading.

uniaxial tension and biaxial tension with shear. The network provides a somewhat accurate prediction on the monotonic region of the curve, however, the model is unable to reproduce the unloading/reloading region. The PRNN starts to predict unloading with the initial, linear elastic stiffness following the assumptions embedded in the J_2 model, and predicts erratically afterwards. This highlights the limitation of the PRNN for describing stiffness degradation in its original design. The network encodes plasticity through the presence of a plasticity model in the material layer. This design gave the network a good bias in Chapter 2, when it could predict unloading behavior in plasticity without seeing it during training. Here, however, the bias is too strong as it prevents the network from describing the stiffness loss that is present in the micromodel.

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

4.5. EXTENDING THE NETWORK WITH COHESIVE MATERIAL POINTS

As shown in Section 4.4, the physically recurrent neural network cannot accurately predict the effect of debonding at the fiber-matrix interface without including all sources of nonlinearity present in the RVE. Therefore, the cohesive zone model from the full-order micro-model has to be implemented in the PRNN as well. This section details the



(a) Uniaxial load case

(b) Biaxial load case with shear

Figure 4.9: Prediction of PRNN trained with 18 fundamental curves with one cycle of unloading on training curves.

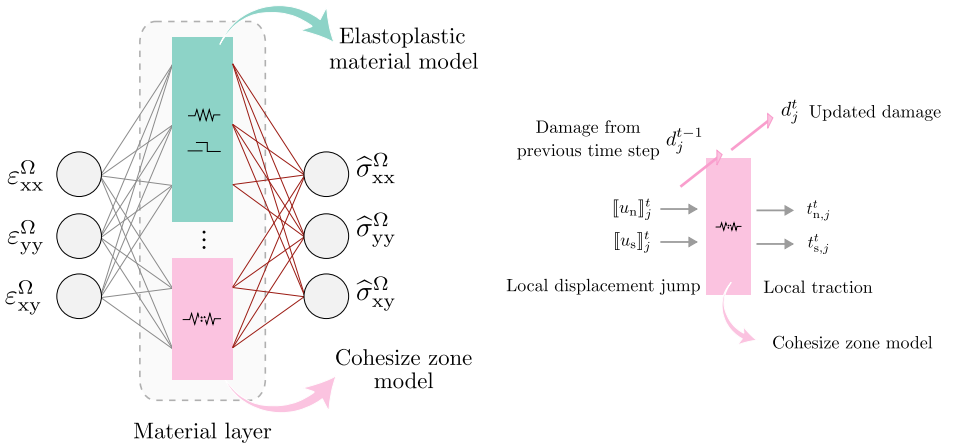
network configurations considered in this study for implementing the CZM within the PRNN framework.

4.5.1. COHESIVE POINTS IN THE EXISTING MATERIAL LAYER

The first design option retains the architecture proposed in Chapter 2 as much as possible. In this design, referred from now on as PRNN_1, there is one material layer containing bulk and cohesive fictitious points. The network is illustrated in Fig. 4.10a with bulk and cohesive points, represented in blue and pink, respectively. The cohesive points relate the local displacement jump vector, with normal and shear components, to a local traction vector, as illustrated in Fig. 4.10b. Similar to the bulk points, the cohesive points also store internal variables to account for history, in this case the damage variable defined as the ratio between dissipated energy and critical energy release rate [14].

4.5.2. COHESIVE POINTS IN SEPARATE LAYER

Instead of having both types of models in the same layer, we also investigate architectures with two material layers: a cohesive and a bulk material layer, each with embedded models, as illustrated in Fig. 4.11. The two architectures considered here consist of one input layer that receives macroscopic strain, two material layers containing the nonlinear models, and one output layer yielding the macro-scale stress predictions. The state



(a) PRNN_1 architecture (b) Fictitious cohesive material point j

Figure 4.10: Architecture with bulk and CZM in the same material layer (PRNN_1) and detail of fictitious cohesive point j.

variables of the cohesive points are densely connected to the bulk points together with the macroscopic strains. To illustrate the connectivity of the layers, Fig. 4.11 highlights how one cohesive point and one bulk point are linked to each other, as well as to the input and output.

Rather than using the output traction values of the cohesive points, damage is used as input to the bulk points. Damage, the internal variable stored in the cohesive points, either increases or remains the same in case of unloading, providing a more monotonic influence on the overall response. This damage variable modifies the local strain value received by the bulk points, resulting in adjusted local stress values for the same level of macroscopic strain. The irreversibility of damage gives rise to a decrease in stiffness during unloading. This design ensures that only the bulk points contribute directly to the stress homogenization procedure, unlike in the architecture described in Section 4.5.1, where tractions rising from the cohesive points are directly connected to the output through the decoder layer. This is more consistent with the homogenization procedure in FE^2, where cohesive tractions do not contribute directly to the macroscopic stress (cf. Eq. (4.3)).

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

$$\boldsymbol{\varepsilon} = \mathbf{W}_{\text{d}} \cdot \mathbf{d} + \mathbf{W}_{\varepsilon\text{b}} \cdot \boldsymbol{\varepsilon}^{\Omega} \tag{4.10}$$

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

In the second method, referred to as PRNN_3 from now on, the damage variables are

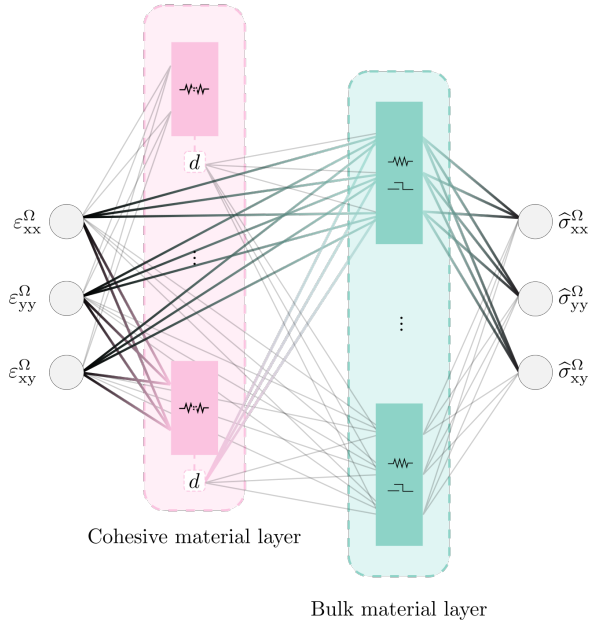


Figure 4.11: Novel architecture PRNN_2 and PRNN_3 with damage as input to bulk point.

used to modify the amplitude of the local strain input to the bulk points. This is achieved by multiplying the local strain input piece-wise by the term $\phi_{sp}(\mathbf{1} + \mathbf{W}_d \cdot \mathbf{d})$, which is forced to attain positive values by applying a SoftPlus activation function ($\phi_{sp}(\cdot)$)

$$\boldsymbol{\varepsilon} = \phi_{sp}(\mathbf{1} + \mathbf{W}_d \cdot \mathbf{d}) \odot (\mathbf{W}_{\varepsilon b} \cdot \boldsymbol{\varepsilon}^{\Omega}). \tag{4.11}$$

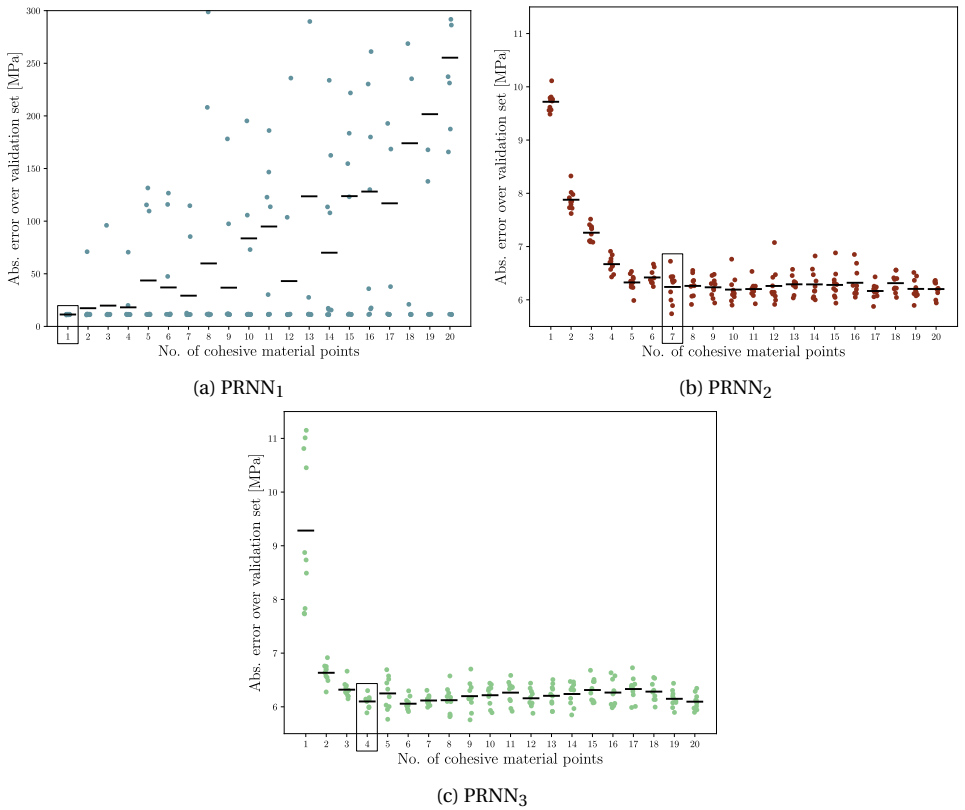
4.6. PERFORMANCE OF PRNN WITH COHESIVE MODEL

In this section, we assess the performance of the PRNNs with the architectures proposed in Section 4.5. The model selection process is presented by analyzing their performance across different training sets and material layer sizes. We evaluate and compare the PRNNs' ability to accurately capture the homogenized response, taking into account microscale damage, under various loading scenarios.

4.6.1. MODEL SELECTION

First we perform model selection for the size of the material layer. For that purpose, networks are trained on 192 GP-based curves (non-monotonic and non-proportional loading) with varying numbers of bulk and cohesive points. The ratio of bulk to cohesive points is kept constant and equal to 4, mirroring the ratio of matrix to cohesive elements in the RVE. The size of the material layer ranges from a minimum configuration of four bulk and one cohesive points to 80 bulk and 20 cohesive points. Figs. 4.12a and 4.12c show the MSE on a validation set with 200 GP-based curves for 10 different initializations in each material layer size, for all three architectures considered in this chapter.

Networks with lowest validation MSE are selected for optimal performance, while prioritizing small networks to avoid overfitting. For $PRNN_1$, networks with 4 bulk points and 1 cohesive point are selected, while for $PRNN_2$ a combination with 28 bulk points and 7 cohesive points is needed. Finally, for $PRNN_3$, 16 bulk and 4 cohesive points are selected. The observation that the validation error increases for increasing network sizes of $PRNN_1$ points at the tendency of this architecture to overfit.



(a) $PRNN_1$ (b) $PRNN_2$

(c) $PRNN_3$

Figure 4.12: Validation error for $PRNN_1$, $PRNN_2$ and $PRNN_3$ trained on 192 GP curves each.

The selected networks are then trained on different training set sizes, ranging from 4 to 192 GP-based curves (non-monotonic and non-proportional loading). Fig. 4.13 displays the MSE on a validation set with 200 GP-based curves across the various training data sizes for the selected material layer sizes of the three architectures studied in this chapter. The solid lines in the figure represent the mean MSE values for each PRNN at different training data sizes. For the first architecture ($PRNN_1$), a training set size of 96 paths is selected. The plateau in Fig. 4.13 indicates that the network in this configuration has reached the limit of its representational power and is too rigid to capture the underlying

physical behavior.

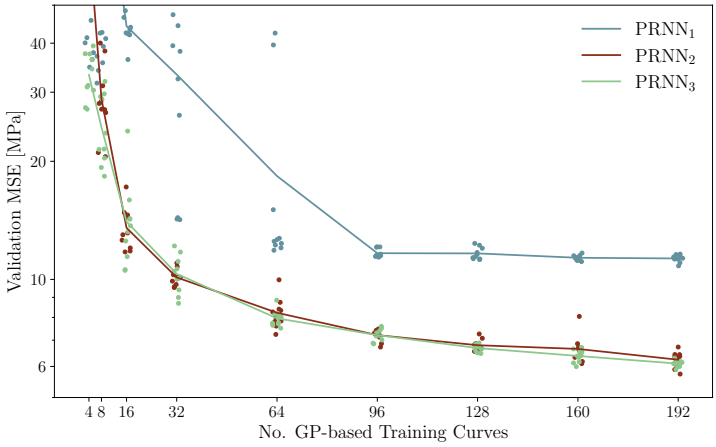


Figure 4.13: Validation MSE for the PRNNs considered across various training data sizes.

On the other hand, $PRNN_2$ and $PRNN_3$ can represent a broader range of material nonlinearity and therefore can reach lower MSE values as dataset size is further increased. Therefore, 192 curves are selected for the latest two architectures. The lowest validation error corresponding to these networks is 11.40 MPa, 5.74 MPa, and 5.89 MPa, achieved with training times of 10, 60, and 20 h respectively.

4.6.2. PREDICTING MICRO-SCALE DAMAGE

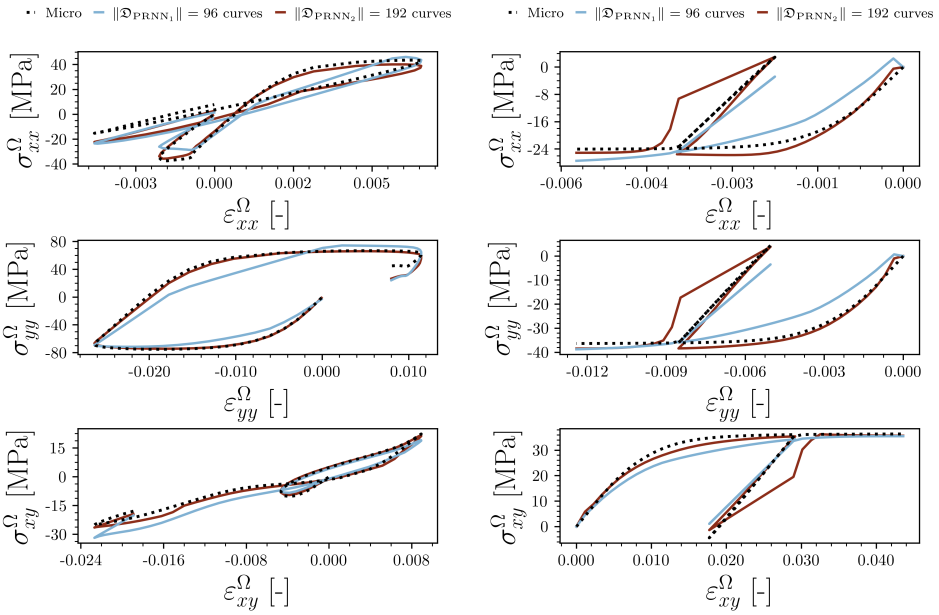
The selected networks are tested on two different datasets: one containing 54 curves from the non-monotonic, non-proportional dataset (the same loading type used for training and validation but in different directions), and another with 54 curves from the random, non-monotonic, and proportional dataset with one cycle of unloading with. The average MSE values on these two test sets are presented in Table 4.1 for each network. It is observed that for the non-monotonic non-proportional test curves, $PRNN_2$ and $PRNN_3$ both outperform $PRNN_1$, with a small difference in accuracy between the two. However, when testing on curves of the proportional type, $PRNN_3$ offers significant additional accuracy over $PRNN_2$.

To illustrate the meaning of these numbers, in the remainder of this section, we compare the performance of the three networks in more detail with stress predictions on individual curves, each time picking representative curves with MSE close to the average MSE from Table 4.1.

First, we illustrate in Fig. 4.14 the performance of $PRNN_1$ and $PRNN_2$. Note how the predictions of $PRNN_1$ on GP-based curves follow the overall trend but with significantly less accuracy compared to $PRNN_2$ (Fig. 4.14a). This observation aligns well with the results shown in Fig. 4.13, where the validation set is also comprised of GP-based curves, emphasizing the significant decrease in validation error when the cohesive points are implemented in a separate layer from the bulk points.

Table 4.1: Average MSE values for the two test datasets, in MPa

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



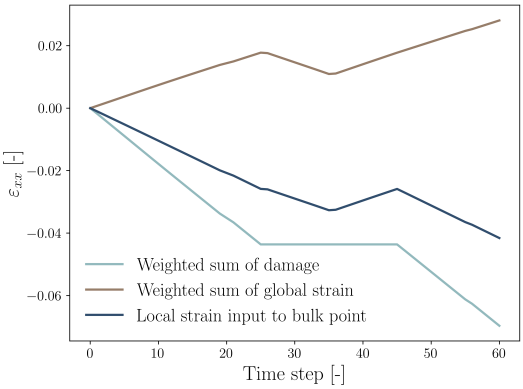
(a) Predictions on GP-based curve

(b) Predictions on curve with one cycle of unloading

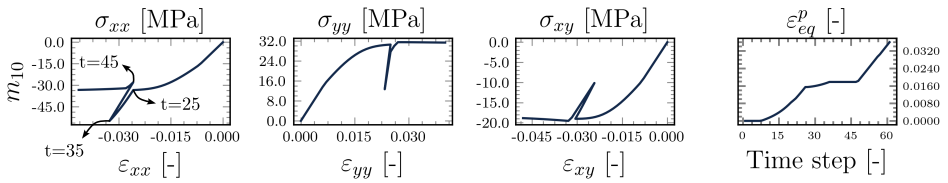
Figure 4.14: Prediction of $PRNN_1$ and $PRNN_2$ on representative test curves.

The difference between $PRNN_1$ and $PRNN_2$ predictions becomes more pronounced when tested on the non-monotonic, proportional dataset. As shown in Fig. 4.14b, $PRNN_1$ predicts poorly. The network not only fails to capture the decrease in stiffness during unloading but also loses accuracy in the monotonic part. It is observed that with $PRNN_1$ there is a preference towards networks with fewer cohesive points. Moreover, small weights connect the normal component of these cohesive points with the output, which is likely due to the large traction values output from the cohesive points in compression. These factors indicate that the network avoids utilizing the cohesive points implemented in the material layer, resulting in unloading with the initial linear stiffness.

Besides the improved accuracy on the test sets, as shown in Table 4.1, $PRNN_2$ shows another advantage. Unloading occurs with a different slope than the initial linear phase (Fig. 4.14b), indicating that the network is able to account for the effect of microscale damage. However, a new problem arises: reloading follows a different path than unload-



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



(b) Response of the bulk point

Figure 4.15: Behavior of one of the bulk points of $PRNN_2$ when predicting on a representative curve from the non-monotonic, propotional test set.

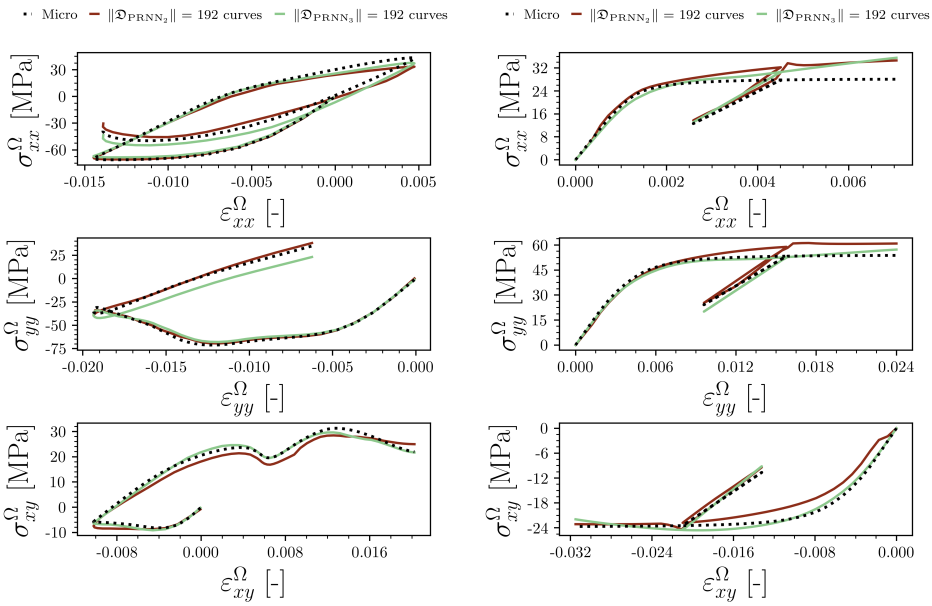
ing. Given the interpretable nature of PRNNs, this phenomenon can be investigated by closely examining the input to one of the bulk points.

Fig. 4.15a illustrates the ε_{xx} component of a particular fictitious bulk material point and its two contributions, one that follows directly from the macroscopic strain ($\mathbf{W}_{\varepsilon b} \cdot \boldsymbol{\varepsilon}^{\Omega}$) and the other that follows from the damage variables from all the cohesive points ($\mathbf{W}_{d} \cdot \mathbf{d}$). Globally, the micromodel is unloading from time step 25, a trend represented by the weighted sum of the global strain values. Meanwhile, the weighted sum of damage is larger than the weighted sum of the global strain and has an opposite sign. Therefore, the final sum used as input to the bulk point prevents the point to follow the global unloading trend (from $t = 25$ to $t = 35$). Instead, the bulk point continues to load while the macroscopic strain is subjected to unloading and only starts unloading once the macroscopic strain is at the reloading branch (from $t = 35$ to $t = 45$). This mismatch between the loading phases leads to further evolution of plastic strain during the macroscopic unloading, as shown in Fig. 4.15b, and causes the undesired change in slopes during unloading and reloading.

When damage is used as an amplifier rather than being simply densely connected to the bulk points, no significant difference in validation errors is found (Fig. 4.13). This is reflected on the prediction of $PRNN_2$ and $PRNN_3$ on a representative GP-based curve in

Fig. 4.16a. However, robustness improves significantly when predicting on curves from the non-monotonic, proportional set. Fig. 4.16b clearly shows that unloading/reloading now takes place along the same path and with a different slope than the initial linear phase, effectively capturing the effect of microscale damage.

To further demonstrate the network's predictive capabilities, Fig. 4.17 shows the prediction of PRNN_3 on a curve from a test set containing non-monotonic, proportional curves with two cycles of unloading. The evolution of damage over time is evident as the slope of the unloading-reloading phase gradually decreases as the loading continues.



(a) GP curve

(b) Curve with one cycle of unloading

Figure 4.16: Prediction of PRNN_2 and PRNN_3 on representative test curves.

4.7. CONCLUSIONS

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

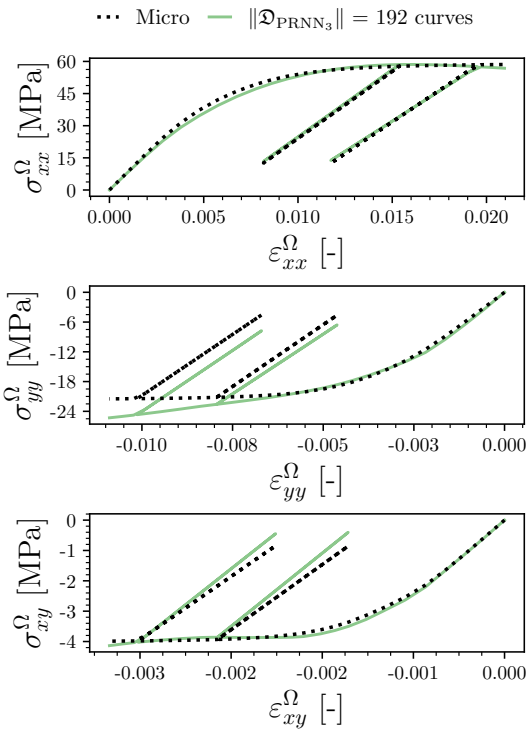


Figure 4.17: Prediction of PRNN_3 on a representative curve with two cycles of unloading.

As a first step to use the PRNN framework for microscale damage, a preliminary study was conducted using the network in its original form with bulk material points only. It was demonstrated that the original PRNN could not describe stiffness degradation, even when trying to overfit on a small set of training curves. These results align well with the general guideline in Section 2.4.2 that all types of nonlinearities present in the RVE need to be included in the network, justifying the need for an extended PRNN architecture that integrates a cohesive zone model.

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

Afterwards, the performance of the proposed PRNNs was evaluated. The three networks were trained with data from non-monotonic, non-proportional (GP-based) curves

and tested on the same type of curves, and on proportional, non-monotonic curves with one cycle of unloading.

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

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

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

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

REFERENCES

[1] K. Wang and W. Sun. "Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning". *Computer Methods in Applied Mechanics and Engineering* 346 (2019), 216–241. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2018.11.026.

[2] Z. Liu. "Deep material network with cohesive layers: Multi-stage training and interfacial failure analysis". *Computer Methods in Applied Mechanics and Engineering* 363 (2020), 112913. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.112913.

[3] Z. Liu. "Cell division in deep material networks applied to multiscale strain localization modeling". *Computer Methods in Applied Mechanics and Engineering* 384 (2021), 113914. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.113914.

[4] 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". *Journal of Computational Physics: X* 9 (2021), 100083. ISSN: 2590-0552. DOI: https://doi.org/10.1016/j.jcpx.2020.100083.

[5] P. Kerfriden, O. Goury, T. Rabczuk, and S. Bordas. "A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics". *Computer Methods in Applied Mechanics and Engineering* 256 (2013), 169–188. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2012.12.004.

[6] J. Oliver, M. Caicedo, A. E. Huespe, J. A. Hernández, and E. Roubin. "Reduced order modeling strategies for computational multiscale fracture". *Computer Methods in Applied Mechanics and Engineering* 313 (2017), 560–595. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.09.039.

[7] J. Schröder. "A numerical two-scale homogenization scheme: the FE2-method". In: *Plasticity and Beyond: Microstructures, Crystal-Plasticity and Phase Transitions*. Ed. by J. Schröder and K. Hackl. Vienna: Springer Vienna, 2014, 1–64. ISBN: 978-3-7091-1625-8. DOI: 10.1007/978-3-7091-1625-8_1.

[8] F. Feyel and J.-L. Chaboche. "FE2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials". *Computer Methods in Applied Mechanics and Engineering* 183.3 (2000), 309–330. ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(99)00224-8.

[9] F. Feyel. "A multilevel finite element method (FE2) to describe the response of highly non-linear structures using generalized continua". *Computer Methods in Applied Mechanics and Engineering* 192.28 (2003). Multiscale Computational Mechanics for Materials and Structures, 3233–3244. ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(03)00348-7.

[10] R. Hill. "Elastic properties of reinforced solids: Some theoretical principles". *Journal of the Mechanics and Physics of Solids* 11.5 (1963), 357–372. ISSN: 0022-5096. DOI: https://doi.org/10.1016/0022-5096(63)90036-X.

[11] M. Raissi, P. Perdikaris, and G. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". *Journal of Computational Physics* 378 (2019), 686–707. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.10.045.

[12] A. Turon, P. P. Camanho, J. Costa, and C. G. Dávila. "A damage model for the simulation of delamination in advanced composites under variable-mode loading". *Mechanics of Materials* 38.11 (2006), 1072–1089. ISSN: 0167-6636. DOI: https://doi.org/10.1016/j.mechmat.2005.10.003.

[13] I. B. C. M. Rocha, P. Kerfriden, and F. P. van der Meer. "Micromechanics-based surrogate models for the response of composites: A critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks". *European Journal of Mechanics, A/Solids* 82 (2020), 103995. ISSN: 09977538. DOI: 10.1016/j.euromechsol.2020.103995.

[14] A. Turon, E. González, C. Sarrado, G. Guillamet, and P. Maimí. "Accurate simulation of delamination under mixed-mode loading using a cohesive model with a mode-dependent penalty stiffness". *Composite Structures* 184 (2018), 506–511. ISSN: 0263-8223. DOI: https://doi.org/10.1016/j.compstruct.2017.10.017.

5

ANISOTROPIC VISCOPLASTIC 3D MODELS UNDER FINITE STRAINS

Previously on Chapter 2, PRNNS were demonstrated to work for micromodels with rate-independent plasticity in 2D problems, capturing loading-unloading behavior without seeing it during training. In this chapter, their applicability is extended to capture time-dependent material behavior. For this purpose, we consider micromodels where the matrix is described by the Eindhoven Glassy Polymer (EGP) model, an advanced elasto-viscoplastic material model for polymers. However, since the constitutive models considered in this chapter are built on top of a 3D finite strain framework, including the EGP, a new architecture suitable for this setting is required. We show how these new and non-trivial features are incorporated into the network and demonstrate that the benefits of the PRNN approach successfully transfer to a much more complex class of constitutive models.

Apart from the shortened introduction and the addition of an appendix with details of the updated Lagrangian formulation, this chapter was integrally extracted from the following publication:

M. A. Maia, I. B. C. M. Rocha, D. Kovačević, and F. P. van der Meer. "Physically recurrent neural network for rate and path-dependent heterogeneous materials in a finite strain framework". *Mechanics of Materials* 198 (2024), 105145. DOI: https://doi.org/10.1016/j.mechmat.2024.105145

5.1. INTRODUCTION

As discussed in Chapter 1 and Chapter 2, in applications involving path-dependent materials, variations of Recurrent Neural Networks (RNN) are the most popular choice of surrogate model. However, other strategies built on Gaussian Processes (GPs) and dimensionality reduction techniques (*e.g.* Proper Orthogonal Decomposition (POD) and Hyper-reduction methods) have also showed potential for reducing the computational costs [1–4]. A pressing issue common in purely data-driven models is their limited ability to extrapolate. This is usually tackled with ever larger training sets and intricate design of experiments that aim to uniformly/densely cover the space of strain paths. A complicating aspect is the curse of dimensionality. Recent works [5, 6] illustrate the hurdles with predicting loading types different from the ones used for training, particularly with RNNs, and how to overcome them with transfer learning strategies.

Another challenge of NNs lies in their black-box nature. One way to address this is to incorporate physics knowledge into the model. Guided by this philosophy, Physics-Informed Neural Networks (PINNs) are likely the most prominent example. Although these networks have been initially designed to solve partial differential equations, the idea of enriching the loss function with extra terms to enforce physics constraints has quickly found its way into the material modeling community [7, 8]. Another way to leverage physical consistency in NNs is to encode the physical knowledge directly in the architecture design [9–11].

Moving away from the recurrency mechanisms in RNNs, transformers rely on self-attention mechanisms to extract correlations among the elements within a sequence. These models have shown improved performance in comparison to other state-of-the-art methods in capturing long-range dependencies in language processing problems [12], but have only recently been applied in the computional homogenization field to predict the response of composite materials with elastoplastic behavior [13, 14]. Beyond the positive assessment in terms of accuracy, a common thread in these works encompass the need of very large datasets (ranging from dozens to hundreds of thousands of curves), the difficulty of training models with millions of parameters and the critical scaling of computational memory space required for both the offline and online phases as the sequence length increases.

When dealing with materials with time-dependency, as it is the case in this chapter, the extra dimensionality related to strain-rate sensitivity adds a new depth to the problem. For clarity, we distinguish time or rate-dependency from path-dependency as the former refers to behavior that is dependent on the duration and speed of the loading, while the latter refers to behavior dependent on the loading sequence and history. In a broader sense, both are framed as history-dependent. In some works, the strain-rate [15] and/or the time increment [16] have been explicitly included in the feature space. In others, a fixed time increment is considered [17, 18].

In [19], a forward Euler discretization was employed to make the stress prediction independent from the time discretization using two feed-forward NNs. The first model learns the rate of change of a set of internal variables learned from the data based on the current strain and the previous set of internal variables, while the second predicts the stress based on the current strain and the internal variables learned by the first NN. In a follow-up work [20], the authors explore how iterated learning can help improve the ac-

curacy of these models in multiscale applications through the inclusion of strain-stress curves extracted from a macroscopic problem of interest, as well as their transferability to other problems.

In [21], a framework based on the dual potential function to describe rate-dependent viscoplastic flow response in metals. The authors take advantage of input-convex NNs to enforce thermodynamic consistency and leverage automatic differentation to compute gradients of the output with respect to the inputs, which are used for solving the implicit time-stepping algorithm employed in their elasto-viscoplastic model. Nevertheless, the method is not suitable for FE simulations yet as arbitrary loading and boundary conditions can take place and only uniaxial deformations were considered.

In all of these works, to train a surrogate for rate-dependence the training data needs to account not only for a good coverage of strains but also strain-rates. To deal with time-dependency in a more seamless manner, we propose to expand the applicability of PRNNs. Previously, the PRNN was demonstrated to work for micromodels with rate-independent plasticity, capturing loading-unloading behavior without seeing it during training. It is anticipated that the same approach can capture rate-dependence. In this paper, we apply the PRNN approach to micromodels where the polymer matrix is described with the Eindhoven Glassy Polymer (EGP) model, an advanced elasto-viscoplastic material model. For this purpose, the following features are added with respect to Chapter 2: 1) time-dependent material behavior, 2) a finite strain formulation, and 3) generalization to 3D space.

A brief description of the microscale analysis based on a finite strain formulation is presented in Section 5.2. In the following, the main changes with respect to the design in Chapter 2, designed and tested for small strains, are discussed in Section 5.3. In Section 5.4, the design of experiments considered for training and testing the network is described, while the numerical applications are organized in two sections. Firstly, in Section 5.5, the accuracy of the network is assessed in a set of numerical experiments over a range of loading scenario based on different training strategies (monotonic vs non-mononotic paths). This chapter is complemented by a brief runtime comparison in Section 5.6 to illustrate the speedup potential of these models. Secondly, in Section 5.7, the network directly replaces the micromodel in the solution of three equilibrium problems, including different strain-rates to cyclic loading and relaxation. Finally, in Section 5.8, the main conclusions from this chapter are presented.

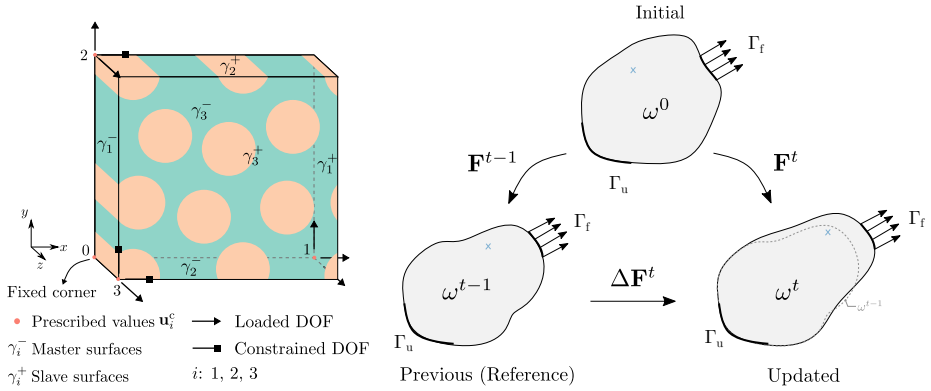
5.2. MICROSCALE ANALYSIS

This chapter focuses on the homogenized behavior of a RVE of a microscopic material with both path and time-dependency. For notation purposes, the superscripts Ω and ω refer to the homogenized (macroscopic) and microscopic quantities, respectively. Let ω denote the RVE domain and consider that periodic boundary conditions (PBC) are applied to simulate the behaviour of a macroscopic bulk material point, as depicted in Fig. 5.1a. In the absence of body forces, the updated Lagrangian formulation, illustrated

in Fig. 5.1b, can be defined by the weak statement of equilibrium:

$$\underbrace{\int_{\omega} \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma} \, d\omega}_{\mathbf{f}^{\mathrm{int}}} - \underbrace{\int_{\Gamma_{\mathrm{u}}} \mathbf{N}^{\mathrm{T}} \mathbf{t}_{\mathrm{p}} \, d\Gamma}_{\mathbf{f}^{\mathrm{ext}}} = \mathbf{0} \tag{5.1}$$

where $\mathbf{N}$ is a matrix with the shape functions used to interpolate the nodal displacements $\mathbf{a}$, $\mathbf{B}$ is the strain-displacement matrix with the gradients of the shape functions with respect to the current coordinates $\mathbf{x}$, $\boldsymbol{\sigma}$ is the Cauchy stress, $\mathbf{t}_{\mathrm{p}}$ are the tractions prescribed on the boundary of the domain Γ_{f}.



(a) Boundary conditions of periodic micromodel ω with constrained dofs to avoid rigid-body motion

(b) Initial (undeformed), previously converged (reference) and updated configurations

Figure 5.1: Micromodel and scheme of configurations used in the updated Lagrangian framework.

With the domain discretized in a Finite Element (FE) mesh, the displacements at the nodal values, known as the degrees of freedom (DOF), are used to describe the displacement field of the micromodel $\mathbf{u} = \mathbf{N}\mathbf{a}$. In this method, Eq. (5.1) is solved iteratively as

$$\mathbf{r} = \mathbf{f}^{\mathrm{int}} - \mathbf{f}^{\mathrm{ext}} = \mathbf{0} \tag{5.2}$$

where $\mathbf{r}$ is the residual vector that vanishes once equilibrium is reached. The iterative procedure involves linearization of $\mathbf{f}^{\mathrm{int}}$ with respect to the DOF vector. In the geometrically nonlinear formulation, this linearization requires accounting for the dependence of $\mathbf{B}$ from Eq. (5.1) on the displacements through a geometric contribution to the stiffness matrix.

The stress in Eq. (5.1) is related to the deformations with a constitutive model $\mathcal{C}^{\omega}$, which, in general, can be described by

$$\boldsymbol{\sigma}, \boldsymbol{\alpha} = \mathcal{C}^{\omega} \left(\mathbf{F}, \boldsymbol{\alpha}^{t-1}, \Delta t \right) \tag{5.3}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^{t-1}$ are the history variables that account for path and rate-dependency at the current and previous time step, Δt is the time increment and $\mathbf{F}$ is the deformation

gradient

$$\mathbf{F} = \mathbf{I} + \nabla \mathbf{u} \tag{5.4}$$

where $\nabla \mathbf{u}$ represents the gradient of the microscopic displacements. Since the deformation gradient is calculated with respect to the initial configuration, its increment can also be easily computed from the current and previous deformation states

$$\Delta \mathbf{F} = \mathbf{F}\mathbf{F}_{t-1}^{-1}. \tag{5.5}$$

For rate-dependent materials, the stress depends on $\Delta\mathbf{F}$ as well as $\mathbf{F}$, which can be achieved with Eq. (5.3) if $\mathbf{F}_{t-1}$ is included in the material history $\boldsymbol{\alpha}$. Upon convergence, the homogenized stresses can be averaged out by integrating the microscopic stresses over the volume ω:

$$\boldsymbol{\sigma}^{\Omega} = \frac{1}{|\omega|} \int_{\omega} \boldsymbol{\sigma} \, \mathrm{d}\omega. \tag{5.6}$$

For completeness, we refer the interested reader on the computational homogenization using the updated Lagrangian formulation to Appendix A. In that material, we cover the transformations among configurations, present a brief proof of the Hill-Mandel condition at the updated configuration, and discuss the homogenized quantities used for equilibrium and material modelling in this work, including the choices made for the neural network discussed in the following.

5.2.1. CONSTITUTIVE MODELS

In this chapter, we consider a composite micromodel made of unidirectional fibers embedded in a matrix material. To describe the constitutive behavior of the matrix, the EGP model is used, while for the fibers, a hyperelastic transversely isotropic model is assigned. These consist of the same choices adopted in [22], where a thorough validation of the material models was carried out for a carbon/PEEK composite material. Here, we only highlight the main aspects of their formulation and focus on how to incorporate them in a PRNN.

The fiber constitutive law is based on the one developed by Bonet and Burton [23] with slight modifications [22]. The constitutive model derives from the strain energy density function and can be split into two components, an isotropic part with a neo-Hookean potential and a transversely isotropic part, with both depending on the right Cauchy-Green deformation tensor

$$\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}. \tag{5.7}$$

The EGP model for the matrix material consists of a rate and path-dependent elasto-viscoplastic, isotropic, 3D constitutive law. In this model, no explicit yield surface is needed since an Eyring-based viscosity function evolves with the stress applied, leading to the viscoplastic flow of the material. The Cauchy stress calculated by the EGP is composed of three contributions: hydrostatic, hardening and driving stress. While the first two parts are defined in more simple terms as they do not depend on the internal variables, in the third part, where viscoplasticity is introduced, a further decomposition can be considered. In this case, the multiple contributions to the driving stress correspond to different molecular (relaxation) processes. Each relaxation process is represented with

a series of Maxwell models (modes) connected in parallel, with a shear modulus in the elastic spring and a stress-dependent viscosity in the dashpot. Here, a single relaxation process is considered and represented with 1 mode.

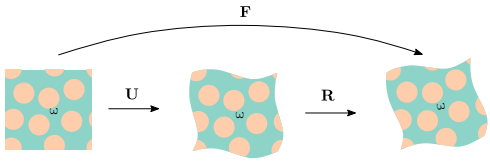


Figure 5.2: Right polar decomposition on deformation gradient **F** resulting in the stretch and rotation tensors **U** and **R**, respectively

For any of the models discussed so far, a helpful tool to deal with the high-dimensionality of the deformation gradient is the polar decomposition theorem. The theorem states that any deformation gradient **F** can be uniquely decomposed into the product of two other tensors: a symmetric one **U** and an orthogonal one **R**, as $\mathbf{F} = \mathbf{RU}$. These two tensors have physical interpretations and are closely related to the principle of material objectivity or material frame indifference. In short, the symmetric tensor represents the deformation (*i.e.* stretches and shear) and the orthogonal tensor represents a rigid body rotation. When applied in this sequence, the final configuration obtained is the same as the one obtained if the deformation gradient was applied directly.

The particular order of stretch and rotation is known as right polar decomposition and is illustrated in Fig. 5.2. From these interpretations and considering the principle of material frame indifference, which states the material response is independent of the observer, one can rewrite stresses as

$$\boldsymbol{\sigma}_{\mathrm{U}}, \boldsymbol{\alpha} = \mathcal{C}^{\omega}\left(\mathbf{U}, \boldsymbol{\alpha}^{t-1}, \Delta t\right) \tag{5.8}$$

$$\boldsymbol{\sigma}_{\mathrm{F}} = \mathbf{R}\,\boldsymbol{\sigma}_{\mathrm{U}}\,\mathbf{R}^{\mathrm{T}} \tag{5.9}$$

where $\boldsymbol{\sigma}_{\mathrm{U}}$ are the unrotated stresses and $\boldsymbol{\sigma}_{\mathrm{F}}$ are the stresses in the original frame of reference.

5.3. PHYSICALLY RECURRENT NEURAL NETWORK

In this section, we present the new architecture of the Physically Recurrent Neural Network (PRNN) to be used in a 3D finite strain framework for micromodels with path and rate-dependent behavior. Having the network in Chapter 2 as the starting point, we highlight and motivate the main changes in comparison to the 2D formulation. In that work, a NN composed of a data-driven encoder, a material layer with embedded physics-based material models and a data-driven decoder is proposed. The data-driven parts learn how the homogenized strain can be dehomogenized and distributed among a small set of fictitious material points and how the stress obtained in these material points can be homogenized again, respectively. With the same core idea, we propose a set of modifications to extend such model to the current application.

Before diving into the details of the novel architecture, training aspects and its use as a constitutive model, we highlight an important change in its input with respect to the 2D formulation in Chapter 2. Here, the surrogate is trained to learn the mapping from stretch (path) to unrotated stress (Eq. (5.8)) and let it be embedded between decomposition and rotation operations to recover the stress in the original frame, as illustrated in Fig. 5.3, instead of mapping deformation gradient to rotated stress directly. With this, the dimensionality of the feature space of the PRNN is reduced from 9 to 6 independent components, alleviating the sampling effort required for training.

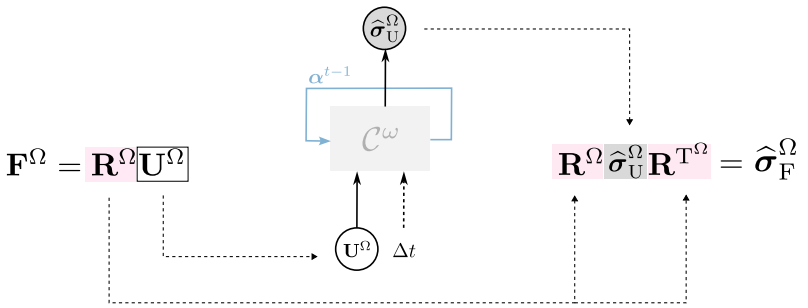


Figure 5.3: Use of PRNN in a general full-order solution setting with $\mathbf{F}^{\Omega}$ and $\widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{F}}$ as input and output, respectively.

5.3.1. ENCODER

The encoder comprises all parameters and operations that convert the homogenized stretch tensor into local fictitious deformation gradients. In the general PRNN architecture illustrated in Fig. 5.4, these correspond to the blue connections. In Chapter 2, the encoder consisted of an arbitrary number of hidden layers fully connected, while in this development a custom layer is proposed to ensure that physical constraints related to the definition of the strain measure are met. Two challenges arise from working with the deformation gradient instead of the small strain vector. Firstly, with the deformation gradient or the stretch, the undeformed state corresponds to the identity and not a null tensor.

In a regular dense layer, if a given set of weights $\mathbf{W}$ were to be applied on the undeformed stretch tensor (*i.e.* $\mathbf{U}^{\Omega} = \mathbf{I}$), the resulting matrix $\mathbf{W}\mathbf{U}^{\Omega}$ would be different from the identity and therefore generate stresses when it should not. To address that, we need to make a few changes to the encoder, starting with the way we treat the input. Now, instead of applying weights to transform a vector with dimension 6, we work on the actual tensor $\mathbf{U}^{\Omega}$ that is 3×3. Note that this is only a *reshaping* operation, and no additional features are needed to fill the tensor.

With that, the weights connecting $\mathbf{U}^{\Omega}$ to the inputs of the material layer can be applied in a similar fashion to the fictitious material points, in groups, to generate the deformation gradients used in that layer. In this case, for each point, a 3×3 weight matrix is needed. Another important change to ensure the zero stress-state comes from the

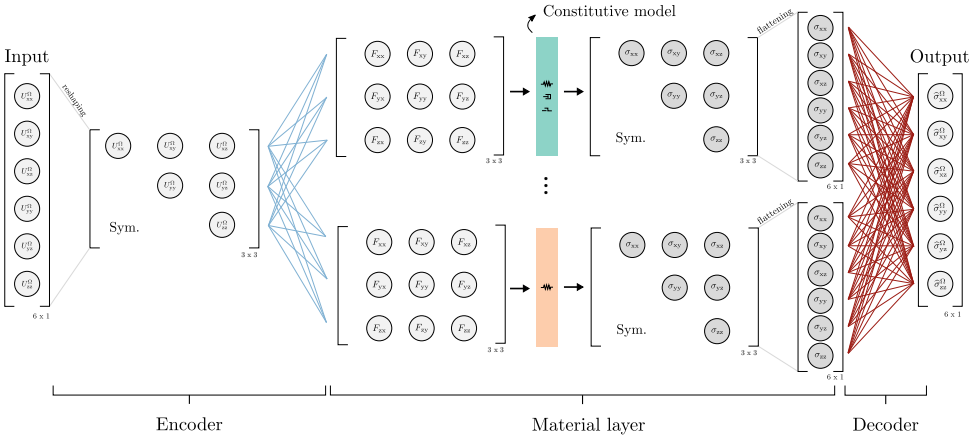


Figure 5.4: New architecture of PRNN for finite strain framework.

definition of the deformation gradient (see Eq. (5.24)). Based on that, we subtract the identity matrix from the homogenized stretch tensor and only then apply the weights to the remaining values. After that transformation, we add the identity back and obtain the final deformation gradient.

Secondly, because the deformation gradient determinant represents the change in volume from the undeformed to the current configuration, the local deformation gradients learned by the network should have strictly positive determinants. One way to avoid negative determinants consists in ensuring that the determinant of the weight matrices applied on $\mathbf{U}^{\Omega} - \mathbf{I}$ to obtain the fictitious local deformation gradients are always positive. This is done by imposing a structured weight matrix $\mathbf{W}_j$ originated from a Cholesky decomposition for each subgroup j. The determinant of the decomposed triangular matrices simplifies to the multiplication of their diagonal elements, so positivity is therefore ensured by applying a softplus function to those diagonal entries. In this case, only 6 learnable parameters are associated to each fictitious material point. The scheme in Fig. 5.5 summarizes how the local strain of one fictitious material point is obtained after the proposed changes.

5.3.2. MATERIAL LAYER

This layer contains the embedded physics-based constitutive models, arranged into a series of fictitious integration (material) points. Because a material model is not a scalar-valued function like typical activation functions (*e.g.* sigmoid, tanh, relu, etc.), a special architecture is required. In that sense, an important change compared to Chapter 2 is the way neurons are interpreted. Here, we group them together in m subgroups, each consisting of a tensor with the same order tensor and dimensions as the deformation gradient in the input layer (3×3 for the present investigation in three dimensions), whereas in Chapter 2 the subgroups consists of vectors of length 3, representing the strain vector in 2D. In this arrangement, each subgroup corresponds to one *fictitious material point*. The basic idea is that the values reaching the subgroup can be seen as a local defor-

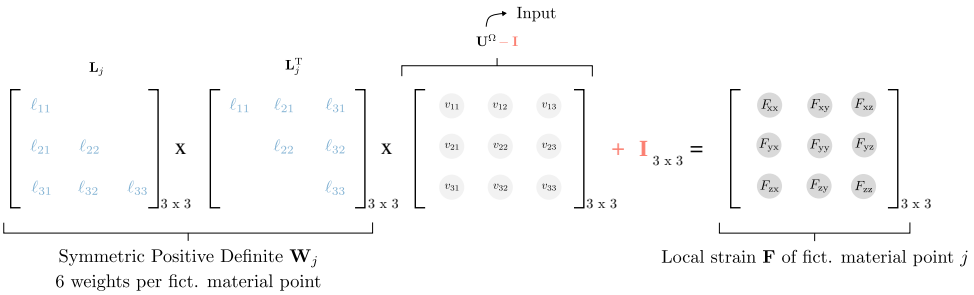


Figure 5.5: Encoder architecture applied to obtain the local strain of a fictitious material point j based on the input $\mathbf{U}^{\Omega}$.

mation learned by the encoder, which will then be evaluated by one of the constitutive models used in the full-order solution with the same material properties.

Once a given constitutive model with its respective material properties is assigned to the subgroup j, say $\mathcal{C}_j^{\omega}$, the next step is to use it to obtain the stresses and the updated internal variables (if any). These internal variables are present in rate and path-dependent material models and are the core of the physics-based memory of the proposed network. However, rate and path-independent constitutive models can also be used in the material layer without further adaptations. A brief discussion on the choice of the constitutive models used in this layer follows at the end of the section.

Consider that $\mathcal{C}_j^{\omega}$ takes as input the deformation gradient $\mathbf{F}$, the internal variables from previous time step $\boldsymbol{\alpha}^{t-1}$ and the increment of time Δt. In the first time step, the internal variables of all material points are properly initialized based on the undeformed state $\boldsymbol{\alpha}_j^0$. In every time step, the current stresses $\boldsymbol{\sigma}$ and updated internal variables $\boldsymbol{\alpha}$ of each subgroup are obtained. These variables are preserved in each subgroup so that in the following load step, when a new $\mathbf{F}$ is fed to the material point, the history of the material can be updated accordingly. A representation of this workflow is shown in Fig. 5.6. Note that the "flattening" operation transforming the 3×3 tensor into a vector with only the 6 independent components, is analogous to the reshaping operation used at the encoder. This condensation does not imply in loss of information since the Cauchy stress tensor is symmetric.

An important aspect illustrated in Fig. 5.6 is that no additional time-related features or trainable parameters are needed to learn the time-dependence. The network learns the strain distribution over the fictitious material points through the encoder, which works the same for all constitutive models. The time increment Δt is passed to the rate-dependent material as additional input, but it has the same value for all material points as would be done in the micromodel simulation. By directly employing the same material models and properties considered in the micromodel with internal variables that naturally follow physics-based assumptions, we can capture the rate and path-dependent behavior in a more straightforward way. With RNNs, the mechanisms behind the evolution of internal variables need to be learnt from the data.

Finally, the user is left with the choice of which constitutive model to employ in the ma-

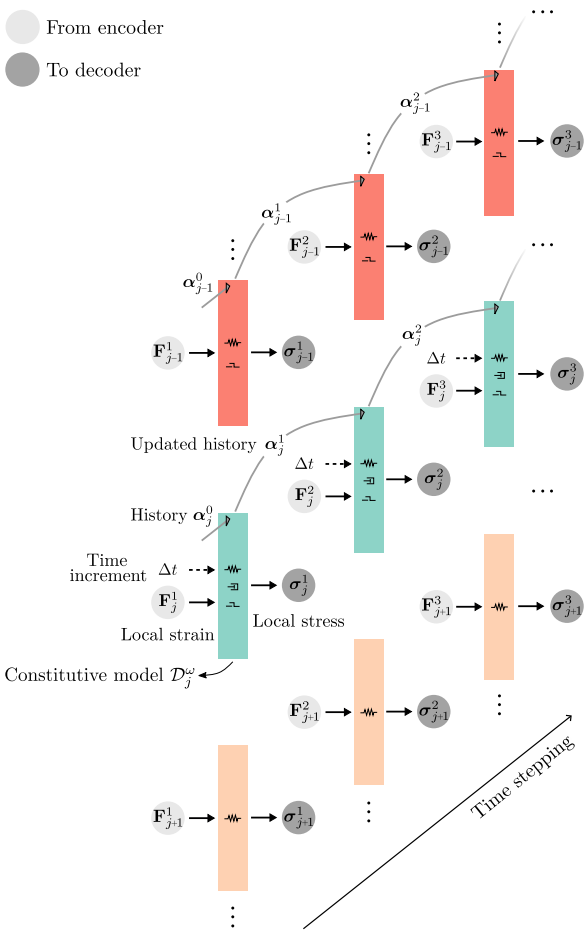


Figure 5.6: Scheme of fictitious material points unrolled in time, each colored box corresponding to a different constitutive model. From top to bottom: path-dependent, path and rate-dependent, and path and rate-independent constitutive models.

terial points. Our recommendation is to employ all nonlinear constitutive models used in the micromodel with their respective known material properties. To illustrate that, consider the composite micromodel studied in the numerical examples of this chapter, in which an orthotropic hyperelastic model is used to describe the fibers and an elasto-viscoplastic model for the matrix. Since both models include nonlinearity in their formulations, we include both types in the material layer. In addition to that, in the present case, each model has distinctive behavior in terms of path and rate-dependence, which emphasizes the importance of both in the network. This topic is further discussed in Section 2.4.4 along with other training aspects and model selection procedure, including the definition of the proportion of the constitutive models in this layer.

5.3.3. DECODER

The decoder comprises all network parameters that work on the outputs of the material layer to obtain the homogenized stresses $\widehat{\boldsymbol{\sigma}}^{\Omega}$. Note that because the outputs of the material layer consist of the stresses from the fictitious material points, the role of the decoder parameters is well aligned with the actual full-order solution. In the micromodel, once the full-field of stresses is obtained, the homogenized stresses are obtained by averaging the stresses over the entire domain. Here, instead of integrating the field with hundreds or thousands of integration points, only a few fictitious material points contribute to the homogenized response where the relative contributions of each fictitious point are learnt from data.

For that purpose, an arbitrary number of neurons and layers can be used. In this study, in particular, for a more direct analogy with the homogenization process, a single dense layer with linear activation and physics-motivated modifications is considered. In this way, the weights of the output layer reflect the relative contribution of each of the material points to the predicted homogenized response. In the actual micromodel, weights come from a numerical integration scheme and are strictly positive. To reflect that, an absolute function $\rho(\cdot)$ is applied element-wise on the weights of the decoder $\mathbf{W}_{\mathrm{d}}$. For the present architecture (see Fig. 5.4), it then follows that the predicted homogenized stress is given by

$$\widehat{\boldsymbol{\sigma}}^{\Omega} = \rho(\mathbf{W}_{\mathrm{d}})\,\mathbf{a} \tag{5.10}$$

where $\mathbf{a}$ corresponds to the concatenation of local stresses from the material points.

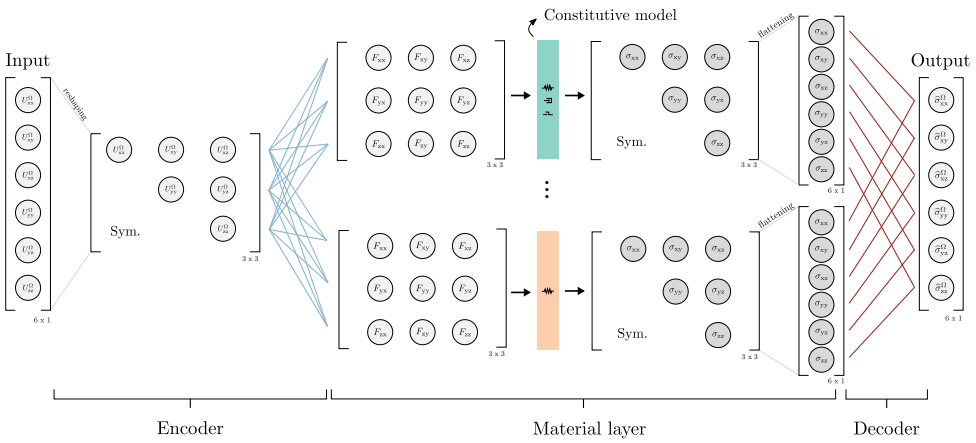


Figure 5.7: PRNN with sparse decoder.

In addition to that, we also investigate the use of a sparsification approach, where instead of having a regular dense layer that connects all components of the local stress tensor to the predicted homogenized stress, only the component-wise contributions are taken into account, as illustrated in Fig. 5.7. For instance, only the stresses σ_{xy} from each of the subgroups are weighted in for obtaining $\widehat{\sigma}^{\Omega}_{\mathrm{xy}}$. This sparsification also brings the decoder closer to the actual homogenization procedure, in which stresses are averaged component-wise.

5.3.4. TRAINING ASPECTS AND ERROR METRICS

The goal of the optimization procedure is to minimize a loss function that quantifies how close the network's prediction are from the actual solution. Here, the standard loss function based on the mean square error (MSE) is used:

$$L = \frac{1}{N_{\text{train}}} \sum_{t=1}^{N_{\text{train}}} \frac{1}{2} \left\| \text{vec}\left(\boldsymbol{\sigma}_t^{\Omega}\right) - \text{vec}\left(\widehat{\boldsymbol{\sigma}}_t^{\Omega}\left(\mathbf{U}_t^{\Omega}, \mathbf{W}, \mathbf{W}_{\text{d}}\right)\right) \right\|^2 \tag{5.11}$$

where N_{train} is the number of loading paths used for training, $\mathbf{W}$ and $\mathbf{W}_{\text{d}}$ are the network parameters for the encoder and decoder, respectively, and vec$(\cdot)$ corresponds to the Voigt representation of the homogenized stress tensor, which consists of 6 components in the 3D case (*i.e.* the "flattening" mentioned in the previous sections). From that, one can compute the gradients of the loss function with respect to the trainable parameters using a backpropagation procedure and then update those accordingly, for which we use the Adam optimizer [24].

The backpropagation here follows the same methodology as in Chapter 2. Note that the gradients of the parameters in the decoder can be obtained based on the conventional backpropagation procedure, but for the ones in the encoder, backpropagation through time is needed. This is a vital aspect of the training and stems from the path-dependency of the material models embedded in the material layer. For completeness, we include the expression for computing the gradients of the weights in the encoder at time step t for a given loading path:

$$\frac{\partial L^t}{\partial \mathbf{W}_j} = \frac{\partial L}{\partial \widehat{\boldsymbol{\sigma}}_t^{\Omega}} \frac{\partial \widehat{\boldsymbol{\sigma}}_t^{\Omega}}{\partial \boldsymbol{\sigma}_j^t} \left\{ \frac{\partial \boldsymbol{\sigma}_j^t}{\partial \mathbf{F}_j^t} \frac{\partial \mathbf{F}_j^t}{\partial \mathbf{W}_j} + \frac{\partial \boldsymbol{\sigma}_j^t}{\partial \boldsymbol{\alpha}_j^t} \frac{\partial \boldsymbol{\alpha}_j^t}{\partial \mathbf{F}_j^t} \frac{\partial \mathbf{F}_j^t}{\partial \mathbf{W}_j} + \frac{\partial \boldsymbol{\sigma}_j^t}{\partial \boldsymbol{\alpha}_j^t} \sum_{\tilde{t}=1}^{t-1} \left[\left(\prod_{\tilde{t}=\tilde{t}+1}^{t} \frac{\partial \boldsymbol{\alpha}_j^{\tilde{t}}}{\partial \boldsymbol{\alpha}_j^{\tilde{t}-1}} \right) \frac{\partial \boldsymbol{\alpha}_j^{\tilde{t}}}{\partial \mathbf{F}_j^{\tilde{t}}} \frac{\partial \mathbf{F}_j^{\tilde{t}}}{\partial \mathbf{W}_j} \right] \right\} \tag{5.12}$$

where $\mathbf{W}_j$ corresponds to the weights associated to the material point j. The gradients related to the internal variables are evaluated using central finite differences. However, if the material models are implemented with automatic differentation support (*e.g.* PyTorch and TensorFlow), these gradients and dependencies can be automatically taken into account with tools such as Autograd and GradientTape, as done with off-the-shelf RNNs.

A potential issue in training with Eq. (5.11) is the large variations of values across the multiple outputs due to the orthotropy of the composite material with high stiffness contrast. In such scenario, one component can disproportionately dominate over the others, leading to unstabilities in the training process and overall poor performance. To mitigate that, each component of $\boldsymbol{\sigma}^{\Omega}$ is normalized to [-1, 1] as follows

$$\sigma_{(\cdot)\,\text{norm}}^{\Omega} = 2 \left(\frac{\sigma_{(\cdot)}^{\Omega} - \min \sigma_{(\cdot)}^{\Omega}}{\max \sigma_{(\cdot)}^{\Omega} - \min \sigma_{(\cdot)}^{\Omega}} \right) - 1 \tag{5.13}$$

where max refers to the maximum absolute homogenized stress values of the component $(\cdot)$ in the training data and min is the negative of that value. The symmetric bounds in each of the components ensures that the zero-stress state from the material points will be reflected in the homogenized stress. Furthermore, to preserve the role of the decoder as the homogenization-like step, the normalization in Eq. (5.13) is also applied to

the local stresses from the fictitious material points. This ensures that all material point stresses are within the same range expected at the output layer. Lastly, no normalization is considered for the inputs, since the range of the features are similar and, more importantly, are compatible with the range expected by the models in the material layer.

For the model selection and performance assessment, we consider two error metrics:

$$\begin{aligned} \text{Absolute error}: & \frac{1}{N_{\text{paths}}} \sum_{i=1}^{N_{\text{paths}}} \frac{1}{L_{\text{path}}} \sum_{t=1}^{L_{\text{path}}} \left\| \text{vec}\left(\boldsymbol{\sigma}_t^{\Omega}\right) - \text{vec}\left(\widehat{\boldsymbol{\sigma}}_t^{\Omega}\right) \right\| \\ \text{Relative error}: & \frac{1}{N_{\text{paths}}} \sum_{i=1}^{N_{\text{paths}}} \frac{1}{L_{\text{path}}} \sum_{t=1}^{L_{\text{path}}} \frac{\left\| \text{vec}\left(\boldsymbol{\sigma}_t^{\Omega}\right) - \text{vec}\left(\widehat{\boldsymbol{\sigma}}_t^{\Omega}\right) \right\|}{\left\| \text{vec}\left(\boldsymbol{\sigma}_t^{\Omega}\right) + \boldsymbol{\varepsilon} \right\|} \end{aligned} \tag{5.14}$$

where N_{paths} refers to the number of loading paths in the validation/test sets, L_{path} is the length of each path and $\boldsymbol{\varepsilon}$ is a stabilizing term with the same dimensions as $\text{vec}\left(\boldsymbol{\sigma}_t^{\Omega}\right)$ filled with 10^{-8} used to avoid division by zero.

To reduce the number of hyper-parameters to be tuned and keep the model selection as simple and straightforward as possible, we define a minibatch as 2 paths, the stopping criterion as the maximum number of epochs (1000) and use the recommend default settings from [24] in the Adam optimizer, including its standard learning rate decay update per iteration. When training the network, the validation set is evaluated every 50 epochs, and the best set of parameters is updated only if the current error is lower than the historical lowest validation error, thus mitigating the risk of overfitting. Further details on the model selection procedure, including the definition of the material layer size, the type of decoder (dense or sparse), and the size of the training set, are presented in Section 5.5.

When choosing which constitutive models to assign to the fictitious material points, we follow the idea of including all sources of non-linearity. In this case, both the fiber and the matrix constitutive models qualify. At this point, it is worth highlighting another aspect that makes having both models in the network important. Although the fiber constitutive model adopted in this chapter only shows non-linearity at very large strains, in our case, it is also the one introducing the transversal isotropy in the micromodel and has distinct behavior from the matrix in terms of path and rate-dependency. Those unique characteristics need to be present in the network so that the encoder and decoder can leverage them into the homogenized stress response.

Related to that is the definition of how many of the fictitious material points are assigned to each of the models. This proportion itself is a hyper-parameter, but to reduce the amount of variables in the upcoming studies, we define a fixed splitting ratio. The hyperelastic and elasto-viscoplastic models correspond to 25 % and 75 % of the material points, respectively, rounding the number of hyperelastic models up when the total number of points is even but not divisible by 4. The higher proportion of points associated to the elasto-viscoplastic model is rooted in the fact that this is the most complex constitutive model in the micromodel, from which we expect higher expressibility. Furthermore, it is also a model with internal variables, 24 per point to be precise, which effectively work as the physics-based memory of the network. Thus, we expect to achieve good performance with smaller networks (*i.e.* more parsimonious PRNNs) compared to splitting ratios that favor hyperelastic models. Other than the difference in the material layer size itself, we expect no significant changes in the overall accuracy of the network

granted model selection has been performed correctly.

Finally, we highlight that the choice of constitutive models, regardless of history-dependence, and their splitting ratio in the material layer do not affect the total number of trainable parameters in the network. For the dense decoder architecture depicted in

5.3.5. USE AS CONSTITUTIVE MODEL

For incorporating the present network as a constitutive model in a microscale analysis that takes as input the homogenized deformation gradient ($\mathbf{F}^{\Omega}$) and the increment of time (Δt) and outputs homogenized stresses $\widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{F}}$, a few additional steps are introduced. First, the polar decomposition theorem is applied on the deformation gradient in order to obtain the rotation $\mathbf{R}^{\Omega}$ and the stretch tensors $\mathbf{U}^{\Omega}$. Once the stretch tensor is obtained and the increment of time is known, the network is used to predict the unrotated stresses $\widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{U}}$ in a forward pass. The rotation tensor is then used to transform the predicted unrotated stresses back into the rotated system.

Obtaining the tangent stiffness matrix is not as straightforward. In this framework, the jacobian of the network is only one part of the tangent stiffness matrix expression for the entire mapping between rotated stresses and the deformation gradient

$$\frac{\partial \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{F}}}{\partial \mathbf{F}^{\Omega}} = \frac{\partial \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{F}}}{\partial \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{U}}} \frac{\partial \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{U}}}{\partial \mathbf{U}^{\Omega}} \frac{\partial \mathbf{U}^{\Omega}}{\partial \mathbf{F}^{\Omega}} + \frac{\partial \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{F}}}{\partial \mathbf{R}^{\Omega}} \frac{\partial \mathbf{R}^{\Omega}}{\partial \mathbf{F}^{\Omega}} \tag{5.15}$$

where the partial derivatives of the homogenized rotation and stretch tensors with respect to the homogenized deformation gradient are given by the expressions derived by Chen and Wheeler [25] and $\partial \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{U}} / \partial \mathbf{U}^{\Omega}$ is given by performing a complete backward pass through the network. Moreover, the partial derivative of the stresses with respect to the unrotated stresses is given by

$$\frac{\partial \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{F}}}{\partial \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{U}}} = \mathbf{R}^{\Omega} \otimes \mathbf{R}^{\Omega} \tag{5.16}$$

where $\otimes$ represents the Kronecker product between two second-order tensors of dimensions $n_{\mathrm{rank}} \times n_{\mathrm{rank}}$, resulting in a second-order tensor of dimensions $n_{\mathrm{rank}}\, n_{\mathrm{rank}} \times n_{\mathrm{rank}}\, n_{\mathrm{rank}}$. Finally, the partial derivative of the stresses with respect to the rotation tensor are evaluated as

$$\frac{\partial \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{F}}}{\partial \mathbf{R}^{\Omega}} = \bar{\mathbf{P}} \left(\mathbf{I} \otimes \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{U}}\, \mathbf{R}^{\Omega^{\mathrm{T}}} \right) + \left(\mathbf{I} \otimes \mathbf{R}^{\Omega}\, \widehat{\boldsymbol{\sigma}}^{\Omega}_{\mathrm{U}} \right) \mathbf{P} \tag{5.17}$$

where $\bar{\mathbf{P}}$ and $\mathbf{P}$ are two permutation matrices given by

$$\begin{aligned} \bar{\mathbf{P}} &= \sum_{i,j} E_{ij} \otimes E_{ij} \\ \mathbf{P} &= \sum_{i,j} E_{ij} \otimes E_{ji} \end{aligned} \tag{5.18}$$

with E_{ij} being a null matrix except for the unit value at $E_{i,j}$.

5.4. DATA GENERATION

In general, surrogate models need to be trained with an extensive amount of data covering several types of loading. This is because it is virtually impossible to have fine control over what types of loading the micromodel will experience upfront even in the simplest scenarios. Therefore, to investigate how well the proposed network can generalize to unseen scenarios, a variety of loading functions and methods for generating the loading paths are considered.

First, we define the geometry and the discretization of the micromodel. In this case, the same composite RVE used in [22], and illustrated in Fig. 5.8, with 9 fibers embedded in a matrix material is adopted. The material models and properties assigned to each of the phases also follow from that work with a minor change in one of the material properties of the matrix. The reinforcements are assumed to be carbon fibers and can be described by the hyperelastic, transversely isotropic material model developed by [23]. For the matrix, the elasto-viscoplastic EGP model is considered with the relaxation spectrum now consisting of one mode (the first). Both of these models are briefly discussed in Section 5.2.1, but for further details on their implementation and numerical validation in the 3D finite strain framework, the reader is directed to the reference paper [22].

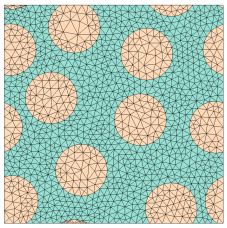


Figure 5.8: Geometry and mesh discretization of micromodel used to generate the data.

To generate the data, two strategies are devised, one producing proportional loading paths, and the other non-proportional loading paths. We use the first type to train and test the network, while the second is reserved for testing only. By proportional we refer to curves in which the loading direction is fixed. For this, we adopt the arc-length formulation with indirect displacement control derived in [26], in which a constant unit load vector is considered and the additional constraint consists in the unsigned sum of the controlled displacements. For stress measures based on the undeformed state, that also entails a constant stress ratio.

For creating proportional paths, three main ingredients are needed: the loading direction $\mathbf{n}$, the loading function λ and the time increment Δt. Previously in Chapter 2, basic load cases (*e.g.* uniaxial and biaxial tension and compression, transverse and longitudinal shear, etc.) were used for training PRNNs subjected to general stress states. Here, due to the increased problem dimensionality, we train with a more general approach of random loading directions. For each path, the unit load vector is obtained by sampling values from 6 independent Gaussian distributions ($X \sim \mathcal{N}(0,1)$) and normalizing them to a unit vector, one for each prescribed corner displacement. As for the time increment,

we set it to $\Delta t = 1\text{s}$ for all time steps. Fixing the time increments allows for a straightforward assessment of the ability of the network to extrapolate to unseen strain-rates.

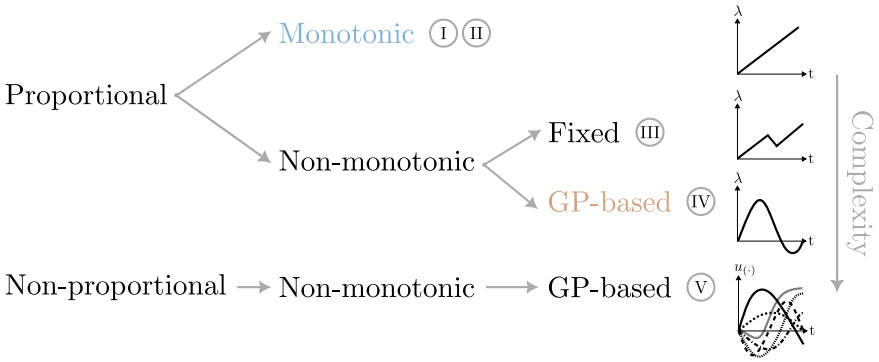
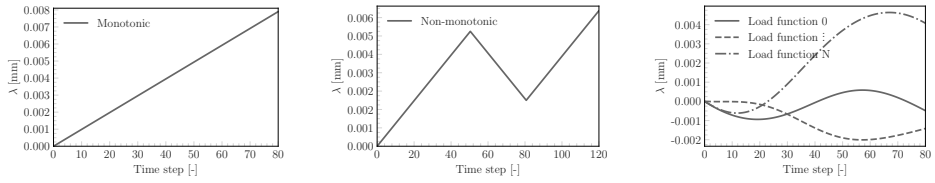


Figure 5.9: Scheme of loading types considered in this chapter, with colored types being used for training and testing, while remaining are for testing.

The last ingredient to create the proportional curves is the loading function λ. We use the two loading functions depicted in Figs. 5.10a and 5.10b as pre-defined monotonic and non-monotonic curves, respectively. Although useful for testing, this non-monotonic set is not as valuable for training since all curves follow the same unloading/reloading behavior. An alternative with more unloading variety is to sample λ from a Gaussian Process (GP) with $\mathrm{X} \sim \mathcal{N}(\mu, \sigma^2)$ and covariance function given by

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x}_p - \mathbf{x}_q\|^2\right) \tag{5.19}$$

where $\mathbf{x}_p$ and $\mathbf{x}_q$ are the time step indices of the sequence of loading function values, σ_f^2 is the variance and ℓ is the length scale. These hyper-parameters control the smoothness and how large the unsign sum of the controlled displacements can be, and are tuned to obtain smooth loading functions, as the ones illustrated in Fig. 5.10c.



(a) Monotonic loading (type I) (b) Non-monotonic loading (type III) (c) Proportional GP-based curves (type IV)

Figure 5.10: Loading functions used to create proportional loading paths.

To create a more diverse set in terms of strain-rate compared to the curves using a single pre-defined loading function, for the proportional GP-based curves, the time increment of each path is drawn from a bounded uniform distribution $\Delta t \sim U(0.01\,\text{s}, 100\,\text{s})$.

Fig. 5.9 shows a summary of the three loading types discussed so far ordered by their level of complexity. In this chapter, we train with two of them, namely monotonic curves (in blue) and proportional GP-based curves (in brown). For testing, we take a step further and generate non-proportional and non-monotonic paths. These are the most complex paths considered and also employ GPs in their formulation. To create these curves, first we switch to a displacement control method and follow a similar procedure as the one employed in Chapter 2. Here, we sample the displacements at the controlling nodes from 6 independent GPs, allowing unloading/reloading to take place at different times across the components of the homogenized deformation gradient. This is illustrated in the bottom right plot of Fig. 5.9, where independent $u_{(\cdot)}$–t functions are plotted for the different components.

For reference, all the types of loading paths studied in the following section are listed below in ascending order of complexity:

- Type I: proportional and monotonic loading path. The direction $\mathbf{n}$ is generated randomly, the loading function λ is as illustrated in Fig. 5.10a with step size $\Delta\lambda = 1 \times 10^{-4}\,\text{mm}$, and $\Delta t = 1\,\text{s}$. In the following sections, data sets using this type of path carry the subscript “mono”;
- Type II: proportional and monotonic loading path with same loading function and step size as Type I, but different strain-rate. Data sets with this type of path carry the subscript “mono" and two variations of superscript, “faster” and “slower”. To generate those, $\Delta t = 0.01\,\text{s}$ and $\Delta t = 100\,\text{s}$ are used, respectively;
- Type III: proportional and non-monotonic loading path with fixed unloading/reloading behavior λ as illustrated in Fig. 5.10b with $\Delta\lambda = 1 \times 10^{-4}\,\text{mm}$, and $\Delta t = 1\,\text{s}$. Data sets with this type of path have the subscript “unl” and the superscript “fixed”;
- Type IV: proportional and non-monotonic loading path with loading function given by a GP with variable step size, and $\Delta t \sim U(0.01\,\text{s}, 100\,\text{s})$. In this case, each loading path follows a different unloading/reloading function. Fig. 5.10c illustrates some of the loading functions generated by this approach with $\ell = 30$ and $\sigma_f^2 = 1 \times 10^{-5}$ as the hyper-parameters of the GP. Data sets with this type of path have the subscript “unl” and the superscript “prop. GP”;
- Type V: non-proportional and non-monotonic loading path with GPs to describe the displacements, and $\Delta t \sim U(0.01\,\text{s}, 100\,\text{s})$. Each controlled displacement in the micromodel is assigned to an independent GP, from which we sample smooth and random functions with variable step size. In this case, the hyper-parameters of the GPs are $\ell = 30$ and $\sigma_f^2 = 2.5 \cdot 10^{-7}$, with the exception of the variance of the GP associated to the displacement in the fiber direction, which is 10 times smaller than the others to prevent excessively high stress values that can dominate the homogenized stress state. Data sets with this type of path have the subscript “unl” and the superscript “non-prop GP”.

5.5. NUMERICAL EXPERIMENTS

In this section, the accuracy of the network is assessed in a set of numerical experiments. The goal is to illustrate the extrapolation properties of the method given the different training strategies. The test cases cover loading directions and strain-rates different from those seen in training, as well as complex unloading/reloading cases. Since we are focusing on the network's accuracy only, the following sections deal with the stretch and the unrotated stresses as their inputs and outputs, respectively.

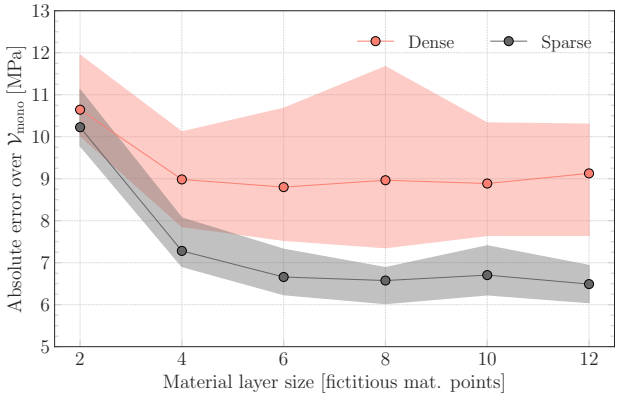


Figure 5.11: Envelope of highest and lowest absolute validation error from 10 PRNNs trained on $\mathcal{D}_{\text{mono}}$ = {144 monotonic curves} over validation set $\mathcal{V}_{\text{mono}}$ = {100 monotonic curves} with different material layer sizes and decoder architectures.

5.5.1. MODEL SELECTION

First, two preliminary studies are carried out for model selection. The first one is used to choose between sparse and dense decoders (see Section 5.3.3), while the second is focused on defining the material layer size. The comparison is carried out with varying size of the material layer each time considering the largest training set with monotonic loading paths. For each combination of decoder architecture and material layer size, 10 random initializations of the PRNN are considered. In each of them, the training set $\mathcal{D}_{\text{mono}}$ consists of 100 monotonic curves randomly selected from a pool of 1000 curves of the same type (Type I). For validation, a fixed set $\mathcal{V}_{\text{mono}}$ with 100 monotonic curves is considered. In Fig. 5.11, the colored areas correspond to the envelope with the highest and lowest absolute errors for each combination, along with the average errors represented by the solid lines with markers. In all cases, we emphasize that the reported errors over validation and test sets correspond to the network parameters associated to the historical best performance during training, as discussed in Section 2.4.4. A marked difference in accuracy between the two types of decoder for all range of material layer size over $\mathcal{V}_{\text{mono}}$ is observed. Therefore, in the remainder of this paper, all networks have a sparse decoder.

The second model selection step is focused on finding an optimal size for the material layer. For this purpose, the material layer size is varied considering a range of different training set sizes. Note that at this stage there is no direct comparison between the two training strategies since their training and validation are of matching types. Similarly to the plot in Fig. 5.11, in Fig. 5.12 we show the envelope of best and worst performances, along with the average absolute errors over the validation set $\mathcal{V}_{(\cdot)}$, which this time consists of either monotonic or proportional GP-based curves. In the following sections, we use the PRNNs with material layer size of 8 for both cases, which corresponds to the point where errors are either the lowest among all training sets or have negligible difference with respect to larger layer sizes.

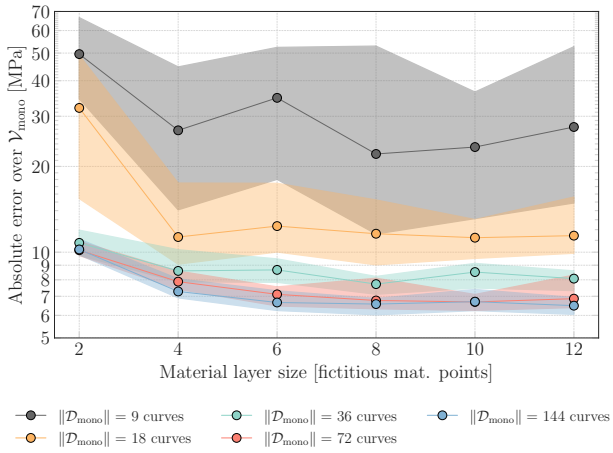
5.5.2. MONOTONIC LOADING

As first test scenario, we consider a test set $\mathcal{T}_{\text{mono}}$ consisting of 100 monotonic curves in random and unseen directions (type I). We evaluate the networks trained on monotonic (type I) and GP-based paths (type IV) over that test set for different training set sizes. Fig. 5.13a shows the lowest absolute and relative errors for both strategies, along with the envelope of absolute errors from 10 initializations. As more data is considered, the error bounds shrink and an optimal training set size can be identified around 72 curves. Although the difference in the lowest errors is still significant, 6.2 MPa (5.4 %) vs 7.6 MPa (6.3 %), more data translates into marginal gain to both. In the breakdown of the error per component in Fig. 5.13b, the largest differences in the accuracy are in the $\sigma^{\Omega}_{\text{yy}}$ and $\sigma^{\Omega}_{\text{zz}}$ components. The overall performance gap between the two training strategies in this scenario is expected since we are testing on the same loading behavior used to generate the training data of one of the strategies. Another aspect to be considered is the fact that the proportional GP-based curves reach lower strain ranges compared to the monotonic paths for the same number of time steps and step size.

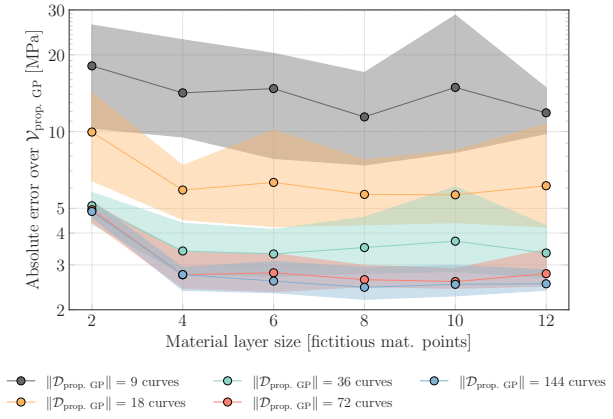
To illustrate the difference in performance, we select a curve from $\mathcal{T}_{\text{mono}}$ with an absolute error close to the best performances from both training strategies. In this case, the prediction error on the curve shown in Fig. 5.14 is around 5.5 MPa and 7.2 MPa for the networks trained on monotonic and proportional GP-based curves, respectively. Note that the accuracy loss stands out more in the components with lower stress magnitude such as $\sigma^{\Omega}_{\text{xx}}$ and $\sigma^{\Omega}_{\text{zz}}$. An explanation for that comes from the choice of the loss function, the mean squared error. Recall that although normalization of the outputs is considered to balance the difference between stress magnitudes among the components, the MSE remains an absolute metric error. As such, values on the higher end of the normalized range can still dominate the loss, leading to a better fit. Neverthess, satisfatory agreement is observed in the remaining components with the network trained on monotonic data, while the network trained on proportional GP-based curves shows more significant errors.

5.5.3. MONOTONIC LOADING WITH DIFFERENT STRAIN-RATES

Next, we test the ability of the PRNN to capture rate-dependency. For that, two new test sets are considered, $\mathcal{T}_{\text{slower}}$ and $\mathcal{T}_{\text{faster}}$, with 100 curves each again in unseen directions (type II). In the first one, the time increment Δt is set to be 100 times larger than the



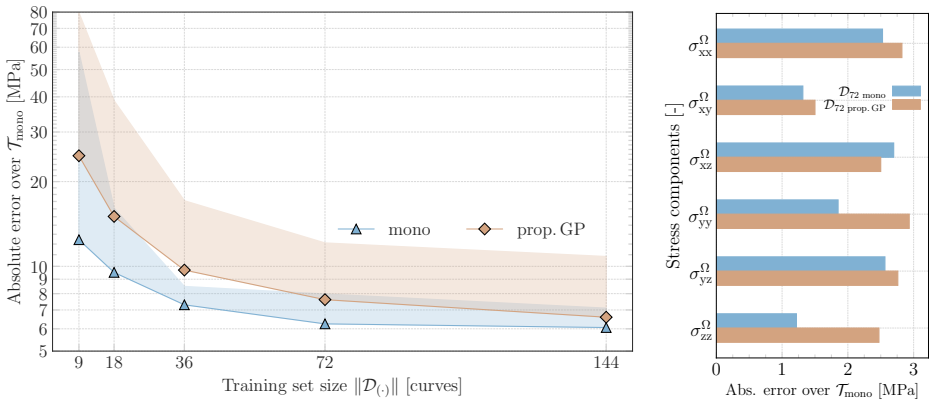
(a) $\mathcal{V}_{\text{mono}}$ = {100 monotonic loading paths}



(b) $\mathcal{V}_{\text{prop. GP}}$ = {100 proportional GP-based loading paths}

Figure 5.12: Envelope of highest and lowest errors in logarithmic scale from 10 initializations of PRNNs trained on different types of loading and material layer sizes over validation set $\mathcal{V}_{(\cdot)}$. Solid lines with markers correspond to the average validation errors.

reference one (1 s) used for generating the monotonic curves for training, and in the second, the time increment is 100 times smaller. The best performances from the 10 PRNNs trained on different types and numbers of curves are summarized in Table 5.1. Again, the slight advantage of the networks trained on monotonic curves is expected since the loading function in both test sets remains monotonic and reaches similar strain levels. As a result, networks trained with proportional GP-based curves show greater benefit from larger sets, as was the case in the previous assessment. Similarly, since the gain is still relatively small compared to doubling the training set size, we continue the



(a) Envelope of highest and lowest errors in log scale from 10 initializations of PRNNs trained on different types of loading over $\mathcal{T}_{\text{mono}}$

(b) Errors per component for best PRNNs trained on 72 curves over $\mathcal{T}_{\text{mono}}$

Figure 5.13: Envelope of absolute errors from 10 PRNNs trained on different sets and evaluated on test set $\mathcal{T}_{\text{mono}}$ = {100 monotonic curves} on the left and absolute errors per component using the best performing networks with 72 curves on the right. Solid lines with markers correspond to the best performances of each training loading type for several training set sizes.

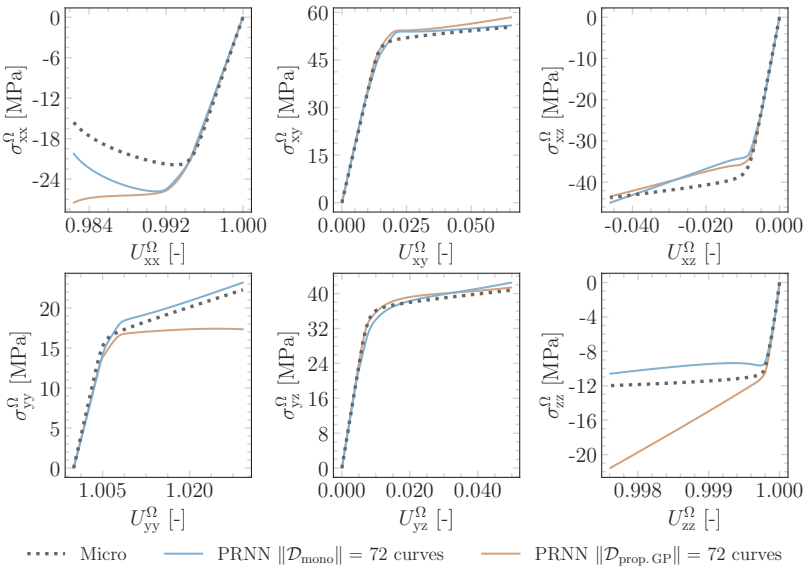


Figure 5.14: Best PRNNs trained on monotonic and GP-based curves on representative curve from test set $\mathcal{T}_{\text{mono}}$.

analysis with the smaller set for both types.

To illustrate the rate-dependent behavior, we use the best networks over each of the test sets and select a representative curve from them to visualize the effect of the different strain-rates (see Fig. 5.15). This is an important milestone of this contribution, especially considering that these strain rates are far from the reference values considered to generate the monotonic curves. The rate dependency in this case is a natural outcome of the elastoviscoplastic model used in the material layer. Encoding rate-dependence in the material layer allows for reproducing this effect without training for it. This is most evident from the error values reported in Table 5.1, where the test errors are of similar magnitude for test sets with unseen strain-rates as for the test set with the same strain rate as used for the training data. In contrast to modern RNNs, our latent variables have physical interpretation, and, more importantly, evolve according to the same physics-based assumptions considered in the micromodel.

Table 5.1: Summary of lowest absolute errors from 10 PRNNs trained on different types of curves over test sets $\mathcal{T}_{\text{mono}}$, $\mathcal{T}_{\text{faster}}$ and $\mathcal{T}_{\text{slower}}$.

Training loading type	Monotonic		Prop. GP	
Training set size	72	144	72	144
Abs. error over $\mathcal{T}_{\text{mono}}$ [MPa]	6.2	6.1	7.6	6.6
Abs. error over $\mathcal{T}_{\text{faster}}$ [MPa]	6.6	6.5	7.5	6.7
Abs. error over $\mathcal{T}_{\text{slower}}$ [MPa]	5.7	5.5	7.1	6.2

5.5.4. UNLOADING/RELOADING BEHAVIOUR

In this section, three types of unloading/reloading paths are tested with data sets from type III, IV and V. In all cases, every scenario is assessed based on a test set with 100 curves. Networks trained with both training strategies (based on type I and type IV curves) are evaluated.

PREDEFINED UNLOADING/RELOADING FUNCTION

Table 5.2 presents the lowest error from 10 networks over the test set of proportional curves with pre-defined unloading $\mathcal{T}_{\text{unl}}^{\text{fixed}}$ (type III). It can be observed that both training strategies lead to similar performances. Note that although the networks trained on proportional GP-based curves can still benefit from a larger training set, we continue the experiments with 72 curves as the gain in accuracy from doubling the training set is minimal. It is also interesting how the networks trained on monotonic paths are still slightly more accurate than the ones that have been trained with unloading. We see this as a result of two subtle advantages: (i) a loading/unloading test function much similar to the monotonic loading paths, especially the first half of the curves in $\mathcal{T}_{\text{unl}}^{\text{fixed}}$, than to the arbitrary unloading in the proportional GP-based curves and (ii) the time increment in the test curves are the same as the ones in the monotonic curves.

While these aspects help elucidate the similar performances, they do not express their significance. These networks have never seen any sort of unloading in training but are

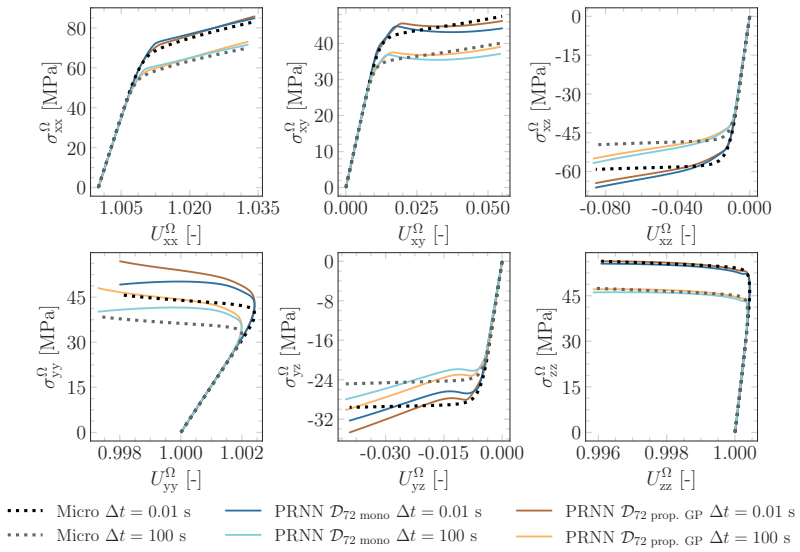


Figure 5.15: Performance of PRNNs trained on monotonic and proportional GP-based curves on test sets with strain-rates 100 times slower and 100 times faster than the one used to create the monotonic training data.

still quite capable of extrapolating to such behavior, correctly accounting for the effect of the plastic deformation. This corroborates the findings in Chapter 2, where a path-dependent material model in the material layer allowed path-dependency to arise naturally. Here, we verify that the method is general and can be extended to account for other non-linearities and time dependencies. Fig. 5.16 shows the predictions on a curve from $\mathcal{T}^{\text{fixed}}_{\text{unl}}$ with representative errors using the best performing network. Note how close the predictions are to each other and the good agreement with respect to the micromodel solution.

Table 5.2: Summary of lowest absolute errors from 10 PRNNs trained on different types of curves over test set $\mathcal{T}^{\text{fixed}}_{\text{unl}}$.

Training loading type	Monotonic		Prop. GP	
Training set size	72	144	72	144
Abs. error over $\mathcal{T}^{\text{fixed}}_{\text{unl}}$ [MPa]	6.7	6.8	7.0	6.5

PROPORTIONAL AND RANDOM UNLOADING/RELOADING

In this experiment, the test set $\mathcal{T}^{\text{prop. GP}}_{\text{unl}}$ is used to represent loading paths with unloading-reloading taking place at random times. These curves consist of the same type of loading used in one of the training strategies, which is similar to the situation discussed in Sec-

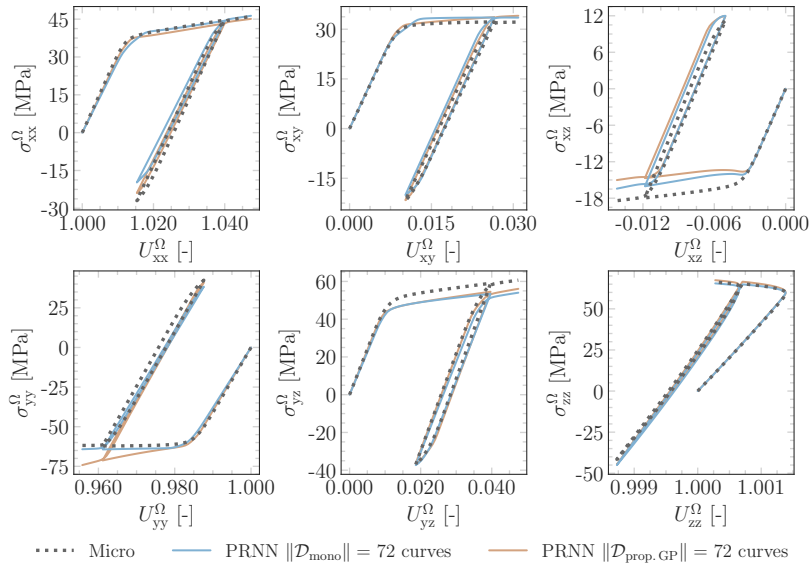


Figure 5.16: Best PRNNs trained on monotonic and proportional GP-based curves on representative curve from test set $\mathcal{T}^{\text{fixed}}_{\text{unl}}$.

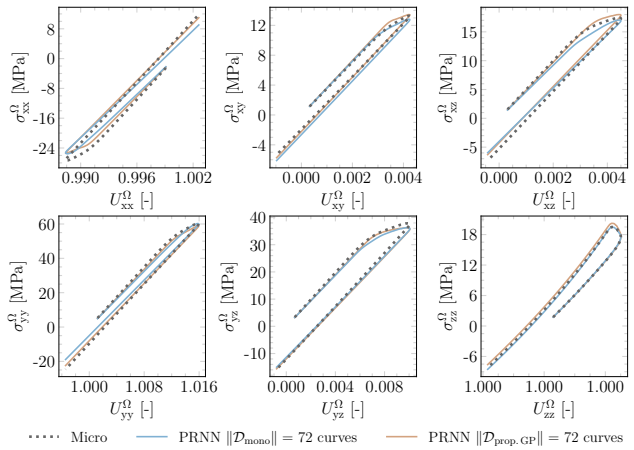
tion 5.5.2. Naturally, this results in lower test errors compared to the networks trained on monotonic loading paths, as shown in Table 5.3. To illustrate the best performance among the 10 networks considered for each strategy, we select a curve $\mathcal{T}^{\text{prop. GP}}_{\text{unl}}$ with errors close to the average lowest absolute error (see Fig. 5.17a), along with the errors per component (see Fig. 5.17b).

Table 5.3: Summary of lowest absolute and relative errors from 10 PRNNs trained on different types of curves over test sets $\mathcal{T}^{\text{prop. GP}}_{\text{unl}}$ and $\mathcal{T}^{\text{non–prop. GP}}_{\text{unl}}$.

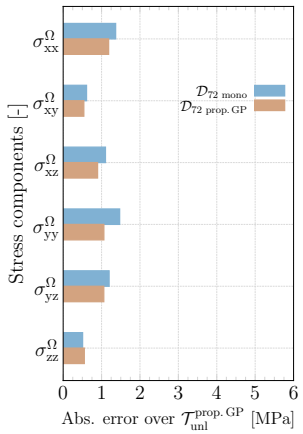
Training loading type	Monotonic		Prop. GP	
Training set size	72	144	72	144
Abs. (and rel.) error over $\mathcal{T}^{\text{prop. GP}}_{\text{unl}}$ [MPa] (%)	3.2 (8.1)	3.4 (7.8)	2.7 (5.6)	2.6 (5.1)
Abs. (and rel.) error over $\mathcal{T}^{\text{non–prop. GP}}_{\text{unl}}$ [MPa] (%)	11.5 (3.4)	12.2 (3.6)	11.0 (3.1)	10.9 (3.0)

NON-PROPORTIONAL AND RANDOM UNLOADING/RELOADING

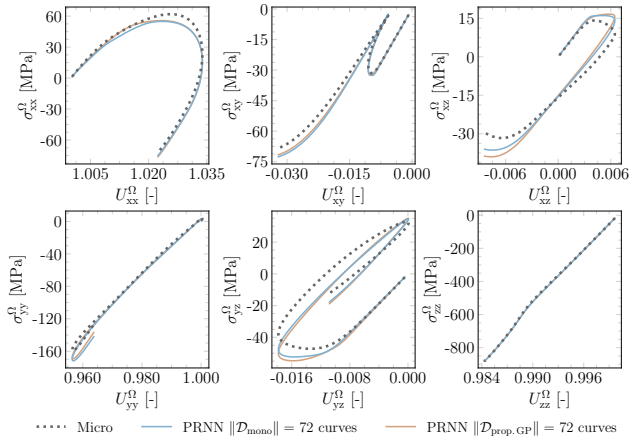
For the last part of the experiments on the accuracy of the network, the test set $\mathcal{T}^{\text{non–prop. GP}}_{\text{unl}}$ is considered. Curves from this set have more complex unloading behavior and significantly higher stress levels compared to the proportional paths in $\mathcal{T}^{\text{prop. GP}}_{\text{unl}}$. This time, the slight gain in accuracy shown in Table 5.3 from training with the proportional non-monotonic data is examined along with the relative errors. This way, we verify that al-



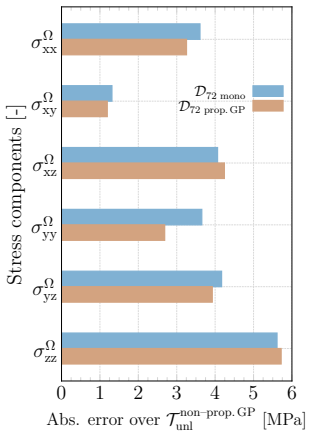
(a) Proportional GP-based curve from test set $\mathcal{T}_{\text{unl}}^{\text{prop. GP}}$



(b) Error per component over $\mathcal{T}_{\text{unl}}^{\text{prop. GP}}$



(c) Non-proportional GP-based curve from test set $\mathcal{T}_{\text{unl}}^{\text{non–prop. GP}}$



(d) Error per component over $\mathcal{T}_{\text{unl}}^{\text{non–prop. GP}}$

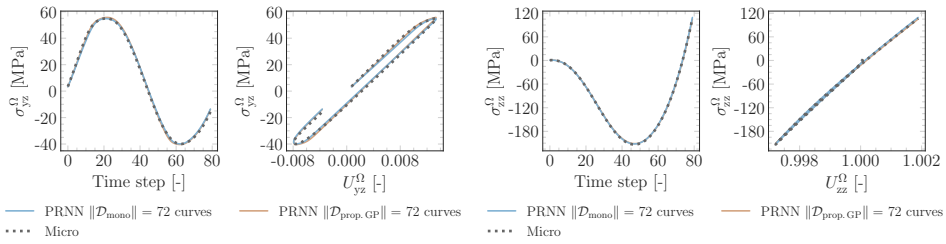
Figure 5.17: Best PRNNs trained on monotonic and GP-based curves on representative curves from two different test sets with random unloading/reloading.

though the absolute test errors have increased, the performances remain consistent with the values seen so far (below 10 %) in terms of relative errors.

In Fig. 5.17c, a representative curve from $\mathcal{T}_{\text{unl}}^{\text{non–prop. GP}}$ illustrates the best performance of both strategies over this set. The difficulty in predicting the lowest magnitude stress (in this case, $\widehat{\sigma}_{\text{xz}}^{\Omega}$) becomes more evident, as well as the variety of unloading, which this time is different in each of the components. While some components go through unloading (*e.g.* $\widehat{\sigma}_{\text{xx}}^{\Omega}$ and $\widehat{\sigma}_{\text{xy}}^{\Omega}$), others are monotonically increasing (*e.g.* $\widehat{\sigma}_{\text{zz}}^{\Omega}$) and reaching high stress

levels, which are naturally followed by higher absolute errors per component as seen in Fig. 5.17d.

An additional curve from $\mathcal{T}_{\text{unl}}^{\text{non–prop. GP}}$ is selected and shown in Fig. 5.18 to highlight another aspect not yet discussed, the orthotropic behavior of the micromodel. Note that the unloading in the z-direction follows the same stress-strain path as the loading, indicating that the elastic fiber is acting as the main load-bearing component. In contrast, the shear stress in yz follows unloading in a different branch due to the development of plastic strains in the matrix.



(a) Component σ_{yz}^{Ω} (b) Component σ_{zz}^{Ω}

Figure 5.18: Orthotropic behavior of selected components in loading path from $\mathcal{T}_{\text{unl}}^{\text{non–prop. GP}}$ illustrated with stress-time and stress-strain views.

Finally, although training with monotonic curves showed consistent and relatively accurate responses in most scenarios, the random and smooth type of loading paths explored in the previous and the current section are deemed to be more general and better representative of arbitrary functions. In both cases, training with 72 proportional GP-based curves has shown better performance and is therefore used to assess the network's capabilities in Section 5.7, where the network is used as a material model in several applications.

5.6. RUNTIME COMPARISON

In this section, we perform a runtime comparison to assess the speed-up of the proposed approach in terms of the homogenized stress evaluation. For that purpose, we continue with the loading type investigated in Section 5.5.4 (type V), and select one model from the 10 initializations trained on 72 proportional GP-based curves to represent the best overall performance. Here, we use the network with the lowest error over $\mathcal{T}_{\text{unl}}^{\text{prop. GP}}$. The choice could also have been based on $\mathcal{T}_{\text{unl}}^{\text{non–prop. GP}}$, but in favor of simplicity, in a case where the experiments presented in Section 5.5 are not carried out, choosing from $\mathcal{T}_{\text{unl}}^{\text{prop. GP}}$ implies a simpler model selection based on a single type of loading.

In this chapter, all simulations, including the data generation and training procedure for the network, were executed on a single core of a Xeon E5-2630V4 processor on a cluster node with 128 GB RAM running CentOS 7. Because we are interested in the final homogenized stress $\boldsymbol{\sigma}_{\text{F}}^{\Omega}$, we include in the PRNN runtime, the time spent in the trans-

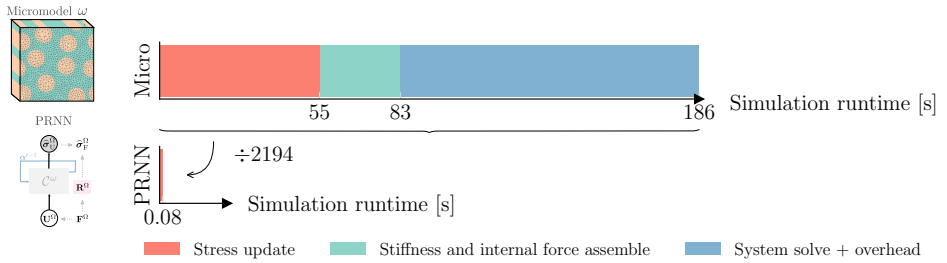


Figure 5.19: Breakdown of simulation runtime using the micromodel and the PRNN averaged over 150 type V loading paths.

formations to bring the predicted homogenized stress back to the original frame, as illustrated in Fig. 5.3. For this comparison, we use as input the converged strain path and time increments from the micromodel simulations. The micromodel mesh is shown in fig. 5.8 and consists of wedge elements integrated with 2 points in the thickness direction, comprising 4992 integration points and 7860 degrees of freedom.

Averaging over the results from 150 simulations, we break down the runtime from the full-order model in the three main parts depicted in Fig. 5.19. With the micromodel, roughly 30 % of the simulation is spent evaluating the constitutive models at the integration points, around 15 % goes to the assembly of the global stiffness matrix and internal force vector and more than half of the total time is spent solving the system, totaling 186 seconds. In contrast, the network needs only 0.08 s to compute the homogenized stress state, which results in a speed-up of three orders of magnitude when compared to the full-order solution.

In terms of offline costs, we show the average times of the two main tasks involved in the training of the networks Table 5.4. First, the time needed to generate a full path of stretches and unrotated stresses, including the polar decomposition and rotation operations; and second, the time spent on training the PRNNs with 8 fictitious material points and 72 proportional GP-based curves itself. It is worth mentioning that, regardless of the offline costs, this section presents only an estimate of the actual speed-up. In the general case, the speed-up depends on several other aspects, such as the robustness of the tangent stiffness matrix, the complexity of the loading case, and the size of the micromodel. In multiscale settings, the gain can be higher since the cost associated with an iteration at the macroscale builds on a much higher execution time when using the micromodel compared to the network, exceeding the sum of the online evaluation and offline costs. To illustrate the potential to achieve higher speed-ups, we include an additional runtime comparison in the last application of Section 5.7.

5.7. APPLICATIONS

In this section, the PRNN trained to surrogate the constitutive behavior of the micromodel in Section 5.5 is tested in applications in which its robustness also plays a role in obtaining the equilibrium path. By robustness, we understand the ability of the net-

Table 5.4: Computational offline costs averaged over 1100 training and validation proportional GP-based curves and 10 PRNNs.

	Stress-strain curve	Training
Av. wall-clock time	3.92 min	20.34 h

work to provide not only accurate stress predictions, as verified in Section 5.5, but also a tangent stiffness matrix that is stable enough for tracing an equilibrium path as close as possible to the one obtained by solving the micromodel. Previously, the entire strain path and time increments obtained from converged micromodel simulations were used as input. Here, the network is directly employed as the material model and therefore the stress prediction at each time step affects the following stress/strain state. In this case, lack of smoothness of the surrogate output may lead the iterative procedure to venture outside the training domain, potentially giving rise to divergence from the true solution.

For all applications in this section, we use the network with the lowest error on $\mathcal{T}_{\text{unl}}^{\text{prop. GP}}$. This time, to simulate its performance as a surrogate model to the micromodel, the network is embedded in a FE mesh that consists of a single 8 node hexahedral element with the same dimensions as the micromodel and one integration point with constitutive response given by the PRNN, as illustrated in Fig. 5.20. To process the deformation gradient $\mathbf{F}^{\Omega}$ into a simpler input space for the network (*i.e.* $\mathbf{U}^{\Omega}$) and obtain the stresses in their original frame of reference ($\widehat{\boldsymbol{\sigma}}_{\text{F}}^{\Omega}$) using $\mathbf{R}^{\Omega}$, we use the scheme in Fig. 5.3. For better readability, we drop the subscript, and refer to the final stresses simply as $\widehat{\boldsymbol{\sigma}}^{\Omega}$. Furthermore, for both the micromodel and the hexahedral element, in addition to the constrained displacements to avoid rigid body motion (see Fig. 5.1a), periodic boundary conditions are applied.

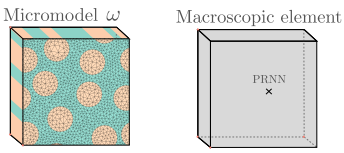
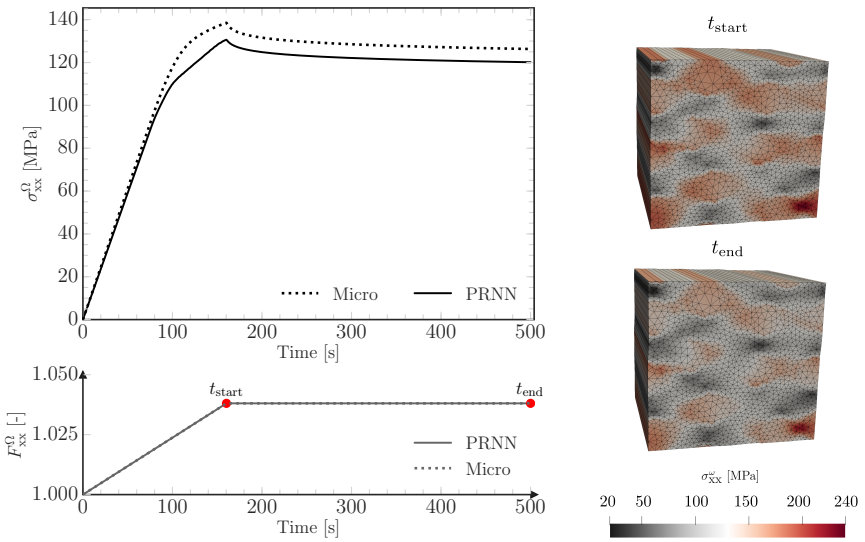


Figure 5.20: Micromodel and PRNN meshes used in the applications.

In the first application, we test the ability of the model to reproduce the stress relaxation phenomenon. In the second, we deal with cyclic loading and in the last application, the network is embedded in the general nonlinear framework developed in [22] to account for off-axis and constant strain-rate loading conditions. For the latter, we also include speed-up measurements to illustrate how aspects such as step size and tangent stiffness smoothness can play a role in increasing or decreasing the speed-up compared to the study in Section 5.6.

5.7.1. RELAXATION

In this study, a loading function to reproduce the stress relaxation phenomenon is devised. For that, the micromodel and the PRNN are loaded until a given strain level is reached $\boldsymbol{\varepsilon}_0^{\Omega}$ at $t = t_0$, when the stress level is $\boldsymbol{\sigma}_0^{\Omega}$. After that, the strain is held constant, while a gradual stress reduction takes place. For that, we use the arc-length control introduced in Section 5.4 and control the stretching in the x-direction, leaving the remaining directions free to deform.



(a) Homogenized stress-time response of micromodel and PRNN on the top and homogenized strain-time on the bottom

(b) Micromodel state at the start and end of constant strain loading

Figure 5.21: Homogenized stress-time response of micromodel and PRNN subjected to uniaxial stretch in x until t = 160 s, when the strain is held constant until the end of the simulation t = 500 s. On the right, the full-field of stresses of the micromodel for the start and end of the constant strain loading (in *red*).

In this example, the micromodel and the homogeneous hexahedral element are loaded with $\|\Delta \mathbf{u}^c\| = 5 \times 10^{-6}$ mm and $\Delta t = 1$ s until $t_0 = 160$ s, when the strain level at that point is held constant until the total time of 500 s is reached and the analysis is terminated, as depicted in the lower plot of Fig. 5.21a. In the upper plot, despite the mismatch in the stress before the start of the constant strain plateau, where the maximum error reaches 11.9 MPa (9%), the overall stress-time response of the micromodel is in relatively good agreement with the network's prediction, with an average error of 6 MPa (5 %). While this case represents a challenging scenario for even modern RNNs due to the long strain repetition, the expected stress decaying behaviour in the prediction comes as an inherent outcome of using a material model that incorporates a spectrum of relaxation times in the material layer. To illustrate the slight difference in the stress state at the begin-

ning and end of the constant strain plateau, we show in Fig. 5.21b two snapshots of the full-field solution.

5.7.2. CYCLIC LOADING

To assess the network's performance on cyclic loading, we continue with the arc-length method and same boundary conditions as the previous application but now the uniaxial stretch at time t is described as

$$F_{xx}^{\Omega} = 1 + \frac{6 \times 10^{-3}}{l} \sin\left(\frac{2\pi}{1000}\, t\right) \tag{5.20}$$

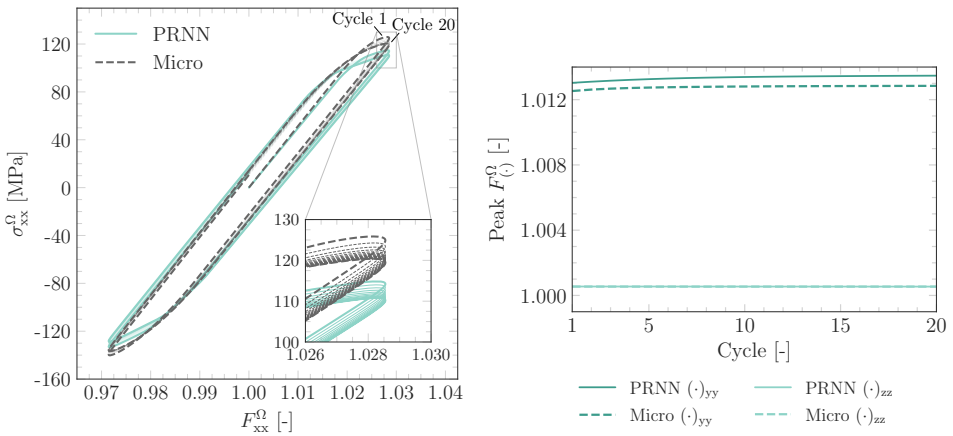
where t is the time step index and $l = 0.021\,\text{mm}$ is the side length of the micromodel. 20 cycles are considered, each consisting of 1000 steps with $\Delta t = 1\,\text{s}$. Fig. 5.22a shows the stress-strain curve for the entire loading history. The network reproduces the reverse plasticity and the hysteresis behavior in the cyclic response. Because Eq. (5.20) consists of a symmetric loading with constant peak and valley strains, a slow stress decay over the cycles takes place. This asymptotic relaxation process can be observed in the inset in Fig. 5.22a and is of similar nature to the one discussed in Section 5.7.1. Overall, good agreement is found between the PRNN and the micromodel solution. This is further assessed by unrolling the stress-strain response in time and extracting the peak and valley quantities.

First, the peak strain values from the diagonal components not controlled by the arc-length are plotted in Fig. 5.22b. In this case, the strain path obtained by the network remains close to the true solution and only minor deviations are observed in the F_{yy} component. Naturally, different loading conditions lead to different levels of accuracy of the strain paths due to the indirect displacement control equation considered here. As for the stresses, the envelopes of maximum and minimum values for the entire loading history are shown in Fig. 5.23. In each, the highest absolute error is marked by double arrows, along with the corresponding relative error. Both absolute and relative errors are within the range of errors obtained in previous sections.

5.7.3. CONSTANT STRAIN-RATE UNDER OFF-AXIS LOADING

For the last application, a dedicated strain-rate based arc-length formulation is used to reproduce the response of unidirectional composites subjected to off-axis loading [22]. In this formulation, two coordinate systems are needed: the global (x and y axes) and the local (1, 2 and 3 axes), as depicted in Fig. 5.24. In the global coordinate system, the initial fiber orientation with respect to the y-axis is defined according to a given off-axis angle χ. The micromodel is then subjected to constant strain-rate ($\dot{\varepsilon}_{yy}$) under uniaxial stress conditions. With that, equivalent homogenized deformation and stress states need to be derived in the local frame, and the transformations between global and local coordinate systems are taken care by the custom arc-length model.

For this study, we embed the network in the *local* frame, with the time increment Δt and the homogenized deformation gradient $\overline{\mathbf{F}}$ as input and the homogenized stress $\overline{\boldsymbol{\sigma}}$ as the output. In Fig. 5.24c, we show the three relevant configurations in this framework. In the simulation, due to the applied loading, the micromodel edge 0–1 tied to the local



(a) Stress-strain curve with first and last cycle highlighted

(b) Peak strains of selected components

Figure 5.22: Stress-strain response of micromodel and PRNN subjected to uniaxial cyclic loading.

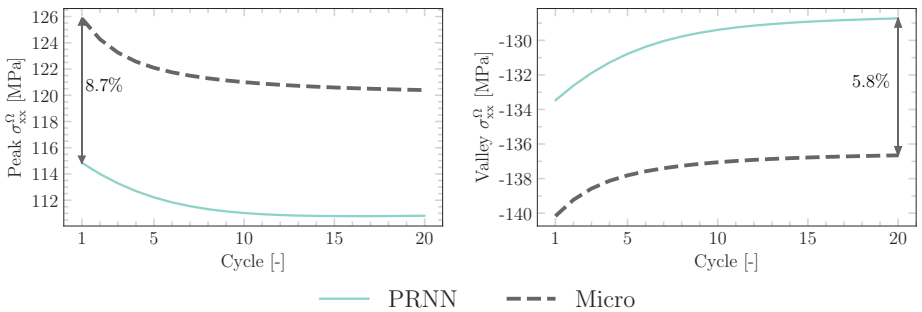
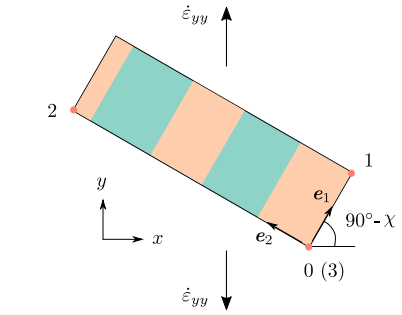
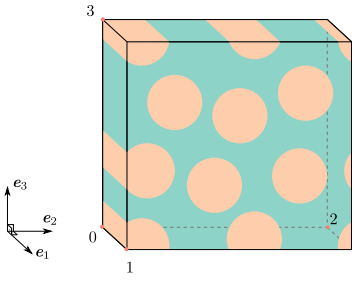


Figure 5.23: Evolution of maximum and minimum stresses for all cycles with double arrows marking the relative error corresponding to the highest absolute difference between the micromodel and PRNN subjected to uniaxial cyclic loading.

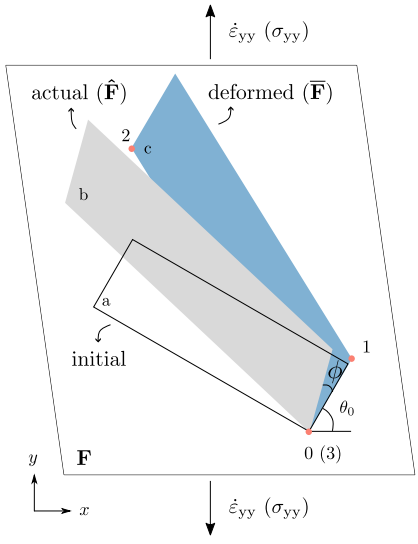
axis $\boldsymbol{e}_1$ should rotate with an angle ϕ with respect to the initial configuration (from “a" to “b"), going from the initial angle θ_0 to a new angle $\theta_1 = \theta_0 + \phi$. However, to avoid rigid-body rotation of the RVE, the controlling node 1 is fixed in the shearing direction, but the angle ϕ is implicitly taken into account through the constraint equation and the unit force vector of the arc-length model. For that reason, configuration “c", in which $\boldsymbol{e}_1$ is always aligned to the initial fiber orientation, is used to evaluate ϕ.



(a) Global and local coordinate systems



(b) Micromodel in the local system



(c) Reorientation of micromodel due to applied loading $\dot{\varepsilon}_{yy}$

Figure 5.24: Global and local coordinate systems with imposed strain-rate $\dot{\varepsilon}_{yy}$ on y direction and off-axis angle χ and reorientation of micromodel due to applied loading $\dot{\varepsilon}_{yy}$ from initial angle θ_0 to $\theta_1 = \theta_0 + \phi$ based on the deformed state [22].

In the local frame, the homogenized deformation gradient $\overline{\mathbf{F}}$ is given by:

$$\overline{\mathbf{F}} = \begin{bmatrix} \overline{F}_{11} & \overline{F}_{12} & 0 \\ 0 & \overline{F}_{22} & 0 \\ 0 & 0 & \overline{F}_{33} \end{bmatrix}. \tag{5.21}$$

To ensure the global constant strain rate condition, a special constraint equation g derived by equating the homogenized deformation gradient component in the global frame F_{yy} to the value imposed from the input is considered

$$g = \underbrace{\overline{F}_{11}\sin(\theta_0)\sin(\theta_1) + \overline{F}_{22}\cos(\theta_0)\cos(\theta_1) + \overline{F}_{12}\cos(\theta_0)\sin(\theta_1)}_{F_{yy}\text{ calculated from micromodel}} - \underbrace{\exp(\varepsilon_{yy}^{t-1} + \dot{\varepsilon}_{yy}\Delta t\ )}_{F_{yy}\text{ imposed from input}} = 0 \tag{5.22}$$

where ε_{yy}^{t-1} is the total strain in the global loading direction from the last converged time step. Another vital part of the framework is related to the update on the unit force vector applied at the controlling nodes. In this case, the geometrically nonlinear effect on the unit force vector comes not only from the change in configuration “a" to “c" but also

from the change in orientation of the micromodel that ϕ introduces. Finally, to relate the stresses from both frames, one can use the load factor λ from the arc-length formulation, which is equivalent to the σ_{yy} stress component in the global frame, to transform it to the local frame:

$$\overline{\boldsymbol{\sigma}} = \sigma_{yy} \begin{bmatrix} \sin^2(\theta_1) & \cos(\theta_1)\sin(\theta_1) & 0 \\ \cos(\theta_1)\sin(\theta_1) & \cos^2(\theta_1) & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{21} & \sigma_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.23}$$

In this contribution, we particularise the framework to $\chi = 45°$ and strain-rates $\dot{\varepsilon}_{yy} = [10^{-5}\,s^{-1}, 10^{-4}\,s^{-1}, 10^{-3}\,s^{-1}]$, resulting in three simulations in total. For more details on the formulation and derivation of the expressions presented in this section, the reader is referred to [22]. Starting with the global stress-strain response, results in Fig. 5.25 show satisfactory agreement with the full-order solution. This is yet another verification of the capability of the network to handle rate-dependency. We also inspect in Fig. 5.26 the evolution of separate pairs of stress and deformation gradient components in the local frame. It is emphasized, that in this simulation, none of these stress and strain components is directly controlled since there is a nonlinear relation where the evolution of the load in local frame depends on the computed deformation, except for the $\bar{\sigma}_{33}$ which is kept at zero. It is observed that all deformation and stress components computed with the PRNN remain close to those coming from the micromodel.

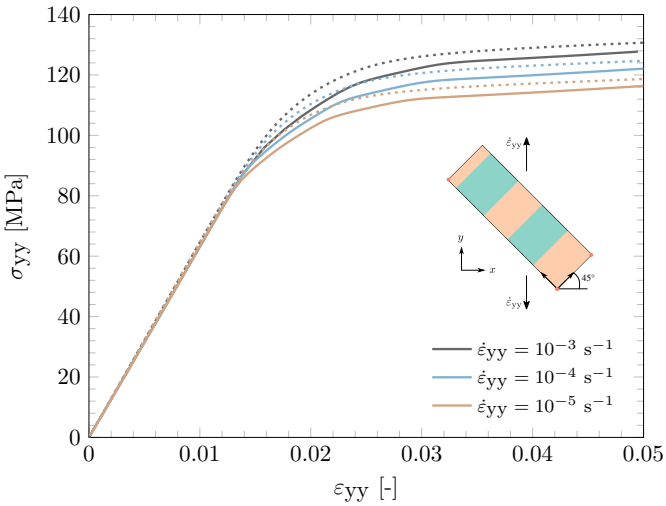
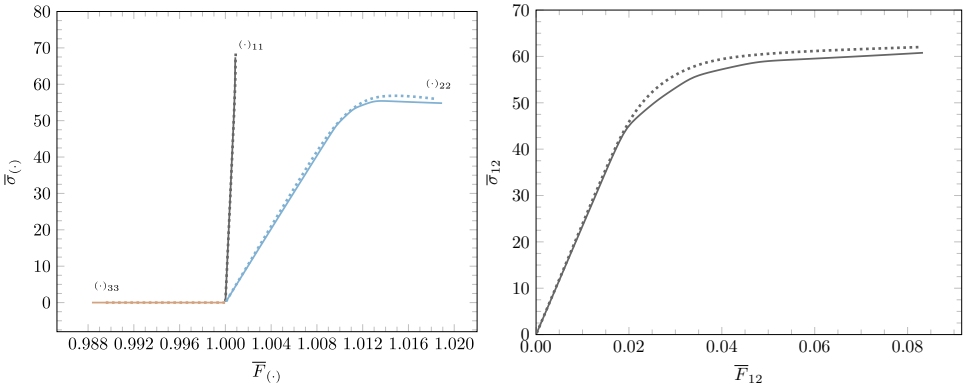


Figure 5.25: Global stress-strain curve from off-axis composite with $\chi = 45°$ and different strain-rates $\dot{\varepsilon}_{yy}$. Solid and dashed lines refer to the micromodel solution and the PRNN prediction, respectively.

A final assessment is made in terms of speed-up. This time, because an adaptive stepping scheme is used, the termination criterion (maximum norm) can be reached with a different number of macroscopic steps depending on the tangent stiffness matrix. For that reason, in Table 5.5, in addition to the breakdown of the total simulation



(a) Diagonal components

(b) Off-diagonal component

Figure 5.26: Stress and deformation in the local system for $\chi = 45°$ and $\dot{\varepsilon}_{yy} = 10^{-4}\,s^{-1}$. Solid and dashed lines refer to the micromodel solution and the PRNN prediction, respectively.

time into the three tasks shown in Fig. 5.19 and the speed-up, we also show the number of steps. With less iterations, speed-ups range from 2900 to 4000, which is significantly higher than the one obtained in Section 5.6 (≈ 2200), where neither the adaptive stepping scheme nor the network's tangent and predictions are used to define the next step in tracing the equilibrium path. Other aspects, such as macroscopic mesh density and algorithmic parameters, can also influence the speed-up and the relative times of each task with respect to the total time. In this particular case with a single macroscopic element, using the PRNN as the homogenized constitutive model means that most of the time is dedicated to evaluating the network. With that, we demonstrate the potential of the proposed approach as a robust and efficient model in a practical application.

Table 5.5: Breakdown of simulation time and speed-up for different strain-rates $\dot{\varepsilon}_{yy}$ and $\chi = 45°$, each averaged over 10 simulations.

$\dot{\varepsilon}_{yy}$ [s^{-1}]	10^{-5}		10^{-4}		10^{-3}	
Type of analysis	Micro	PRNN	Micro	PRNN	Micro	PRNN
N_{steps} [-]	293	95	290	95	279	65
Stress evaluation [s] (%)	165 (15)	.222 (59)	160 (15)	.219 (59)	160 (14)	.168 (60)
Stiff. and int. force assemble [s] (%)	91.6 (8)	.0117 (3)	89.4 (8)	.0116 (3)	91.4 (8)	.00875 (3)
System solve + overhead [s] (%)	843 (77)	.142 (38)	843 (77)	.142 (38)	872 (78)	.105 (37)
Total simulation time [s]	1099	.375	1092	.373	1123	.282
Speed-up [-]	2929		2932		3980	

5.8. CONCLUDING REMARKS

A novel Physically Recurrent Neural Network (PRNN) architecture has been developed to accelerate the microscale analysis of path and rate-dependent heterogeneous materials. The formulation follows the core idea in Chapter 2, where the homogenized response of a micromodel is obtained by a network with constitutive models embedded in one of its layers. In this *material layer*, we have *fictitious material points* with the same constitutive models and properties as used in the micromodel. The values passed from encoder to the material layer are interpreted as (fictitious) local strains, which are input to the constitutive model assigned to the material points, yielding (fictitious) local stresses. These local stresses are subsequently transformed by a decoder to obtain the homogenized stress.

What distinguishes the present methodology from the state-of-the-art surrogate models, particularly the ones based on RNNs, is the strong physics-based assumptions built into the model. Here, history-dependency is a natural outcome of the embedded material models. This is because, in addition to the local stress, the material model assigned to a fictitious material point is also in charge of updating its own internal variables (if any), which are stored from one time step to another. Therefore, PRNNs naturally inherit rich memory mechanisms from the constitutive models, bypassing the need to learn these latent dynamics from data.

While the concept of having few fictitious material points representing the homogenized response of a micromodel remains at the core of the method, a new architecture is required to extend the applicability of the network to 3D problems in a finite strain framework. Among the key changes compared to Chapter 2 are the use of the polar decomposition theorem and the principle of material objectivity. With the former, the deformation gradient can be uniquely decomposed into two tensors, namely stretch and rotation. The network is then used to learn the mapping between stretch and unrotated stress, from which the stress in the global coordinate frame is retrieved using the principle of material objectivity.

For the numerical examples, we considered a unidirectional composite micromodel with rate-dependent plasticity in the matrix and hyperelasticity in the fibers. Two different training strategies (monotonic vs non-monotonic) were considered. When creating the monotonic curves, a single value of time increment was considered so that we could clearly illustrate the exceptional ability of the network to extrapolate to strain rates far from the ones seen during training. We have also tested the performance of the network on curves with increasingly complex unloading behavior. In this case, although the networks trained on monotonic data could capture unloading behavior and performed well in most of the considered scenarios, training on non-monotonic curves led to better performance overall. Comparing the number of curves of the network selected for the numerical applications with previous developments, now we need twice as many curves to train a PRNN that is twice as big. This linear scaling should not be expected given the exponential increase nature from the curse of dimensionality, yet we can still achieve it.

In Section 5.7, we shifted our focus to applications where the PRNN is directly replacing the micromodel in the solution of the equilibrium problem. In the first application, we demonstrated that the network can reproduce relaxation, which can be a difficult behavior to capture with RNNs due to the long repetition of the input (*i.e.* constant strain).

In our case, since the constitutive models in the network have such behavior in their formulation, the homogenized response also reflected it robustly. In the second example, cyclic loading was considered, again showing the ability of the network to extrapolate to loading conditions and direction different than those trained for. For the last application, the special arc-length formulation proposed in [22] to account for off-axis loading and constant strain-rate conditions was employed. We particularised the framework to one off-axis angle and three different strain-rates and showed good agreement with the actual micromodel for a case where the network and its tangent are used to compute the solution of a nonlinear problem.

To assess the network's potential to accelerate micromodel simulations, we investigated two scenarios. Firstly, the network was used to predict stresses based on the converged strain paths from 150 micromodel simulations, leading to a stress evaluation 2200 times faster compared to the full-order model. Then, we assessed the speed-up on a problem in which the PRNN was directly involved in tracing the solution. In that case, the constant strain-rate application was used as a reference. It was observed that the lower number of steps needed when using the PRNN as the material model led to speed-ups even higher, between 2900 and 4000 for the different strain-rates. In summary, the proposed network provides an efficient model that can describe the rate-dependent, orthotropic response of thermoplastic composites in large deformations. Trained on data generated with a micromodel, the PRNN response remains close to that of the micromodel for a wide range of loading scenarios, including those outside the training range.

APPENDIX A. COMPUTATIONAL HOMOGENIZATION WITH UPDATED LAGRANGIAN FORMULATION

In this section, the finite strains formulation briefly presented in Section 5.2 is further detailed, with a focus on the coupling between the macro- and microscopic levels. For finite strains, the two main formulations are the total Lagrangian and updated Lagrangian. They are based on three configurations: the undeformed configuration t_0, and the beginning t_{start} and the end t_{end} of the time step t. In the total Lagrangian, the reference configuration in which mechanical equilibrium is calculated is fixed and consists of the undeformed configuration, that is $t_{\text{start}} = t_0$. On the other hand, in the updated Lagrangian, the reference configuration is continuously updated after each time increment, so that the end of the previous time step is the reference for the current one (i.e., $t_{\text{start}} = (t-1)_{\text{end}}$). A common strain measure used in both formulations is the deformation gradient $\mathbf{F}^{(\cdot)}_{t_0 \to t_{\text{end}}}$, either as primary, or auxiliary strain measure, which can be defined as

$$\begin{aligned}
\mathbf{F}^{(\cdot)}_{t_0\to(\star)} &= \boldsymbol{\nabla}^{(\cdot)}_{\mathbf{X}} \mathbf{x}^{(\cdot)}_{(\star)} \\
&= \boldsymbol{\nabla}^{(\cdot)}_{\mathbf{X}} \left(\mathbf{X}^{(\cdot)} + \mathbf{u}^{(\cdot)}_{t_0\to(\star)} \right) \\
&= \boldsymbol{\nabla}^{(\cdot)}_{\mathbf{X}} \mathbf{X}^{(\cdot)} + \boldsymbol{\nabla}^{(\cdot)}_{\mathbf{X}} \mathbf{u}^{(\cdot)}_{t_0\to(\star)} \\
&= \mathbf{I} + \boldsymbol{\nabla}^{(\cdot)}_{\mathbf{X}} \mathbf{u}^{(\cdot)}_{t_0\to(\star)}
\end{aligned} \tag{5.24}$$

where $\boldsymbol{\nabla}$ is the gradient operator, $\mathbf{X}$ refers to the material coordinates (i.e. undeformed configuration), $\mathbf{x}$ refers to the spatial coordinates, given by the sum of $\mathbf{X}$ and $\mathbf{u}_{t_0\to(\star)}$,

which correspond to the displacements from t_0 to $(\star)$, with $(\star)$ referring to either t_{start} or t_{end}, and $(\cdot)$ referring to any of the two scales considered in this work, macro (Ω) and micro (ω).

Similar to the homogenization theory applied to small strains, we start by employing the averaging relations. We define the macroscopic deformation gradient tensor as

$$\mathbf{F}^{\Omega}_{t_0\to(\star)} = \frac{1}{V_0}\int_{V_0} \mathbf{F}^{\omega}_{t_0\to(\star)}\,\mathrm{d}V_0 \tag{5.25}$$

where V_0 refers to the volume of the RVE at the undeformed configuration t_0. In the total Lagrangian formulation, such configuration is used as reference to derivatives, equilibrium and averaging of quantities from micro-to-macro, with the work conjugated stress-strain measures being the macroscopic first Piola-Kirchhoff $\mathbf{P}^{\Omega}_{t_{\text{end}},t_0}$, calculated at t_{end} and expressed at the undeformed configuration t_0, and the macroscopic deformation gradient $\mathbf{F}^{\Omega}_{t_0\to t_{\text{end}}}$, respectively.

In this chapter, however, we adopt the *updated Lagrangian* formulation. The two formulations are equivalent, and can be obtained one from another, with the only fundamental difference being the configuration of reference. In the updated Lagrangian, equilibrium is based on the updated configuration (i.e. t_{end}), and therefore we employ the Cauchy stress $\boldsymbol{\sigma}^{(\cdot)}_{t_{\text{end}}}$. Similarly to the averaging operation in Eq. (5.25), we define the macroscopic Cauchy stress tensor as

$$\boldsymbol{\sigma}^{\Omega}_{t_{\text{end}}} \equiv \frac{1}{v}\int_{v} \boldsymbol{\sigma}^{\omega}_{t_{\text{end}}}\,\mathrm{d}v \tag{5.26}$$

where v is the volume at t_{end}. Both equations are based on the so-called Hill-Mandel principle, which requires that the macroscopic volume average of variation of work performed on the RVE is equal to the local variation of work on the macroscale. Formulated in terms of work conjugated stress-strain quantities in the current configuration, the Hill-Mandel principle can be expressed as

$$\frac{1}{v}\int_{v} \delta\boldsymbol{\varepsilon}^{\omega}_{t_{\text{start}}\to t_{\text{end}}} : \boldsymbol{\sigma}^{\omega}_{t_{\text{end}}}\,\mathrm{d}v = \delta\boldsymbol{\varepsilon}^{\Omega}_{t_{\text{start}}\to t_{\text{end}}} : \boldsymbol{\sigma}^{\Omega}_{t_{\text{end}}} \tag{5.27}$$

where $\delta\boldsymbol{\varepsilon}^{(\cdot)}_{t_{\text{start}}\to t_{\text{end}}}$ is a virtual strain given by

$$\begin{aligned}\delta\boldsymbol{\varepsilon}^{(\cdot)}_{t_{\text{start}}\to t_{\text{end}}} &= \left(\boldsymbol{\nabla}_{\mathbf{x}^{(\cdot)}_{t_{\text{start}}}}\,\delta\mathbf{u}^{(\cdot)}_{t_{\text{start}}\to t_{\text{end}}}\right)_{\text{sym}} \\ &= \frac{1}{2}\left(\left(\boldsymbol{\nabla}_{\mathbf{x}^{(\cdot)}_{t_{\text{start}}}}\,\delta\mathbf{u}^{(\cdot)}_{t_{\text{start}}\to t_{\text{end}}}\right) + \left(\boldsymbol{\nabla}_{\mathbf{x}^{(\cdot)}_{t_{\text{start}}}}\,\delta\mathbf{u}^{(\cdot)}_{t_{\text{start}}\to t_{\text{end}}}\right)^{\mathrm{T}}\right)\end{aligned} \tag{5.28}$$

where $\delta\mathbf{u}^{(\cdot)}_{t_{\text{start}}\to t_{\text{end}}}$ is the virtual displacement associated with $\delta\boldsymbol{\varepsilon}^{(\cdot)}_{t_{\text{start}}\to t_{\text{end}}}$. Note that Eq. (5.28) is similar to the strain tensor in small strains, but here the derivatives are taken with respect to the spatial coordinates at configuration t_{start}, instead of the material coordinates at configuration t_0.

Considering that the principle of separation of scales holds, the microscopic displacements can be additively split as

$$\delta\mathbf{u}^{\omega}_{t_{\text{start}}\rightarrow t_{\text{end}}} = \left(\boldsymbol{\nabla}_{\mathbf{x}^{\Omega}_{t_{\text{start}}}}\delta\mathbf{u}^{\Omega}_{t_{\text{start}}\rightarrow t_{\text{end}}}\right)\mathbf{x}^{\omega}_{t_{\text{start}}} + \delta\tilde{\mathbf{u}}^{\omega}_{t_{\text{start}}\rightarrow t_{\text{end}}} \tag{5.29}$$

where the first term is linearly related to the macroscopic displacements and the second term corresponds to a fluctuation field caused by the microscopic inhomogeneities. Thus, replacing Eq. (5.29) in Eq. (5.27) and using the fact that the Cauchy stress tensor is symmetric, we obtain

$$\begin{aligned}
&\frac{1}{v}\int_{v}\left(\boldsymbol{\nabla}_{\mathbf{x}^{\omega}_{t_{\text{start}}}}\left(\left(\boldsymbol{\nabla}_{\mathbf{x}^{\Omega}_{t_{\text{start}}}}\delta\mathbf{u}^{\Omega}_{t_{\text{start}}\rightarrow t_{\text{end}}}\right)\mathbf{x}^{\omega}_{t_{\text{start}}} + \delta\tilde{\mathbf{u}}^{\omega}_{t_{\text{start}}\rightarrow t_{\text{end}}}\right)\right)_{\text{sym}} : \boldsymbol{\sigma}^{\omega}_{t_{\text{end}}}\,\mathrm{d}v \\
&= \underbrace{\frac{1}{v}\int_{v}\underbrace{\boldsymbol{\nabla}_{\mathbf{x}^{\Omega}_{t_{\text{start}}}}\delta\mathbf{u}^{\Omega}_{t_{\text{start}}\rightarrow t_{\text{end}}}}_{\text{constant over } v} : \boldsymbol{\sigma}^{\omega}\,\mathrm{d}v}_{\text{Term A}} + \underbrace{\frac{1}{v}\int_{v}\boldsymbol{\nabla}_{\mathbf{x}^{\omega}_{t_{\text{start}}}}\tilde{\mathbf{u}}^{\omega}_{t_{\text{start}}\rightarrow t_{\text{end}}} : \boldsymbol{\sigma}^{\omega}_{t_{\text{end}}}\,\mathrm{d}v}_{\text{Term B vanishes with PBC}}.
\end{aligned} \tag{5.30}$$

Rearranging Term A in Eq. (5.30) by removing the constant term out of the integral, we obtain

$$\begin{aligned}
&\boldsymbol{\nabla}_{\mathbf{x}^{\Omega}_{t_{\text{start}}}}\delta\mathbf{u}^{\Omega}_{t_{\text{start}}\rightarrow t_{\text{end}}} : \underbrace{\frac{1}{v}\int_{v}\boldsymbol{\sigma}^{\omega}_{t_{\text{end}}}\,\mathrm{d}v}_{\boldsymbol{\sigma}^{\Omega}\text{ as defined in Eq. (5.26)}} \\
&= \boldsymbol{\nabla}_{\mathbf{x}^{\Omega}_{t_{\text{start}}}}\delta\mathbf{u}^{\Omega}_{t_{\text{start}}\rightarrow t_{\text{end}}} : \boldsymbol{\sigma}^{\Omega}_{t_{\text{end}}}.
\end{aligned} \tag{5.31}$$

On the right-hand side of Eq. (5.27), we use again the fact that the Cauchy stress tensor is symmetric to arrive at

$$\begin{aligned}
&\left(\boldsymbol{\nabla}_{\mathbf{x}^{\Omega}_{t_{\text{start}}}}\delta\mathbf{u}^{\Omega}_{t_{\text{start}}\rightarrow t_{\text{end}}}\right)_{\text{sym}} : \boldsymbol{\sigma}^{\Omega} \\
&= \boldsymbol{\nabla}_{\mathbf{x}^{\Omega}_{t_{\text{start}}}}\delta\mathbf{u}^{\Omega}_{t_{\text{start}}\rightarrow t_{\text{end}}} : \boldsymbol{\sigma}^{\Omega}_{t_{\text{end}}}
\end{aligned} \tag{5.32}$$

which is exactly the same expression as Eq. (5.31) derived from the variation of microscopic work, fulfilling the Hill-Mandel condition.

It is worth mentioning that strains and stresses can be retrieved from other configurations, such as the standard relation connecting Cauchy and first Piola-Kirchhoff

$$\boldsymbol{\sigma}^{\omega}_{t_{\text{end}}} = \frac{\mathbf{F}^{\omega}_{t_0\rightarrow t_{\text{end}}}\mathbf{P}^{\omega}_{t_{\text{end}},t_0}}{\det(\mathbf{F}^{\omega}_{t_0\rightarrow t_{\text{end}}})} \tag{5.33}$$

where $\mathbf{P}^{\omega}_{t_{\text{end}},t_0}$ is the microscopic first Piola-Kirchhoff stress tensor calculated at t_{end} expressed at the undeformed configuration t_0. This relation always holds at the local (microscopic) level. Nevertheless, when used to compute volume averages, the non-linear nature of the transformations, in general, makes results differ from the ones obtained based on the macroscopic counterparts [27, 28]. In other words, the following alterna-

tives would not match in the general case

$$\frac{1}{v}\int_{v}\boldsymbol{\sigma}^{\omega}_{t_{\text{end}}}\,\mathrm{d}v \neq \frac{\left(\frac{1}{V_0}\int_{V_0}\mathbf{F}^{\omega}_{t_0\rightarrow t_{\text{end}}}\mathrm{d}V_0\right):\left(\frac{1}{V_0}\int_{V_0}\mathbf{P}^{\omega}_{t_{\text{end}},t_0}\mathrm{d}V_0\right)}{\det\left(\frac{1}{V_0}\int_{V_0}\mathbf{F}^{\omega}_{t_0\rightarrow t_{\text{end}}}\mathrm{d}V_0\right)}. \tag{5.34}$$

A recent study by de Souza Neto and Feijóo [29] clarified that the inequality in Eq. (5.34) is only verified in the specific case where Uniform Boundary Conditions are adopted to model the RVE. In any case, the choice of the primary strain-stress measure needs to be done carefully, taking into account aspects such as convenience of the implementation, experimental results and work-conjugacy.

We emphasize that while the overall procedure is incremental, the constitutive modeling response of the materials is formulated in terms of the total deformation gradient ($\mathbf{F}^{(\cdot)}_{t_0\rightarrow t_{\text{end}}}$), which can be promptly retrieved from $\Delta\mathbf{F}^{(\cdot)}_{t_0\rightarrow t_{\text{end}}}$ using $\mathbf{F}^{(\cdot)}_{t_0\rightarrow t_{\text{start}}}$, known from the previous time step

$$\mathbf{F}^{(\cdot)}_{t_0\rightarrow t_{\text{end}}} = \Delta\mathbf{F}^{(\cdot)}_{t_{\text{start}}\rightarrow t_{\text{end}}}\,\mathbf{F}^{(\cdot)}_{t_0\rightarrow t_{\text{start}}}. \tag{5.35}$$

As discussed in Section 5.2.1, in case of rate-dependency, such as the EGP model, both the total and the increment of deformation gradient are needed. For convenience, the PRNN was designed to take the total deformation gradient as its input for two reasons: 1) $\mathbf{F}^{\Omega}_{t_0\rightarrow t_{\text{end}}}$ is the state measure required to compute the consistent tangent stiffness matrix and 2) it conveniently contains all necessary kinematic information, as the increment of deformation gradient required for advancing the rate-dependent internal state variables can be easily derived from the current and previous total states.

REFERENCES

[1] J. Oliver, M. Caicedo, A. E. Huespe, J. A. Hernández, and E. Roubin. "Reduced order modeling strategies for computational multiscale fracture". *Computer Methods in Applied Mechanics and Engineering* 313 (2017), 560–595. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.09.039.

[2] F. Ghavamian, P. Tiso, and A. Simone. "POD–DEIM model order reduction for strain-softening viscoplasticity". *Computer Methods in Applied Mechanics and Engineering* 317 (2017), 458–479. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.11.025.

[3] I. B. C. M. Rocha, F. P. van der Meer, and L. J. Sluys. "Efficient micromechanical analysis of fiber-reinforced composites subjected to cyclic loading through time homogenization and reduced-order modeling". *Computer Methods in Applied Mechanics and Engineering* 345 (2019), 644–670. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2018.11.014.

[4] 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". *Journal of Computational Physics: X* 9 (2021), 100083. ISSN: 2590-0552. DOI: https://doi.org/10.1016/j.jcpx.2020.100083.

[5] E. Ghane, M. Fagerström, and M. Mirkhalaf. “Recurrent neural networks and transfer learning for predicting elasto-plasticity in woven composites”. *European Journal of Mechanics - A/Solids* 107 (2024), 105378. ISSN: 0997-7538. DOI: https://doi.org/10.1016/j.euromechsol.2024.105378.

[6] H. L. Cheung and M. Mirkhalaf. “A multi-fidelity data-driven model for highly accurate and computationally efficient modeling of short fiber composites”. *Composites Science and Technology* 246 (2024), 110359. ISSN: 0266-3538. DOI: https://doi.org/10.1016/j.compscitech.2023.110359.

[7] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes. “A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics”. *Computer Methods in Applied Mechanics and Engineering* 379 (2021), 113741. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.113741.

[8] R. Arora, P. Kakkar, B. Dey, and A. Chakraborty. *Physics-informed neural networks for modeling rate- and temperature-dependent plasticity*. 2022. DOI: 10.48550/arxiv.2201.08363.

[9] F. Masi and I. Stefanou. “Multiscale modeling of inelastic materials with Thermodynamics-based Artificial Neural Networks (TANN)”. *Computer Methods in Applied Mechanics and Engineering* 398 (2022), 115190. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.115190.

[10] M. Eghbalian, M. Pouragha, and R. Wan. “A physics-informed deep neural network for surrogate modeling in classical elasto-plasticity”. *Computers and Geotechnics* 159 (2023), 105472. ISSN: 0266-352X. DOI: https://doi.org/10.1016/j.compgeo.2023.105472.

[11] K. Garanger, J. Kraus, and J. J. Rimoli. “Symmetry-enforcing neural networks with applications to constitutive modeling”. *Extreme Mechanics Letters* 71 (2024), 102188. ISSN: 2352-4316. DOI: https://doi.org/10.1016/j.eml.2024.102188.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[13] Y. Zhongbo and P. L. Hien. “Pre-trained transformer model as a surrogate in multiscale computational homogenization framework for elastoplastic composite materials subjected to generic loading paths”. *Computer Methods in Applied Mechanics and Engineering* 421 (2024), 116745. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2024.116745.

[14] E. Pitz and K. Pochiraju. “A neural network transformer model for composite microstructure homogenization”. *Engineering Applications of Artificial Intelligence* 134 (2024), 108622. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2024.108622.

[15] J. Wen, Q. Zou, and Y. Wei. “Physics-driven machine learning model on temperature and time-dependent deformation in lithium metal and its finite element implementation”. *Journal of the Mechanics and Physics of Solids* 153 (2021), 104481. ISSN: 0022-5096. DOI: https://doi.org/10.1016/j.jmps.2021.104481.

[16] W. Ge and V. L. Tagarielli. “A computational framework to establish data-driven constitutive models for time- or path-dependent heterogeneous solids”. *Scientific Reports* 11.1 (2021). DOI: 10.1038/s41598-021-94957-0.

[17] F. Ghavamian and A. Simone. “Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network”. *Computer Methods in Applied Mechanics and Engineering* 357 (2019), 112594. ISSN: 00457825. DOI: 10.1016/j.cma.2019.112594.

[18] G. Chen. “Recurrent neural networks (RNNs) learn the constitutive law of viscoelasticity”. *Computational Mechanics* 67.3 (2021), 1009–1019. DOI: 10.1007/s00466-021-01981-y.

[19] B. Liu, E. Ocegueda, M. Trautner, A. M. Stuart, and K. Bhattacharya. “Learning macroscopic internal variables and history dependence from microscopic models”. *Journal of the Mechanics and Physics of Solids* 178 (2023), 105329. ISSN: 0022-5096. DOI: https://doi.org/10.1016/j.jmps.2023.105329.

[20] Y. Zhang and K. Bhattacharya. “Iterated learning and multiscale modeling of history-dependent architectured metamaterials”. *Mechanics of Materials* 197 (2024), 105090. ISSN: 0167-6636. DOI: https://doi.org/10.1016/j.mechmat.2024.105090.

[21] A. Eghtesad, J. Tan, J. N. Fuhg, and N. Bouklas. “NN-EVP: A physics informed neural network-based elasto-viscoplastic framework for predictions of grain size-aware flow response”. *International Journal of Plasticity* 181 (2024), 104072. ISSN: 0749-6419. DOI: https://doi.org/10.1016/j.ijplas.2024.104072.

[22] D. Kovačević and F. P. van der Meer. “Strain-rate based arclength model for nonlinear microscale analysis of unidirectional composites under off-axis loading”. *International Journal of Solids and Structures* 250 (2022), 111697. ISSN: 0020-7683. DOI: https://doi.org/10.1016/j.ijsolstr.2022.111697.

[23] J. Bonet and A. J. Burton. “A simple orthotropic, transversely isotropic hyperelastic constitutive equation for large strain computations”. *Computer Methods in Applied Mechanics and Engineering* 162.1 (1998), 151–164. ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(97)00339-3.

[24] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/arxiv.1412.6980.

[25] Y.-C. Chen and L. Wheeler. “Derivatives of the stretch and rotation tensors”. *Journal of Elasticity* 32.3 (Sept. 1993), 175–182. DOI: 10.1007/bf00131659.

[26] I. B. C. M. Rocha, P. Kerfriden, and F. P. van der Meer. “Micromechanics-based surrogate models for the response of composites: A critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks”. *European Journal of Mechanics, A/Solids* 82 (2020), 103995. ISSN: 09977538. DOI: 10.1016/j.euromechsol.2020.103995.

[27] S. Nemat-Nasser. “Averaging theorems in finite deformation plasticity”. *Mechanics of Materials* 31.8 (1999), 493–523. ISSN: 0167-6636. DOI: https://doi.org/10.1016/S0167-6636(98)00073-8.

[28] V. Kouznetsova. “Computational homogenization for the multi-scale analysis of multi-phase materials”. PhD thesis. 2002. DOI: 10.6100/IR560009.

[29] E. de Souza Neto and R. Feijóo. “On the equivalence between spatial and material volume averaging of stress in large strain multi-scale solid constitutive models”. *Mechanics of Materials* 40.10 (2008), 803–811. ISSN: 0167-6636. DOI: https://doi.org/10.1016/j.mechmat.2008.04.006.

6

BRIDGING EXPERIMENTS AND MULTISCALE SIMULATIONS

In the final contribution of this thesis, the developments in Chapter 5 are employed to bridge constant strain-rate and creep experiments on unidirectional thermoplastic composites under off-axis loading and multiscale simulations. Previously, these experiments were modeled as single-scale micromechanical simulations under the assumption of macroscopic homogeneity [4, 5]. However, simulations with low-off axis angles showed significant discrepancies with the experiments, leading to the hypothesis that the mismatch was caused by macroscopic inhomogeneity. Testing this hypothesis would require a multiscale approach, which is computationally prohibitive in this case, leaving the matter unresolved until now.

With PRNNs, the multiscale problem can be reformulated as a surrogate-based multiscale problem, where the network is used to predict the homogenized response of the micromodel. With this, we contribute to a better understanding of the experiments by testing the hypotheses raised in [4] and [5]. Additionally, we explore new features of the PRNN for transfer learning in terms of material properties extrapolation, promoting more efficient training. Finally, this chapter showcases the robustness of the network to obtain macroscopic convergence with an implicit solver through a series of simulations.

Apart from the shortened introduction, this chapter was integrally extracted from the following source material:

M. A. Maia, I. B. C. M. Rocha, D. Kovačević, and F. P. van der Meer. *Surrogate-based multiscale analysis of experiments on thermoplastic composites under off-axis loading.* 2025. arXiv: 2501.10193 [math.NA]. URL: https://arxiv.org/abs/2501.10193

6.1. INTRODUCTION

Our starting point in this chapter is the experiments and contributions in [1–3]. In those works, constant strain-rate and creep experiments were modeled at the microscale under the assumption that macroscale variations are negligible. In [1], the authors laid the foundations of a custom framework based on a single-scale micromodel simulation with special boundary conditions to emulate a uniaxial stress state in the global frame for a unidirectional fiber-reinforced composite (FRC) with rate-dependent response under arbitrary off-axis loading. Later, the formulation was expanded to include failure analysis for constant strain-rate [2], constant stress [3] and cyclic loading [4]. The micromodel-based formulation was able to accurately reproduce the experimental results in most cases, with the exception of small off-axis angle scenarios, in which a significant difference was observed already prior to failure. A possible reason for the mismatch is that the stress state in the specimen is not uniform on the macroscale. However, testing this hypothesis with the same high-fidelity micromodel would require a concurrent multiscale approach (FE^2), which is computationally intractable in this case. Now with the PRNN developed in Chapter 5, a surrogate-based multiscale analysis comes within reach.

It is worth mentioning that other studies have highlighted similar challenges when modeling off-axis experiments through full-order microscale simulations and/or FE simulations. Wan *et al.* [5] pointed to the difficulty of running tests and reproducing off-axis specimens with fibre angles smaller than 45° using a single micromechanical model. The strain inhomogeneity matter was discussed in [6] through the lens of the Digital Image Correlation method. In that work, however, the authors focused on the characterization of the material for different experimental setups, leaving out details on the computational modeling. Recently, another alternative to FE^2 was proposed in [7]. The authors developed a mesoscopic extension of the elasto-viscoplastic model used in [2] with anisotropic pressure-dependent behavior to model the experiments at a coupon-level, capturing the off-axis strain inhomogeneities efficiently. In more complex architectures, such as braided [8] and woven [9] composites, studies investigating the failure mechanisms have modeled off-axis experiments as multiscale problems. In [8], FE^2 was shown to outperform direct numerical simulations, but the reported CPU times grew exponentially with mesh refinement at the macroscale, even with the small number of integration points (2–6) and no history-dependence.

Hierarchical modeling has also been explored for modeling FRCs under off-axis loading [10–13], with examples including stress amplifications factors computed from predefined reference points in the RVE [10, 11] and the use of asymptotic homogenization to compute homogenized coefficients from a simplified RVE with linear viscoelastic matrix to reproduce creep experiments with transverse fibers [12]. These approaches are computationally efficient, as the factors/coefficients are computed only once (or a few times); however, their validity is tied to the specific microstructural states sampled offline, making their extensions to the scenario explored here (models with rate- and path-dependent behavior) challenging.

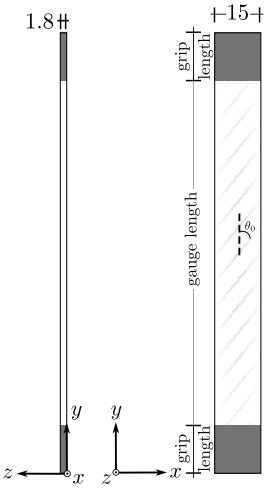
As for the research line based on surrogate modeling followed in this thesis, several applications have been proposed in the past few years for both unidirectional [5, 6, 14–20] and woven composites [13, 21–23]. Although part of these works validate numerical simulations with experimental data [5, 13, 14, 16, 21–24], only few works deal with off-

axis loading [5, 16, 22, 23], and these neglect strain-rate effects.

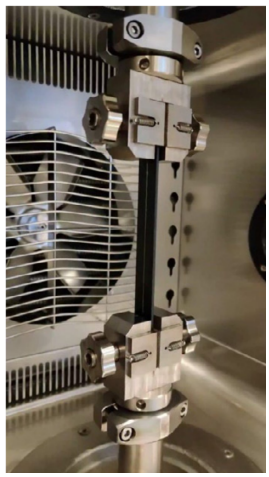
In this chapter, with the PRNN proposed in Chapter 5 as a microscale surrogate, we can now model the off-axis constant strain-rate and creep experiments from [2, 3] as a surrogate-based multiscale problem. The key novelty of this work lies in applying the network to solve an engineering problem that would be computationally intractable with FE^2, while accounting for complex material behavior using elasto-viscoplastic isotropic and hyperelastic transversely isotropic models. This approach allows direct evaluation of the macroscopic strain and stress variability hypotheses from [2, 3], making it possible to illustrate when macroscopic uniformity assumptions hold or fail, providing insights for future material testing and design. We also discuss a new PRNN transfer learning feature that takes advantage of the embedded constitutive models in the network to avoid (re-)training when going from one RVE with a given set of material properties to another RVE with the same geometry and constitutive models but different properties.

In the following, we discuss the test setup and methods in Section 6.2. Then, we assess the performance of the surrogate-based approach on a selection of experiments in Sections 6.3 and 6.4, showcasing the robustness of the network for multiscale analysis. Finally, we draw the main conclusions of this chapter in Section 6.5.

6.2. METHODS



(a) Geometry with dimensions in mm



(b) Setup used for creep testing [3, 25]

Figure 6.1: Schematic representation of UD carbon/PEEK composite system specimen and creep testing setup.

In this section, we outline the most relevant aspects of the experimental and numerical setups used to obtain the strain-stress response of a thermoplastic composite system over different off-axis angles, strain-rates and stress levels.

6.2.1. EXPERIMENTAL SETUP

Coupons were made of carbon fiber reinforced unidirectional tapes and PEEK matrix with fiber volume fraction of 0.4. The geometry of the coupons manufactured for the constant strain-rate experiments is illustrated in Fig. 6.1a, where θ_0 is the off-axis angle. The numerical counterpart models only the gauge length, which is 120 mm for the constant strain rate experiments. In these tests, the crosshead speeds were kept constant corresponding to strain-rates ranging between 10^{-6} s^{-1} and 10^{-3} s^{-1}. To record the stress-strain relationship, clip-on expensometers were used. Crosshead displacements were converted to engineering strain, while the resulting force was used to compute the engineering stress. For the creep experiments, the gauge length was 100 mm for the off-axis angles 90° and 45°, and 120 mm for the smaller angles 30° and 15°. In these experiments, constant force was applied on the coupon, from which the engineering stress was calculated and the crosshead displacements gave the engineering strain. Further details on the experiments setup can be found in [2, 3], that is based on [25].

6.2.2. MULTISCALE PROBLEM FORMULATION

To replicate the experiments discussed in the previous section, a concurrent multiscale approach (FE^2) can be used. In this case, the macroscopic domain Ω is discretized into a FE mesh, with a periodic representative volume element (RVE) ω nested to each integration point. The RVE consists of another FE model that characterises the heterogeneous material at a length scale significantly lower than the macroscopic one.

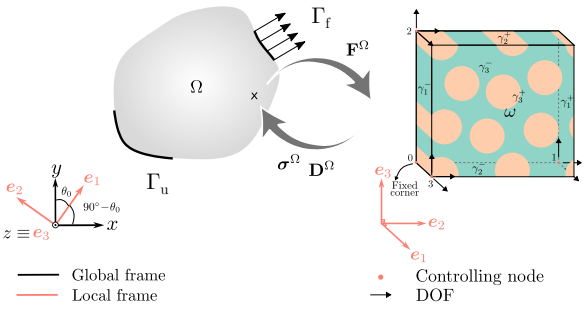


Figure 6.2: Scheme of concurrent multiscale framework with two scales (macro Ω and micro ω) for composite material with off-axis loading.

In this chapter, the updated Lagrangian (UL) formulation is adopted [26]. As discussed in Appendix A of Chapter 5, this means that the incremental equilibrium problem is solved using the work-conjugate pair of Cauchy stress and the spatial gradient of the displacement increment from the reference to the updated configuration, while the constitutive model is formulated in terms of the total deformation gradient.

At the microscale, regular constitutive models can be assigned to each of the phases. To solve the micromodel problem, periodic boundary conditions based on the macroscopic deformation gradient $\mathbf{F}^\Omega$ are employed. Further, if the local coordinate system of the RVE, in which fiber direction is always parallel to $\boldsymbol{e}_1$, is not aligned with the global

one, we relate the two coordinate systems through a transformation matrix:

$$\mathbf{Q}_0(\theta_0) = \begin{bmatrix} \cos(90^\circ - \theta_0) & \sin(90^\circ - \theta_0) & 0 \\ -\sin(90^\circ - \theta_0) & \cos(90^\circ - \theta_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

where θ_0 marks the angle between the y-axis and the fiber direction, as depicted in Fig. 6.2. This matrix allows $\mathbf{F}^\Omega$ to be transformed from the global to the local coordinate system using the standard change of basis of second-order tensors [26]:

$$\mathbf{F}_\mathrm{L}^\Omega = \mathbf{Q}_0 \mathbf{F}^\Omega \mathbf{Q}_0^\mathrm{T} \tag{6.2}$$

where the subscript "L" refers to the local coordinate system. Once the microscopic problem has converged, a computational homogenization procedure is performed to bridge the two scales:

$$\boldsymbol{\sigma}_\mathrm{L}^\Omega(\mathbf{x}^\Omega) = \frac{1}{|\omega|}\int_\omega \boldsymbol{\sigma}^\omega(\mathbf{x}^\omega)\,\mathrm{d}\omega \tag{6.3}$$

where $\mathbf{x}^\Omega$ and $\mathbf{x}^\omega$ are the spatial coordinates of the material points at the micro and macroscales in the updated configuration t_end. Similarly, to transform the stress from the local to the global coordinate system, we use

$$\boldsymbol{\sigma}^\Omega = \mathbf{Q}_0^\mathrm{T} \boldsymbol{\sigma}_\mathrm{L}^\Omega \mathbf{Q}_0. \tag{6.4}$$

The formulation is completed with the definition of the macroscopic constitutive tangent $\mathbf{D}^\Omega$. For that, automatic differentiation, perturbation methods based on finite differences or condensation procedures can be used depending on the memory allocation and computational efficiency requirements [27, 28]. In this chapter, our reference solutions are experiments and we do not perform the full-order FE^2 due to the exceedingly high computational cost. The main bottleneck, in this case, comes from the coupling between the two equilibrium problems, where the solution of the macroscopic displacement field defines the boundary condition for the RVEs, which in turn provide the missing homogenized constitutive model, requiring iterative solves of a large number of microscopic FE problems. In Section 6.2.4 we elaborate on the alternatives to this approach explored here.

6.2.3. CONSTITUTIVE MODELS

In this section, we discuss the two constitutive models used in the composite RVE adopted for the applications. These are the same models as in Chapter 5 and reference works [1–3], and we therefore skip their derivation and only summarize their main features and the novelties explored in this chapter. The first constitutive model is a hyperelastic transversely isotropic model based on the formulation by [29] with slight modifications from [1], which is assigned to the fibers and is referred to as $\mathcal{C}^\omega_\mathrm{fiber}$. To describe this model, five elastic constants are needed: the Young's modulus in the preferential stiffness direction and in the plane of isotropy, the shear modulus and the Poisson's ratio in the plane perpendicular to the isotropic plane and the Poisson's ratio in the plane of isotropy.

For the matrix, we use the Eindhoven Glassy Polymer (EGP), a rate and path-dependent elasto-viscoplastic model, here referred to as $\mathcal{C}^\omega_\mathrm{matrix}$. The Cauchy stress from this model

consists of three contributions. The first is a hydrostatic one, which is hyperelastic, and depends on the bulk modulus κ as

$$\boldsymbol{\sigma}_{\mathrm{h}} = \kappa\,(J-1)\,\mathbf{I} \tag{6.5}$$

where J is the determinant of the deformation gradient. The second contribution is a hardening part that represents polymer chain reorientation, and depends on the hardening modulus G_{r} as

$$\boldsymbol{\sigma}_{\mathrm{r}} = G_{\mathrm{r}}\tilde{\mathbf{B}}^{\mathrm{d}} \tag{6.6}$$

where $\tilde{\mathbf{B}}^{\mathrm{d}}$ is the deviatoric part of the isochoric left Cauchy-Green deformation tensor.

Finally, the third contribution of the EGP comes from the driving stress, which is the rate and path-dependent component. This component is given by the sum of an arbitrary number of Maxwell elements, also known as *modes*, connected in parallel. The main benefit of a multi-mode formulation is the higher flexibility of the model in fitting the pre-yield response [30]. With EGP, multiple relaxation *processes* can also be considered to represent the thermorheologically complex behavior of the material [31]. For two processes, say α and β, the driving stress can be further split into two contributions as

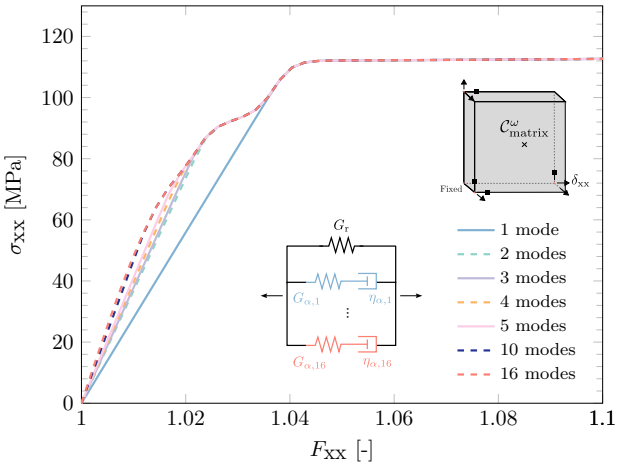
$$\begin{aligned}\boldsymbol{\sigma}_{\mathrm{s}} &= \boldsymbol{\sigma}_{\mathrm{s},\alpha} + \boldsymbol{\sigma}_{\mathrm{s},\beta} \\ &= \sum_{i=1}^{a} \boldsymbol{\sigma}_{\mathrm{s},\alpha,i} + \sum_{j=1}^{b} \boldsymbol{\sigma}_{\mathrm{s},\beta,j} \\ &= \sum_{i=1}^{a} G_{\alpha,i}\tilde{\mathbf{B}}^{\mathrm{d}}_{\mathrm{e},\alpha,i} + \sum_{j=1}^{b} G_{\beta,j}\tilde{\mathbf{B}}^{\mathrm{d}}_{\mathrm{e},\beta,j}\end{aligned} \tag{6.7}$$

where $G_{x,k}$ is the shear modulus, $\tilde{\mathbf{B}}^{\mathrm{d}}_{\mathrm{e},x,k}$ is the deviatoric part of the isochoric elastic left Cauchy-Green deformation tensor, with k referring to the mode, x to the process, and a and b are the total number of modes considered in the processes α and β, respectively.

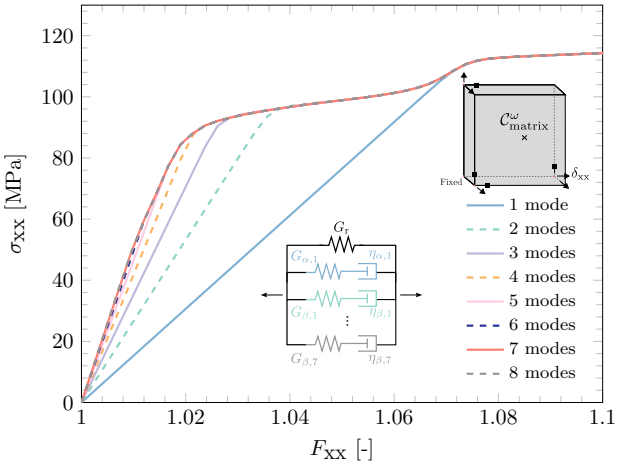
We highlight that the EGP modes are not standard linear Maxwell elements. In each dashpot k, $\tilde{\mathbf{B}}^{\mathrm{d}}_{\mathrm{e}x,k}$ is calculated by integrating an evolution equation in which the following constitutive relation is introduced to define the rate of plastic deformation as a non-Newtonian flow rule

$$\mathbf{D}_{\mathrm{p},x,k} = \frac{\boldsymbol{\sigma}_{\mathrm{s}x,k}}{2\eta_{x,k}(\overline{\tau}_{\mathrm{s},x}, T, p, S_x, \eta_{0x,k}, \tau_{0x}, \lambda_x)} \tag{6.8}$$

where $\eta_{x,k}$ is the viscosity function that depends on the equivalent stress applied $\overline{\tau}_{\mathrm{s},x} = \sqrt{1/2\,\boldsymbol{\sigma}_{s,x} : \boldsymbol{\sigma}_{s,x}}$, the temperature T, the hydrostatic pressure p, the thermodynamic state parameter S_x, which accounts for two competing mechanisms (physical aging and mechanical rejuvenation), and given initial viscosity $\eta_{0x,k}$, characteristic shear stress τ_{0x} and pressure dependency parameter λ_x. This also means no explicit yield function is assumed. As a result, plastic flow is always present, but its magnitude depends strongly on the (highly nonlinear) viscosity. It is worth mentioning that we use the common spring–dashpot analogy only as a mechanical representation of the elastic and rate-dependent mechanisms, but the dashpot strain in EGP represents practically irreversible, stress-induced plastic flow.



(a) Material properties from $\boldsymbol{\mu}^{\dot{\varepsilon}}_{\text{matrix}}$



(b) Material properties from $\boldsymbol{\mu}^{\text{creep}}_{\text{matrix}}$

Figure 6.3: Stress-strain curves using $\mathcal{C}^{\omega}_{\text{matrix}}$ with different number of modes from $\boldsymbol{\mu}^{\dot{\varepsilon}}_{\text{matrix}}$ and $\boldsymbol{\mu}^{\text{creep}}_{\text{matrix}}$ considering uniaxial loading with $\delta_{xx} = 5 \times 10^{-5}$ mm and $\Delta t = 1$ s. Arrows on the element (gray box) are degrees of freedom that are either constrained (square tip) or free (triangular tip).

In this study, we adopt the same set of material properties obtained in [2, 3], where the authors used the experiments reproduced in this work for calibration. We use the symbol $\boldsymbol{\mu}^{(\cdot)}_{(\star)}$ to refer to the properties of a given material, where $(\star)$ is either the *matrix* or the *fiber*, and $(\cdot)$ refers to constant strain-rate $(\dot{\varepsilon})$ or *creep* experiments. For the constant strain-rate case, 1 process and 16 modes were needed for the best fit with experiments,

while for creep, 2 processes and 8 modes were enough, from which 1 mode belongs to α and 7 to β. For illustrative purposes, we plot in Fig. 6.3 the response of a single material point under uniaxial loading with $\mathcal{C}^{\omega}_{\text{matrix}}$ as the constitutive model and different number of modes. Note how in both scenarios, although the pre-yield regime response changes significantly with more modes, the post-yield response (plastic regime) is the same. This originates from the model definition where the equivalent plastic strain is computed based on equivalent stress associated with the mode of highest viscosity (*i.e.* longest relaxation time). With the modes in descending order of magnitude of viscosity, that always corresponds to the first one.

6.2.4. ALTERNATIVES TO A FULL MULTISCALE FORMULATION

In Fig. 6.4, we illustrate two alternatives to a full multiscale formulation: the single-scale micromechanical approach proposed in [1–3], and our contribution, the surrogate-based multiscale approach. The single-scale model consists in a custom constant strain-rate arclength [2]. Although highly efficient and accurate in many scenarios, the assumptions made to simplify the entire specimen macroscopic domain to a single macroscopic point restrict its validity to uniform macroscopic fields. This limitation motivates the use of a surrogate-based multiscale approach, which allows macroscopic variability and avoids the prohibitive cost of a full multiscale simulation.

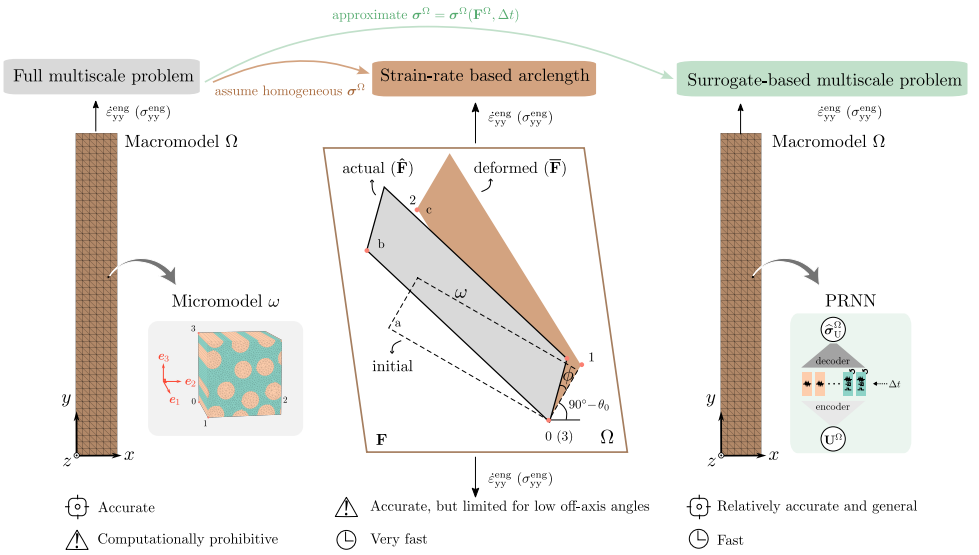


Figure 6.4: Alternative approaches to replace full multiscale problem (left) for modeling composites under off-axis loading: single-scale micromechanical simulation with special boundary conditions (center) and surrogate-based multiscale problem where PRNN replaces the micromodel to solve the macroscopic problem (right).

We emphasize that the single-scale micromodel approach is used here only as a refer-

ence to (i) help identify the impact of the macroscopic variability in the surrogate-based multiscale simulations in improving the fit with experiments, and to (ii) examine the hypothesis in [2, 3] regarding the mismatch at lower off-axis angles. As mentioned in Section 6.2.3, we also adopt the same material parameters calibrated with the single-scale approach for consistency. This way, differences in the results between the two approaches cannot be attributed to different parameter identification strategies. Other calibration strategies that included the surrogate-based multiscale approach in their framework could in principle be employed, but they fall outside the scope of the present contribution.

A slight change was implemented with respect to the equations shown in the original formulations for the constant strain-rate case due to the different choice of strain and stress measures [1, 2]. This change was motivated by the type of quantities reported in the experiments. Instead of imposing a constant true strain-rate to achieve a uniaxial true (Cauchy) stress, here we impose, at the global frame, a constant engineering strain-rate ($\dot{\varepsilon}^{\text{eng}}_{\text{yy}}$) to achieve uniaxial engineering stress ($\sigma^{\text{eng}}_{\text{yy}}$) using

$$\varepsilon^{\text{eng}}_{\text{yy},t} = \varepsilon^{\text{eng}}_{\text{yy},t-1} + \dot{\varepsilon}^{\text{eng}}_{\text{yy}} \Delta t \tag{6.9}$$

where $\varepsilon^{\text{eng}}_{\text{yy},t}$ and $\varepsilon^{\text{eng}}_{\text{yy},t-1}$ are the engineering strains applied at the current and previous time steps at the global frame, respectively.

For modeling creep, the authors in [3] incorporated the use of engineering strains and stresses in their formulation. The new model was adapted from their previous developments, with the main changes regarding the inclusion of the creep stress in the external force components for a force-controlled analysis and the consideration of the following condition to emulate the loading observed in the experiments:

$$\sigma^{\text{eng}}_{\text{yy,t}} = \min\left(\sigma^{\text{eng}}_{\text{yy},t-1} + \dot{\sigma}^{\text{eng}}_{\text{yy}} \Delta t, \sigma^{\text{max}}_{\text{yy}}\right) \tag{6.10}$$

where $\dot{\sigma}^{\text{eng}}_{\text{yy}}$ is the engineering stress-rate at the loading phase before reaching the maximum (constant) stress level $\sigma^{\text{max}}_{\text{yy}}$.

Although we do not discuss the micromodel-based formulations in detail, it is worth mentioning an important aspect we will later contrast with the surrogate-based simulations: the (potential) reorientation of the microstructure during the loading process, represented by the angle ϕ in Fig. 6.4. This reorientation leads to the reduction of the initial off-axis angle, creating a stiffening effect as the fibers try to align with the global loading direction y, and is especially evident in cases with lower off-axis angles, where even small angle variations can lead to significantly different stress states. The procedure to compute this angle is detailed in [2, 3], and depends on a series of transformations between frames of the deformation gradient (see Fig. 6.4). Here we highlight that, in the experiments, these rotations vary across the coupon, as they are constrained by the presence of the clamps of the testing machine.

6.2.5. PHYSICALLY RECURRENT NEURAL NETWORK (PRNN)

In contrast to the aforementioned microscale-based approach, we replace the RVE with a PRNN trained on stress-strain snapshots (see Fig. 6.4). We summarize in Algorithm 2 the

main coordinate system transformations needed to handle global and local frames when evaluating the homogenized stress response and the chain rule used for computing the macroscopic tangent stiffness matrix.

Algorithm 2: Evaluation of macroscopic integration point using the PRNN

Input : homogenized deformation gradient at global frame $\mathbf{F}^{\Omega}$, time increment Δt, initial off-axis angle θ_0

Output: homogenized stress $\widehat{\boldsymbol{\sigma}}^{\Omega}$, tangent stiffness matrix $\mathbf{D}^{\Omega}$

1 compute transformation matrix $\mathbf{Q}_0$ according to Eq. (6.1) based on θ_0

2 transform strain from global to local frame: $\mathbf{F}_{\mathrm{L}}^{\Omega} \leftarrow \mathbf{Q}_0 \mathbf{F}^{\Omega} \mathbf{Q}_0^{\mathrm{T}}$

3 perform polar decomposition on strain: $\mathbf{R}_{\mathrm{L}}^{\Omega}, \mathbf{U}_{\mathrm{L}}^{\Omega} \leftarrow \texttt{polarDecomposition}(\mathbf{F}_{\mathrm{L}}^{\Omega})$

4 apply PRNN to stretch tensor: $\widehat{\boldsymbol{\sigma}}_{\mathrm{U}}^{\Omega}, \mathbf{D}_{\mathrm{U}}^{\Omega} \leftarrow \texttt{PRNN}(\mathbf{U}_{\mathrm{L}}^{\Omega}, \Delta t)$

5 retrieve stress in the original coordinate system at the local frame: $\widehat{\boldsymbol{\sigma}}_{\mathrm{L}}^{\Omega} \leftarrow \mathbf{R}_{\mathrm{L}}^{\mathrm{T}} \widehat{\boldsymbol{\sigma}}_{\mathrm{U}}^{\Omega} \mathbf{R}_{\mathrm{L}}^{\Omega}$

6 transform stress from local to global frame: $\widehat{\boldsymbol{\sigma}}^{\Omega} \leftarrow \mathbf{Q}_0^{\mathrm{T}} \widehat{\boldsymbol{\sigma}}_{\mathrm{L}}^{\Omega} \mathbf{Q}_0$

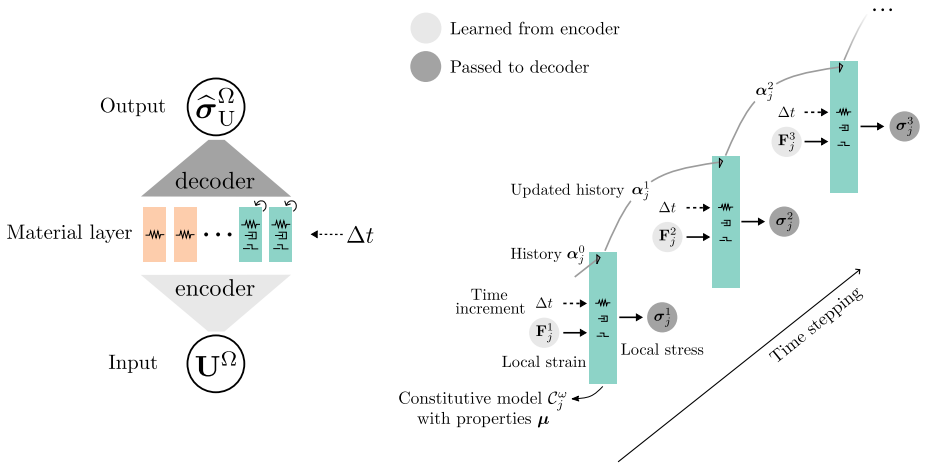
7 compute tangent stiffness matrix: $\mathbf{D}^{\Omega} \leftarrow \frac{\partial \widehat{\boldsymbol{\sigma}}^{\Omega}}{\partial \widehat{\boldsymbol{\sigma}}_{\mathrm{L}}^{\Omega}} \left(\frac{\partial \widehat{\boldsymbol{\sigma}}_{\mathrm{L}}^{\Omega}}{\partial \widehat{\boldsymbol{\sigma}}_{\mathrm{U}}^{\Omega}} \frac{\partial \widehat{\boldsymbol{\sigma}}_{\mathrm{U}}^{\Omega}}{\partial \mathbf{U}_{\mathrm{L}}^{\Omega}} \frac{\partial \mathbf{U}_{\mathrm{L}}^{\Omega}}{\partial \mathbf{F}_{\mathrm{L}}^{\Omega}} + \frac{\partial \widehat{\boldsymbol{\sigma}}_{\mathrm{L}}^{\Omega}}{\partial \mathbf{R}_{\mathrm{L}}^{\Omega}} \frac{\partial \mathbf{R}_{\mathrm{L}}^{\Omega}}{\partial \mathbf{F}_{\mathrm{L}}^{\Omega}} \right) \frac{\partial \mathbf{F}_{\mathrm{L}}^{\Omega}}{\partial \mathbf{F}^{\Omega}}$

8 **return** $(\widehat{\boldsymbol{\sigma}}^{\Omega}, \mathbf{D}^{\Omega})$

For details on implementation, training aspects or architectural choices, the reader is directed to Chapter 5. Here, we highlight the main features of the network. In the PRNN, we preserve the constitutive models used in the micromodel, as well as their material properties, and embed them in an encoder-decoder architecture, as illustrated in Fig. 6.5a. Through the encoder, we learn a set of values that we interpret as the local strain of *fictitious microscopic material points*. Then, the constitutive model associated to each point is used to compute stresses and, in case of a history-dependent model, updated internal variables. Storing the internal variables from one time step to another, as illustrated in Fig. 6.5b for a rate and path-dependent model $\mathcal{C}_j^{\omega}$, allows history dependence to arise naturally. Compared to the full-order model, the encoder learns a task analogous to the solution of the original microscale equilibrium problem, and the decoder corresponds to a stress homogenization operation.

Having constitutive models in the network also raises a few constraints that regular NNs do not need to meet. For example, since the output of our encoder are deformation gradients, these must have positive determinant. For that purpose, and taking into account the zero strain-stress state ($\mathbf{U}^{\Omega} = \mathbf{I} \rightarrow \widehat{\boldsymbol{\sigma}}_{\mathrm{L}}^{\Omega} = \mathbf{0}$), we propose to constrain the weights from the encoder to be a symmetric matrix per material point. This arrangement avoids negative determinants and only requires 6 learnable parameters per material point. For the decoder, we employ a sparse architecture, in which only the matching stress components contribute to the macroscopic stress. Therefore, including encoder and decoder, each material point is associated with 12 learnable parameters.

As for the constitutive model itself, both models present in the RVE, $\mathcal{C}_{\text{matrix}}^{\omega}$ and $\mathcal{C}_{\text{fiber}}^{\omega}$, are embedded in the material layer. The splitting ratio between them is a hyper-parameter, but here we follow with the proportion from Chapter 5: the hyperelastic and elasto-viscoplastic models correspond to 25 % and 75 % of the material points, respectively, rounding the number of hyperelastic models up when the total number of points



(a) Compact representation of PRNN

(b) Fictitious material point with history-dependent constitutive model unrolled in time

Figure 6.5: Architecture of PRNN and fictitious material point unrolled in time.

is even but not divisible by 4.

6.2.6. TRANSFER LEARNING

In this section, we take a step further and explore how one can leverage the design choices in the material layer of a PRNN to transfer from one set of material properties ($\boldsymbol{\mu}^{\mathrm{A}}$) to another ($\boldsymbol{\mu}^{\mathrm{B}}$). For this particular work, the goal is to use the networks trained in Chapter 5, which were then used to predict the homogenized response of an RVE with constitutive models $\mathcal{C}^{\omega}_{\text{fiber}}$ and single-mode $\mathcal{C}^{\omega}_{\text{matrix}}$, to now predict the response of the same RVE considering a multi-mode $\mathcal{C}^{\omega}_{\text{matrix}}$.

The solution to that if a regular data-driven surrogate model was used would involve either training from scratch with the full 16 modes data or use some transfer learning strategy, such as warm start. In both cases, additional computational effort would be necessary to (re)train the surrogates with different properties. Here, because material properties in the constitutive model are embedded in the material layer, we can keep the network trained on a single mode from the previous chapter and simply update the material properties in the network without any retraining. This scheme is illustrated in Fig. 6.6 with the models used in this work, but the approach is general and applicable to other constitutive models. The ability to extrapolate to different material properties in the online phase is yet another benefit of having an explainable function in the latent space.

For PRNNs, in particular, another reason for extrapolating from the single-mode networks comes from the fact that each mode in $\mathcal{C}^{\omega}_{\text{matrix}}$ is associated to 15 internal variables, therefore to account for a full relaxation spectrum with 16 modes, a total of 240 internal variables would be needed per material point evaluated by this constitutive model.

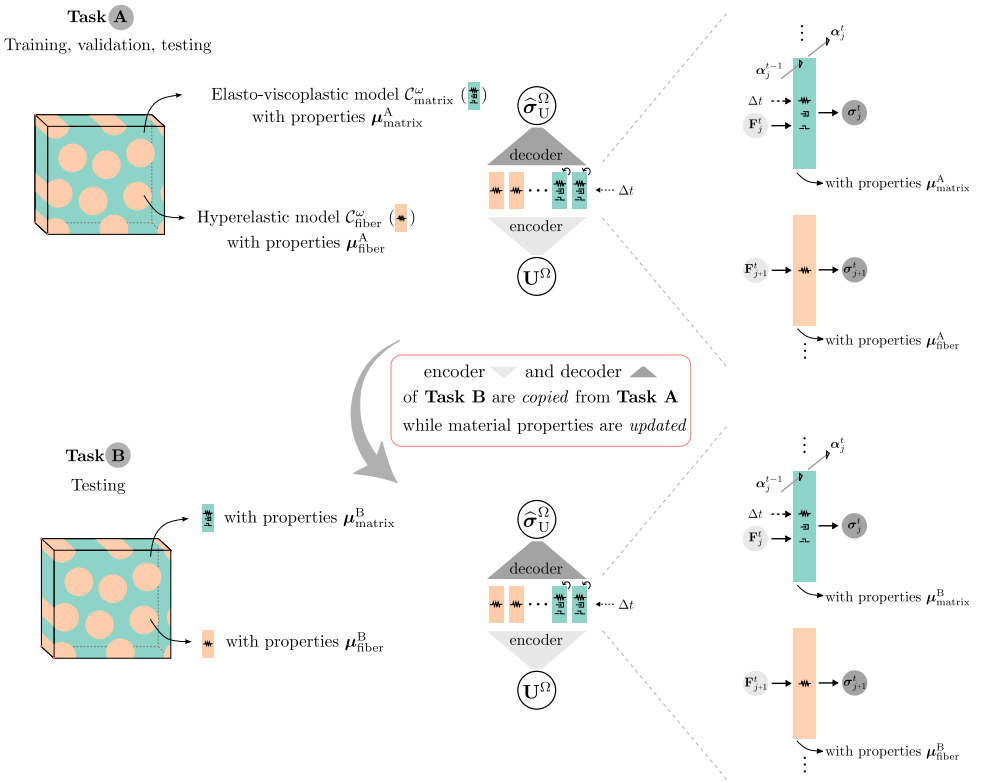


Figure 6.6: From training with micromodel with a set of properties $\boldsymbol{\mu}^{\text{A}}_{(\cdot)}$ to extrapolating to micromodel with set of properties $\boldsymbol{\mu}^{\text{B}}_{(\cdot)}$ without retraining.

Such large sum of internal variables combined with the use of finite differences to compute the derivatives related to the backpropagation in time would slow down training significantly, as well as the time for obtaining the training loading paths.

6.3. CONSTANT STRAIN-RATE EXPERIMENTS

In this section, we apply the transfer learning approach in Section 6.2.6 to go from a PRNN with single-mode $\mathcal{C}^{\omega}_{\text{matrix}}$ model to a multi-mode one, and present the results obtained with the PRNN-based approach to the multiscale problem, from now on referred to as "FEPRNN", for modeling constant strain-rate experiments. The mesh adopted for these simulations is shown in Fig. 6.4, and consists of 576 (linear) wedge elements integrated with 1 point, with the PRNN discussed in Section 6.3.1 as the homogenized constitutive model.

To simulate the loading and boundary conditions of the experiments, we employ a displacement controlled analysis, with the displacements in all three directions being constrained at the bottom and top surfaces of the coupon, except for the y direction

at the top, where the specimen is pulled at a constant engineering strain-rate. We employ the adaptive stepping scheme from [32] to deal with potential convergence issues, using time increments between 0.1 s and 100 s, and compare results with the experimental stress-strain curves for a range of off-axis angles and strain-rates. In most cases, the micromodel response, referred as to "Micro" in the plots, with and without the fiber rotation update ϕ, is also plotted for reference. In addition to that, we include a brief study on the use of oblique end-tabs as a possible adaptation to the experiments for a more uniform strain distribution.

6.3.1. FROM SINGLE TO MULTI-MODE PRNN

To assess the performance of the networks from Chapter 5 trained with single-mode properties in extrapolating to the RVE with the full relaxation spectrum $\boldsymbol{\mu}^{\dot{\varepsilon}}_{\text{matrix}}$, we consider the networks trained on the largest training set from that work, that is, 144 proportional curves with loading/unloading/reloading based on Gaussian Process (GP), from now on referred to as GP-based paths, with 8 material points in total (from which 6 are evaluated by $\mathcal{C}^{\omega}_{\text{matrix}}$ and 2 by $\mathcal{C}^{\omega}_{\text{fiber}}$).

In this particular application, the update in the material properties offers a further possibility for maximizing efficiency. Instead of considering the full relaxation spectrum at once, we investigate a gradual mode addition in the online phase. The idea comes from the fact that the latest modes have increasingly small contributions to the elastic regime, as illustrated in Fig. 6.3, and could, in principle, be left out from the PRNN without loss of accuracy, leading to an even faster model evaluation.

For this task, we select the best network over test set $\mathcal{T}_1$, which consists of 150 curves with the same loading type and material properties as the ones used for training. This loading type is deemed to be representative of various loading conditions as each curve has a different loading/unloading/reloading behavior, as well as time step size, in an unseen direction. Then, on a new test set $\mathcal{T}_{16}$ with 150 curves generated using the 16 modes in $\mathcal{C}^{\omega}_{\text{matrix}}$, we assess the accuracy of the chosen network by updating the material properties considering a gradual addition of modes in the fictitious material points evaluated by $\mathcal{C}^{\omega}_{\text{matrix}}$, as shown in Fig. 6.7. Clearly, as modes are added, accuracy is increased, reaching the lowest error around 9 modes with relative error of 5.9 %.

6.3.2. COMPARISON WITH EXPERIMENTAL RESULTS

Previously, we had an indicative of the smallest number of modes needed in the PRNN for accurately extrapolating to an RVE with 16 modes in the matrix, which is the reference in the micromodel-based approach in [2]. Here, we follow a similar exercise in terms of gradually increasing the number of modes, this time comparing the surrogate-based approach directly with the experiments for a final decision. For this, we select the experiment with $\theta_0 = 90°$ and $\dot{\varepsilon}^{\text{eng}}_{\text{yy}} = 10^{-4}\,\text{s}^{-1}$. With the fibers perpendicular to the loading direction, the matrix becomes the primary load-bearing component, making the contribution of each mode more pronounced in the overall homogenized response, and therefore the ideal case for assessing the extrapolation capabilities of the network trained on the first mode only. Fig. 6.8 shows the stress-strain curves and the relative mean error of the stresses predicted by the FEPRNN in comparison to the experiments.

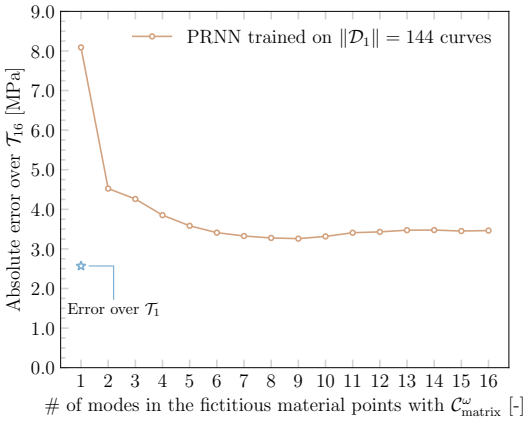


Figure 6.7: Accuracy of PRNN trained on 144 proportional GP-based curves and a single mode over test set $\mathcal{T}_{16}$ considering different number of modes in the fictitious material points evaluated by $\mathcal{C}^{\omega}_{\text{matrix}}$.

These errors are calculated by evaluating the surrogate-based multiscale response at the strain values from the experiment, using linear interpolation between the two nearest pairs of strain-stress values when needed according to

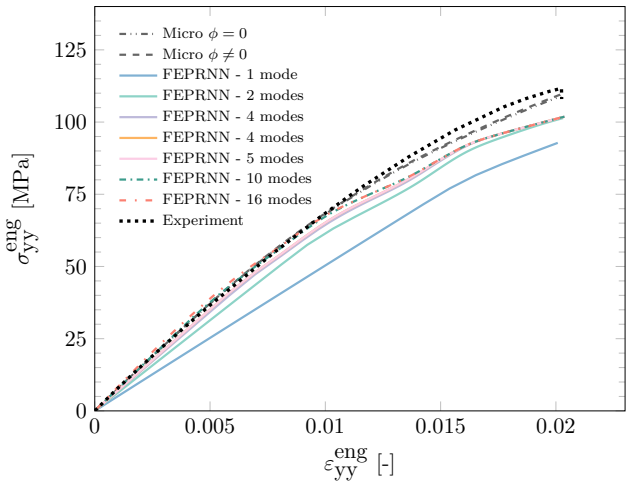
$$\text{Relative mean error:} \frac{1}{n^{\dot{\varepsilon}}_{\text{exp}}} \sum_{i=1}^{n^{\dot{\varepsilon}}_{\text{exp}}} \frac{|\sigma^{\text{eng}}_{\text{yy,FEPRNN}}(\varepsilon^{\text{eng}}_{\text{yy,exp},i}) - \sigma^{\text{eng}}_{\text{yy,exp}}(\varepsilon^{\text{eng}}_{\text{yy,exp},i})|}{|\sigma^{\text{eng}}_{\text{yy,exp}}(\varepsilon^{\text{eng}}_{\text{yy,exp},i})|}, \tag{6.11}$$

where $n^{\dot{\varepsilon}}_{\text{exp}}$ is the number of pairs of strain-stress available in the experiment. Note in Fig. 6.8a how the stress-strain curves change only slightly with five or more modes, with relative errors between 5 % and 9 %. Based on that, for the remainder of this section, we use five modes in all FEPRNN constant strain-rate simulations, as this represents a good balance between accuracy and efficiency.

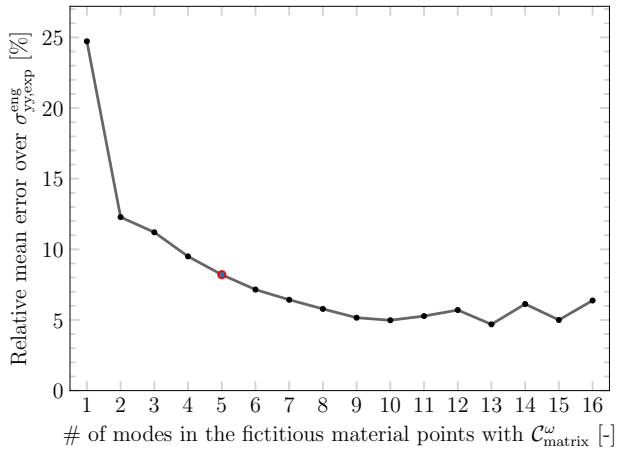
Next, we investigate the worst case scenario in [2], namely off-axis angle $\theta_0 = 15°$ and strain-rate $\dot{\varepsilon}^{\text{eng}}_{\text{yy}} = 10^{-4}\,\text{s}^{-1}$. In Fig. 6.9a, despite the slight mismatch around $\epsilon^{\text{eng}}_{\text{yy}} = 0.01$, the FEPRNN response closely follows the experiment in the remaining parts of the curve, resulting in a relative mean error of 5.4 %. An overview of how the FEPRNN and micromodel-based solutions (with and without the reorientation angle ϕ) compare with the experiment for different off-axis angles is shown in Fig. 6.9b. It becomes clear how the reorientation can impact heavily the final response depending on the initial off-axis angle, and how, for most cases, the FEPRNN response shows better agreement with the experiments than the micromodel-based solution, except for $\theta_0 = 90°$. With fibers perpendicular to the loading direction, the matrix is the primary load-bearing component, making the contribution from each mode more pronounced, while the network (trained on the first mode only) relies on extrapolation to predict the multi-mode response. At the same time, at lower stress magnitudes, the FEPRNN can be slightly less accurate due

to our choice of loss function, namely the mean squared error, during training. Nevertheless, the overall agreement remains satisfactory, with mean relative error of 8.2 %. The relative mean errors of the 30° and 45° cases are 3.8 % and 2.9 %, respectively.

Although the micromodel-based results are not new, it is only now possible to verify 1) how well the reorientation angle computed from that solution matches with the average angle computed from the PRNN-based simulation and 2) how well the assumption

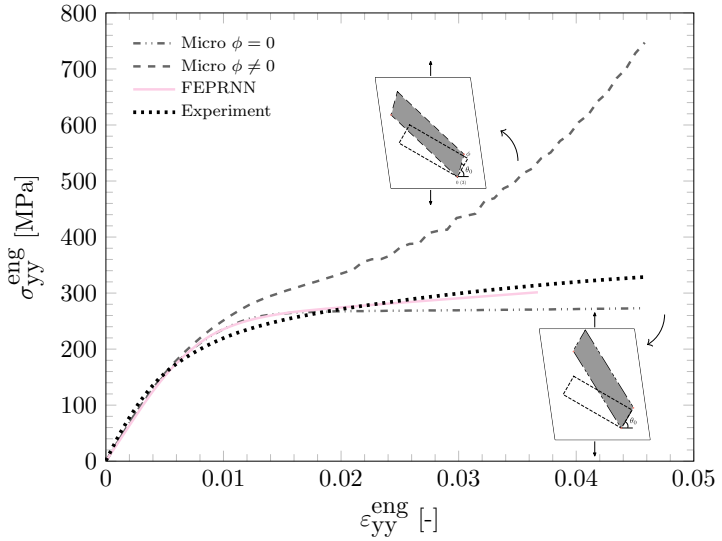


(a) FEPRNN with different number of modes extracted from $\boldsymbol{\mu}^{\dot{\varepsilon}}_{\text{matrix}}$

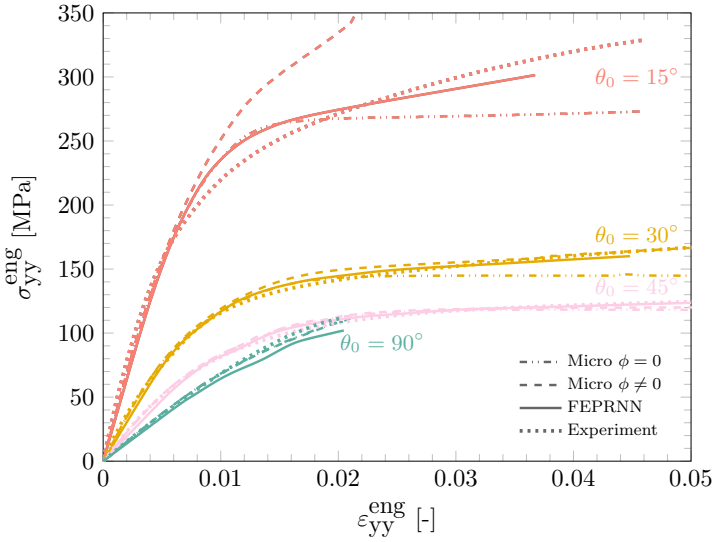


(b) Relative mean errors of FEPRNN with respect to experiment

Figure 6.8: Stress-strain curves for off-axis angle 90° and $\dot{\varepsilon}^{\text{eng}}_{\text{yy}} = 10^{-4}\,\text{s}^{-1}$ obtained by the micromodel-based solution (16 modes) and the FEPRNN with different number of modes in the fictitious material points evaluated by $\mathcal{C}^{\omega}_{\text{matrix}}$ compared to experiment.



(a) Varying boundary conditions in micromodel for $\theta_0 = 15°$



(b) Different off-axis angles and micromodel boundary conditions

Figure 6.9: Stress-strain curve for $\dot{\varepsilon}_{yy}^{eng} = 10^{-4}\,s^{-1}$ using different methods and boundary conditions for multiple off-axis angles.

of macroscopic homogeneity holds. For this end, we show in Fig. 6.10 the full field of strains in the loading direction for different off-axis angles. Note how the variation in the macroscopic strain becomes larger as the off-axis angle θ_0 decreases. Clearly, the assumption behind the micromechanical analysis, that the specimen is in a macroscop-

ically uniform state, becomes increasingly inaccurate as θ_0 decreases.

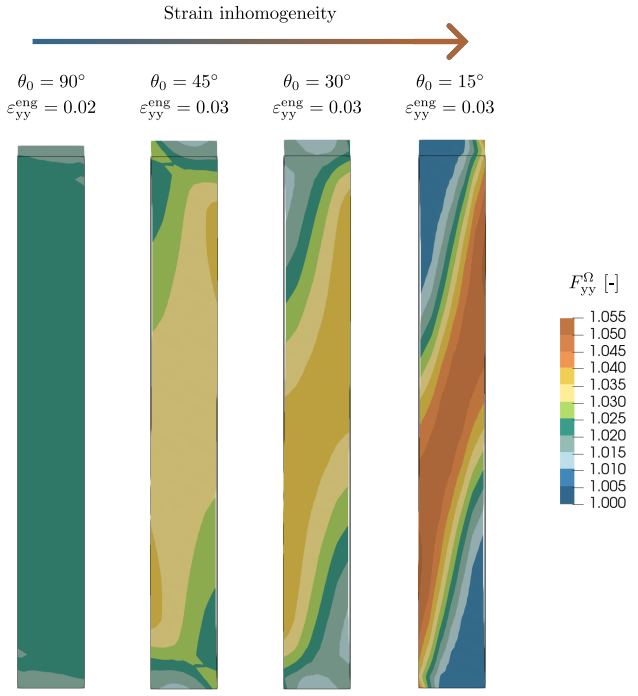


Figure 6.10: Macroscopic strains from FEPRNN simulations for $\dot{\varepsilon}_{yy}^{eng} = 10^{-4}\,s^{-1}$ and different off-axis angles.

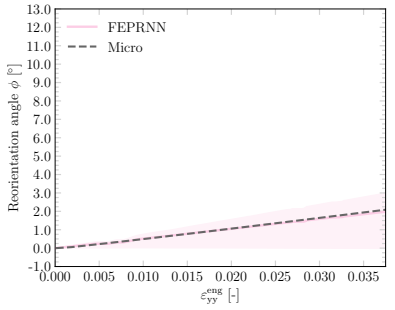
To further illustrate the problem, we calculate the reorientation angle ϕ at each of the macroscopic integration points with the surrogate-based multiscale approach. For this, we make a parallel with the deformation states in Fig. 6.4 and imagine that there is an equivalent shape of $\hat{\mathbf{F}}^{\Omega}$, whose side 0 – 1 is not moving in direction 2, and therefore $\overline{F}_{21}^{\Omega} = 0$. From that condition, the angle ϕ can be expressed as

$$\phi = \arctan(\hat{F}_{21}^{\Omega} / \hat{F}_{11}^{\Omega}) \tag{6.12}$$

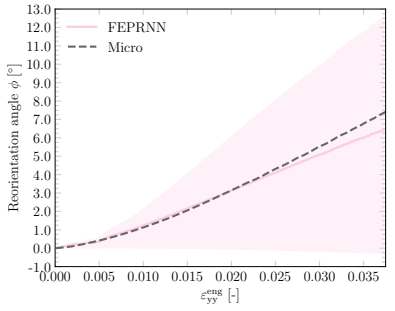
where $\hat{F}_{21}^{\Omega}$ and $\hat{F}_{11}^{\Omega}$ are components of the deformation gradient in the local frame (see Eq. (6.2)). Based on this expression, we compute the simple average of ϕ over the entire macroscopic domain for the off-axis angles 45° and 15° and compare it with the one calculated by the custom arclength model, as shown in Fig. 6.11. In addition to the average, the envelope corresponding to the highest and lowest angles over the entire specimen are plotted (the pink shaded areas). Note how the angle variation is dramatically higher in the 15° case. For both angles, the highest values are located at the center of the specimen (see Fig. 6.11c) and the lowest ones, close to zero, near the grips, where movement is restricted.

In the $\theta_0 = 45°$ case, the mean reorientation angle follows remarkably well the angle computed from the arclength model, which is also translated in the good visual agree-

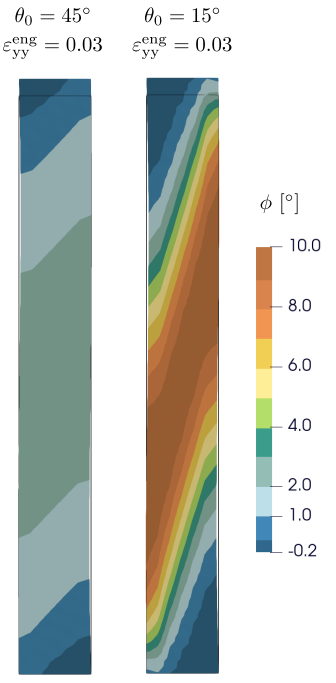
ment seen in Fig. 6.9. For the 15° composite, the envelope of the angle ϕ becomes increasingly larger and the single micromodel loses representativeness for the average response of the specimen, more strongly in the averaged stress-strain curve of fig. 6.9a than in the mean orientation angle in fig. 6.11b.



(a) $\theta_0 = 45°$



(b) $\theta_0 = 15°$



(c) Snapshots of full field of angle ϕ using PRNN

Figure 6.11: Reorientation angle for different initial off-axis angles for constant strain-rate experiment with $\dot{\varepsilon}_{yy}^{eng} = 10^{-4}\,s^{-1}$. Shaded areas in a) and b) represent the envelope of maximum and minimum values over the entire macroscopic domain, while solid lines correspond to the average.

In addition to that, the multiscale simulations showed the presence of shear stresses, suggesting that the boundary conditions in the experiments are not only far from the assumptions in the arclength model, but also from its own initial goal, a uniaxial tensile test. Finally, we illustrate the generality of the framework with two other strain rates for the 30° case: $10^{-3}\,s^{-1}$ and $10^{-5}\,s^{-1}$. Fig. 6.12 shows the good match of the two engineering stress-strain curves obtained using the PRNN as the constitutive model with respect to the experimental curves. In both cases, the maximum error is around 5 %, at the last time step.

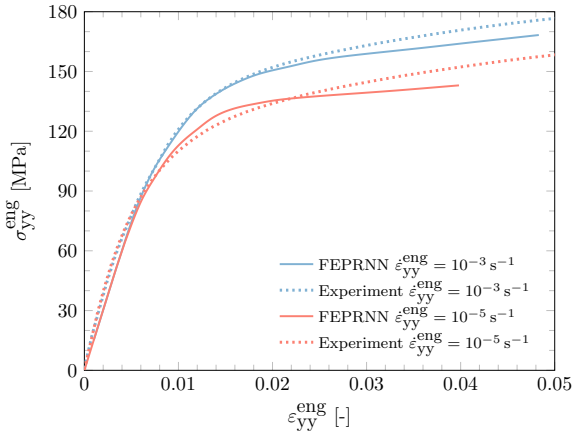


Figure 6.12: Stress-strain curves for off-axis angle $\theta_0 = 30°$ and different strain-rates $\dot{\varepsilon}_{yy}^{eng}$.

6.3.3. INVESTIGATION INTO ALTERNATIVE SETUPS

With an efficient multiscale framework, we can also investigate the influence of the specimen geometry and boundary conditions on the response, which could be interesting in case one seeks to calibrate material models based on engineering stresses and strains coming directly from test frame readings. First, we analyze an alternative geometry with oblique end-tabs. The tab design proposed by Sun and Ilsup [33] is illustrated in Fig. 6.13, with β given by

$$\beta = \operatorname{acot}(-S_{16}/S_{11}) \tag{6.13}$$

where S_{16} and S_{11} are the compliance coefficients with respect to the global coordinate system. For linear elasticity, this design ensures a macroscopically homogeneous stress states under uniaxial loading. In the nonlinear case, the compliance coefficients change from one time step to another, but we adopt the values obtained at the undeformed configuration with 5 modes, resulting in $\beta = 32°$.

To illustrate the effectiveness of the oblique end-tabs in creating a nearly uniform stress-state, we proceed with the most challenging case explored so far: off-axis angle of 15° and strain-rate of $10^{-4}\,s^{-1}$. For comparison, we plot the stress-strain curves for σ_{yy}^{eng} and σ_{xy}^{eng} for the two types of end-tabs in Fig. 6.14. Note that the oblique design effectively reduces the shear to almost zero, which also results in a lower engineering stress level in the loading direction (axial).

Another way to minimize the rise of shear stresses is to allow the specimen to move laterally. Fig. 6.15 shows the comparison between the different types of end-tabs and boundary conditions with respect to the experiments. For the oblique design, allowing for lateral movement has little to no impact on the outcome since the shear stress measured at the top surface and across the entire coupon was already limited (see Fig. 6.14). In contrast, with the straight end-tabs, allowing lateral movement reduces the engineering stress by about 10 %. These relative differences can be further explained by analysing the strain/stress distributions of each setup.

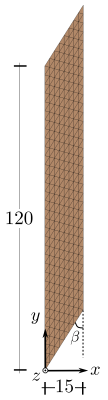
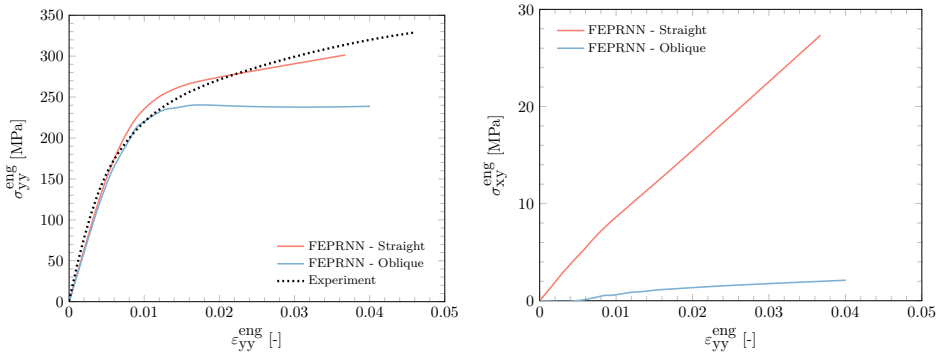


Figure 6.13: Alternative coupon geometry with oblique-end tabs.



(a) Axial stress

(b) Shear stress

Figure 6.14: Stress-strain curves for $\theta_0 = 15°$ and $\dot{\varepsilon}^{eng}_{yy} = 10^{-4}\,s^{-1}$ using straight vs oblique end-tabs.

We plot in Fig. 6.16 the macroscopic stress distributions from least to most uniform at $\varepsilon^{eng}_{yy} = 0.03$. Note the stress concentration near the corners of the specimen, the higher stress band formed in the diagonal of the coupon, which is aligned with the fiber direction, and the slight in-plane bending in the simulations with straight end-tabs. Although stress concentration is reduced when allowing lateral movement, the material near the edge of the straight end-tabs still experiences geometric discontinuity, and the fully constrained lower surface continues to limit Poisson's contraction. As a result, the Straight + Lateral alternative behaves as an intermediate case between the Straight and the oblique end-tabs configurations in terms of stress uniformity. The use of oblique end-tabs alone seems to almost eliminate any stress concentration, proving an efficient method to achieve uniform distributions even in the presence of material non-linearity. The slight bend at the top right corner of the Oblique + Lateral coupon stems from al-

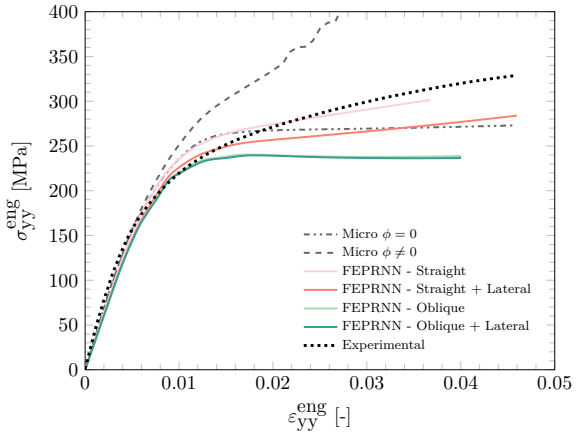
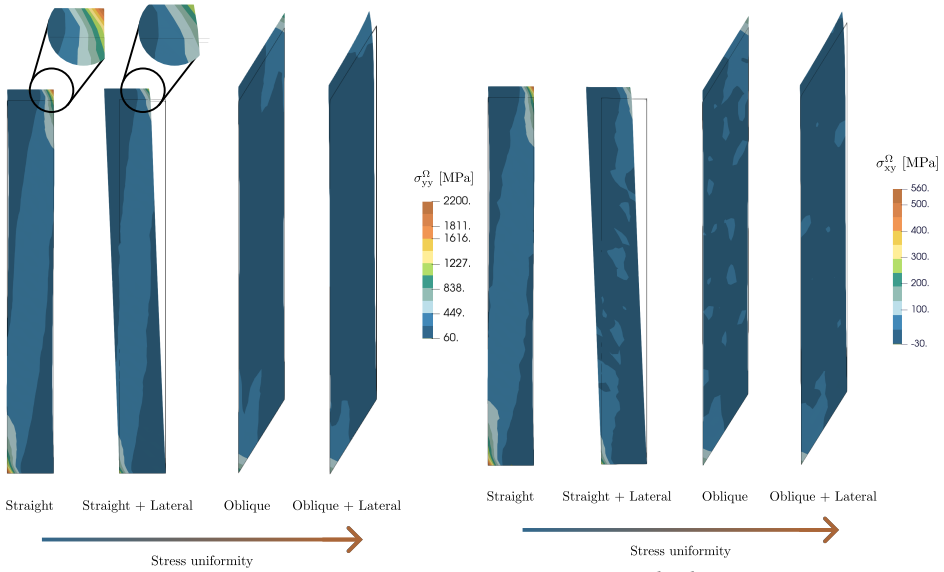


Figure 6.15: Stress-strain curves for different end-tab designs and boundary conditions for $\theta_0 = 15°$ and $\dot{\varepsilon}_{yy}^{eng} = 10^{-4}\,s^{-1}$.

lowing lateral movement at the top surface, while displacements at the bottom surface are fully constrained.

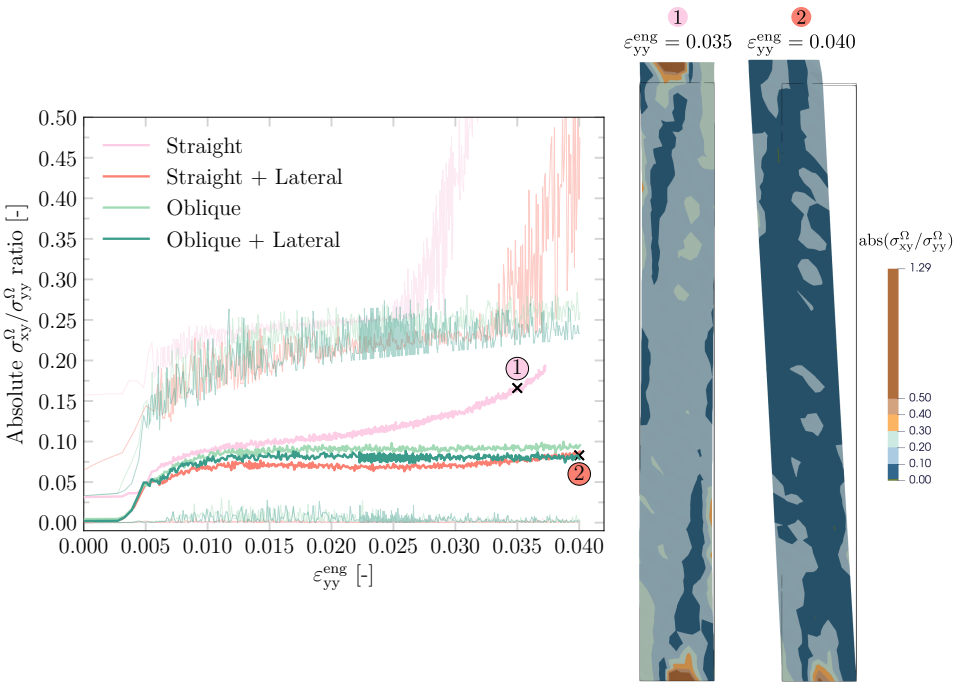


(a) Axial stress

(b) Shear stress

Figure 6.16: Full field of stresses predicted by FEPRNN at the same strain level ($\varepsilon_{yy}^{eng} = 0.03$) on specimen with straight and oblique end-tabs with and without lateral movement allowed for $\theta_0 = 15°$ and $\dot{\varepsilon}_{yy}^{eng} = 10^{-4}\,s^{-1}$.

Next, similar to the reorientation angle analysis in Fig. 6.11, we compute the ratio between shear and axial stress at each integration point of the macroscopic domain for all setups. Fig. 6.17 shows the average, maximum, and minimum values obtained. Lower average values with narrow variance bounds are desirable to approach a uniform uniaxial stress state. In this study, all three alternatives to the Straight setup lead to lower average ratios, but only those with oblique end-tabs do not exhibit a surge in maximum ratios as loading progresses. We illustrate in Fig. 6.17b the regions causing the spikes at the strain levels marked in Fig. 6.17a. These consist of the areas near the fully constrained grips, close to the corners that undergo strain/stress localization.



(a) Shear and axial stress ratio for different setups. Lines with reduced opacity correspond to the highest and lowest values over the entire macroscopic domain, while solid lines correspond to the average

(b) Stress ratio over coupons with different boundary conditions

Figure 6.17: Absolute ratio between shear and axial stresses predicted by FEPRNN for simulations with different end-tabs geometry and boundary conditions for $\theta_0 = 15°$ and $\dot{\varepsilon}^{eng}_{yy} = 10^{-4}\,s^{-1}$.

The trends discussed in this section are in line with other studies [33–37] based on ply-level properties and/or based on DIC with similar end-tab geometries and loading fixtures. These works discuss the over- or underestimation of elastic properties in terms of the error between the apparent and actual axial modulus. The actual modulus is computed from known unidirectional properties, while the apparent modulus is based on the average stress over the central area of the specimen. In general, setups with straight-

end tabs, smaller off-axis angles, and lower aspect ratios lead to the highest errors due to the extensional-shear coupling effects that result in highly inhomogeneous strain/stress fields, here illustrated by the left-most coupon in Fig. 6.16.

In [36], the authors proposed a new test framework with rotating grips in combination with straight end-tabs that produces virtually the same stress distributions and apparent modulus compared to those measured with oblique end-tabs and fixed grips. The lateral and rotational capabilities of the fixture allowed the specimen to realign freely during axial loading, reducing the stress concentrations otherwise induced by the straight end-tabs. In our simulations, the Straight + Lateral configuration similarly reduces inhomogeneity, although not as effectively. The rotating devices, however, require careful setup to allow free rotation without introducing additional constraints or measurement errors.

6.4. CREEP EXPERIMENTS

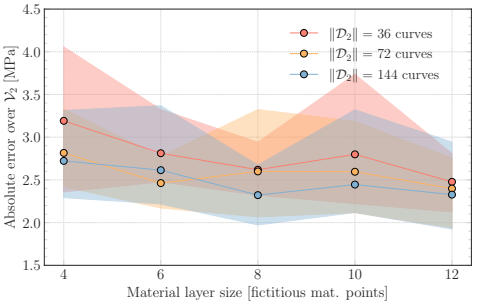
In this section, we follow the same structure as in the previous study, but adapt it to the creep experiments. We employ the adaptive stepping scheme [32] with larger bounds for the time increments ($\Delta t = [0.001\,\mathrm{s}, 300\,\mathrm{s}]$) as the creep simulations are significantly longer and more complex. They include sharp changes in strain-rate between loading and constant stress phases and wide variation within the latter. The boundary conditions in Section 6.3 are kept the same, except that instead of imposing a constant strain-rate on the top surface of the specimen, we use a force-controlled algorithm to load it according to Eq. (6.10).

This represents a harder test of the PRNN's capabilities, since the network was trained using a single randomly sampled time increment per loading path, while creep simulations span several orders of magnitude both in terms of total and time increments, and include distinct loading phases never seen during training. Using smooth GP-based paths is a Design of Experiments strategy we have explored in Chapter 5 to avoid biasing the network toward any particular loading scenario, while keeping sufficient variability for generalization to unseen loading behavior.

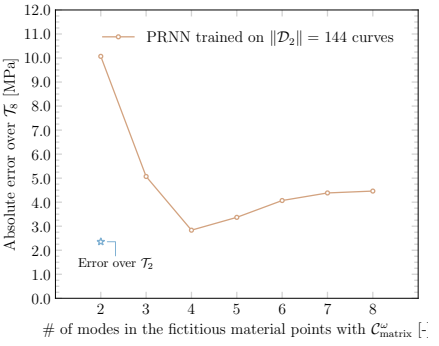
6.4.1. FROM ONE RELAXATION SPECTRUM TO ANOTHER

In [3], a new relaxation spectrum for $\mathcal{C}^{\omega}_{\text{matrix}}$ with a total of 2 processes and 8 modes was calibrated directly on the creep experiments. In these experiments, no extensometer was used, therefore measurements implicitly take into account the compliance effect of the machine grips. This effect can be considerable depending on the off-axis angle. In addition to that, the shear modulus G_{12} was changed from 45 MPa to 5 MPa for $\mathcal{C}^{\omega}_{\text{fiber}}$ [3]. Apart form these properties, since the same micromodel was used, the straightforward option here would be to update the material properties in both $\mathcal{C}^{\omega}_{\text{matrix}}$ and $\mathcal{C}^{\omega}_{\text{fiber}}$ and test the PRNN performance with gradually increasing modes until convergence, in the same fashion as in Section 6.3.1. This time, however, direct transfer to the new set of properties is not possible. To better understand why, we generate several test sets considering different combinations of matrix and fibers properties and found that the direct update of the material properties works, but only on a certain range around the original properties used for training. This study can be found in the Appendix A.

Hence, for modeling the creep experiments, we take a step back and train the PRNNs from scratch with the new set of properties $\boldsymbol{\mu}^{\text{creep}}_{\text{fiber}}$ and the first 2 modes (the first from each process) from $\boldsymbol{\mu}^{\text{creep}}_{\text{matrix}}$. For the training, the same approach and optimization parameters and hyper-parameters as in Chapter 5 are used with two modifications: 1) when creating the proportional GP-based curves, the time increment in each curve is sampled from a log-uniform distribution $\Delta t \sim U\,[10^{-3}\,\text{s}, 10^{3}\,\text{s}]$, and 2) the maximum number of epochs was increased to 2000. Fig. 6.18a shows the envelope of highest and lowest errors over a validation set $\mathcal{V}_2$ with 150 proportional GP-based curves for the different initializations, and material layer and training set sizes. From that, we select the networks with 12 fictitious material points (9 evaluated by $\mathcal{C}^{\omega}_{\text{matrix}}$ and 3 by $\mathcal{C}^{\omega}_{\text{fiber}}$) trained on 144 proportional GP-based curves.



(a) Envelope of highest and lowest errors from 10 initialization of PRNNs trained on different number of proportional GP-based curves and material layer size over validation set $\mathcal{V}_2$



(b) Accuracy with mode addition in PRNN with 6 material points trained on the first two modes over test set with the full relaxation spectrum $\mathcal{T}_8$

Figure 6.18: Model selection procedure for PRNNs trained on the micromodel with material properties calibrated on creep experiments.

Then, we apply the same reasoning as discussed in Section 6.2.6 and Section 6.3.1 to go from a network trained on 2 modes from $\boldsymbol{\mu}^{\text{creep}}_{\text{matrix}}$ to one that predicts the response of a micromodel with the full relaxation spectrum. For this purpose, first we select the network with the lowest error over a test set $\mathcal{T}_2$ with 150 proportional GP-based curves and same material properties as in the training. Then, on a new test set $\mathcal{T}_8$ that considers an RVE with the new full relaxation spectrum, we test the accuracy of that network with a varying number of modes (see Fig. 6.18b). In this case, the errors reach a minimum around 3 MPa ($\approx 10\,\%$) with 4 modes. Similar to the methodoloy adopted in Section 6.3.1, we repeat this study for the surrogate-based multiscale simulations to make a final choice on the number of modes we consider in the PRNN, having the fitness with the experiments as our guide.

6.4.2. COMPARISON WITH EXPERIMENTAL RESULTS

We start the discussion again with the scenario for which fiber reorientation is not relevant and where the impact of the extrapolation in terms of material properties is more evident, *i.e.* $\theta_0 = 90°$. Fig. 6.19a shows the strain-time FEPRNN responses for the maximum engineering stress of $\sigma_{yy}^{max} = 97$ MPa, as well as the micromodel results from [3], without the consideration of micro-cracking. Note how the strain responses converge towards the one obtained with the full relaxation spectrum (8 modes) with only a few modes and close to the experimental curve. We quantify the differences, similarly to Eq. (6.11), between the strains predicted by our surrogate-based approach and those obtained in the experiment during the constant stress phase in Fig. 6.19b to define the smallest number of modes we need to consider in the PRNN using the following metric:

$$\text{Relative mean error:} \frac{1}{n_{exp}^{creep}} \sum_{i=1}^{n_{exp}^{creep}} \frac{|\varepsilon_{yy,FEPRNN}^{eng}(t_{exp,i}) - \varepsilon_{yy,exp}^{eng}(t_{exp,i})|}{|\varepsilon_{yy,exp}^{eng}(t_{exp,i})|} \tag{6.14}$$

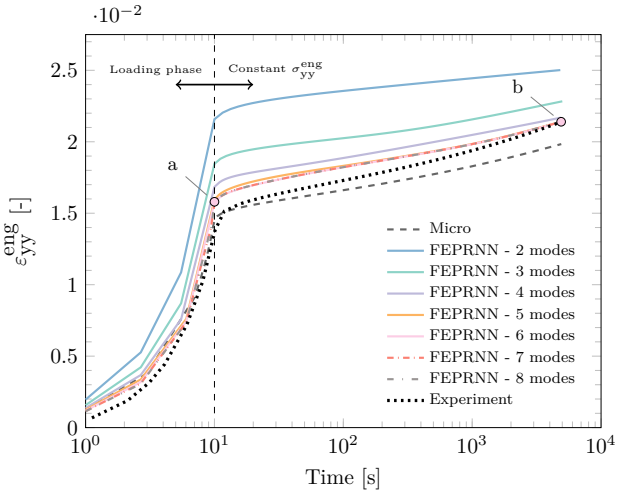
where n_{exp}^{creep} is the number of pairs of time-strain available in the experiment. Based on the plateau around the response with six modes ($\approx 4.1\,\%$) in Fig. 6.19b, we employ this number of modes in all FEPRNN creep simulations from here on.

To further demonstrate the generality of the approach, we consider two different σ_{yy}^{max}: 92 MPa and 101 MPa. The strain-time curves are shown in Fig. 6.20, while the relative errors calculated over the constant engineering stress phase are summarized in Table 6.1. In both cases, the lowest error occurs at the last time step, with average errors around 9.0 % and 5.6 %, respectively. The statistics for the previous case discussed are also included in Table 6.1 for reference.

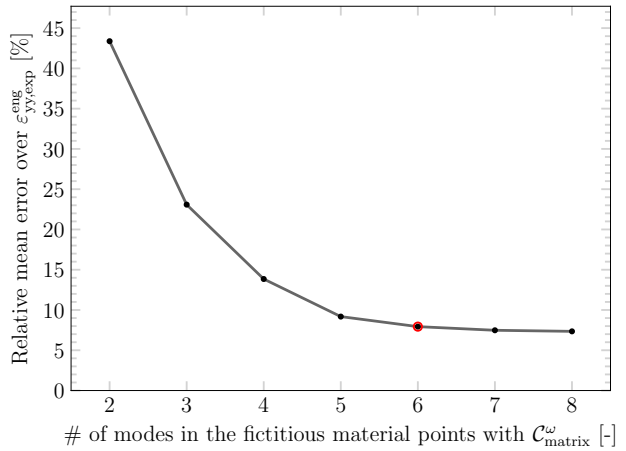
Table 6.1: Summary of relative errors between FEPRNN and creep experiments for $\theta_0 = 90°$ and different maximum stress levels.

	Relative error [%]			
σ_{yy}^{max} [MPa]	Lowest	Average	Highest	Last time step
92	7.1	9.0	10.9	7.1
97	0.1	4.2	14.5	0.1
101	2.0	5.6	14.5	2.0

Next, we investigate the $\theta_0 = 15°$ case, a scenario where the effect of the reorientation angle update ϕ on the micromodel results is significant. Therefore, we plot in Fig. 6.21 the two alternatives associated with this approach — with ($\phi \neq 0$) and without ($\phi = 0$) the reorientation — for comparison with the FEPRNN results, where reorientation of the material follows naturally and varies across the coupon. Despite the offset with respect to the experiment and the convergence issue at the later stages of the numerical simulation, the FEPRNN response lies between the two bounds of the micromodel solution, as expected. The lowest, average and maximum relative errors over the constant engineering stress phase are 6.0 %, 13.1 % and 14.9 %, respectively. In this case, the surrogate-based multiscale simulation is vital to account for the macroscopic variations that the micromodel-based approach cannot. These variations are illustrated in



(a) FEPRNN with different number of modes extracted from $\boldsymbol{\mu}_{\text{matrix}}^{\text{creep}}$



(b) Relative errors of FEPRNN compared to constant strain-rate experiment

Figure 6.19: Strain-time curves for $\theta_0 = 90°$ and $\sigma_{yy}^{max} = 97$ MPa using the micromodel-based solution (8 modes) and FEPRNN with different number of modes in the fictitious material points evaluated by $\mathcal{C}_{\text{matrix}}^{\omega}$ compared to experiment.

Fig. 6.22 by the macroscopic strain fields snapshots at times "c" and "d" marked in the simulations shown in Fig. 6.21, along with the snapshots at times "a" and "b" marked in Fig. 6.19a with $\theta_0 = 90°$, which, in contrast, show a nearly uniform distribution across the entire domain, which complies with the assumption adopted for the micromodel solution.

For $\theta_0 = 45°$ and $\theta_0 = 30°$, the FEPRNN significantly undershoots the engineering

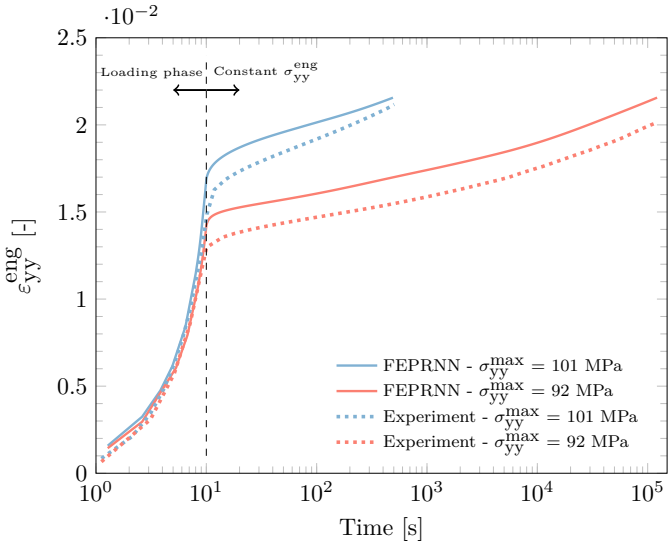


Figure 6.20: Strain-time curves for $\theta_0 = 90°$ and different stress levels σ_{yy}^{max}.

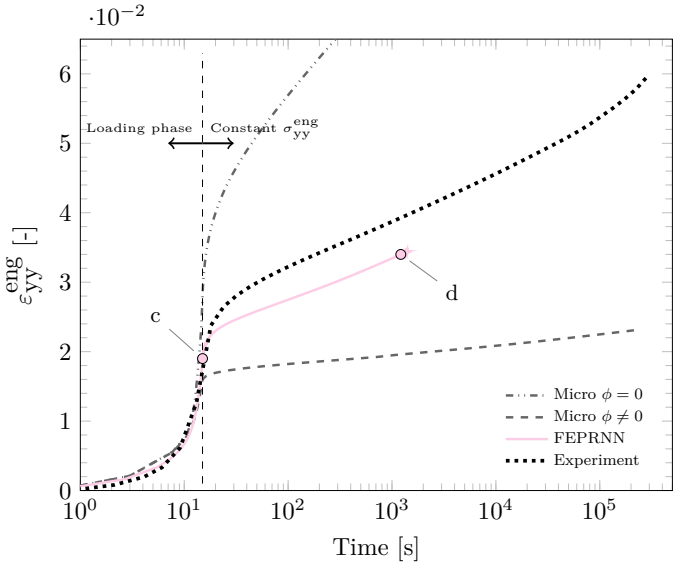


Figure 6.21: Strain-time curves for $\theta_0 = 15°$ and $\sigma_{yy}^{max} = 290$ MPa.

strain, following at best the micromodel response, as shown in Fig. 6.23. The general trend across angles reflects the fact that the material properties were calibrated from creep tests performed without an extensometer, which implicitly include the machine-grip compliance to different extents depending on the off-axis angle. As a result, when

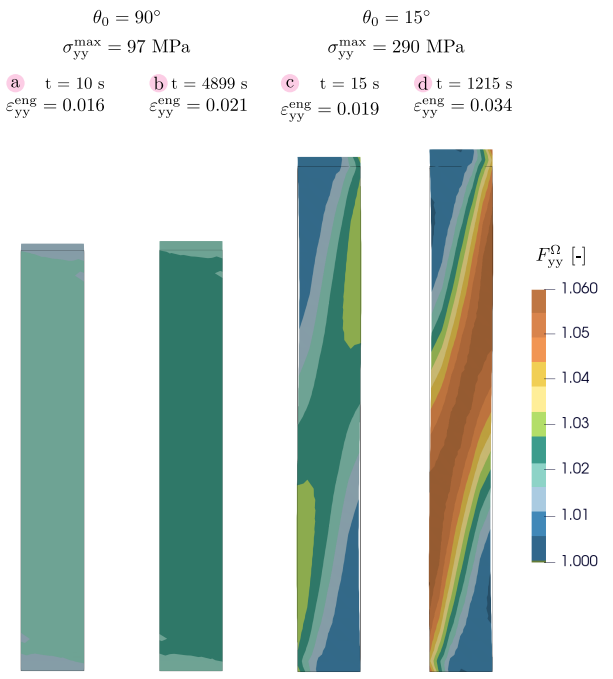
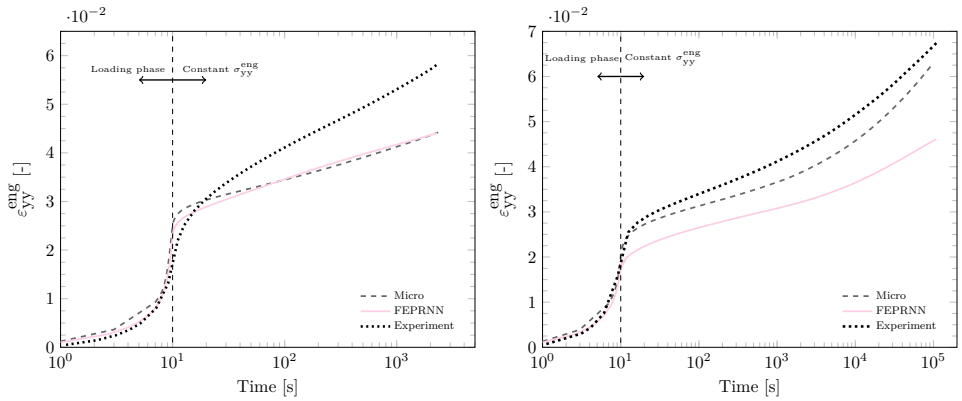


Figure 6.22: Macroscopic strain field predicted by FEPRNN at different times of simulations with different initial off-axis angles.

these properties are used in a multiscale simulation, we end up correcting the same machine-grip effect twice. As noted in [3], improving the underlying material calibration through more accurate measurements would probably improve these predictions. Nevertheless, the present results indicate that the surrogate framework already captures the main features of the creep response and, more importantly, it establishes a robust framework for future developments. Future research directions include extentsometer-based re-calibration of the relaxation spectrum, surrogate-driven parameter (re-)identification, and targeted sensitivity analyses of the material properties with respect to the material response.

6.5. CONCLUSION

In this chapter, we address the challenges observed in [2, 3] to model constant strain-rate and creep experiments on unidirectional composites under off-axis loading. In those works, a simplified and efficient approach was proposed based on a single macroscopic point with special boundary conditions to reproduce an (assumed) uniaxial stress state. While successful in many cases, the framework showed limitations with lower off-axis angles. The main hypothesis is that the assumption of macroscopic uniform deformation did not hold in the experiments. To test this hypothesis with the same high-fidelity micromodel, a full multiscale approach would be necessary, which remains computa-



(a) $\theta_0 = 30°$, $\sigma_{yy}^{max} = 154$ MPa

(b) $\theta_0 = 45°$, $\sigma_{yy}^{max} = 110$ MPa

Figure 6.23: Strain-time curves for $\theta_0 = 30°$ and $\theta_0 = 45°$ and loading time $t = 10$ s.

tionally prohibitive. However, with the development of the PRNN, a surrogate model recently tested on the same micromodel and constitutive models as in this chapter (see Chapter 5), a surrogate-based multiscale analysis becomes within reach. Building on these contributions, it is now possible to verify the hypotheses raised in [2] and to gain insights into the potential macroscopic strain/stress distribution in the experiments.

The PRNNs have a hybrid architecture that combines data-driven components and physics-based constitutive models to leverage better generalization properties with limited datasets. They have been used to model the homogenized response of micromodels with a range of constitutive laws, from linear elastic to elasto-viscoplastic models, including the ones considered in this chapter. With embedded models in the latent space, extrapolation is possible not only in terms of loading types but also in terms of material properties. Here, we harness this feature to simplify and reduce the computational cost of the offline phase by training with a fraction of the set of properties used in the references [2, 3]. Specifically, we generate and train the network with the first mode of each process in the elasto-viscoplastic model. Because each mode is associated with many internal variables, and these are at the core of the backpropagation in time, training with fewer modes translates into a more efficient process. We then show that these networks can be successfully transferred to our main task without any retraining effort by directly updating the material properties of the constitutive model. This feature is illustrated in the multiscale applications with the gradual mode addition.

From the constant strain-rate experiments, and assuming that the good match between them and the multiscale simulations implies a faithful characterization of the former, we summarize the following findings:

- Due to the tension/shear coupling, the constrained movement of the specimen by the machine grips, and the straight end-tabs, significant shear is present when the coupon is loaded in tension, except for the extreme cases $\theta_0 = 0°$ and $\theta_0 = 90°$. Allowing free shear deformation of the coupon eliminates the shear stress measured

at the top surface, and significantly reduces the stress concentration problem near the grips, but it is insufficient to achieve uniform strain and stress distributions;

- While it is generally known from experiments monitored by Digital Image Correlation that off-axis loading can lead to inhomogeneous strain fields, the surrogate-based simulations in this chapter validate and illustrate in detail how severe these inhomogeneities can be in lower off-axis angles. In our case, for $\theta_0 = 15°$, strain and stress concentrations manifest near the grips of the specimen, making the response deviate significantly from a representative uniaxial stress-strain state;
- We verify the hypotheses raised in the reference work of the micromechanical model [2] to explain the lack of fitting for $\theta_0 = 15°$. The experimental response is indeed a combination of material points that rotate to align with the loading direction (at the center of the specimen) and material points that have little to no rotation (near the surface of the end-tabs). For the remaining studied off-axis loading cases, there is moderate ($\theta_0 = 30°$ and $\theta_0 = 45°$) to negligible ($\theta_0 = 90°$) strain variation;
- To achieve a more uniform stress-strain uniaxial state in these experiments, we investigate the use of oblique end-tabs. It is confirmed that these also help reduce the stress concentration near the grips and greatly reduce the manifestation of shear stress across the specimen even with material nonlinearity.

For the creep experiments, we observe no accuracy gain in the multiscale simulations for the intermediary off-axis angles ($\theta_0 = 30°$ and $\theta_0 = 45°$) compared to the micromodel solution. Our results highlight the pitfalls of calibrating the material parameters from experiments without reliable strain measurements and under the assumption of homogeneous stress fields. For the two remaining cases, opposite scenarios are observed: one where the strain distribution is quite uniform with little machine grip effect ($\theta_0 = 90°$) and the other with large macroscopic variability ($\theta_0 = 15°$). For the former, we obtain a relatively good fit at different stress levels with average relative errors ranging from 4.2 % to 9.0 %. Whereas for $\theta_0 = 15°$, despite the convergence issues, the approximate response follows the same trend as the experiment, with an average error of 13.1 %. These findings, along with the insights from the constant strain-rate experiments, showcase the potential of PRNNs to enable efficient and robust multiscale analysis under variable loading conditions.

APPENDIX A. LIMITATIONS ON THE TRANSFER LEARNING FOR CREEP EXPERIMENTS

The following combination of material properties was considered to generate test sets of 150 proportional GP-based curves each: full relaxation spectrum calibrated for the creep experiments, referred as $\boldsymbol{\mu}^{\text{creep}}_{\text{matrix}}$, and varying shear modulus G_{12} in the material properties of the fibers $\boldsymbol{\mu}^{\text{creep}}_{\text{fiber}}$. The only difference between $\boldsymbol{\mu}^{\dot{\varepsilon}}_{\text{fiber}}$ and $\boldsymbol{\mu}^{\text{creep}}_{\text{fiber}}$ is the G_{12}, which changed from 45 MPa to 5 MPa.

With the network parameters obtained training on the first mode of $\boldsymbol{\mu}^{\dot{\varepsilon}}_{\text{matrix}}$ and $\boldsymbol{\mu}^{\dot{\varepsilon}}_{\text{fiber}}$, we update the material properties in the fictitious material points with $\mathcal{C}^{\omega}_{\text{fiber}}$ to match the

corresponding shear modulus being tested, while the modes used in $\mathcal{C}^{\omega}_{\text{matrix}}$ were gradually updated with increasing number of modes from the new spectrum. Although the test sets have slightly different stress-strain ranges, the comparison in terms of absolute errors in Fig. 6.24 is meaningful and reveals a few insights.

Firstly, going from the 1st mode of one relaxation spectrum ($\boldsymbol{\mu}^{\dot{\varepsilon}}_{\text{matrix}}$) to a multi-mode prediction of another ($\boldsymbol{\mu}^{\text{creep}}_{\text{matrix}}$) is possible. Second, the best performance is achieved around 5 modes. And lastly, in this case, the change in the shear modulus is the most important parameter in determining how well we can extrapolate. For the value considered in the creep simulations ($G_{12} = 5$ MPa), the lowest error observed is, unfortunately, still quite high, around 9 MPa.

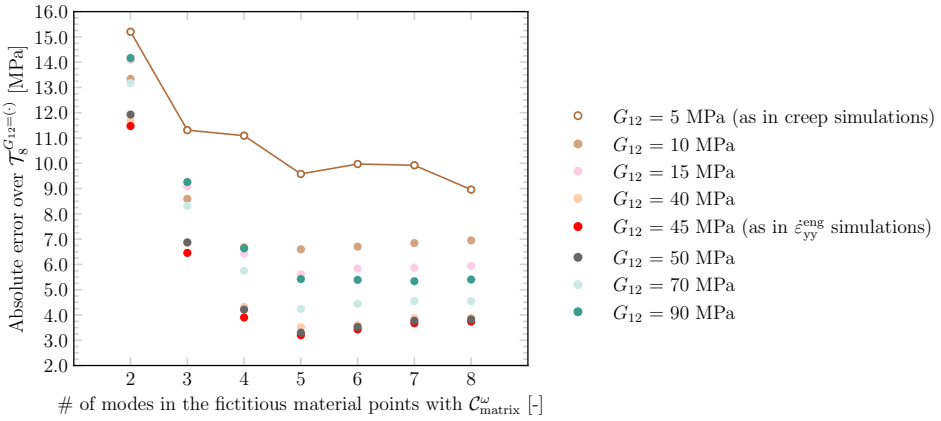


Figure 6.24: Accuracy with gradual mode addition in the fictitious material points evaluated by $\mathcal{C}^{\omega}_{\text{matrix}}$ of PRNN trained on micromodel simulations with a single mode from $\boldsymbol{\mu}^{\dot{\varepsilon}}_{\text{matrix}}$ over test sets with micromodel simulations with 8 modes from $\boldsymbol{\mu}^{\text{creep}}_{\text{matrix}}$ and different fiber shear moduli.

REFERENCES

[1] D. Kovačević and F. P. van der Meer. "Strain-rate based arclength model for nonlinear microscale analysis of unidirectional composites under off-axis loading". *International Journal of Solids and Structures* 250 (2022), 111697. ISSN: 0020-7683. DOI: https://doi.org/10.1016/j.ijsolstr.2022.111697.

[2] D. Kovačević, B. K. Sundararajan, and F. P. van der Meer. "Microscale modeling of rate-dependent failure in thermoplastic composites under off-axis loading". *Engineering Fracture Mechanics* 276 (2022), 108884. ISSN: 0013-7944. DOI: https://doi.org/10.1016/j.engfracmech.2022.108884.

[3] D. Kovačević, B. K. Sundararajan, and F. P. van der Meer. "Micromechanical model for off-axis creep rupture in unidirectional composites undergoing finite strains". *Composites Part A: Applied Science and Manufacturing* 176 (2024), 107860. ISSN: 1359-835X. DOI: https://doi.org/10.1016/j.compositesa.2023.107860.

[4] D. Kovačević, P. Hofman, I. B. C. M. Rocha, and F. P. van der Meer. "Unifying creep and fatigue modeling of composites: A time-homogenized micromechanical framework with viscoplasticity and cohesive damage". *Journal of the Mechanics and Physics of Solids* 193 (2024), 105904. ISSN: 0022-5096. DOI: https://doi.org/10.1016/j.jmps.2024.105904.

[5] L. Wan, K. Zhang, J. Chen, A. Li, J. Wu, and D. Yang. "Experimental testing and micromechanical modelling of unidirectional CFRP composite laminae under multiaxial loading conditions". *Composite Structures* 357 (2025), 118889. ISSN: 0263-8223. DOI: https://doi.org/10.1016/j.compstruct.2025.118889.

[6] Y. Ma, Y. Li, and L. Liu. "Off-Axis Tension Behaviour of Unidirectional PEEK/AS4 Thermoplastic Composites". *Applied Sciences* 13.6 (2023). ISSN: 2076-3417. DOI: 10.3390/app13063476.

[7] P. Hofman, D. Kovačević, F. van der Meer, and L. Sluys. "A viscoplasticity model with an invariant-based non-Newtonian flow rule for unidirectional thermoplastic composites". *Mechanics of Materials* 211 (2025), 105507. ISSN: 0167-6636. DOI: https://doi.org/10.1016/j.mechmat.2025.105507.

[8] X. Song, J. Zhou, K. M. Yeoh, D. Zhang, S. Zhang, K. Raju, X. Chen, Z. Guan, W. J. Cantwell, and V. B. C. Tan. "Multiscale modelling of residual thermal stresses and off-axis bending in 3D braided composites". *Composites Part B: Engineering* 308 (2026), 112972. ISSN: 1359-8368. DOI: https://doi.org/10.1016/j.compositesb.2025.112972.

[9] Z. Lu, Y. Zhou, Z. Yang, and Q. Liu. "Multi-scale finite element analysis of 2.5D woven fabric composites under on-axis and off-axis tension". *Computational Materials Science* 79 (2013), 485–494. ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2013.07.003.

[10] X. Li, Z. Guan, Z. Li, and L. Liu. "A new stress-based multi-scale failure criterion of composites and its validation in open hole tension tests". *Chinese Journal of Aeronautics* 27.6 (2014), 1430–1441. ISSN: 1000-9361. DOI: https://doi.org/10.1016/j.cja.2014.10.009.

[11] Z. Liu, Z. Guan, R. Tan, J. Xu, and X. Li. "Multiscale Analysis of CFRP Laminates with MMF3 Criterion under Different Off-Axis Loading Conditions". *Materials* 11.11 (2018). ISSN: 1996-1944. DOI: 10.3390/ma11112255.

[12] M. Katouzian and S. Vlase. "Creep Response of Carbon-Fiber-Reinforced Composite Using Homogenization Method". *Polymers* 13.6 (2021). ISSN: 2073-4360. DOI: 10.3390/polym13060867.

[13] X. Han, K. Huang, T. Zheng, J. Ding, J. Zhou, H. Liu, L. Zhang, and L. Guo. "A highly efficient ANN-based multiscale failure model for hierarchical fiber-reinforced composites". *Thin-Walled Structures* 217 (2025), 113850. ISSN: 0263-8231. DOI: https://doi.org/10.1016/j.tws.2025.113850.

[14] J. Lu, P. Zhu, Q. Ji, Q. Feng, and J. He. "Identification of the mechanical properties of the carbon fiber and the interphase region based on computational micromechanics and Kriging metamodel". *Computational Materials Science* 95 (2014), 172–180. ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2014.07.034.

[15] F. Ghavamian and A. Simone. "Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network". *Computer Methods in Applied Mechanics and Engineering* 357 (2019), 112594. ISSN: 00457825. DOI: 10.1016/j.cma.2019.112594.

[16] J. Gao, M. Shakoor, G. Domel, M. Merzkirch, G. Zhou, D. Zeng, X. Su, and W. K. Liu. "Predictive multiscale modeling for Unidirectional Carbon Fiber Reinforced Polymers". *Composites Science and Technology* 186 (2020), 107922. ISSN: 0266-3538. DOI: https://doi.org/10.1016/j.compscitech.2019.107922.

[17] Y. Ismail, L. Wan, J. Chen, J. Ye, and D. Yang. "An ABAQUS® plug-in for generating virtual data required for inverse analysis of unidirectional composites using artificial neural networks". *Engineering with Computers* 38.5 (Oct. 2021), 4323–4335. ISSN: 1435-5663. DOI: 10.1007/s00366-021-01525-1.

[18] K. A. Kalina, L. Linden, J. Brummund, and M. Kästner. "FE$^{\text{ANN}}$: an efficient data-driven multiscale approach based on physics-constrained neural networks and automated data mining". *Computational Mechanics* 71.5 (2023), 827–851. ISSN: 1432-0924. DOI: 10.1007/s00466-022-02260-0.

[19] Y. Zhou and S. J. Semnani. "A machine learning based multi-scale finite element framework for nonlinear composite materials". *Engineering with Computers* (Apr. 2025). ISSN: 1435-5663. DOI: 10.1007/s00366-025-02121-3.

[20] Y. Yamanaka, N. Hirayama, and K. Terada. "Surrogate Computational Homogenization of Viscoelastic Composites". *International Journal for Numerical Methods in Engineering* 126.6 (2025), e70008. DOI: https://doi.org/10.1002/nme.70008. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.70008.

[21] G. Kubo, T. Matsuda, H. Nagaoka, and Y. Sato. "Development and Validation of Multiscale Thermo-Elasto-Viscoplastic Analysis Method for Plain-Woven Composites". *Key Engineering Materials* 794 (Feb. 2019), 78–88. ISSN: 1662-9795. DOI: 10.4028/www.scientific.net/kem.794.78.

[22] C. Zhang, Z. Bian, T. Chen, T. Q. Bui, J. L. Curiel-Sosa, and C. Mao. "Data-driven deep learning models for predicting off-axis tensile damage of 2.5D woven composites at elevated temperatures". *Thin-Walled Structures* 209 (2025), 112944. ISSN: 0263-8231. DOI: https://doi.org/10.1016/j.tws.2025.112944.

[23] C. Zhang, Z. Bian, T. Q. Bui, and J. L. Curiel-Sosa. "A tree-based machine learning surrogate model for predicting off-axis tensile mechanical properties of 2.5D woven composites at high temperatures". *Composite Structures* 360 (2025), 119044. ISSN: 0263-8223. DOI: https://doi.org/10.1016/j.compstruct.2025.119044.

[24] X. Han, K. Huang, T. Zheng, J. Zhou, H. Liu, Z. Li, L. Zhang, and L. Guo. "An ANN-based concurrent multiscale damage evolution model for hierarchical fiber-reinforced composites". *Composites Science and Technology* 259 (2025), 110910. ISSN: 0266-3538.

[25] B. Sundararajan. "Matrix dominated failure in continuous carbon fibre reinforced Poly(ether ether ketone)". PhD thesis. Netherlands: University of Twente, Apr. 2024. ISBN: 978-90-365-6039-9. DOI: 10.3990/1.9789036560405.

[26] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary. *Nonlinear finite elements for continua and structures*. John wiley & sons, 2014.

[27] İ. Temizer and P. Wriggers. "On the computation of the macroscopic tangent for multiscale volumetric homogenization problems". *Computer Methods in Applied Mechanics and Engineering* 198.3 (2008), 495–510. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2008.08.018.

[28] J. C. Zhu, M. B. Bettaieb, and F. Abed-Meraim. "Comparative study of three techniques for the computation of the macroscopic tangent moduli by periodic homogenization scheme". *Engineering with Computers* 38.2 (2020), 1365–1394. ISSN: 1435-5663. DOI: 10.1007/s00366-020-01091-y.

[29] J. Bonet and A. J. Burton. "A simple orthotropic, transversely isotropic hyperelastic constitutive equation for large strain computations". *Computer Methods in Applied Mechanics and Engineering* 162.1 (1998), 151–164. ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(97)00339-3.

[30] L. van Breemen, E. Klompen, L. Govaert, and H. Meijer. "Extending the EGP constitutive model for polymer glasses to multiple relaxation times". *Journal of the Mechanics and Physics of Solids* 59.10 (2011), 2191–2207. ISSN: 0022-5096. DOI: https://doi.org/10.1016/j.jmps.2011.05.001.

[31] L. C. A. van Breemen, T. A. P. Engels, E. T. J. Klompen, D. J. A. Senden, and L. E. Govaert. "Rate- and temperature-dependent strain softening in solid polymers". *Journal of Polymer Science Part B: Polymer Physics* 50.24 (2012), 1757–1771. DOI: https://doi.org/10.1002/polb.23199.

[32] F. P. van der Meer. "Mesolevel Modeling of Failure in Composite Laminates: Constitutive, Kinematic and Algorithmic Aspects". *Archives of Computational Methods in Engineering* 19.3 (2012), 381–425. DOI: 10.1007/s11831-012-9076-y.

[33] C. Sun and C. Ilsup. "An oblique end-tab design for testing off-axis composite specimens". *Composites* 24.8 (1993), 619–623. ISSN: 0010-4361. DOI: https://doi.org/10.1016/0010-4361(93)90124-Q.

[34] M. Nemeth, C. Herakovich, and D. Post. "On the Off-Axis Tension Test for Unidirectional Composites". *Journal of Composites Technology and Research* 5 (May 1982). DOI: 10.1520/CTR10794J.

[35] M. Kawai, M. Morishita, H. Satoh, S. Tomura, and K. Kemmochi. “Effects of end-tab shape on strain field of unidirectional carbon/epoxy composite specimens subjected to off-axis tension”. *Composites Part A: Applied Science and Manufacturing* 28.3 (1997), 267–275. ISSN: 1359-835X. DOI: https://doi.org/10.1016/S1359-835X(96)00122-4.

[36] Y. Xiao, M. Kawai, and H. Hatta. “An integrated method for off-axis tension and compression testing of unidirectional composites”. *Journal of Composite Materials* 45.6 (2011), 657–669. DOI: 10.1177/0021998310377936.

[37] M. Merzkirch and T. Foecke. “10° off-axis testing of CFRP using DIC: A study on strength, strain and modulus”. *Composites Part B: Engineering* 196 (2020), 108062. ISSN: 1359-8368. DOI: https://doi.org/10.1016/j.compositesb.2020.108062.

7

Conclusion

Work it harder, make it better
Do it faster, makes us stronger
More than ever, hour after hour
Work is never over

Daft Punk

In this thesis, a novel class of neural networks, PRNNs, was introduced for accelerating multiscale simulations of complex materials. PRNNs achieve this by replacing the homogenized response of the Representative Volume Element (RVE) in FE^2 and micromechanical simulations. Their design directly addresses the main challenges in surrogate material modelling: the black-box nature, the poor extrapolation properties, and the data-hungriness of purely data-driven models, as well as the limited generalization of highly specialized architectures to a specific class of materials.

To tackle these issues, we devised a hybrid approach that builds on the high flexibility of data-driven models while preserving key aspects of the computational homogenization procedure by embedding classical constitutive models in an encoder-decoder network architecture. The network was tested in a variety of scenarios that cover different levels of complexity in terms of 1) constitutive models (from linear elasticity to complex elasto-viscoplasticity), 2) loading cases (from monotonic to cyclic to creep and relaxation), and 3) analysis frameworks (from 2D small strains to 3D finite strains). The consistently good performance of PRNNs in these studies ratifies the method as a reliable tool to accelerate multiscale simulations and to enable its adoption to solve real-world problems. The main contribution of this thesis is the development of a non-intrusive and robust surrogate model that is general in the sense that the framework is unbound to a specific class of material model.

In the following, we summarize the five main advantages of the proposed approach:

- **Interpretability**: By incorporating well-established constitutive models in NNs, PRNNs ensure that predictions remain interpretable and grounded in physics rather than relying on intricate and opaque approximations. This transparency is explored in many ways throughout the chapters, from the identification of potential pitfalls when choosing the architecture (as in Chapter 2 and Chapter 4) to the direct comparison of the latent space with microscopic quantities, as in Chapter 3.

- **Extrapolation capabilities**: The proposed approach inherits from FE^2 the idea that the complex behavior of heterogeneous materials can be accurately described by letting simpler constitutive models that represent the microscopic constituents interact. In the PRNN, this interaction is described by the data-driven components, which are learned based on snapshots of the homogenized responses of the micromodel, but the embedded models are kept untouched. This means the network can reproduce the complex behavior encapsulated in the physics-based models without the need to (re)learn the time-dependencies from the data.

- **Data efficiency**: By offloading part of the learning process to known physical laws, PRNNs require significantly less training data while still achieving high accuracy. This makes them practical for real-world applications where extensive datasets can be difficult to obtain.

- **Robustness**: Throughout the chapters, we applied the network in a series of equilibrium problems to showcase its robustness to obtain macroscopic convergence with an implicit solver. From simple monotonic and cyclic loading to creep and relaxation, different applications demonstrated the consistency of this feature under general loading conditions in different analysis frameworks.

- **Generality**: Last, but not least in importance, is generality. While particularised in this thesis for NNs, the hybrid nature of this approach allows modularity, meaning an arbitrary data-driven model could be used as the first of the two main components, provided the second and most important part is preserved: the physics-based constitutive models. In this regard, the approach is general and can be employed to account for any effects by adapting the constitutive model embedded in the material layer. This thesis explored elastoplasticity, linear and nonlinear elasticity, stiffness degradation, hyperelasticity, orthotropy, and elasto-viscoplasticy.

Based on these features, we now highlight the main achievements and conclusions from each chapter:

PRNNs generalize to unseen arbitrary loading behavior based on relatively small datasets with monotonic loading paths. Compared to a Bayesian RNN, the proposed approach outperformed it using 64 times less data. A brief study on a micromodel with different combinations of constitutive models and material properties demonstrated the generality of the method, although limited to linear and nonlinear elasticity and elastoplasticity. As a constitutive model, the network showed robustness in FE^2 problems, with speed-ups of up to four orders of magnitude.

For a linear encoder and decoder, we illustrated the effect of a new weight normalization constraint on bringing the stress distribution closer to the true distribution from the micromodel. The constraint also acts as a regularization technique, preventing overfitting and enabling robust training with limited datasets. With just two curves, PRNNs can accurately predict and extrapolate to different loading conditions. We have also leveraged the latent space to retrieve information from the microscopic level and incorporated it into a multi-task approach.

The first key extension towards including damage models in PRNNs was presented. Starting from the original architecture that only considered bulk constitutive models — which failed to capture stiffness degradation resulting from microscale debonding, we proposed a series of modifications to better represent the increased complexity in the micromodel. The final PRNN architecture was successfully trained on non-proportional and non-monotonic loading and tested over different loading conditions.

We extended the PRNN to deal with path and rate-dependent heterogeneous materials in a 3D finite strain framework. To reduce the high dimensionality of the deformation gradient, we applied polar decomposition, tasking the network with the mapping between stretch and unrotated stress. The network was employed as the surrogate model for a unidirectional composite micromodel with rate-dependent plasticity in the matrix and hyperelasticity in the fibres. The network extrapolated well to various strain-rates and unloading/reloading scenarios, and was later applied to solve a micromodel-based simulation with off-axis loading.

With PRNNs as the homogenized constitutive model, we enabled a surrogate-based multiscale framework that addressed unanswered questions about off-axis experiments on unidirectional thermoplastic composites. It was previously demonstrated in [1, 2] that these experiments can be well captured by a single micromodel, except for small off-axis angles. The hypothesis was that the mismatch in these cases was due to the non-uniform stress/strain states at the macroscale. A multiscale simulation was necessary to test this hypothesis with the same high-fidelity micromodel, which is not (computationally) feasible in this case. With the surrogate-based multiscale simulations, we confirmed the significant macroscopic variations in the tests and showed that accounting for them improves model-experiment agreement for small angles without affecting large-angle accuracy. Moreover, the framework proved robust enough to achieve macroscopic convergence with an implicit solver across several constant strain-rate and creep simulations.

Although PRNNs have been applied to a comprehensive set of problems, their flexibility is far from exhausted. The developments discussed here pave the way for expanding their applicability to more complex material systems and loading conditions. In the following sections, we briefly discuss opportunities for further development, generalization, and new applications in promising areas, with some already underway.

7.1. COLLABORATIONS AND IMPACT

Parallel with the developments presented in this thesis, multiple collaborators within and outside TU Delft have pushed the boundaries of PRNN's versatility in new applications. In the following, we briefly discuss how these ideas advance the state of the art in their respective fields and how they build on the developments discussed in this work.

- **Multiscale analysis of woven composites using Hierarchical PRNNs (HPRNNs)** [3]: Journal paper (preprint). University of Gothenburg, SE.

  In the full-order computational homogenization scheme, woven materials are modelled in three scales: macro, meso and micro, as illustrated in Fig. 7.1a. The RVE at the mesoscale is composed of warp and weft yarns embedded in an elastoplastic matrix, while the RVE at the microscale is treated as a unidirectional composite. However, the interaction between the matrix and the yarns introduces substantial complexities across scales and results in extreme computational effort, rendering the full-order solution infeasible.

  To enable these simulations, we propose a bottom-up approach. Firstly, we use the micro-RVE to generate pairs of homogenized strains and stresses. These are used to train PRNNs in the same fashion discussed throughout this thesis. Then, the optimal parameters of this network, referred to as warp-PRNN in Fig. 7.1a, are copied, generating what we call weft-PRNN. The inputs of this network are obtained by rotating the inputs from the warp-PRNN and transforming the outputs accordingly. These two models are combined with the elastoplastic model that describes the matrix material in the meso-RVE and micro-RVE in a so-called *module*.

Finally, a second PRNN is devised to combine the response of several of these modules leveraging an encoder and decoder tasked with the de-homogenization of the macroscale strain into fictitious mesoscopic strains and the homogenization of the fictitious mesoscopic stresses into the macroscale stress, as depicted in Fig. 7.1a. Adopting HPRNNs for both scale transitions avoids nonphysical behavior often observed in predictions from pure data-driven RNNs and transformers. This results in better generalization under complex cyclic loading conditions.

- **Accelerating multiscale modelling of delamination with a suite of surrogate models** [4]: Journal paper (in preparation). TU Delft, NL.

  In this work, we are concerned with modelling crack propagation at the mesoscale while keeping track of the micromechanical evolution of plasticity and distributed cracking ahead of the crack tip in laminated composites. This requires a concurrent multiscale approach, which is computationally prohibitive. The solution explored here involves combining two surrogates in a single framework, as PRNNs are not suitable for capturing softening at the macroscale.

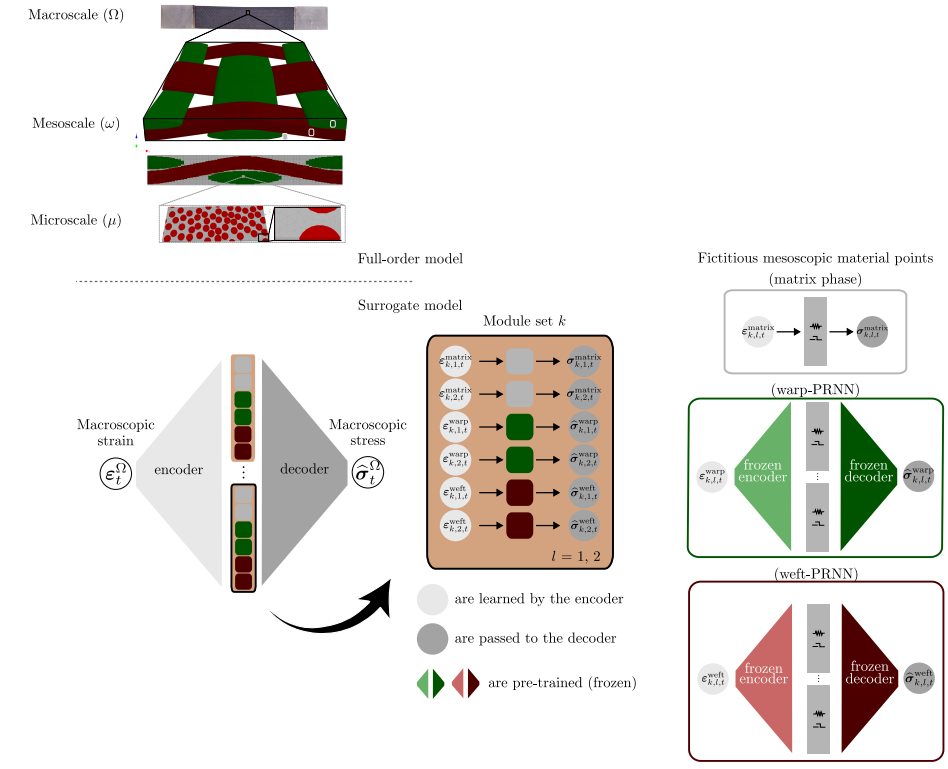
  For example, to simulate mixed-mode bending tests, we consider three domains, as illustrated in Fig. 7.1b. In the regions where softening can take place, we employ a GP-based active learning framework conditioned on a small number of micromodel computations observations of strains and stresses. The same framework is applied to model crack growth but conditioned on observations of cohesive tractions and displacement jumps. Finally, for the surrounding bulk domain, we employ PRNNs. This combination allows us to perform FE^2 simulations that would otherwise be out of reach and help elucidate energy dissipation mechanisms due to delamination.

- **Efficient optimization of composites using ML-based techniques**: Grant. Universidade Federal do Ceara (UFC), BR.

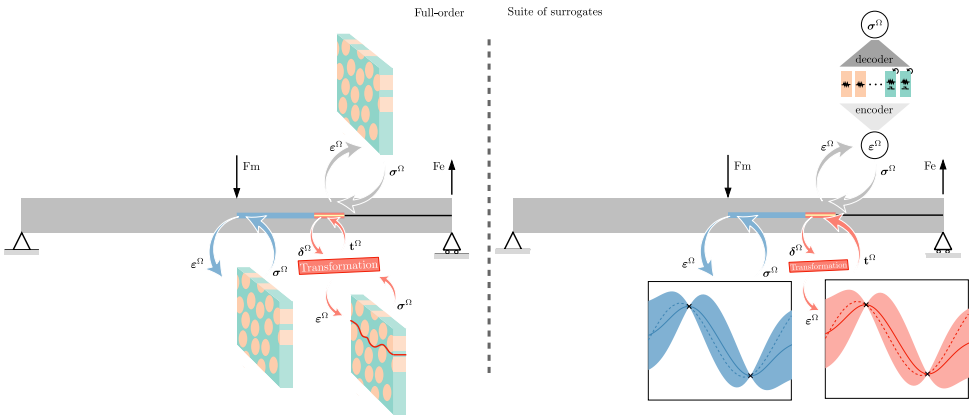
  The goal of this grant is to establish and develop a partnership between two laboratories: the Statistical Learning for Intelligent Material Modeling Laboratory (SLIMM Lab/TU Delft, NL) and the Laboratorio de Mecanica Computacional e Visualizacao (LMCV/UFC, BR). The idea is to build on the strengths of each research group to develop an efficient method for the simulation and optimization of composite materials.

  Optimization techniques are needed to fully exploit the potential of composites. However, due to the high computational cost of nonlinear analyses, optimization is often performed using simplified methods without considering phenomena happening at lower scales, which can lead to unsafe and inefficient solutions. A potential solution is to employ PRNNs to efficiently represent the nonlinear behavior of composites at the microscale.

  For this project, PRNNs also need to be extended to deal with other types of composite materials, namely functionally graded materials. The overarching goal is to integrate the network and other ML-based techniques into an efficient optimization framework and validate the optimal solutions through laboratory experiments.



(a) HPRNN for homogenization of woven composites



(b) Suite of surrogate models to capture delamination in mixed-mode bending test

Figure 7.1: Schematic representation of collaborative works that expand the applicability of PRNNs.

In addition to these ongoing collaborations, the following three MSc theses connect with the core idea of PRNNs:

- **PRNNs to predict microscale debonding in composite materials [5]**:

  The goal of this MSc thesis was to extend PRNNs to account for distributed damage in the form of microscale debonding. For this purpose, we employed cohesive zone models (CZMs) at the interface elements between fibre and matrix in a composite RVE. This arrangement allows stiffness degradation without softening at the macroscale. The main problem of the original architecture discussed in Chapter 2 was the inability to reproduce that behavior, even when CZMs were included in the material layer. Therefore, several architectures were proposed to address this limitation, leading to a modified layout that leveraged the damage variable to modify the fictitious strains fed to the bulk models. This study resulted in the publication discussed in Chapter 4.

  The researcher has continued advancing the applicability of PRNNs in a PhD at the University of Porto (Portugal) with collaboration from the former MSc supervisory team at TU Delft. In the most recent development, PRNNs trained on a micromodel with fixed material properties, as presented in Chapter 2, were demonstrated to maintain accuracy with varying properties. This opened the door for the model to be used to propagate uncertainty in multiscale frameworks, speeding up simulations on overly coarse macroscopic meshes by three orders of magnitude and enabling simulations with finer meshes that were previously inaccessible.

- **Multi-task approach to predict maximum hydrostatic stress using PRNNs [6]**:

  The goal of this MSc thesis was to explore the potential of PRNNs, trained initially to predict macroscopic stress, to extract information from the microscale. Specifically, we were interested in retrieving the maximum microscopic stress, which has been recently used as a failure indicator, from the pool of fictitious stress states in our latent space. Though the PRNN cannot predict failure yet, having a threshold maximum stress to compare with can help set up alternatives to deal with the problem. For example, once we reach this threshold, we could switch to another surrogate model or a full-order micromodel at the macroscale point where this happens or remove the element from the mesh. The main developments and key takeaways from this study were included in the publication discussed in Chapter 3.

- **FE model as building blocks of NNs to model lattice materials [7]**:

  The goal of this MSc thesis was to predict the homogenized constitutive response of 2D lattice materials. For that, we built on the idea of having smaller representations of the macroproblem embedded in the network to capture geometric nonlinearity. In the RVEs discussed throughout this thesis, the building blocks and primary (often only) source of nonlinearity were the constitutive models, with no specific concern for geometric nonlinearity. In contrast, for the lattice application, the constitutive models were quite simple (linear elasticity), while geometric nonlinearity was the central phenomenon to be captured.

The basic building block was changed to a Finite Element (FE) model, specifically a cantilever beam, as illustrated in Fig. 7.2. Each beam can be discretized in as many elements as needed and inherits all the same material properties and constitutive models as the beams that compose the high-fidelity unit cell. Although the findings in [7] do not satisfy the requirements for practical application yet, they indicate that introducing FE models improved the extrapolation properties and interpretability of the surrogate model compared to regular FNNs.

Given further development, these networks could be the key to efficiently handling more complex scenarios as the required changes are relatively straightforward to incorporate, e.g., dynamics and history-dependent behavior. This is possible because the state of the FE models is known at every time step, which includes potential internal variables of the constitutive models.

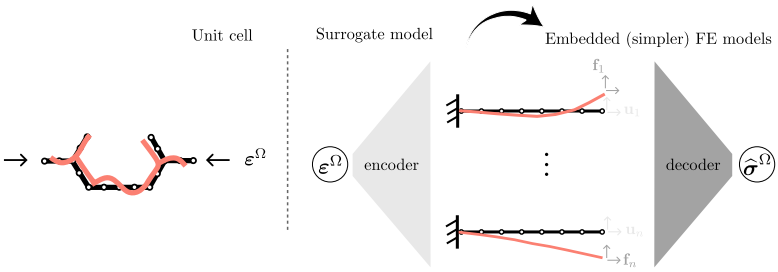


Figure 7.2: FE models as building blocks embedded in PRNN architecture, as in [7].

7.2. FUTURE RESEARCH DIRECTIONS

1. **Different RVE morphologies, micromodel geometry, and material properties**: While a wide range of constitutive model behaviors and loading conditions were considered in this thesis, the numerical examples only dealt with two-phase composites with fixed micromodel geometry. A straightforward application would be to test it with more than two phases and different RVE morphologies (e.g., amorphous, porous, and Voronoi-Tessellated). Another avenue unexplored is the extension of PRNNs to deal with geometrical features (e.g., volume fraction and radius and shape of inclusions) and variable material properties. Although the latter topic is briefly discussed in Chapter 6, we only touched the surface of the potential of having such a feature without explicitly training for it. Limitations and implications have not been thoroughly discussed yet.

2. **Robustness to noisy data and active learning**: PRNNs were not tested with noisy data. Therefore, strategies for improving their robustness to such conditions could be explored (e.g., dropout and bayesian approach). Another avenue for impact would be the integration of PRNNs in an active learning framework to guide the adaptive selection of loading paths, material properties, and/or constitutive models. The careful incorporation of new data has not been considered so far, but it has

the potential to achieve more efficient training schemes and improve the design of space exploration.

3. **Multi-physics**: PRNNs have been primarily applied to mechanical systems, but many real-world problems involve multiple physical phenomena (e.g., thermal, moisture diffusion, aging mechanisms, etc.), and PRNNs have not been investigated in these multi-physics contexts.

4. **Strain localization**: In this thesis, softening was only considered at the microscale for modelling debonding in a composite RVE (see Chapter 4). In practice, however, micro-cracks appear scattered and can coalesce to form arbitrary paths that must be propagated to the macroscale, challenging the fundamental assumptions of FE^2. In recent years, a framework with enhanced localization criterion for macroscopic cohesive failure with less intrusive changes to FE^2 compared to the literature was proposed [8]. However, the computational cost associated with these simulations is enormous, which is precisely where PRNNs could thrive. Extending these networks from bulk to cohesive homogenization can enable the analysis that would otherwise be out of reach.

5. **Material model calibration of heterogeneous materials**: A promising research direction is the integration of PRNNs for automated material model calibration. While automated frameworks, such as the Efficient Unsupervised Constitutive Law Identification and Discovery (EUCLID) [9] made important advances in the context of homogeneous materials, the generalization to heterogeneous materials remains an open challenge. The large microscopic variability, the interaction between phases, and the limited number of experiments are only some of the hurdles. PRNNs could help address these difficulties in different ways depending on how much knowledge of constitutive models, material properties, and micromodel geometry we have *a priori*.

   From an assumed set of constitutive models and a sufficiently large and fixed micromodel geometry, a PRNN that generalizes to different material properties could serve as the constitutive model in the surrogate-based multiscale simulation that reproduces the experimental conditions, similar to what we explored in Chapter 6. In this case, an outer optimization loop is required to identify the material properties.

   Moving towards a more general approach naturally introduces more complexity. For example, if no information on the material geometry can be assumed, the geometric features would have to be a part of the optimization loop, and a more general PRNN, one that can extrapolate both in terms of material properties and material geometry, would be required. In case the constitutive models are also uncertain, one option would be to work on a small pool of constitutive model candidates. Based on the findings in Chapter 3, where minimal training sets provided reasonable extrapolation, an ensemble of PRNNs trained on different constitutive model combinations could be considered.

   Alternatively, a fully automated framework could be developed where the three main ingredients are identified simultaneously. In this case, networks trained on-

the-fly with combinations obtained by an active learning strategy could be employed to handle the increased complexity of the optimization problem. In all cases, the robustness of PRNNs to noisy data is paramount and demands attention.

6. **Many-query applications**: The applications in FE^2 illustrated the potential of PRNNs in speeding up computationally intensive simulations, but other many-query applications could benefit immensely from it. Integrating the proposed model in other many-query frameworks, such as material design optimization, sensitivity analysis, topology optimization, uncertainty quantification, and inverse modelling, requires development. From the identification of optimal microscopic features (e.g., fibre orientation, volume fraction, etc.) to the quantification of material property variability, PRNNs can pottentially be extended beyond their current role as surrogate models to actively guide design, calibration, and decision-making processes in next-generation material design and optimization processes.

REFERENCES

[1] D. Kovačević, B. K. Sundararajan, and F. P. van der Meer. "Microscale modeling of rate-dependent failure in thermoplastic composites under off-axis loading". *Engineering Fracture Mechanics* 276 (2022), 108884. ISSN: 0013-7944. DOI: https://doi.org/10.1016/j.engfracmech.2022.108884.

[2] D. Kovačević, B. K. Sundararajan, and F. P. van der Meer. "Micromechanical model for off-axis creep rupture in unidirectional composites undergoing finite strains". *Composites Part A: Applied Science and Manufacturing* 176 (2024), 107860. ISSN: 1359-835X. DOI: https://doi.org/10.1016/j.compositesa.2023.107860.

[3] E. Ghane, M. A. Maia, I. B. C. M. Rocha, M. Fagerström, and M. Mirakhalaf. *Multiscale Analysis of Woven Composites Using Hierarchical Physically Recurrent Neural Networks*. 2025. arXiv: 2503.04901 [physics.comp-ph]. URL: https://arxiv.org/abs/2503.04901.

[4] L. Ke, I. B. C. M. Rocha, M. A. Maia, and F. P. van der Meer. *Accelerating Multiscale Modeling of Delamination With a Suite of Surrogate Models*. Talk at the 9th European Congress on Computational Methods in Applied Sciences and Engineering. 2024. URL: https://eccomas2024.org/event/contribution/b5301968-b3a6-11ee-ac5b-000c29ddfc0c.

[5] N. Kovács. "Physically Recurrent Neural Networks for Cohesive Homogenization of Composite Materials". MSc thesis. TU Delft, 2023.

[6] A. M. C. M. van Gils. "Predicting maximum stresses in composite microstructures using surrogate models". MSc thesis. TU Delft, 2025.

[7] P. J. van IJzendoorn. "Multiscale Modelling of Lattice Materials: a novel approach using Beam Neural Networks". MSc thesis. TU Delft, 2024.

[8] L. Ke and F. P. van der Meer. "A computational homogenization framework with enhanced localization criterion for macroscopic cohesive failure in heterogeneous materials". *Journal of Theoretical, Computational and Applied Mechanics* (2022). ISSN: 2726-6141. DOI: 10.46298/jtcam.7707.

[9] M. Flaschel, S. Kumar, and L. De Lorenzis. "Automated discovery of generalized standard material models with EUCLID". *Computer Methods in Applied Mechanics and Engineering* 405 (2023), 115867. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.115867.

RESEARCH OUTPUTS

JOURNAL ARTICLES

6. M. A. Maia, I. B. C. M. Rocha, and F. P. van der Meer. "Towards sparsity and interpretability in Physically Recurrent Neural Networks". In preparation.

5. E. Ghane, M. A. Maia, I. B. C. M. Rocha, M. Fagerström, and M. Mirakhalaf. *Multiscale Analysis of Woven Composites Using Hierarchical Physically Recurrent Neural Networks*. 2025. arXiv: 2503.04901 [physics.comp-ph]. URL: https://arxiv.org/abs/2503.04901[†]

4. M. A. Maia, I. B. C. M. Rocha, D. Kovačević, and F. P. van der Meer. *Surrogate-based multiscale analysis of experiments on thermoplastic composites under off-axis loading*. 2025. arXiv: 2501.10193 [math.NA]. URL: https://arxiv.org/abs/2501.10193

3. N. Kovács, M.A. Maia, I. B. C. M. Rocha, C. Furtado, P. P. Camanho, and F. P. van der Meer. "Physically Recurrent Neural Networks for computational homogenization of composite materials with microscale debonding". *European Journal of Mechanics - A/Solids* 112 (2025), 105668. ISSN: 0997-7538. DOI: https://doi.org/10.1016/j.euromechsol.2025.105668

2. M. A. Maia, I. B. C. M. Rocha, D. Kovačević, and F. P. van der Meer. "Physically recurrent neural network for rate and path-dependent heterogeneous materials in a finite strain framework". *Mechanics of Materials* 198 (2024), 105145. DOI: https://doi.org/10.1016/j.mechmat.2024.105145

1. M. A. Maia, I. B. C. M. Rocha, P. Kerfriden, and F. P. van der Meer. "Physically recurrent neural networks for path-dependent heterogeneous materials: Embedding constitutive models in a data-driven surrogate". *Computer Methods in Applied Mechanics and Engineering* 407 (2023), 115934. DOI: https://doi.org/10.1016/j.cma.2023.115934

AWARDS

1. Early Career Researcher Prize at the 1st Workshop on Physics Enhancing Machine Learning Machine Learning Applied Solid Mechanics, IOP, London, UK 2022.

[†]Collaborative work, not part of this thesis.

WORKSHOPS

3. M. A. Maia, I. B. C. M. Rocha, and F. P. van der Meer. *A frame-invariant physically recurrent neural network for microscale analysis of rate and path-dependent heterogeneous material.* Invited talk at the 2nd Workshop on Physics-enhancing Machine Learning in Applied Solid Mechanics. 2023. URL: https://iop.eventsair.com/asm2023/programme

2. M. A. Maia, I. B. C. M. Rocha, P. Kerfriden, and F. P. van der Meer. *Neural Networks with Embedded Physics-based Material Models to Accelerate FE^2 Simulations of Path-dependent Heterogeneous Materials.* Invited talk at the 1st Workshop on Physics-enhancing Machine Learning in Applied Solid Mechanics. 2022. URL: https://iop.eventsair.com/asm2022/programme

1. M. A. Maia, I. B. C. M. Rocha, P. Kerfriden, and F. P. van der Meer. *Accelerating multiscale finite element simulations using neural networks with embedded physics-based material models.* Invited talk at the 24th Engineering Mechanics Symposium. 2021. URL: https://engineeringmechanics.nl/2021/05/18/twenty-fourth-engineering-mechanics-symposium

CONFERENCE ARTICLES

1. M. A. Maia, I. B. C. M. Rocha, and F. P. van der Meer. "Neural networks meet physics-based material models: Accelerating concurrent multiscale simulations of path-dependent composite materials". In: *Proceedings of the 20th European Conference on Composite Materials: Composites Meet Sustainability*. EPFL Lausanne, 2022, 891–898

CONFERENCE ABSTRACTS

11. M. A. Maia, A. M. C. M. van Gils, I. B. C. M. Rocha, and F. P. van der Meer. *Exploring the latent space of Physically Recurrent Neural Networks in the low-data regime.* Talk at the Euromech Colloquium 656 Data-driven mechanics and physics of materials. 2025

10. M. A. Maia, A. van Gils, I. B. C. M. Rocha, and F. P. van der Meer. *An Interpretable Multi-task Approach for Physically Recurrent Neural Networks.* Talk at the 3rd IACM Digital Twins in Engineering Conference (DTE 2025) & 1st ECCOMAS Artificial Intelligence and Computational Methods in Applied Science (AICOMAS 2025). 2025. URL: https://dte_aicomas_2025.iacm.info/event/contribution/6920d520-8faa-11ef-b344-000c29ddfc0c

9. I. B. C. M. Rocha, M. A. Maia, N. Kovács, D. Kovačević, P. Kerfriden, and F. P. van der Meer. *Hybrid surrogate modeling for multiscale simulations with Physically Recurrent Neural Networks.* Invited key-note talk at the 3rd Workshop on Physics-enhancing Machine Learning in Applied Solid Mechanics. 2024. URL: https://iop.eventsair.com/asm2024/programme

8. L. Ke, I. B. C. M. Rocha, M. A. Maia, and F. P. van der Meer. *Accelerating Multiscale Modeling of Delamination With a Suite of Surrogate Models.* Talk at the 9th European Congress on Computational Methods in Applied Sciences and Engineering. 2024. URL: https://eccomas2024.org/event/contribution/b5301968-b3a6-11ee-ac5b-000c29ddfc0c

7. M. A. Maia, I. B. C. M. Rocha, D. Kovačević, and F. P. van der Meer. *A physically recurrent neural network for rate dependent composite materials.* Talk at the 9th European Congress on Computational Methods in Applied Sciences and Engineering. 2024. URL: https://eccomas2024.org/event/contribution/d6cbc547-974c-11ee-8a2d-000c29ddfc0c

6. N. Kovács, M. A. Maia, I. B. C. M. Rocha, C. Furtado, P. P. Camanho, and F. P. van der Meer. *Physically Recurrent Neural Networks for Computational Homogenization of Composite Materials with Microscale Debonding.* Talk at the 9th European Congress on Computational Methods in Applied Sciences and Engineering. 2024. URL: https://eccomas2024.org/event/contribution/de91820b-9788-11ee-8a2d-000c29ddfc0c

5. E. Ghane, M. A. Maia, I. B. C. M. Rocha, M. Fagerström, and M. Mirkhalaf. *Efficient Multiscale Analysis of Woven Composites Using Physics-based Hierarchical Neural Networks.* Talk at the 9th European Congress on Computational Methods in Applied Sciences and Engineering. 2024. URL: https://eccomas2024.org/event/contribution/72111f5d-b227-11ee-ac5b-000c29ddfc0c

4. N. Kovács, M. A. Maia, I. B. C. M. Rocha, C. Furtado, P. P. Camanho, and F. P. van der Meer. *Physically Recurrent Neural Networks for Computational Homogenization of Composite Materials with Microscale Debonding.* Talk at the 21st European Conference on Composite Materials (ECCM). 2024. URL: https://eccm21.site.calypso-event.net/programme/programme-vue-synoptique/zoom/abstract-zoom.htm?zoom=659a692f-61a7-ee11-a8d4-005056ac07b9

3. M. A. Maia, I. B. C. M. Rocha, and F. P. van der Meer. *Physically recurrent neural networks for microscale analysis of rate-dependent off-axis unidirectional laminates.* Talk at the 9th ECCOMAS Thematic Conference on the Mechanical Response of Composites. 2023. URL: https://composites2023.cimne.com/event/contribution/e65b5d71-a640-11ed-b019-000c29ddfc0c

2. M. A. Maia, I. B. C. M. Rocha, and F. P. van der Meer. *Neural networks with embedded physics-based material models to accelerate multiscale finite element simulations.* Talk at the 8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS2022). 2022. URL: https://eccomas.org/2021/01/22/3542/

1. M. A. Maia, I. B. C. M. Rocha, and F. P. van der Meer. *Neural networks meet physics-based material models: accelerating FE2 simulations of path-dependent composite materials.* Talk at the 20th European Conference on Composite Materials (ECCM20). 2022. URL: https://eccm20.org/

SUPERVISED MSC THESES

3. A. M. C. M. van Gils. “Predicting maximum stresses in composite microstructures using surrogate models”. MSc thesis. TU Delft, 2025

2. P. J. van IJzendoorn. “Multiscale Modelling of Lattice Materials: a novel approach using Beam Neural Networks”. MSc thesis. TU Delft, 2024

1. N. Kovács. “Physically Recurrent Neural Networks for Cohesive Homogenization of Composite Materials”. MSc thesis. TU Delft, 2023

CURRICULUM VITÆ

Marina ALVES MAIA

09-05-1994 Born in Fortaleza, Brazil.

EDUCATION

2012–2017 Bachelor in Civil Engineering
Department of Civil Engineering
Universidade Federal do Ceara, Brazil
Thesis: Multi-objective optimization of composite risers using bioinspired algorithms
Promotor: Prof. dr. Evandro Parente Junior

2018–2020 Master of Science in Computational Mechanics
Department of Structural Engineering
Universidade Federal do Ceara, Brazil
Thesis: Sequential approximate optimization of composite structures
Promotor: Prof. dr. Evandro Parente Junior

2021–2025 PhD candidate in Computational Mechanics
Department of Civil Engineering and Geosciences
Delft University of Technology, the Netherlands
Copromotor: Dr. Iuri B. C. M. Rocha
Promotor: Dr. ir. Frans P. van der Meer