

Data-driven Analysis of Microstructure-Property Relation in Functional Materials

Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
Genehmigte Dissertation von Binbin Lin aus Zhejiang, China
Tag der Einreichung: 01.06.2023, Tag der Prüfung: 17.07.2023

1. Gutachten: Prof. Bai-Xiang Xu
2. Gutachten: Prof. Sarbajit Banerjee
Darmstadt, Technische Universität Darmstadt



TECHNISCHE
UNIVERSITÄT
DARMSTADT

MECHANICS of
FUNCTIONAL
MATERIALS



Materials and Earth
Sciences Department
Mechanics of Functional
Materials

Data-driven Analysis of Microstructure-Property Relation in Functional Materials

Accepted doctoral thesis by Binbin Lin

Date of submission: 01.06.2023

Date of thesis defense: 17.07.2023

Darmstadt, Technische Universität Darmstadt

Bitte zitieren Sie dieses Dokument als:

URN: urn:nbn:de:tuda-tuprints-244751

URL: <http://tuprints.ulb.tu-darmstadt.de/24475>

Jahr der Veröffentlichung auf TUprints: 2023

Dieses Dokument wird bereitgestellt von tuprints,

E-Publishing-Service der TU Darmstadt

<http://tuprints.ulb.tu-darmstadt.de>

tuprints@ulb.tu-darmstadt.de

Die Veröffentlichung steht unter folgender Creative Commons Lizenz:

Namensnennung – Weitergabe unter gleichen Bedingungen 4.0 International

<https://creativecommons.org/licenses/by-sa/4.0/>

This work is licensed under a Creative Commons License:

Attribution–ShareAlike 4.0 International

<https://creativecommons.org/licenses/by-sa/4.0/>

To my loved ones and those who have been part of my
journey...

Acknowledgements

First and foremost, I would like to begin by expressing my deep appreciation to Prof. Bai-Xiang Xu for granting me the chance to pursue this research in her group. I am grateful for the trust she has placed in me and for involving me in numerous projects that have contributed significantly to my personal development. Thank you for dedicating your time and effort, as well as engaging in countless discussions, personal meetings, and idea exchanges, which have made this work possible.

Further, I would like to express my gratitude to Prof. Sarbajit Banerjee, Prof. Hongbin Zhang, and Prof. Robert Stark for devoting their time to review and evaluate this thesis. Their expertise and valuable input have greatly contributed to the quality and refinement of this work, and I am sincerely appreciative of their efforts.

I would also like to express my gratitude to the DFG (Deutsche Forschungsgemeinschaft) and NHR4CES (National High Performance Computing Center for Computational Engineering Science) for their financial support, which has made my research possible. Additionally, I would like to acknowledge the Lichtenberg supercomputer at TU Darmstadt for providing the computational resources that have greatly accelerated the progress and outcomes of my research.

This work benefits from many internal and external collaborations, I would like to thank the colleagues from the Functional Paper Research project: Julia, Jan-Lukas, the colleagues from Texas A&M University for the battery studies: David, Luis, Justin, the colleagues from the NHR4CES project: Setareh, Janis, Daniel, Sherri, and the Glass project colleagues: Chen Xing, Miriam. Thank you all for your great cooperation.

I would like to thank my colleague and buddy Yangyang for his support on all technical issues as well as his spirit and tireless sharing of his knowledge about science. Many thanks also go to Yang Bai for his support during my initial phase of developing the cohesive zone model, and to Christoph for setting up the latex template during my final writing phase. Thanks also go to Somnath and Nick for their general technical support.

A big applause goes to Maren Arnold for all kinds of administrative support. I would also like to thank my current and former master students: Mozhdeh, Sebastian, Josquin, Nima, Jean, Michael, Ruxin. Some of them are now PhD students and are pursuing their own PhDs. I am sure that the learning and teaching has always been mutual, thank you all for your contributions.

I had a great pleasure working in the Mechanics of Functional Materials Group and enjoyed the time with former members Bai Yang , Ziqi, Liu Yao, Wang Shuai, Yangbin, Peter, Mamun, Liu Wei, Qihua, Yi Min, Habib, Dominik, Hamid, Xiandong. I still enjoy every day with Ruan Hui, Xianglong, Armin, Abdullah, Yifan, Mozhdeh, Xiangfeng, Wan Xin, Unai, Runxin, Shahed, Patrick, Zeeshan, Jiajun, Nick, Somnath, Timi, Bai-Xiang, Yangyang, Sebastian, Jean, Micheal, thank you all for being part of this great journey.

Hier gilt mein Dank auch meinen ehemaligen LehrerInnen Frau Bartels, Frau Dunkel und den Herren Bartelmes, Schlosser und Lüdicke, die mich gefordert und gefördert haben und ohne die meine akademische Laufbahn sicherlich anders verlaufen wäre.

I would like to thank all of my friends for supporting me and all the good moments in my free time.

借此也谢谢我的父母, 何文正, Julia, 二姨, 姑姑和姑父们, 我的岳父岳母, 红星。

最后谢谢我可爱的小伙伴Kiki和我挚爱的妻子Mina, 感谢你们无时无刻的陪伴♡

愿大家都身体健康, 一生平安。
May you all be blessed with love, happiness and health.

Abstract

The interplay between structure and property is a fundamental research topic in materials science and engineering. Materials possess diverse microstructures, and effectively characterizing, representing, and correlating them with properties poses significant challenges. As a result, the understanding of the microstructure-property relation relies primarily on empirical approaches, which limits its application in materials optimization and design. However, the emergence of machine learning and data science methods in recent years has provided powerful tools with immense potential to advance materials research and design principles. These approaches offer promising opportunities to develop materials that meet future needs and have the capability to revolutionize traditional methods of materials research.

This thesis focuses on the application of machine learning techniques to explore the relationships between microstructure and properties. Three prototype microstructural systems are studied: nanowire structure in lithium-ion cathode material, fibrous network structure in paper material, and grain/phase structure in dual-phase steel. The present work investigates different forms of microstructure representation across multiple microstructure levels. These include the use of deep neural networks to derive geometric descriptors to characterize nanowire morphology based on particle-level microscopy images, the derivation of descriptors from the complex fibrous network structure of paper materials at the network level, and the use of image-based latent features at the microstructure domain level for dual-phase steel.

The material properties considered in this work are electrochemical properties obtained from experimental assessments, as in the case of battery cathode material, or from sound physical simulation data generated by sophisticated material models and simulations, as demonstrated for paper material and dual-phase steel. This thesis convincingly demonstrates that the use of machine learning-based techniques enables effective microstructural characterization, extraction of microstructural features, rapid prediction of material response, and ultimately the establishment of microstructure-property relations to facilitate improved material optimization and design.

Zusammenfassung

Die Wechselwirkung zwischen Struktur und Eigenschaften ist ein grundlegendes Forschungsthema in der Materialwissenschaft und Werkstofftechnik. Werkstoffe weisen unterschiedliche Mikrostrukturen auf, deren Charakterisierung, effektive Darstellung, und Korrelation mit den Eigenschaften eine große Herausforderung darstellt. Daher beruht das Verständnis der Wechselwirkung zwischen Mikrostruktur und Eigenschaften hauptsächlich auf empirischen Ansätzen, was die Anwendung bei der Optimierung und Konstruktion von Werkstoffen einschränkt. Das Aufkommen von Methoden des maschinellen Lernens und der Datenwissenschaft in den letzten Jahren hat jedoch leistungsstarke Werkzeuge mit einem enormen Potenzial für die Weiterentwicklung der Materialforschung und der Konstruktionsprinzipien hervorgebracht. Diese Ansätze bieten vielversprechende Möglichkeiten für die Entwicklung von Werkstoffen, die den Anforderungen der Zukunft gerecht werden, und haben das Potenzial, die traditionellen Methoden der Materialforschung zu revolutionieren.

Diese Arbeit konzentriert sich auf die Anwendung von Techniken des maschinellen Lernens zur Erforschung der Beziehungen zwischen Mikrostruktur und Eigenschaften. Drei prototypische mikrostrukturelle Systeme werden untersucht: Nanodrahtstruktur in Lithium-Ionen-Kathodenmaterial, faserige Netzwerkstruktur in Papiermaterial und Korn/Phasenstruktur in Dualphasenstahl. In der vorliegenden Arbeit werden verschiedene Formen der Mikrostrukturdarstellung über mehrere Mikrostrukturebenen hinweg untersucht. Dazu gehören die Verwendung von tiefen neuronalen Netzen zur Ableitung geometrischer Deskriptoren zur Charakterisierung der Nanodrahtmorphologie auf der Grundlage von Mikroskopiebildern auf Partikelebene, die Ableitung von Deskriptoren aus der komplexen faserigen Netzwerkstruktur von Papiermaterialien auf der Netzwerkebene und die Verwendung von bildbasierten latenten Merkmalen auf der Ebene der Mikrostrukturdomäne für Dualphasenstahl.

Bei den in dieser Arbeit betrachteten Materialeigenschaften handelt es sich um elektrochemische Eigenschaften, die aus experimentellen Bewertungen gewonnen werden, wie im Fall des Batteriekathodenmaterials, oder um solide physikalische Simulationsdaten, die durch fortschrittliche Materialmodelle und -simulationen generiert werden, wie für Papiermaterial und Dualphasenstahl der Fall ist. Diese Arbeit legt überzeugend dar,

dass der Einsatz von auf maschinellem Lernen basierenden Techniken eine effektive mikrostrukturelle Charakterisierung, die Extraktion von mikrostrukturellen Merkmalen, eine schnelle Vorhersage der Materialreaktion und schließlich die Erstellung von Mikrostruktur-Eigenschafts-Beziehungen ermöglicht, um eine verbesserte Materialoptimierung und -gestaltung zu erreichen.

Contents

Acknowledgements	I
Abstract	V
1 Introduction	1
1.1 Overview & Motivation	1
1.2 Outline	7
2 Fundamentals and Background	11
2.1 General Materials Aspects	11
2.1.1 Microstructure classes	16
2.1.2 Microstructure representations	18
2.1.3 Concept of micromechanics	23
2.2 Machine Learning Fundamentals	26
2.2.1 Basic concept of machine learning	26
2.2.2 Learning paradigms & task objectives	27
2.2.3 Data and pre-processing	29
2.2.4 Machine learning algorithms and models	31
2.3 Continuum Mechanics and Numerical Approximation	48
2.3.1 Kinematics	48
2.3.2 Concept of stress	51
2.3.3 Constitutive equation	53
2.3.4 Momentum balance principle	54
2.3.5 Variational (weak) form of governing equation	57
2.3.6 Finite element approximation	58
3 Deep Learning-enabled Geometry-Lithiation Correlation in Battery Electrode Material	63
3.1 V_2O_5 nanowires as battery cathode material and electrochemical properties	63



3.2	Data generation and methodologies	66
3.2.1	V ₂ O ₅ nanowires & experimental imaging	66
3.2.2	Lithiation property data	69
3.2.3	Mask R-CNN model & training	70
3.3	Instance segmentation of experimental images	79
3.4	Particle geometry-lithiation correlation analysis	90
3.5	Perspectives on data extension	98
4	Data-driven Microstructure Sensitivity Study of Fibrous Paper Material	101
4.1	Fiber network structure of paper material and mechanical properties . . .	101
4.2	Data generation and methodologies	106
4.2.1	Fiber network generation & feature determination	106
4.2.2	Fiber network model & cohesive FE simulation	110
4.2.3	Shallow ML models & training	115
4.3	Correlation and sensitivity analysis	116
4.4	Perspectives on data extension	121
5	End-to-end Image-based Correlation Learning in Dual-phase Steel	125
5.1	Dual-phase steel material and mechanical properties	125
5.2	Data generation and methodologies	128
5.2.1	Experimental imaging & microstructure phase segmentation . . .	128
5.2.2	Image-based meshing & elastic-plastic FE simulation	128
5.2.3	U-net model & training	133
5.3	End-to-end microstructure-to-field prediction and correlation analysis . .	135
5.4	Perspectives on data extension	138
6	Conclusion and Outlook	141
	Bibliography	146
	List of Figures	171
	List of Tables	173
	Abbreviations	175
	Curriculum Vitae	179
	List of Publications	181

1 Introduction

1.1 Overview & Motivation

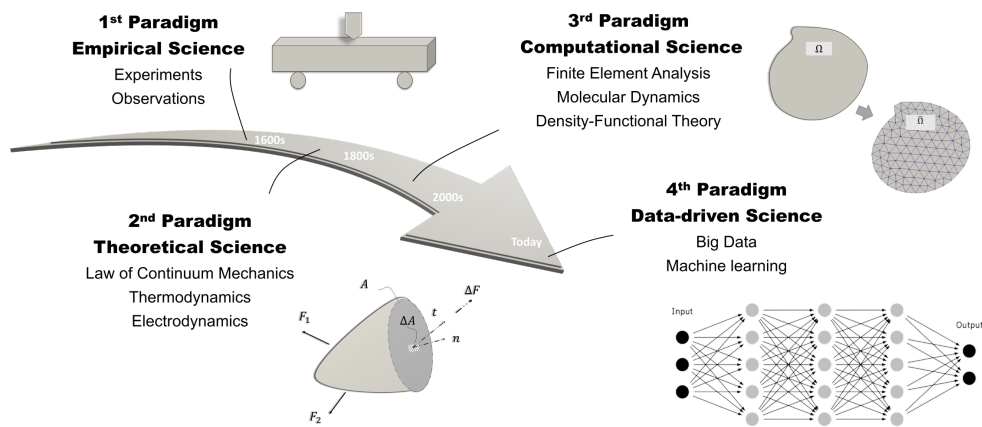


Figure 1.1: The evolution of scientific paradigms involves a profound transformation, shifting from conventional empirical science to complex computational approaches that enhance the understanding in scientific disciplines and create better research methodologies. The advent of artificial intelligence and machine learning opens up new avenues, enabling innovative possibilities that reshape the existing scientific framework.

Artificial intelligence (AI) and machine learning (ML) are driving significant economic and social transformations, and they have become crucial technologies in various research domains, including material science and engineering [1]. The integration of ML and data-driven techniques into scientific research methodology has given rise to what is known as the fourth research paradigm [2], extending the third paradigm of computational science. Commonly, the computational science paradigm employs tools

at multiple length scales, such as density functional theory (DFT), molecular dynamics (MD), and finite element (FE) analysis, to advance scientific research in the field of material science and engineering. The data-driven science paradigm, is often built upon experimental or computational data and is becoming increasingly dominant in the material science community. It combines traditional research paradigms, including established experimental and computational methods, with data science approaches such as machine learning. This integration has found numerous applications, particularly in computational material simulation and design at various scales [3]. Examples include ML-based DFT calculations [4, 5, 6], MD simulations [7, 8], and FE or multi-scale simulations [9, 10, 3]. The synergy among different research paradigms has shown great potential in solving previously complex problems. ML models can now learn and make meaningful predictions without extensive prior knowledge. This new approach shifts the focus of computational simulation from passive, parameter-based feature analysis to a more active, comprehensive, and automated exploration of relevant data, facilitating knowledge transfer across different domains of expertise. Consequently, it enables a deeper understanding of the intricate physics and underlying complex processes in materials research and design.

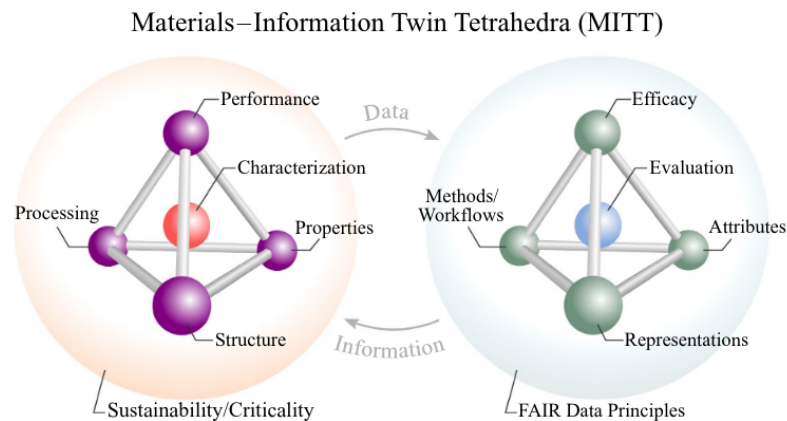


Figure 1.2: Materials tetrahedron and its information tetrahedron twin - emphasizing the workflow from materials data generation and processing to information and knowledge extraction for materials development and design process. Adapted from [11] (CC BY 4.0).

More closely, the fundamental principle in materials science and engineering revolves

around the materials tetrahedron, which aims to investigate and utilize processing-structure-property-performance (PSPP) relationships to advance material development and design [11, 12]. In a typical life cycle of material development, the processing stage involves the preparation and transformation of raw materials into a form that can be utilized. Techniques such as melting and casting are employed, depending on the specific material under consideration. Through these processing methods, a diverse range of materials with distinct structures and properties can be obtained. The term "structure" refers to the organization of a material at various length scales following the processing stage. This encompasses the arrangement of atoms and molecules within the material, as well as the arrangement of larger-scale features like grains, phases, and other structural elements. "Property" describes the characteristics exhibited by a material when subjected to external stimuli. This can include thermal properties like thermal conductivity, capacity, and electrical property such as electrical conductivity or dielectric constant, and mechanical properties such as elasticity and strength. "Performance" pertains to how well a material performs in a specific application, taking into account factors such as fatigue strength, thermal stability, and other considerations specific to its intended use. Lastly, "characterization" refers to the process of quantitatively measuring the attributes associated with processing, structure, properties, and performance. By comprehending the interconnections among these elements, materials can be designed, developed, and produced with customized properties to fulfill specific requirements. To accelerate this process, the extensive use of information science methods is desired in conjunction with the data-driven science approach. These methodologies enable the generation of data and information flows that promote the exchange and streamline the creation of knowledge about PSPP relationships. As Deagen et al. [11] emphasized, the concept of the materials tetrahedron can be extended to the realm of information science, leading to the notion of an "information tetrahedron." This parallel concept highlights the similarities between materials science and information science, as depicted in Fig.1.2. For instance, the analogy between processing in materials science and workflow/methods in information science can be observed in the procedures involved in acquiring, integrating, and analyzing digitized information. In the context of information science, representations serve as mappings of material compositions and structures to data structures, ontologies, spatio-temporal representations, models, and other relevant constructs. Concrete examples of contextualized workflows/methods include correlative characterization [13], which involves the integration of information from multiple analytical techniques to correlate features and properties in a specific region or time of interest, as well as inverse design [4] and microstructure image-based physical simulations [14]. Different types of representations can include physical descriptors as well as different data formats, such as 2D pixel and 3D voxel representations

from experimental imaging. While additional analogies can be established, the primary objective is to establish a comprehensive infrastructure for materials data and informatics that encompasses interconnected elements. This infrastructure aims to facilitate collaboration among experts from diverse disciplines and enable the effective utilization of these resources. In order to ensure widespread adoption within the materials community, these resources need to demonstrate efficient digital representations and methodologies. Additionally, they should establish trust through robust cyber-physical infrastructures and adhere to the principles of FAIR data, which emphasize Findability, Accessibility, Interoperability, and Reusability [15].

Within the process chain of establishing PSPP relations, the interaction between material structure and properties stands out as a crucial topic in materials research. Although material structure encompasses various length scales, this study specifically concentrates on the microstructure and its relation to material properties. The term "microstructure"¹ in this context refers to the structure of the material at the microscopic level, where individual features of the material become visible under a microscope. This scale is commonly defined as the range at which microscopic observations can be made. When designing microstructural material systems, it is crucial to understand the relationship between microstructure and material properties, or referred to as the microstructure-property relation (MPR) in this work. This understanding involves a bi-directional process. In the forward process, the focus is on predicting material properties based on a given microstructure, which is denoted as cause-driven or effect-driven. On the other hand, the inverse design process aims to tailor and enhance a microstructure to meet specific application and performance requirements, known as goals- or means-driven [17]. By controlling the characteristics of the microstructure, it is possible to influence and achieve desired material properties and performance for various application scenarios.

It should be noted, however, that the microstructural features are inherently different depending on the specific material that is being examined. Materials can be broadly classified based on their chemical composition and atomic structure, including metals, ceramics, polymers, composites, and more advanced materials such as semiconductors and bio-composites [12]. Figure 1.3 illustrates scanning electron microscopy images of three microstructures from different material class and exemplary applications. It can be observed that the microstructure of different materials exhibits significant diversity in terms of particle² morphology, spatial arrangement, and overall phase composition at

¹The term microstructure in this work refers to the notation of work [16] as micro-structured materials. That is, the structure appears relates to the average domain size. See later Section 2.1 within the context of structure scales.

²A particle can be considered as the smallest instance at considered scale in the microstructural system

these levels within the microstructural scale. These variations in microstructural systems across different material types provide a fascinating motivation for studying complex microstructures, predicting material properties, and establishing correlations and design principles for advanced functional materials.

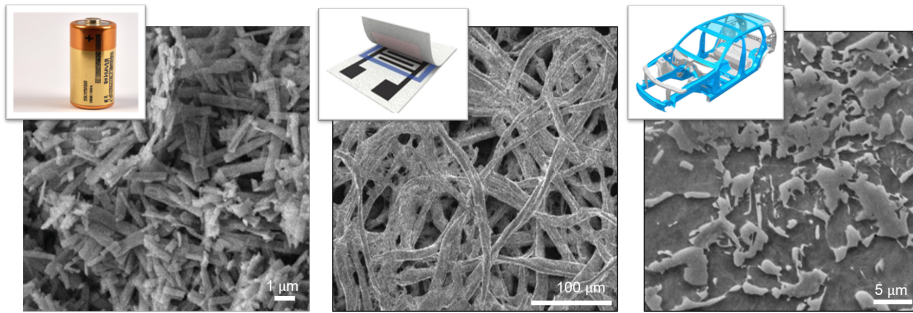


Figure 1.3: Microstructural systems considered in this work and their exemplary material products. **Left:** Microstructure of a battery cathode material [18], reproduced with permission; **Middle:** Fibrous microstructure of a paper-based micro-fluidic device [19], reproduced with permission; **Right:** Microstructure of a dual-phase steel material used in automotive chassis [20].

In pursuit of these objectives, important considerations in material property prediction and design principles revolve around effectively representing the design space of a heterogeneous microstructure system using a limited number of design variables, efficiently reconstructing statistically equivalent microstructures for design evaluation, and rapidly searching for the optimal microstructure design to achieve desired material properties [21]. To explore these aspects, the initial step often involves characterizing³ and developing various forms of microstructure representations, as well as reconstructing digital representations of material microstructures. The initial step typically involves the acquisition of raw microscopy data using advanced imaging techniques. Common experimental characterization methods employed to describe the

without further refining the chemical compositions and structural details of the particle. This can be equivalently a grain, a phase, a fiber or even a defect. (if relevant to the microstructural system)

³To provide clarity, the term "characterization" typically refers to the process of examining the structure of a material through experimental procedures. It involves analyzing various aspects of the material's microstructure. On the other hand, the term "representation" pertains to the output or outcomes of the characterization procedure. This may involve employing advanced image analysis techniques to obtain an appropriate description or depiction of the microstructure.

microstructure include Scanning Electron Microscopy (SEM) [22], X-ray Photoelectron Spectroscopy (XPS) [23], X-ray Diffraction (XRD) [24], X-ray Ptychography (XRP) [25], Focused Ion Beam (FIB) [26], Transmission Electron Microscopy (TEM) [27], Atomic Force Microscopy (AFM) [28], and many others. Subsequently, advanced image analysis tools are required to pre-process and analyze the image data to extract the relevant features to represent the microstructure. Clearly, the task of characterizing and representing the underlying microstructure of a material is challenging and highly dependent on the specific material being studied. The characterization process involves the use of different imaging techniques tailored to the material, followed by the processing and analysis of large amounts of image data. This requires the use of robust, efficient, and flexible tools that go beyond traditional algorithms designed for specific applications. Such tools should be capable of capturing fine details in complex microstructural systems that exhibit variations in particle morphology, spatial arrangement, and overall composition. As mentioned earlier, the use of machine learning techniques in image analysis tools has become increasingly prevalent, offering enhanced capabilities and accelerating the feature analysis process. This enables the automation of characterization processes in materials research [29, 30, 31]. Regardless of the specific imaging techniques used to analyze image data, the results of image characterization and analysis, referred to as microstructure representation in this context, often involve the derivation of physical or geometric descriptors aimed at capturing the morphological features of the microstructure. For example, in the case of polycrystalline materials, descriptors such as grain size, shape, and orientation can serve as suitable indicators of the microstructure under consideration. Describing the microstructure of the material using descriptors is an established method due to its high interpretability and connection to physical causality. Other representation methods include statistical correlation functions, which provide statistical characterizations of the geometric spatial arrangement, and latent-feature-based methods, which are closely related to machine learning approaches [21].

Despite the diversity of material systems and their corresponding microstructural representations, the combination of data science approaches and material simulation methods holds great promise and has already demonstrated numerous applications in the field of MPR. In this study, several interesting applications are presented that use machine learning approaches in conjunction with physics-based data obtained from finite element material simulations. These applications aim to characterize the microstructure, predict the material properties, and determine the MPR for the material classes shown in Figure 1.3. State-of-the-art deep learning methods are used for instance/semantic segmentation, which provides the appropriate data input representations for ML models, enabling MPR model construction and correlation analysis. Notably, significant effort is devoted to the generation and analysis of physics-based simulation data and, in

particular, their workflow automation in the presence of a large dataset. In addition, this study explores the potential and limitations of emerging machine learning techniques that utilize physics-based simulation data for MPR studies and beyond. Overall, this research highlights the promising integration of machine learning and physics-based simulation in the study of microstructure-property relationships, and also discusses future opportunities and potential limitations associated with these techniques.

1.2 Outline

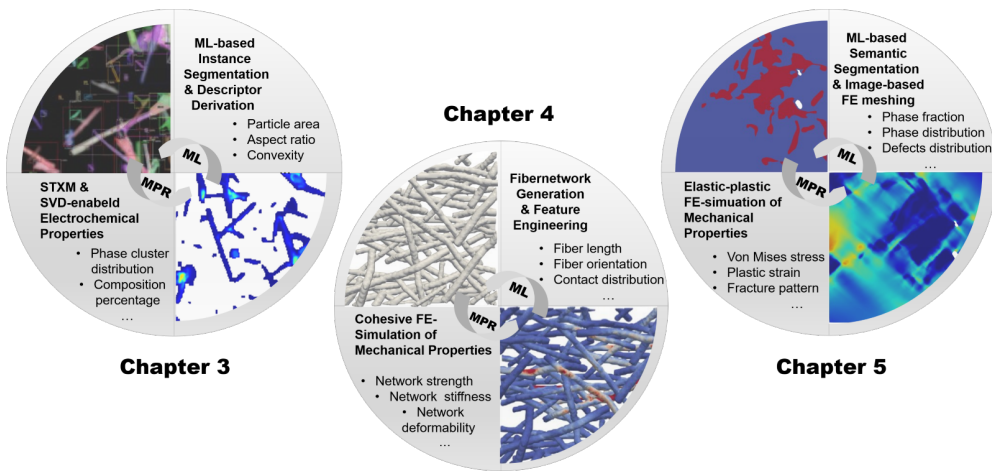


Figure 1.4: Overview of the chapters with considered microstructural systems. While different materials are subject to study, their MPR using machine learning techniques covering various aspects are the common ground of this study. **Left:** An ensemble of segmented nanowire particles of a battery cathode material and their lithiation phase pattern; **Middle:** Synthetically generated fiber network and its mechanical FE simulation showing the stress field; **Right:** An image-based microstructure model of a dual-phase steel and its FE simulation showing the plastic strain field.

The present work investigates different representations of microstructures for three prototypical material classes: nanowire structures of battery cathode materials, fibrous network structure of paper materials, and phase structure of dual-phase steels. Each chapter focuses on the relationship between these microstructures and their respective

material properties. The structure of each main chapter follows a similar format. It begins with an introduction to the material class and an overview of its microstructural system and the specific property of interest. The next section outlines the methods used to obtain the microstructure representation as input and the property data as output. This comprehensive data generation and methodology section includes several subsections, including computational and experimental methods for generating and acquiring the input and output data. It also covers additional steps such as data cleaning, feature engineering and selection, and detailed finite element modeling and simulation to build a comprehensive material property database as output. Next, the chapter introduces the machine learning models used based on the specific representation of microstructures and properties. The training and testing results of these ML models are presented. Following this, the results section provides a data-driven evaluation of the input and output data, along with an analysis of the MPR. It provides valuable insights into the material systems and concludes with a summary specific to the class of materials studied. In addition, each chapter includes interesting perspectives on data extensions and related work performed during the study period, stemming from various collaborations.

An overview of each chapter is provided below: Chapter 2 provides the foundation for the main chapters 3-5 of this study. In section 2.1, general aspects of the structure length scale, different material classes, microstructural systems and the concept of MPR are covered. The focus of the MPR analysis is to examine in detail different types of microstructure representations that serve as input for ML models. Section 2.2 delves into the basic concepts of ML models and workflows. It explains the working pipelines of ML models and explores different types of ML goals. To facilitate understanding and support the overall study framework, simplified synthetic data in the context of materials science are used to introduce the models. Following the introduction to ML concepts, section 2.3 provides an overview of the continuum mechanical framework. It also discusses the numerical discretization scheme using the finite element method, which is used to calculate the mechanical properties of the microstructure systems in the Chapters 4 and 5 .

Chapter 3 focuses on the V_2O_5 nanowire particles as a cathode material for Lithium-Ion Batteries (LIBs) as a canonical system. The chapter begins by introducing the electrochemical properties of the cathode material and highlighting the significance of the phase-transition phenomenon and the coupling between geometry and chemistry. Next, a deep neural network model for particle segmentation is developed. This model specifically addresses the task of particle segmentation in optical density based spectromicroscopy image data. The algorithm for this task is presented along with the experimental dataset to be characterized. The main emphasis of this chapter is on the

development of a synthetic dataset that is used to train the state-of-the-art instance segmentation model, the Mask Regional Convolutional Neural Networks (Mask-RCNN) model. This synthetic dataset helps to overcome the challenge of obtaining a large amount of labeled data, which is typically difficult for complex microscopy data in the field of materials science. After demonstrating the capability and performance of the image analysis tool for different types of spectromicroscopy image data, it is applied to characterize a set of electrochemically lithiated nanowires imaged by scanning transmission X-ray microscopy (STXM). The machine learning segmentation model provides microstructural descriptors that are then correlated with the electrochemical property, in particular, the lithiation pattern of the electrode nanowire particles at the particle level. Finally, the influence of geometry on lithiation properties is investigated by correlation analysis and the results are discussed.

Chapter 4 focuses on the microstructures of fibrous paper material, especially large sets of fiber networks. The mechanical properties of fibrous paper material are calculated using finite element simulation, which serves as the database for this chapter. The main objective of this chapter is to explore how to generate and derive descriptors for the 3D fiber network microstructure and characterize its mechanical properties using a cohesive finite element model. This model captures the fiber/fiber debonding behavior within the deformation of the fibrous paper material system. Once the data set is generated, dimensionality reduction techniques, such as hierarchical clustering, are used to obtain distinctive microstructural descriptors. These descriptors are then used to construct a reduced surrogate model for the MPR. Using the well-trained surrogate model, sensitivity analysis is performed to investigate the influence of design parameters in the fiber network considering the presence of high structural variations. The chapter also discusses strategies to improve the mechanical properties of the fibrous paper material based on the results of the sensitivity analysis.

Chapter 5 focuses on the MPR study of a typical dual-phase steel commonly used in automotive applications. This chapter follows an end-to-end approach in analyzing the material. The representation of the microstructure in this chapter is straightforward and circumvents the need to derive descriptors. Instead, semantically segmented micrographs of the DP steel are used as input data. These micrographs are obtained by applying convolutional neural networks for image segmentation. The corresponding output is the mechanical properties of the material, specifically the mechanical stress/strain field quantities. These quantities are calculated by elastic-plastic finite element simulations based on the segmented micrographs. To enable this end-to-end approach, a fast and accurate image-based finite element meshing step is incorporated. This step transforms the segmented micrographs into finite element meshes, which are then used for subsequent finite element calculations of the mechanical field quantities.

Using input and output data in the form of field or image maps generated on the fly at different positions within a large microstructure image, an advanced convolutional neural network model known as U-Net is trained. This model facilitates the prediction of the MPR in a straightforward manner. In other words, it acts as a surrogate model that takes a micrograph image as the input and seamlessly produces the corresponding mechanical field maps as the output.

Lastly, future perspectives, challenges, and potential limitations of each chapter are summarized and discussed in relation to the overall MPR research framework. This involves considering the potential advancements and areas for improvement in each chapter's specific field of study. It also includes identifying the challenges that may arise in applying the proposed methodologies and addressing any limitations that could impact the effectiveness or generalizability of the results. By examining these aspects, the overall MPR research framework can be further refined and enhanced, paving the way for future developments and advancements in the field. These discussions provide valuable insights into the potential directions for future research and highlight the areas where additional efforts and collaborations may be needed to overcome the identified challenges and limitations.

2 Fundamentals and Background

2.1 General Materials Aspects

The study of the relationship between structure and properties is a fundamental aspect of material science. The term "structure" refers to the arrangement of a material at various length scales, and it significantly influences the material properties. According to Callister et al. [12], structures can be classified into different scales. At the atomic scale, the atomic structure describes the interaction between electrons and atoms, encompassing the arrangement of atoms and molecules within the material. Moving to the next larger scale, the arrangement of grains, phases (in the case of crystalline materials), and other structural features are referred to as microscale structures. Additionally, depending on the structural characteristics of the material, an intermediate scale known as the mesoscale can be identified as the next larger scale. Finally, macroscale structures include structural elements that are visible to the naked eye. In materials science, the term "microstructure" commonly refers to the structure of a material at the microscopic level, where individual features of the material can be observed using a microscope. In the context of this study, the term "microstructure" can be alternatively referred to as "microstructured materials" according to Bargmann et al. [16]. This term emphasizes that the structure is associated with the average domain size and is not limited to the micrometer range. The unit of length can vary and include nanometer or millimeter scales, depending on the geometry of the constituents, such as their morphology, orientation, and spatial distribution.

Although this work specifically focuses on microscale features, it is important to note that the structure of a material is inherently multiscale and can be influenced by various factors at each scale. For instance, in polycrystalline metals, see Fig. 2.1, the structural characteristics encompass lattice structure and lattice defects at the nanoscale, as well as grain/phase formation, impurities, and the presence of larger pores and cracks at the meso- and macroscale. Similarly, in natural cellulose-based materials (see Fig. 2.2), the fibers form meso- and microscale web-like structures, while individual cellulose fibers exhibit hierarchical structures consisting of macrofibrils and microfibrils. At an

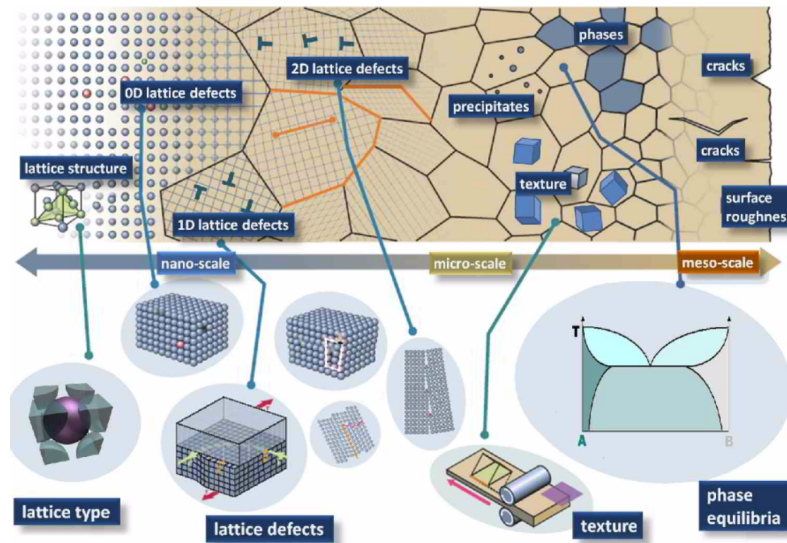


Figure 2.1: Multiscale nature and structural characteristics of metals and alloys [32].

even smaller scale, the underlying components are long chains of cellulose molecules. Therefore, understanding the structure across different length scales is crucial, it provides valuable insight into material behavior and helps material researchers in designing new materials with specific properties or optimizing the performance of existing materials.

To provide a contextual background for the materials investigated in this study, the conventional classification of materials based on chemical composition and atomic structure, as described in the work by Callister et al. [12], is briefly introduced. Metals, being one of the earliest and most commonly used engineering materials since ancient times, are chemical elements that exist in pure form or as alloys, which are mixtures of two or more metallic elements. Metals generally exhibit a crystalline solid structure characterized by an ordered arrangement of atoms and a high degree of symmetry. They possess notable properties such as high electrical and thermal conductivity, high density, and ductility. Many metals are also known for their strength and toughness, making them indispensable in construction, automotive, aerospace, and other applications. There are numerous metallic elements, and they vary in terms of their chemical reactivity, which refers to their ability to lose electrons when interacting with other elements. Highly reactive metals include lithium and potassium. Lithium, for example, is widely used in LIBs. Further, Metals are also divided into transition metals and non-transition

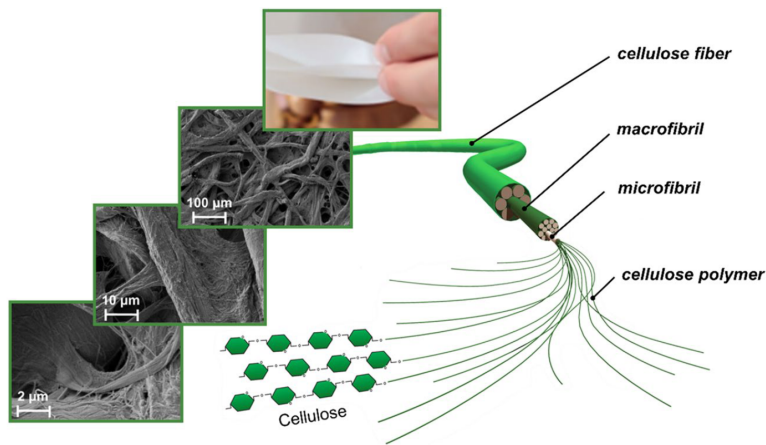


Figure 2.2: Paper material at different scales [33], adapted with permission (CC BY 4.0).

metals based on their atomic and electronic properties. A major distinction between these two categories is the number of valence electrons. Transition metals typically have valence electrons in the d-orbital, which allows them to form various compounds with different chemical properties. In Chapter 3, vanadium pentoxide (V_2O_5) is introduced as a cathode material for LIBs, where vanadium serves as an example of a transition metal capable of forming compounds with oxygen. Consequently, the microstructure of a metal can be influenced by various factors, including the chemical composition of the metal, the processing conditions utilized during its fabrication, the presence of impurities or defects, and the presence of distinct phases or structural features. One of the most commonly observed microstructural features in metals is grain structure, which refers to the arrangement of grains or crystals within the material. The size, shape, and distribution of these grains play a critical role in determining the properties and performance of the metal. In addition, metals can have other microstructural features such as dislocations, defects, or precipitates at various length scales, which can have significant effects on the strength and ductility of the material. An illustrative example of such a microstructure can be found in DP steel, which is discussed in Chapter 5. DP steel is a metal material commonly used in automotive applications and is characterized by its distinct grain and phase microstructure.

Polymers represent another significant class of materials. They consist of large molecules composed of repeating units known as monomers. These materials are organic and carbon-based, existing in various forms such as plastics, fibers, and rubber.

Polymers possess distinctive characteristics including flexibility, strength, and the ability to be shaped or molded into different forms. They are also known for their low density, low melting point, and low electrical and thermal conductivity. Polymers encompass a wide range of types, including natural polymers like proteins and cellulose, as well as synthetic polymers such as polyethylene and polypropylene. In the microstructure of polymers, one common feature is the arrangement of polymer chains. Polymers can exhibit either amorphous or crystalline structures. Amorphous polymers have disordered chains without a regular arrangement, while crystalline polymers have ordered chains with a regular arrangement. The degree of crystallinity significantly influences the properties and performance of a polymer. Other microstructural features in polymers include defects in the polymer chains, such as chain ends or cross-links, as well as the incorporation of additives or fillers dispersed within the polymer matrix. Chapter 4 focuses on the study of the fiber network structure formed by cellulose fibers as a microstructural element. Unlike defects in polymer chains or crystallinity, fibers are considered as homogeneous structures in this work. The microstructural features of the fiber network, such as shape, length, and spatial arrangement of fibers, are analyzed w.r.t. MPR. The fiber network is regarded as the microstructural system due to its size and overall characteristics.

Another major category of materials is ceramics. Ceramics are non-metallic, inorganic materials made from compounds of a metal and a non-metal. They are known for their high hardness, strength, melting point, and low electrical and thermal conductivity. The properties of ceramics can be tailored to specific applications through careful selection of composition, processing of raw materials, and manufacturing and firing conditions. Composites are another important class of materials, consisting of two or more different materials, each with different properties. Composites are designed to combine the advantageous properties of the individual materials, resulting in a material with specific performance requirements. They are used in a variety of industries, including aerospace, construction, and automotive. Composites can be classified into different types, such as metal-ceramic composites and polymer-fiber composites, each with its own unique microstructure. Section 2.1.3 provides more examples and detailed descriptions of microstructural features, which are categorized based on their geometric appearance. In addition to these commonly used engineering materials, there is ongoing development and design of highly advanced smart materials that are tailored for specific applications. These materials include switchable and adaptable semiconductors, as well as bio-composite based materials, which represent advancements in technology and material design.

In the field of materials science, the term "structure" is often associated with the molecular or crystal structure of the material [34, 35], and it plays a crucial role in

determining intrinsic properties such as band gap and formation energies. These intrinsic properties are independent of the amount of material present at the macroscopic level and depend primarily on the chemical composition and atomic structure of the material. In the present context of MPR, the material properties are considered to be extrinsic. This means that the properties of a microstructural system depend on the size, shape and volume fractions of its heterogeneous constituents. The material properties are influenced by the specific arrangement and interactions of these constituents within the microstructure.

In general, a property is a characteristic of a material that relates to the nature and extent of its response to a specific stimulus. For example, when a specimen is subjected to a force, it undergoes deformation. The deformation of a stiff material differs from that of a soft material under the same force, indicating the material's stiffness as a property. Similarly, when a material is exposed to heat, it can conduct and store heat, resulting in properties such as thermal conductivity and heat capacity. In general, solid materials can be divided into six major property classes: mechanical, electrical, thermal, magnetic, optical, and deteriorative properties. Mechanical properties describe how a material responds to mechanical forces or loads. They provide insight into the behavior of the material under various applied forces, including tension, compression, bending, and shear. Examples of mechanical properties include strength, toughness, elasticity, and ductility. Electrical properties refer to a material's response to electrical stimuli, such as applied electric fields. They include properties such as electrical conductivity, which describes a material's ability to conduct electricity, and dielectric constant, which describes a material's ability to store electrical charge. Thermal properties relate to a material's behavior with respect to heat transfer. They include thermal conductivity, which quantifies a material's ability to conduct heat, and heat capacity, which describes its ability to store heat. Deteriorative characteristics relate to the chemical reactivity of materials. While many other properties are equally important, the focus of this work is primarily on the mechanical properties of the paper and steel microstructural systems, which are presented in Chapters 4 and 5. Additionally, in Chapter 3, the electrochemical property of a battery electrode material is studied in detail. The electrochemical property of a battery electrode refers to its ability to undergo redox reactions (electron transfer reactions) when the battery is charged and discharged. These reactions involve the transfer of electrons between the cathode and the anode and are responsible for storing and releasing electrical energy in the battery. More importantly, intercalation or lithiation occurs in the battery cathode material, which is the insertion of lithium ions into the lattice structure of the cathode material. The ability of the cathode material to accommodate ions in close relation to the particle geometry is studied in the respective chapter. This property is also referred to as the lithiation phase pattern or cluster in this

work, which represents the concentration field of lithium ions in the particles. In other words, this property describes the amount of lithium ion uptake in the cathode particles under a given load condition.

Overall, the material properties depend on the problem at hand and can be investigated under different types of external stimuli. The main interest of this work lies in the mechanical and electrochemical properties and their MPR investigations. Therefore, the detailed material properties under investigation will be introduced in the corresponding chapters later on.

2.1.1 Microstructure classes

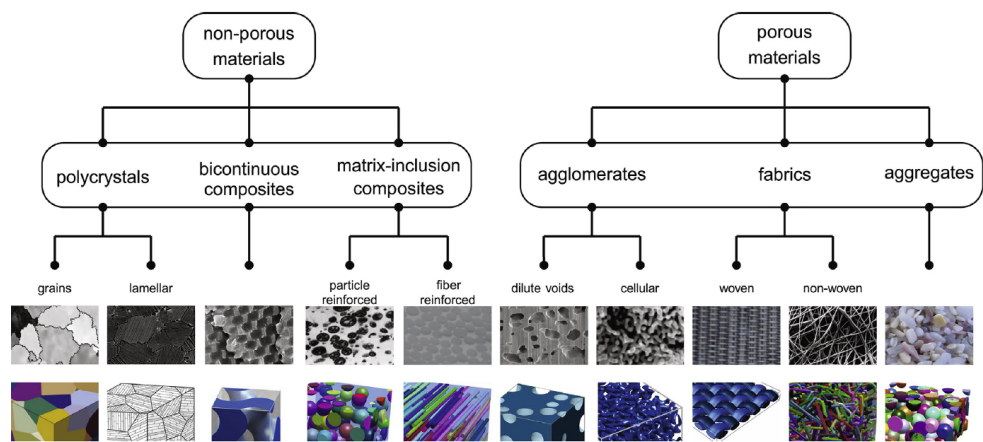


Figure 2.3: Classification of microstructures into porous and non-porous solids and its subdivisions [16], adapted with permission (CC BY-NC-ND).

In many cases, materials exhibit heterogeneous structures at various length scales, which may not necessarily be at the microscale. While materials can be classified based on their chemical compositions, this section provides an overview of different microstructure classes based on their geometric appearance and arrangement, as described by Bargmann et al. [16]. Afterward, different representation forms, regardless of microstructure classes, are presented. The microstructures can be broadly categorized into non-porous and porous media, considering the "space-filling property." This classification focuses on how geometric objects like grains, phases, or fibers fill a given

volume within the microstructure. The following summary provides a brief overview of this classification scheme. Non-porous solids are divided into three main categories: polycrystals, bicontinuous composites, and matrix-inclusion composites. Polycrystal materials, such as metals, alloys, or ceramics, consist of grains with grain boundaries that connect different grains. These grains have similar compositions except for their crystallographic orientations. Important microstructural features in polycrystals include grain size, shape, orientation, distribution, and grain boundary characteristics. Advanced imaging techniques like electron backscatter diffraction (EBSD) [36] and X-ray diffraction contrast tomography (DCT) [37] can be used to determine grain orientation and crystallographic properties. Lamellar microstructures are similar to grain structures, but with grains further separated into thin lamellae. In this case, the grains are referred to as colonies. This microstructure is commonly observed in alloys subjected to specific heat treatments, such as some certain two-phase titanium alloys. Structural features of lamellar microstructures include the number of lamellae lay-ups, the thickness of the lamellae, and their waviness within the grains. Bicontinuous microstructures are composite materials consisting of two continuous, interpenetrating, and interconnected phases. These two phases are typically arranged in a pattern that forms a network-like structure. Bicontinuous microstructures are often designed to combine the advantageous characteristics of both phases, resulting in a material with specific performance requirements. One example is a metal-matrix composite where a metal phase and a second phase of ceramic or polymer material form the bicontinuous structure. Important structural features of bicontinuous microstructures include the volume fraction of each phase, the presence of open and dead pores, and the path length of one phase relative to the other. Matrix-inclusion composites are composite materials consisting of a matrix phase and one or more inclusion phases. The matrix phase is usually continuous, enveloping and holding together the inclusion phases. Inclusion phases in matrix-inclusion composites can exist as particles, fibers, or platelets, and they are dispersed throughout the matrix phase. Matrix-inclusion composites are commonly used in applications where the combination of the two phases enhances the overall performance compared to using a single-phase material. Fiber-reinforced plastics, where fibers are dispersed within a plastic matrix, are one example of matrix-inclusion composites. These fibers are often aligned in a specific direction to provide improved strength and stiffness in that direction. Another example is a metal matrix composite, where inclusion particles are dispersed within a metallic matrix. An aluminum metal matrix with silicon carbide inclusions for aerospace applications is a typical composition in such composites.

To effectively design microstructured material systems by understanding the relationship between microstructure and material properties, a critical challenge is to characterize the material system and identify microstructural feature parameters that

can be correlated with material properties. For clarification, the term "characterization" is often used to describe the material in experimental characterization techniques that result in microstructure descriptors. We use the term "representation" to refer to the end product of the characterization process. This term can be viewed as the theoretical description of the microstructures and is the input data to the machine learning models for MPR. Various types of microstructure representations have been proposed in the literature. Commonly used methods include the use of a set of geometric descriptors or statistical correlation functions, each of which offers its own advantages [21]. More recently, machine learning-based approaches have been used to represent features for material design [38]. These approaches leverage the capabilities of machine learning algorithms to extract relevant information and patterns from microstructural data.

2.1.2 Microstructure representations

In this subsection, a brief review of the available methods in the literature for microstructure representation is presented. These methods aim to capture the essential characteristics of microstructures and to enable their integration into computational models and ML models for property prediction and materials design.

Descriptor-based representation

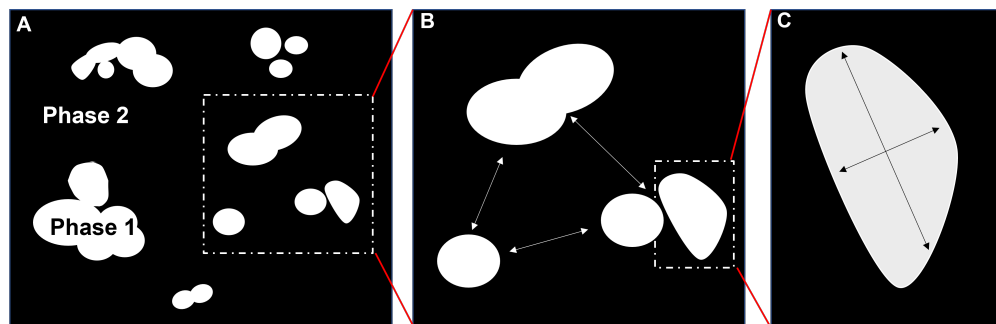


Figure 2.4: Hierarchical levels of descriptor-based representations: A: Composition level; B: Dispersion level; C: Particle geometry level.

The feature or design space is typically the entire microstructural domain and can usually be represented in reduced form, i.e., using a limited number of descriptors

sufficient to capture the relevant features of the material. This is a common practice when characterizing the microstructure of the material. The difficulty lies in identifying the critical microstructural features that influence the material properties and selecting a set of descriptors that can accurately quantify these features. In the descriptor-based representation, the microstructural features can be characterized at three different levels [21], namely the composition, dispersion, and particle phase geometry levels. Using a two-phase composite as an example, composition describes the phases of a material and their relative volume fractions. This descriptor can be used to distinguish between different components of a material, such as the volume fraction of inclusion particles in a two-phase composite. Dispersion descriptors describe the spatial relationship and neighborhood status of inclusions in a material. These descriptors can include nearest neighbor distance, number of filler clusters, and other measures of the distribution of inclusions within the material. Geometry or morphology descriptors describe the shapes of inclusions in a material. They can include size, surface area, aspect ratio, surface-to-volume ratio, roundness, eccentricity, elongation, rectangularity, and many others. These descriptors are typically used to describe inclusions at the lowest level of microstructure. These features can be either deterministic or statistical. Fig. 2.4 shows the microstructural descriptors at these three levels. Furthermore, the descriptor-based representation allows a parametric optimization approach to search for the optimal microstructure design that satisfies the target material properties. The descriptors have a high degree of physical meaning and are interpretable. However, reducing the representation to a handful of descriptors can lead to underrepresentation of the actual design space and loss of information. Using a carefully selected set of microstructural descriptors can help explore the design space more efficiently and identify materials with the desired properties and performance. Further work [39] attempts to define the so-called "global parameters". There are eight such parameters, namely line length, surface area, volume fraction, line curvature, line torsion, surface curvature, number, and connectivity. Whenever a physical property can be directly related to one of these parameters, the relationship is insensitive to the value of any other geometric property of the structure. Average geometric properties of the microstructure can be obtained by taking ratios of the fundamental parameters. These can be used to formulate structure-property relationships when the average is composed of uniquely defined components. While this universal concept is interesting, some of these global parameters are usually difficult to characterize for complex microstructural systems.

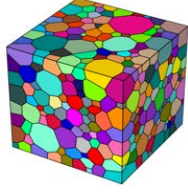
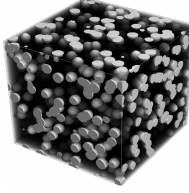
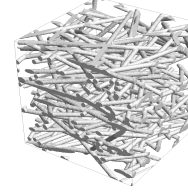
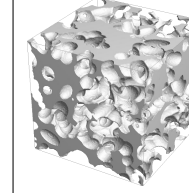
Micro-structure classes	Polycrystalline material	Composite material (Matrix-Inclusion system)	Fibrous material	Cellular material
Level of descriptors				
Composition	-	Volume fraction	Grammage or porosity	Solid volume phase percentage or porosity
Dispersion	<ul style="list-style-type: none"> - Texture of the polycrystal - Grain orientation distribution 	<ul style="list-style-type: none"> - Average distance of inclusion particles - Number of inclusions 	<ul style="list-style-type: none"> - Fiber orientation - Anisotropy - Contact distribution - Number of contacts 	<ul style="list-style-type: none"> - Tortusity - Pore distribution - Number of open or closed pores
Particle	<ul style="list-style-type: none"> - Grain size - Grain shape - Grain boundary length - ... 	<ul style="list-style-type: none"> - Inclusion size - Aspect ratio - Roundness - ... 	<ul style="list-style-type: none"> - Fiber length - Fiber diameter - Fiber curl - ... 	<ul style="list-style-type: none"> - Ligament diameter - Pore size - Surface area - ..

Figure 2.5: Descriptor-based representations for exemplary microstructure classes.

Statistical correlation function-based representation

A more general type of representation is the use of so-called statistical correlation functions. Among several types of correlation functions [40], the most commonly used correlation function is the n-point correlation and its simplified variants.

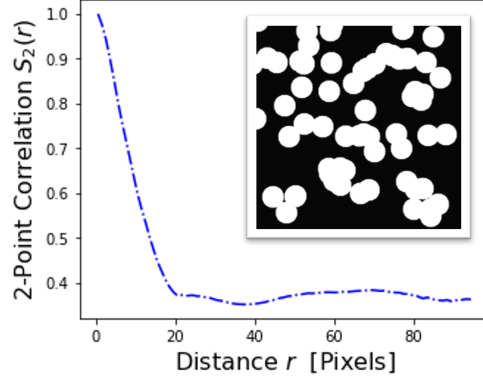
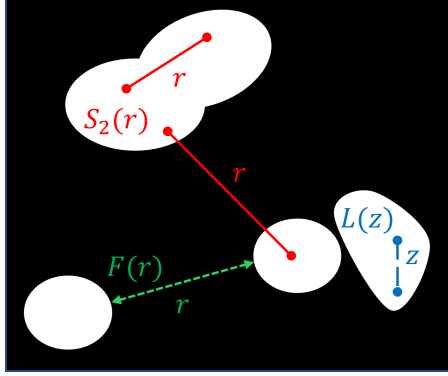
$$I(\mathbf{x}) \begin{cases} 1, & \text{if } \mathbf{x} \text{ in the inclusion phase} \\ 0, & \text{otherwise.} \end{cases} \quad (2.1)$$

The n-point correlation function can then be defined as [40]:

$$S_n(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) = \langle I(\mathbf{x}_1)I(\mathbf{x}_2)\dots, I(\mathbf{x}_n) \rangle \quad (2.2)$$

where the angular bracket is interpreted as the probability of n points in the same inclusion phase at locations $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ are found. However, due to the high computational cost of the n-point correlation, two-point correlation functions are frequently utilized instead. The simplification to two-point correlation is straightforward and can be read as:

$$S_2(\mathbf{x}_1, \mathbf{x}_2) = \langle I(\mathbf{x}_1)I(\mathbf{x}_2) \rangle = S_2(r) \quad (2.3)$$



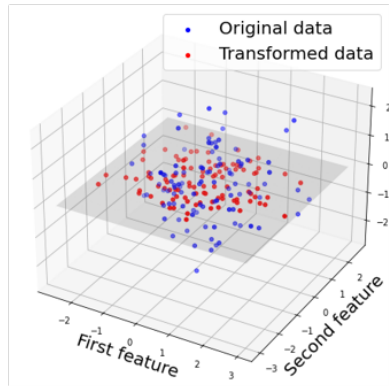
- (a) Exemplary illustration showing different correlation functions: Two-point correlation $S_2(r)$; Surface correlation $F(r)$; Lineal path correlation.
- (b) Normalized two-point correlation function for an exemplary microstructural image of a two-phase material, matrix phase shown in black and inclusion phase in white, respectively.

Figure 2.6: Examples of statistical correlation functions

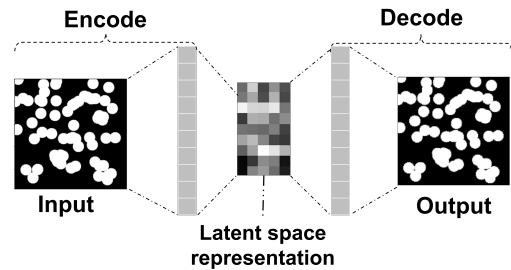
with $r = |\mathbf{x}_2 - \mathbf{x}_1|$ as the Euclidean distance between two points in the same phase. The $S_2 = (r)$ function is computed as the probability of all possible distances for paired two points in the same inclusion phase, see Fig. 2.6b. In other words, it describes the probability of finding two particles at a certain distance from each other changes with the distance. Other examples include the surface correlation function $F(r)$:

$$F(r) = \langle \mathcal{M}(\mathbf{x}_1)\mathcal{M}(\mathbf{x}_2) \rangle \quad (2.4)$$

with \mathcal{M} as the interface identification function. The surface correlation function describes the statistical relationship between the distances between pairs of points on the surface of a material phase. It is often used to relate the interface properties of the material. The lineal path function is another example that describes the probability that a line segment of length z lies entirely in a material phase. $L(z)$ thus contains degenerate connectedness information along a path in that material phase. There are many other correlation functions that provide information about the relative positions of different phases within the microstructure and can be used to describe the microstructure in greater detail. See the work by Torquato [40] for further details.



(a) PCA reduces a tree-feature descriptor space to a plane of two principal components, with the transparent plane to be the principal planes or dimensions



(b) Schematic structure of an autoencoder. The encoding path takes the image as the input and flattens to a 1-D array for encoding. The encoded structure can be seen as the latent space representation that is then subject to decoding to reconstruct the input image.

Figure 2.7: Examples of latent space representations.

Image- and latent space-based feature representation

In machine learning, a latent space generally refers to a lower-dimensional representation of the data because the relationships between the data in this space are not necessarily directly observable in the original data. The idea is that the latent space features capture the underlying patterns and relationships in the data that can be used for relational interpretation without explicitly defining features. It can be considered as compact representation of the data that preserves important information, relationships, and structure. In dimensionality reduction techniques, such as Principal Component Analysis (PCA) or Singular Value Decomposition (SVD), the latent space is represented by a set of principal components or singular values, see Fig. 2.7a. Generally, both dimensionality reduction techniques are used to reduce the number of strongly correlated descriptors to a subset of less or uncorrelated descriptors, thus increasing the training efficiency in machine learning models [38]. In the present context in Chapter 3, SVD and PCA are applied to a spectral dataset to obtain distinctive phase clusters of lithium content in cathode particles. In representation learning techniques, such as Autoencoder (AE) and Convolutional Neural Networks (CNN), the latent space is represented by a series of encoded features or mapping functions, with the aim of learning an efficient representation while compressing the dataset, removing noises and capturing

the underlying structures and patterns in the data [41]. For example, in an AE, see Fig. 2.7b, the latent representations are learned through encoding, that is, in a most simple case, a hidden neural network layer with a reduced number of neurons than the input neurons. See Section 2.2 for detailed information about neural networks. The decoding operations can be seen as a new hidden layer to reconstruct the original data [42]. The encoding and decoding operations can be also carried out in form of convolutional operations, where the convoluted feature maps can be considered as latent space feature representations. This is performed in Chapter 5 for the DP steel microstructures. For generative models such as Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN), the latent space is equally represented by a set of latent variables, that are typically modeled as a set of normally distributed random variables based on the input data [43, 44]. These are a few examples of latent space features, and there are many more techniques and models that use latent spaces in different ways. The choice of latent space features to use depends on the specific problem and the type of data being analyzed.

2.1.3 Concept of micromechanics

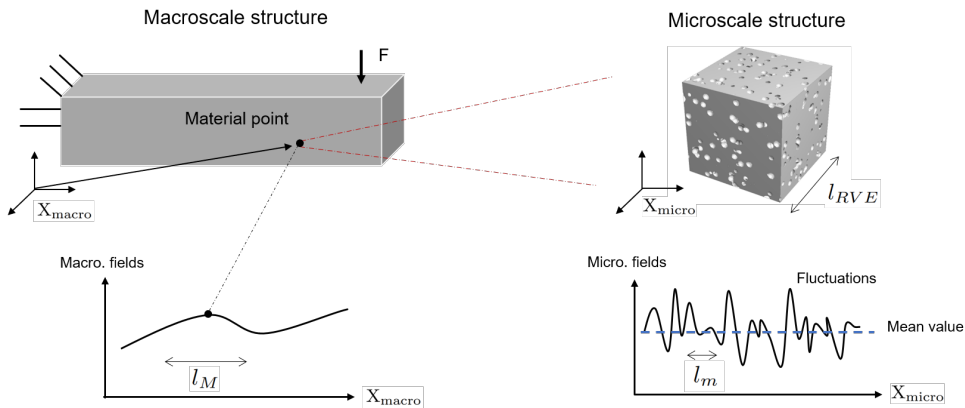


Figure 2.8: Characteristic length and decoupling of the scales. Recreated from Micromechanics lecture notes by Bai-Xiang Xu.

In the presence of a heterogeneous microstructural system, independent of materials or external stimuli, and whose properties are to be predicted, general continuum

mechanical theories do not consider individual microstructural features in the material. On the contrary, micromechanics is a concept for resolving the underlying detailed structural components and spatial arrangement of a macroscopic material that captures the fluctuations of a field variable of interest given by the microstructural heterogeneity. It uses classical continuum mechanical theory, but treats each microstructural phase constituent equivalently as a continuum. Micromechanics thus provides a detailed view into the microstructure of a considered material and aims to understand the origin of the resulting macroscopic material properties by analyzing the underlying microstructures. It uses the concept of the Representative Volume Element (RVE), a unit of volume whose effective property is equivalent to the macroscopic material property. In essence, the macroscopic fields (e.g. stress and strain) at a material point are given by the average of the microscopic stress and strain fields over any RVE located at the same material point, resulting in a boundary value problem at the microscale for the RVE, see Fig. 2.8. To use the RVE concept, the scales must be decoupled [45, 46, 47]:

$$l_m \ll l_{RVE} \ll l_M, \quad (2.5)$$

where l_m is the characteristic length of the microstructure, at which the microscopic field of interest changes (e.g. through pores or defects, as can be found in Fig. 2.8). l_{RVE} is the dimension of the RVE, that captures the statistical distributions of representative microstructural features. l_M denote the characteristic length associated with macroscopic solids, at which the macroscopic quantities change. In particular, the fluctuations of the field variables on l_M are not considered on the macroscopic scale, but only the mean values. Conversely, the gradient of the macroscopic field is not considered on the microscopic scale. In many microstructural systems such as in composite systems mentioned earlier, the RVE is approximated as a unit cell and considered periodic. In this case, l_{RVE} can be given by the periodic length. However, most real microstructures are non-periodic, so determining an appropriate size of RVE to capture the statistical representativeness is a crucial aspect for computational modeling and simulation and has been discussed for many material and microstructural systems [48, 49, 50]. The size of the RVE depends strongly on the material of interest, the characteristics of the structural features and the considered properties. A major drawback of RVE lies in the assumption that an RVE must be spatially invariant and large enough to contain the necessary number of microstructural features and must be a representative microstructure of the entire material. This implies that if the size of the RVE becomes infinite $\frac{l_{RVE}}{l_m} \rightarrow \infty$, the scatter in an average material response vanishes, and the related problem becomes a conventional, deterministic continuum mechanics problem, and suffers from high computational cost [50]. To mitigate this issue, Statistical Volume Elements (SVE)

are commonly used instead, which take a smaller unit size than conventional RVE but larger than the characteristic length of the microscale structure. By doing so, the randomness¹ in the microstructure can be captured and correlated to quantify the macroscopic variation in material properties [51]. The SVE size is then characterized by a dimensionless parameter:

$$\delta = \frac{l_{RVE}}{l_m}. \quad (2.6)$$

In particular, SVE transforms to RVE when $\delta \rightarrow \infty$. In practice, SVE also depends on the characteristic microscale length, and δ must be determined for the heterogeneous microstructure system at hand. In fact, the smaller the chosen δ , the more random microstructure configurations must be considered to ensure that the variance and sensitivity are captured for the material properties under consideration. For examples, δ was determined to be approximately 20 for a porous steel alloy material with about 100 random microstructures of a given SVE size in [50].

Finally, it should be noted that although no direct homogenization was performed in the current work, by means of SVE, this work demonstrates how microstructural characteristics influence the material property of interest for various material systems, leveraging different microstructure representations and machine learning techniques. In other words, while mean values are of interest for multi-scale simulations via homogenization, MPR in this work aims to understand and characterize the origin of fluctuations in microscopic response caused by various microstructural features.

¹The randomness of a microstructure is characterized by considering multiple number of realizations of a given volume of microstructure [49].

2.2 Machine Learning Fundamentals

2.2.1 Basic concept of machine learning

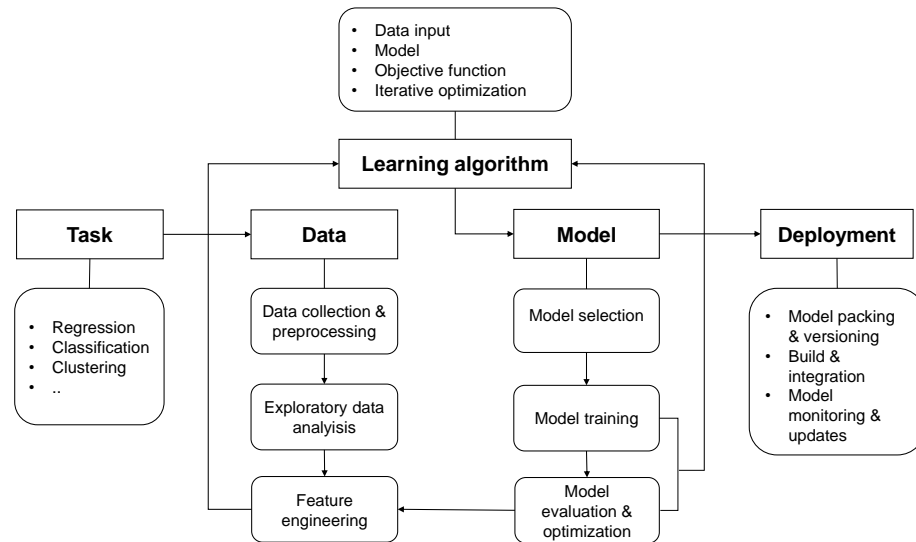


Figure 2.9: Machine learning workflow: The main components of a machine learning workflow pipeline are: Task, Data, Model, Deployment. The learning algorithm "drives" the model to "learn" the data by iteratively optimizing the objective function.

Most generally, ML can be defined as an algorithm that learns from experience (E) with respect to some class of tasks (T) and performance measure (P) if its performance on task T as measured by P improves with experience E without explicit programming [52]. A typical workflow of any machine learning pipeline basically consists of the following steps.

- *Task and objective definition.* The first step is to set and define the task that the machine learning model will perform (e.g., classification, regression, clustering), including possible objective function formulations.
- *Data collection and processing.* The next step is to collect the data that will be used to train the model. Clean and preprocess the data to remove missing values,

outliers, or any inconsistencies in the dataset. Often, exploratory data analysis is performed to gain initial understanding and correlation patterns in the dataset in order to proceed with the feature engineering step, i.e., selecting and engineering the relevant features that will be used as inputs to the model. This step can significantly increase the efficiency of model training and interpretation of results. The data is then separated into training and test or validation sets for later model training and evaluation steps.

- *Model selection, training, and evaluation.* In this step, the machine learning model is selected or developed depending on the actual task to be solved and the data collected. The model itself refers to the mathematical formulation that takes the input and produces the output, regardless of the type of output. In most popular machine learning literature, model training usually refers to an optimization problem specified by iteratively minimizing an objective function using a gradient-descent based method. This step can be specified as the learning algorithm. After the model is trained on the training data, the performance of the trained model is evaluated on a separate test dataset based on predefined metrics. Based on the evaluation results, the model is fine-tuned by adjusting the hyperparameters, features, or models. These training and evaluation steps are repeated until adequate performance is achieved.
- *Model deployment.* Once the model is well-trained and tested, it can be deployed to production. From a software engineering perspective, the next steps include model packaging, versioning, and integration into the production environment. The performance of the model must be monitored over time based on incoming data. Updates to the model must be made based on new incoming data or changing requirements of the specified task.

The following sections present the above steps from a general perspective, but also emphasize the specific subject of study related to the main chapters.

2.2.2 Learning paradigms & task objectives

As shown earlier, different tasks can be formulated by the machine learning models. Based on different tasks and objectives, machine learning can be broadly categorized into three learning paradigms: *Supervised Learning*, *Unsupervised Learning*, and *Reinforcement Learning*. This section briefly introduces the main learning paradigms and their tasks or objectives.

Supervised learning Supervised learning, the most commonly used approach, where the model is provided with an example set of paired *inputs* x_i and *output* y_i data, known as *training* dataset labeled by the user or domain expert. The input dataset x_i is a M -dimensional array or in \mathbb{R}^M in general case, with $i = 1, 2, \dots, N$ and N as the number of samples. They are also called *features*, *descriptors* or *independent variables*, stored in a matrix \mathbf{X} of size $M \times N$ or written as $\mathbb{R}^{M \times N}$. Similarly, one output data sample y can be a one-dimensional or multi-dimensional array, which is referred to as *response*, *property* or *dependent variable*. If $y_i \in \mathbb{R}$, then the problem is referred to as a single-output problem, otherwise a multi-output problem. The input feature data x_i are generally multi-dimensional. Depending on the type of the output data, either a *continuous* or *categorical* and *discrete* value (or array), a *regression* or *classification* problem is formulated, respectively. An example in the context of the MPR of a polycrystalline material could be the (average) grain size, (average) grain orientation as feature inputs x , and the strength of the material as output y for a regression problem. Alternatively, it can be formulated as a classification problem that classifies the inputs x as $y = 0$ or 1 , denoting a brittle or ductile material class, respectively. Independent of the type of the problem, the supervised approach seeks to minimize the so-called *cost*, *loss* or *objective function* L , i.e:

$$\min L := \|f(\mathbf{x}) - y\| \quad (2.7)$$

with $\hat{y} = f(\mathbf{x})$ denotes the model f that maps the input x to the output \hat{y} , and $\|\cdot\|$ arbitrary norm function that maps $\mathbb{R}^N \rightarrow \mathbb{R}$. The algorithm for the actual "learning" is presented in the model and algorithm Section 2.2.4.

Unsupervised learning In unsupervised learning, the model is only provided with a set inputs x_i without any labeled output. The goal is to discover possible "hidden" patterns in the data. Typical approaches include *clustering*, *dimensionality reduction*, *graph structure discovery*, *imputation tasks* and so on. More mathematically, unsupervised learning can be formulated as a self-mapping:

$$\min L := \|f(\mathbf{x}) - \mathbf{x}\| \quad (2.8)$$

where $f(\mathbf{x})$ can be seen as a density estimation model $p(\mathbf{x}|\Theta)$ to determine the model as a probabilistic estimate with parameter Θ yet to specify. From a probabilistic point of view, according to [53], the difference from the supervised case is that the density estimate is conditional on $p(y|\mathbf{x}, \Theta)$ due to the given label y . In a later section, the clustering and dimensionality reduction approaches are applied as preprocessing procedures to the dataset. In Chapter 3, the spectral image data used for the feature output was obtained using principal component analysis and singular value decomposition, both defined as

unsupervised learning algorithms. In Chapter 4, the input microstructure features are clustered to filter out co-linear features with dimension reduction using the hierarchical clustering algorithm prior to model training.

Reinforcement learning Reinforcement learning, along with the other two machine learning paradigms, represents a fundamentally different approach, where an agent or machine takes actions in an environment and tries to maximize reward through trial and error. This approach allows the machine to make decisions without human intervention or explicitly programmed rules to perform a task. Some examples include autonomous driving [54], robot control [55], and strategy games such as Google’s famous AlphaGo [56]. Recent publications [57, 58] present some of the ideas for using deep reinforcement learning for digital material design and demonstrate the striking potential of this third paradigm of machine learning. This approach is beyond the scope of the current work, so the reader is referred to recent literature for more information.

2.2.3 Data and pre-processing

Data is the essential component of the entire machine learning pipeline and is often critical to the success of a machine learning model. While the Material Genome Initiative [59] has been in existence for several years with a focus on the synergy of experiment, theory, and computation on crystal structure databases, microstructure databases have been less common due to the inherent diversity of microstructure features, sizes, and the difficulty in establishing an overall framework to represent the structural properties in a unified manner. Only a few databases of special material systems have recently been established for nanocomposites [60], mechanical metamaterials [61]. However, general purpose data storage platforms such as Kadi4mat [62] and data exchange protocols of the National Research Data Infrastructure for Engineering sciences (NFDI4Ing) [63] make it possible to collect and store the data bases respecting the FAIR principles.

In the present study, the data depend on the representation of the microstructures, such as microstructure feature descriptors or microscopy images in pixelated form. In Chapter 3, image analysis using a deep learning model is performed on a set of microscopy images to obtain the morphological descriptors as an input dataset for the ML model. While the descriptors are feature arrays, the deep learning model developed in this chapter took mask images and mask contour points as an unstructured dataset in the form of dictionaries. In Chapter 4, microstructural feature descriptors are used and derived from different three-dimensional voxelized fiber network datasets as the input dataset. The Chapter 5 uses segmented microstructures and simulated strain and stress

field maps in the form of images. Since different types of data formats are used in this work, some common data preprocessing steps are presented below. Specific data formats and preprocessing procedures are given in the respective chapters. In a general workflow, a significant amount of data will be re-recorded and will undergo steps such as cleaning, transformation, and feature extraction or data reduction. Incorrect data preprocessing often leads to inadequate model training and misinterpretation of prediction results. The steps include:

- Data cleaning:
 - *Missing values.* The methods for handling missing values can generally be either by removing the data samples or manually imputing the missing numbers, or by using numerical methods such as interpolation or calculated averages. Other popular imputation methods for filling in missing values can be found in [64].
 - *Noisy data or outliers.* This problem can have several sources, but is often related to instrumental settings in the context of experimental characterization. Popular methods include binning, which is a technique in which the data are sorted and partitioned into equally sized bins, and then the noisy data are replaced by the bin mean or bin median. Regression analysis can be used to fit a trend in the data to smooth out the noise in the data set. Clustering can be used to group the data into clusters and remove the outliers that are not in the clusters.
- Data transformation:
 - *Normalization.* In this context, it refers to transforming feature descriptors to be on a similar scale so that the efficiency of model training and performance is increased. Common methods include scaling the values in a certain range, such as standard, min-max, or logarithmic scaling.
 - *Aggregation and integration.* Newly retrieved data often needs to be aggregated into existing databases. This step often requires converting the data format to the desired database format.
- Data reduction:
 - *Feature extraction.* Feature extraction or engineering can significantly reduce the data set by selecting or generating important features from the raw data related to the specific task. This step usually requires domain knowledge to extract the characteristics and properties of the collected data. The main

goal is to use the extracted features to improve the performance of the model and the quality of the machine learning predictions. Many approaches, such as principal components, hierarchical clustering, and sensitivity analysis, can help reduce correlated features or re-create features that are important to the actual machine learning model and task to reduce unnecessary computational time and resources.

- *Data compression.* The goal of data compression is to reduce the size of data without losing information for storage and processing efficiency. Common methods, such as one-hot encoding, represent categorical data by representing it as vectors of zeros and ones.
- *Data discretization.* Discretizing continuous numbers into discrete intervals can help further reduce data storage and simplify the problem formulation. For example, given a range of numbers from 0 to 100, discretizing into categories such as "below 50" or "above 50" can change the regression task to a classification task, thus increasing the correlation of input variables with target variables, instead of using continuous numbers.

2.2.4 Machine learning algorithms and models

In this section, the learning algorithm is first demonstrated using the supervised learning paradigm. An example of a simple linear regression is given. Then, a clustering task is presented as an example of the unsupervised learning paradigm. The machine learning models can generally be divided into conventional or shallow and deep learning models. The main difference lies in the feature extraction steps. Feature extraction in shallow machine learning is a manual process that requires domain knowledge of the data and the step of feature engineering. In contrast, deep learning models can extract features from raw data with little or no preprocessing effort [65, 66]. The basics of these models used in the main chapters are presented below. At the end of this section, common issues in the model performance evaluation step, such as over- and under-fitting, are discussed.

For the machine learning pipeline, the input data is divided into different parts for training, validating, and testing the model. The training dataset is the portion of the data used to fit the model. The validation dataset evaluates a model's fit to the training dataset while tuning the model's hyperparameters. The test dataset is the held-out dataset and is used to evaluate the generalization performance of the trained model. To train the model, the actual learning is done by minimizing the loss function, usually defined as the difference between the model prediction data, denoted as \hat{y}_i , and the target training

data y_i . An optimally learned machine model should predict accurately on the test output data. A model is then said to generalize well or to have learned to perform the tasks. The goodness or accuracy of the model can be evaluated with different kinds of score metrics, depending on the task. The simplest example of supervised machine learning is linear regression, which will now be introduced.

Example: Linear regression In the first data collection step, given that a series of input data points as x_1, x_2, \dots, x_N , and the corresponding output data points as y_1, y_2, \dots, y_N measured from an experiment, one assumes that the linear mapping $f : \mathbb{R} \rightarrow \mathbb{R}$ given by the equation $f(x) = mx + b$ with $m, b \in \mathbb{R}$ can capture the general relation between the input and output data. m, b are referred to as the slope and intercept of the line. See Fig 2.10. The training of the model in this context is to find the linear line parameterized

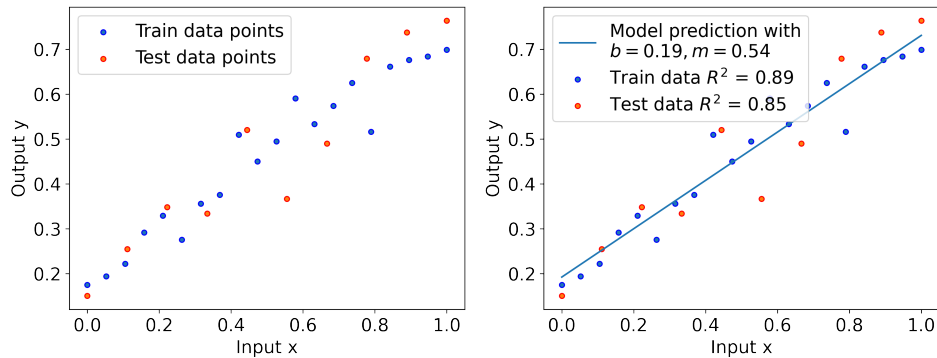


Figure 2.10: Scatter plot of exemplary data points; a: Separated into train and test data; b: Linear regression fit with R^2 -score.

by m, b that fits or comes as close as possible to the input data points. This is generally equivalent to the minimization problem: find m, b such that the difference between the model output and the output data, (take the mean squared error (MSE) as an example norm for the difference between these quantities):

$$\min_{m,b} L = \frac{1}{N} \sum_i^N (\hat{y}_i - y_i)^2 \quad (2.9)$$

with $\hat{y}_i = f(x_i)$. The equation 2.9 is also referred to the loss function in previous context. To minimize the loss function, a gradient descent method can be applied. The gradient

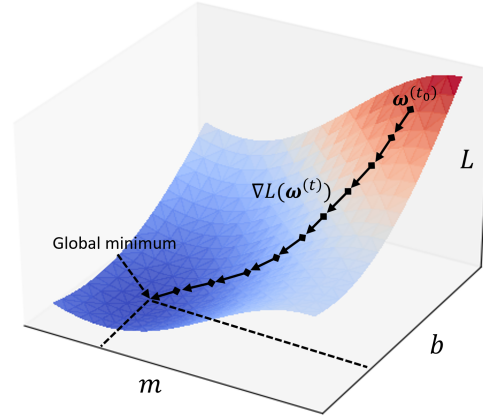


Figure 2.11: Illustrative example of the 2D gradient descent for the linear regression problem, where the model parameters m, b are the search space and the loss function to be minimized. $\nabla L(\mathbf{w}^{(t)})$ denotes the search direction and the learning rate γ the length of the search vector.

descent method is an iterative optimization algorithm that finds a local minimum of a continuous and differentiable function. The brief idea is to start at a random point and use the gradient of the loss function as the direction to descend toward the local minimum by taking a small step each iteration. The loss function fulfills the above condition in the current linear regression setting. It has to be minimized with respect to the parameter vector $\mathbf{w} = [m, b]^T$, and the computation rule is denoted as:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \gamma \nabla L(\mathbf{w}^{(t)}), \quad (2.10)$$

with γ being the learning rate, the subscript t for the iteration step, $-\nabla L(\mathbf{w})$ the direction in which L decrease with the parameter vector \mathbf{w} , tol the error tolerance, n the number of iterations. An outline of the algorithm can be found in 1. Once the defined stopping criteria are met, the error tolerance is reached and the local minimum is returned. The trained model with the optimal parameter set \mathbf{w} can be returned and used to predict unseen data.

To evaluate the model's performance in the present linear regression setting, the R^2 score can be used. The R^2 is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.11)$$

Algorithm 1 Gradient descent

Require: : $x_i, y_i, \mathbf{w}^{(t_0)}, \gamma, tol, n$
for $t = 1$ to n **do**
 Determine $\nabla L(\mathbf{w}^{(t)})$
 Update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \gamma \nabla L(\mathbf{w}^{(t)})$
 Evaluate $L(\mathbf{w}^{(t+1)})$
 if $e(\mathbf{w}^{(t+1)}) > tol$ **then**
 Set $t = t+1$
 else
 Return $\mathbf{w}^{(t+1)}$ as the optimal parameters
 end if
end for

with y_i being the ground truth data value, and \hat{y}_i being the model predicted value, \bar{y} the mean value of all output data. It is also referred to as *coefficient of determination*, which describes the proportion of the variation in the dependent variable explained by the independent variables. The R^2 score can be used for training and test datasets to evaluate the model performance. This is evaluated for the toy dataset in Fig. 2.10b. Further discussion of model performance metrics can be found later in model evaluation section . Note that the present setting can be easily extended to multidimensional by considering y_i as a vector in the form of:

$$\hat{\mathbf{y}}_i = \mathbf{W}^T \mathbf{x}_i, \quad (2.12)$$

with subscript i being the sample index and \mathbf{W}^T the weight matrix containing the linear coefficients.

Example: K-means clustering As a typical example of unsupervised learning, one considers the problem of clustering data into groups. Exemplary data can be material properties, such as strengths and failure strains of the material. The task can be to group the data points into different types of materials. In the supervised case, there may be two classes of material given, classified into a brittle and a ductile class. In the unsupervised case, one can arbitrarily choose the number of clusters or classes of materials to see if there are possible subpopulations within the data. The K -means algorithm groups the data by partitioning samples \mathbf{x}_i into K clusters of equal variance,

minimizing a function denoted as the inertia [64]:

$$\operatorname{argmin}_K \sum_i^N (\mu_j - x_i)^2 \quad (2.13)$$

with N denotes the number of the samples, μ_j the mean of the samples in each cluster, and the subscript $j = 1, 2, \dots, K$ with $K \leq N$, respectively. Fig. 2.12a shows an example of synthetic data points, where x_1 can be considered as some numbers of failure strain and x_2 as ultimate strength for given material samples. Fig. 2.12b-d shows the model prediction with $K = 2, 3, 4$. Here, 0 and 1 are the cluster labels representing exemplary brittle and ductile material classes, while cluster labels 2 and 3 denote possible further subclasses within the given dataset.

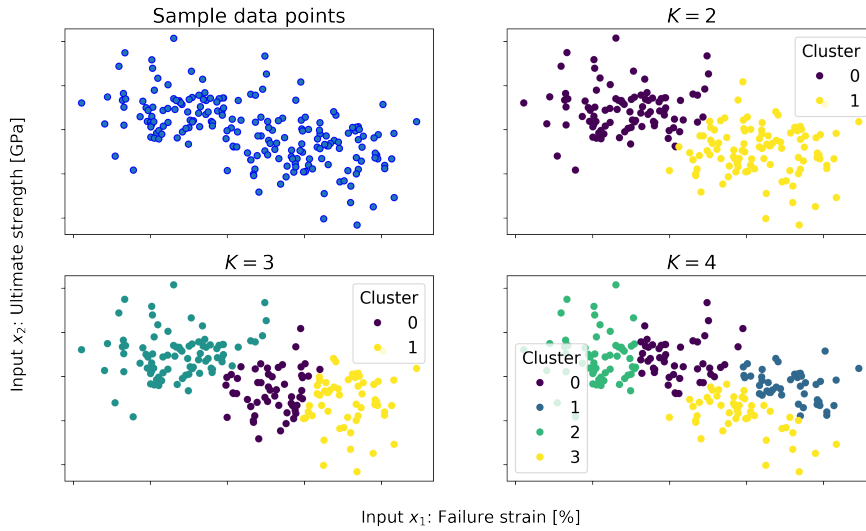


Figure 2.12: K-means clustering for a synthetic dataset with failure strain and ultimate strength as exemplary input parameters. **a**: Sample data points; **b-d** show the model prediction with $K = 2, 3, 4$ while 0, 1 are the cluster labels representing the global brittle and ductile material class, respectively. Cluster labels 2, 3 denote possible sub-classes.

Conventional (shallow) machine learning models

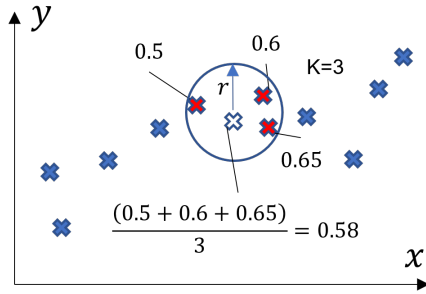
In general, models are distinguished in their model formulation by whether a predefined number of parameters is specified. If so, they are generally called *parametric* models. On the other hand, if the model does not have a fixed number of parameters and the number increases with the amount of data, they are called *nonparametric* models [53]. In Chapter 4, some classical machine learning models are used for MPR. Therefore, their model equations are presented in this section. Since all case studies in this thesis are of the regression type, the focus is on the use of the presented models for this type of problem. Without loss of generality, the models can also be applied to classification problems. First, the lasso model is introduced as a regularized linear model representing the class of linear models. Second, K-Nearest Neighbors (KNN), Decision Trees (DTs), representing the class of nonparametric models, and Gradient Boosting (GB), representing the class of ensemble models, are introduced. These models have been implemented in the open source machine learning library scikit-learn [64].

Lasso In essence, the Lasso model is a linear model with an added regularization term. The optimization problem and the loss function can be written as:

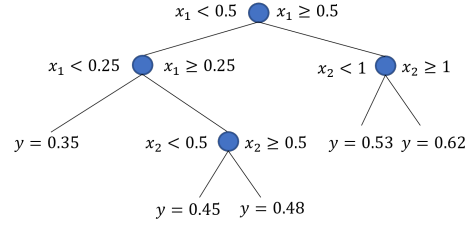
$$\operatorname{argmin}_{\mathbf{W}^T} L := \frac{1}{2N} \sum_i^N (\mathbf{W}^T \mathbf{x}_i - \mathbf{y}_i)^2 + \alpha \|\mathbf{W}^T\|_1 \quad (2.14)$$

with $\hat{\mathbf{y}}_i = \mathbf{W}^T \mathbf{x}_i$ as the linear model, \mathbf{W}^T the coefficient matrix, α a constant, $\|\cdot\|_1$ the L_1 norm of the coefficient matrix, where the last term is also referred to as the penalty or regularization term. Adding regularization as a constraint to the minimization process usually increases convergence and improves solution finding. There are other possible regularization terms, such as using the L_2 norm of the weight matrix [64]. The optimization process can follow the same gradient descent steps as shown earlier in this chapter.

K-Nearest Neighbours The KNN model can be used for both classification and regression tasks, depending on whether the data is discrete or continuous. For classification, the class prediction is computed from a simple majority vote of the nearest neighbors of each point in the training set, i.e., a query point is assigned to the data class that has the most representatives among the nearest neighbors of the point [64], depending on the *weighting* or *distance* function. For regression, the target property is predicted by local interpolation of the target values associated with the nearest neighbors in the training



(a) Visualization of KNN algorithm for regression type of problem: Exemplary dataset for $K=3$ and using uniform weighting to predict the unknown value.



(b) Visualization of DT-model for a synthetic dataset. The tree predicts the response based on x_1 and x_2 . Starting at the top node, the values of the features are compared to the thresholds to decide which branch to be followed. When the branches reach a leaf node, the response is set to the corresponding value.

Figure 2.13: Examples of shallow learning models

set. The user must specify K as an integer value, i.e., how many neighbors to select. Several weighting functions can be used for prediction. The most common are uniform weights, where all points in each neighborhood are weighted equally, see Fig. 2.13a. The second is to weight the points by the inverse of their distance. In this case, closer neighbors of a query point will have a greater influence than neighbors that are farther away. The simplest algorithm for finding neighbors is brute-force computation of the distances between all pairs of points in the data set. The model becomes inefficient as the sample size increases, so fast implementations other than brute force are needed.

Decision trees DT-based models are a nonparametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules derived from data features. Mathematically, given the training vectors $\mathbf{x}_i \in \mathbb{R}^M$, and the label $y_i \in \mathbb{R}$, a decision tree recursively splits the features s.t. the samples with the similar target values are grouped together. As shown in Fig. 2.13b, the data at the top node can be denoted by Q_m with N_m samples. For each split $\theta(j, t_m)$ of the node with feature j , and threshold t_m , the data is divided into $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$ subsets:

$$Q_m^{left}(\theta) = \{(\mathbf{x}_i, \mathbf{y}_i) | x_j < t_m\}, \quad Q_m^{right}(\theta) = \{(\mathbf{x}_i, \mathbf{y}_i) | x_j \geq t_m\}. \quad (2.15)$$

The quality of a candidate split of node is computed by minimizing the loss function L at both ends to obtain an optimal set of θ^* :

$$\theta^* = \operatorname{argmin}_{\theta} G(Q_m, \theta) := \frac{N_m^{left}}{N_m} L(Q_m^{left}) + \frac{N_m^{right}}{N_m} L(Q_m^{right}), \quad (2.16)$$

by splitting the nodes until maximum depth (specified by user) is reached. L is defined:

$$L(Q_m) = \frac{1}{N_m} \sum_{y \in Q_m} (\bar{y}_m - y_i)^2 \quad (2.17)$$

with $\bar{y}_m = \frac{1}{N_m} \sum_{\hat{y}_i \in Q_m} \hat{y}_i$, where \bar{y}_m is the mean value of prediction values at leaf nodes \hat{y}_i . A tree can be thought of as a piecewise constant approximation. The deeper the tree, the more complex the decision rules and the better and smoother the prediction model. DTs are easy to interpret, their decision logic can be visualized. However, as the tree gets deeper, the model tends to overfit the data. Predictions from decision trees are not smooth and continuous, resulting in poorer performance when interpolating and extrapolating predictions [64].

Gradient Boosting The GB model is an ensemble method with the goal of combining the predictions of multiple base estimators to improve generalizability and robustness over a single base estimator [64]. The base estimator can be any shallow model, such as a DTs-based model. In the ensemble method, two types of ensembles are available. The first is the *averaging method*, which is to build several estimators independently and take their average predictions as the final output prediction, with the goal of reducing the variance of the model compared to a single estimator. The second, the boosting method, builds on basic estimators that are applied sequentially, with the goal of reducing the bias of the combined estimator. GB belongs to the second class of boosting methods and is briefly introduced below. It builds an additive model of the form [64]:

$$\hat{y}_i = F_M(\mathbf{x}_i) = \sum_{m=1}^M h_m(\mathbf{x}_i) \quad (2.18)$$

where h_m are the DTs as base estimators of fixed size. M corresponds to the number of base estimators used in the method and is a hyperparameter specified by the user. The GBRT follows a form of greedy function:

$$F_M(\mathbf{x}_i) = F_{M-1}(\mathbf{x}_i) + h_m(\mathbf{x}_i) \quad (2.19)$$

where a newly added tree h_M is fitted to minimize a sum of losses L_m , given the previous ensemble function $F_{M-1}(\mathbf{x}_i)$:

$$h_{m+1} = \operatorname{argmin}_h L_m := \operatorname{argmin}_h \sum_i^n L(F_{M-1}(\mathbf{x}_i) + h_m(\mathbf{x}_i), \mathbf{y}_i), \quad (2.20)$$

where L is the loss, e.g. least squares for the regression problem, or a log-loss for the classification problem. The minimization scheme can again follow the gradient descent method, more details can be found in [64].

Deep learning models

Neural network, or deep learning, models have been considered a universal tool for approximating nonlinear, multivariate models, suitable for both regression and classification problems. Advanced supercomputing technology and computations using graphics cards have enabled these models to scale up to extremely large problems, which are common in most computer vision tasks. Various state-of-the-art neural network models have been applied in the current work, such as U-net and R-CNN, which also uses the concept of convolutional neural networks (CNN). Their basic network structure and working principles are therefore briefly introduced.

Fully-connected feed-forward neural networks A Fully-Connected Feedforward Neural Network (FCFFN), or Multilayer Perceptron (MLP), is a neural network that learns a generally nonlinear mapping $\mathcal{N} : \mathbb{R}^I \rightarrow \mathbb{R}^O$, where I, O are the input and output dimensions, respectively. The network consists of an input layer, H numbers of the hidden layer, and an output layer, which has the following nested form:

$$\hat{\mathbf{y}}_i = \mathcal{N}(\mathbf{x}_i) = \phi^O \left(\phi^{O-1} \left(\dots \left(\phi^H \left(\dots \left(\phi^I \right) \right) \right) \right) \right) = \phi^O \circ \phi^{O-1} \circ \dots \circ \phi^H \dots \circ \phi^I, \quad (2.21)$$

for $H = O, O - 1, \dots, I$, with the first layer $\phi^I = \mathbf{x}_i$ and, last layer $\phi^O = \hat{\mathbf{y}}_i$ as the in- and outputs, respectively. Their values are stored in the so-called *neural units* or *neurons*. Consequently, the number of neurons corresponds to the dimension of the in- and output dimensions, see Fig. 2.14. ϕ^H is a hidden layer with their layer units:

$$\phi_k^H = a^H(\mathbf{W}_k^H \phi^{H-1}) = z_k \quad (2.22)$$

where ϕ_k^H is the k -th neuron of the hidden layer ϕ^H . k is the number of the neurons in the hidden layer, a hyper-parameter as same as the number of hidden layers H that

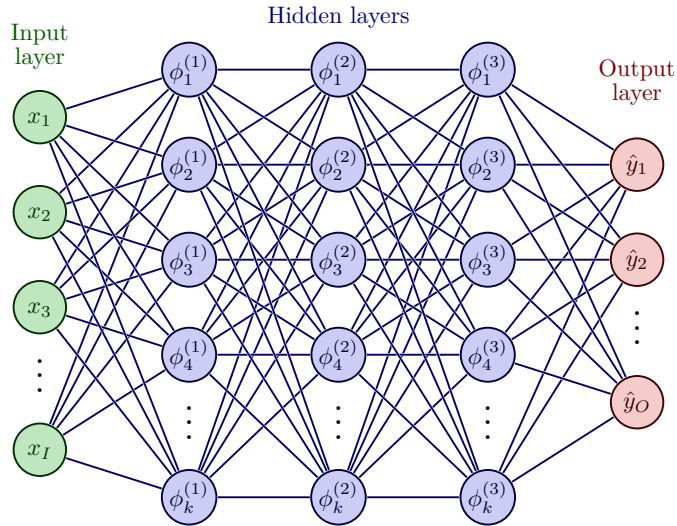


Figure 2.14: Network structure of a multilayer perceptron model, generated with TikZ package (CC BY 4.0).

need to be specified by the user. \mathbf{W}_k is the weight vector parameters for the k -th neuron. $a^H : \mathbb{R} \rightarrow \mathbb{R}$ is the so-called activation function that introduced non-linearity to the neural networks, see Fig. 2.15. The most common activation functions are summarized in Table 2.1. Depending on the task at hand, different activation functions can be used, see [41] for a more comprehensive list of activation functions and their corresponding usage.

Convolutional neural networks CNNs are neural networks commonly used in image processing applications. A CNN does not differ in architecture from an FCNN, which also consists of an input layer, hidden layers, and an output layer. However, in CNN, the hidden layers include layers that perform so-called convolutions. Convolution is an operation that performs a dot product of the convolution kernel with the layer's input matrix, typically an input image. A convolution kernel can be thought of as an image processing filter that modifies the input image matrix. Depending on the kernel, different spatial features can be captured. The convolution operation creates multiple feature maps, which are modified representations of the input matrix for the algorithm to learn. According to [67], given the input x^h in the h -layer of the third order tensor with a size

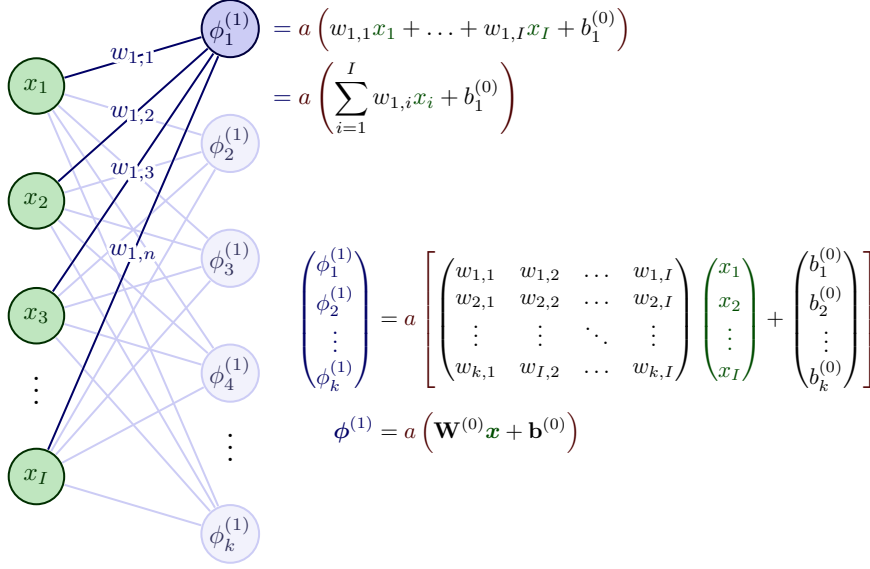


Figure 2.15: Forward computation of the first layer in the MLP model.

of $H^h \times W^h \times D^h$, with H^h, W^h, D^h being the height, width, and depth (number of channels), respectively. The convolution kernel K of size $H \times W \times D^h$ is applied by sliding along the input, from left to right, top to bottom, and performing the dot product with the input, summing up the output along channels of size D^h , see the exemplary convolution operation for a sample location in Fig. 2.17a. If M convolution kernels are used, then $K \in \mathbb{R}^{H \times W \times D^h \times M}$. The convolution result $x^{h+1} \in \mathbb{R}^{H^{h+1} \times W^{h+1} \times M}$, with $H^{h+1} = H^h - H + 1, W^{h+1} = W^h - W + 1$ with:

$$x_{i^{h+1}, j^{h+1}, m}^{h+1} = \sum_{i=0}^H \sum_{j=0}^W \sum_{d^h=0}^{D^h} K_{i,j,d^h,m} \cdot x_{i^{h+1}+i, j^{h+1}+j, d^h}^h + b_{i^{h+1}, j^{h+1}, m}. \quad (2.23)$$

The above step is carried out for all convolution kernels $0 \leq m \leq M$ and all spatial locations $0 \leq i^{h+1} \leq H^{h+1}, 0 \leq j^{h+1} \leq W^{h+1}$. $x_{i^{h+1}+i, j^{h+1}+j, d^h}^h$ refers to the element of x^h of the index triple $(i^{h+1} + i, j^{h+1} + j, d^h)$. $b_{i^{h+1}, j^{h+1}, m}$ is the bias term that is added to the output of the next layer. The feature maps typically contain different levels of detail, such as shapes and edges. Notably, the convolution operation captures features in a spatially localized field, thus reducing model complexity in terms of neuron

Name	Definition
Sigmoid	$a(z) = (1 + \exp(-z))^{-1}$
Tanh	$a(z) = \tanh(z)$
Softplus	$a(z) = \log_e(1 + \exp(z))$
ReLU	$a(z) = \begin{cases} 0, & \text{for } z < 0 \\ z, & \text{for } z > 0 \end{cases}$
Leaky ReLU	$a(z) = \begin{cases} 0.01z, & \text{for } z < 0 \\ z, & \text{for } z > 0 \end{cases}$
ELU	$a(z) = \begin{cases} z, & \text{for } z < 0 \\ \alpha \exp(z) - 1, & \text{for } z > 0 \end{cases}$
Softmax	$a(z) = \frac{\exp(z_j)}{\sum_l^k \exp(z_l)}$

Table 2.1: Some common activation functions [41]

requirements compared to FCFNN. Furthermore, the feature maps then contribute to the input of the next layer, usually a pooling layer. The goal of the pooling layer is to extract significant features from the feature map and reduce the size of the feature map to significantly reduce the computational cost. when the Max Pooling is applied to the previous output $\mathbf{x}^{h+1} \in \mathbb{R}^{H^{h+1} \times W^{h+1} \times M}$, it takes the largest number of the convoluted feature map given a certain pooling matrix size of $(H^p \times W^p)$, that is:

$$x_{i^{h+2}, j^{h+2}, m}^{h+2} = \max_{(0 \leq i \leq H^p, 0 \leq j \leq W^p)} \{x_{i^{h+1}, j^{h+1}, m}^{h+1}\} \quad (2.24)$$

and leads to $\mathbf{x}^{h+2} \in \mathbb{R}^{H^{h+2} \times W^{h+2} \times M}$, with $H^{L+2} = H^{L+1}/H^p, W^{L+2} = W^{L+1}/W^p$. Other pooling strategies such as Average Pooling and Sum Pooling are available [67]. Furthermore, the Fully Connected (FC) layer is placed before the output layer. The FC layer takes the inputs from the previous layer as a flattened vector and fully connects the model weights before passing them to the output layer. The FC layer, usually several in a row, is the same as hidden layers as in the case of the FCFNN model. Finally, depending on the nature of the actual problem, activation functions are used to activate the last FC layer to obtain the final output.

Other advanced neural networks As motivated earlier, many state-of-the-art deep learning models from computer science have been adapted, developed, and applied in re-

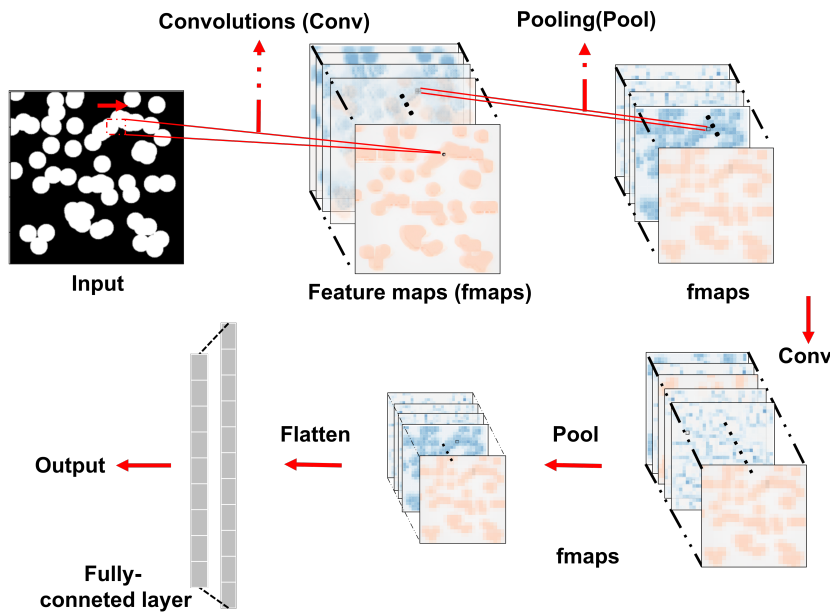


Figure 2.16: A common CNN-Structure.

sponse to the new research interest in the materials science and engineering community. For example, Recurrent Neural Networks (RNNs), originally used for language modeling and speech recognition, have been applied to spatiotemporal microstructure evolution prediction [69]. Generative Adversarial Networks (GANs) are another kind of neural networks designed to study a collection of training examples and learn the probability distribution that generated them [44] and have found application in microstructure generation [70], and design [71]. Now, due to the rise of ChatGPT [72], a chat robot based on transformer models, more and more research work is being done to explore the possible potential of transformer models in similar field of materials research [73, 74, 75]. These new models represent current, ongoing research efforts, and the reader is referred to the appropriate literature.

Model evaluation and performance

As introduced earlier for the linear regression example, different metrics can be used to evaluate the performance of the model, depending on the task at hand. Some commonly

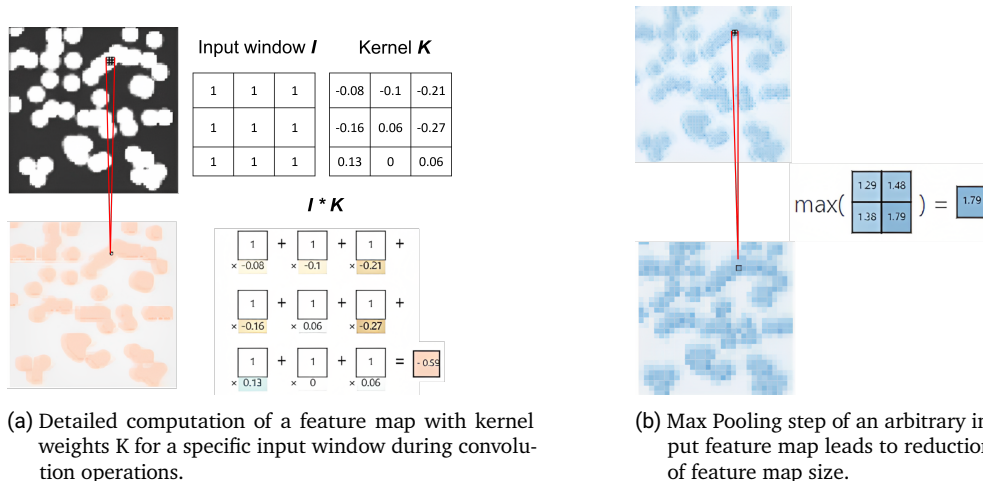


Figure 2.17: Exemplary computation of convolution and max pooling operation, visualized with CNN Explainer [68].

used regression metrics are listed in Table 2.2. More specific metrics, such as those for the particle segmentation task in chapter 3, will be introduced in the appropriate sections.

Furthermore, regardless of the task to be solved, a common problem is that while the trained model may be the best fit to the training data, it may have low generalization power to unseen data, see Fig. 2.19 for an illustrative example. This is due to the problem in the model selection step that the chosen model may be too simple or too complicated, thus *underfitting* or *overfitting* the data. Overfitting is a problem that generally indicates good performance on the training data, but poor generalization to other or test data. Underfitting indicates poor performance on the training data and poor generalization to other data sets. Both problems imply lower generalization power of the selected model and are undesirable effects. Monitoring training and test loss functions is essential in practice to detect and counteract such a problem. The main techniques for solving common overfitting problems can be briefly divided into two perspectives. From a data perspective, the first step may be to sample a larger and more representative amount of dataset. However, this can lead to high costs in sampling new data sets. Typically, sampling of experimental data, especially in the context of materials science, is extremely expensive. The costly sampling step can be mitigated by the so-called *data augmentation* technique. This technique helps to increase the data set

Name	Definition
Mean absolute error	$MAE = \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $
Mean squared error	$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
Mean abs. percentage error	$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{ y_i - \hat{y}_i }{ y_i }$
Max error	$MA = \max y_i - \hat{y}_i $
Coefficient of determination	$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$

Table 2.2: Some common regression metrics [64]; y_i : Ground truth value; \hat{y}_i : Model predicted value, n : Number of samples; \bar{y} : Mean of the ground truth values

by including synthetic samples in the training, which was one of the techniques used in chapter 3 of this thesis. In addition, the feature data may be too complex to train on. Applying dimensionality reduction techniques to feature data helps to reduce the complexity of the data. This technique is part of unsupervised learning and was used in Chapter 4 to reduce correlated microstructural features for the fiber network in paper materials. From the model perspective, many *model regularization* techniques generally help to penalize the model complexity by introducing a regularization term in the loss function, as demonstrated for the lasso model, thus alleviating overfitting in the model training process. On the other hand, it is relatively easy to solve an underfitting problem in daily practice. The most effective method is to increase the complexity of the model. As shown in Fig. 2.19, increasing the complexity of the model effectively reduces the bias in the test dataset, but it is observed that a too complex model tends to overfit the dataset, thus reducing the generalization power of the model for the unseen dataset. Another practical approach is called K-fold cross-validation. This technique helps to identify the overfitting problem and checks the robustness of the model. As shown in Fig. 2.19, the incoming data is divided into a training and a test part. The training data set is then further divided into K-fold sub-sets, with one of the K-sets being used to validate the model. This is done for k iterations, where the data set is permuted each time. These techniques can be used together for hyperparameter tuning [64], where the fine-tuned model can be applied to the hold-out test data for a final generalization performance check.

In conclusion, while an acceptable solution can be found quickly for the current regression setting with a simplified toy dataset, it must be noted that the problem and dataset are complex in nature with multi-dimensional inputs and outputs. From a computational point of view, the user often needs to set the maximum number of

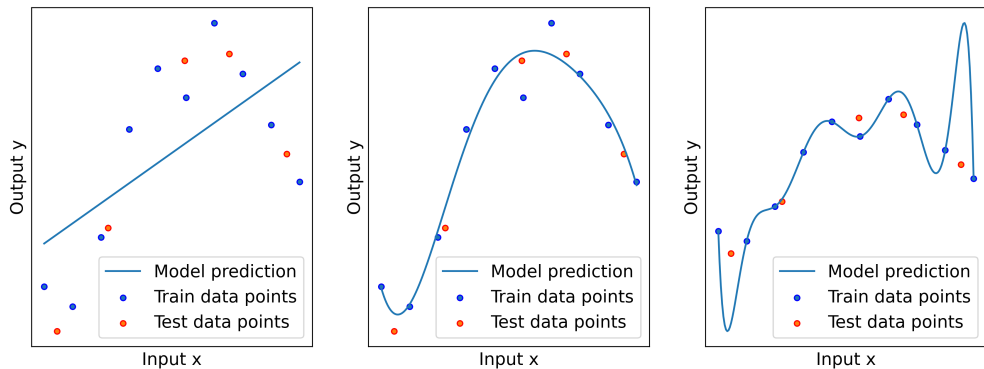


Figure 2.18: Exemplary data points fitted with different degree of polynomial basis functions; **Left:** An underfitted dataset showing high bias between train and test data points; The bias is observable for the distance between the train, test data points and prediction curve; **Middle:** A good fit between the prediction curve and the train and test data; **Right:** The prediction curve overfits the train data, high variance can be observed for the test data.

iterations and an error tolerance for the given optimization problem. The number of iterations is often reached before a minimum can be found. From a more mathematical point of view, the nature of the problem may not guarantee the uniqueness or even the existence of a global minimum. Thus, finding a local minimum is a more common practice. While many ML models can be used, it depends heavily on the objective, data, computational resources, and many other factors. As the No Free Lunch theorem states, there is generally no single straightforward ML model or optimization algorithm for any given predictive modeling problem. For more practical applications of ML, see e.g. [76].

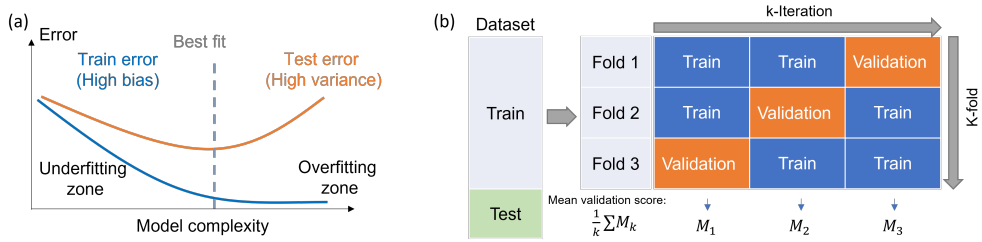


Figure 2.19: Model evaluation schemes. **Left:** Illustrative example of the trade-off between underfitting and overfitting; while the training error decreases with increasing model complexity, the test error gradually increases, implying typical overfitting behavior; **Right:** Exemplary K-fold cross-validation scheme for K=3. The input data is first divided into training and test data. For cross-validation, the training data is again divided into k-fold data sets for training and validation. This process is repeated in k iterations to obtain a mean cross-validated performance score. The validated model is then applied to the test dataset to verify the generative power of the model.

2.3 Continuum Mechanics and Numerical Approximation

In the following, the basic principle of continuum mechanics and the FE approximation scheme are introduced, as they form the basis of the simulation techniques used in the main chapters.

2.3.1 Kinematics

Configurations in Lagrangian and Eulerian framework

One considers a continuous deformable body that occupies a region in the three-dimensional Euclidean space \mathbb{R}^3 . The subsets of \mathbb{R}^3 occupied by a body \mathcal{B} are called its configurations such as a mapping $\varphi : \mathcal{B} \rightarrow \mathbb{R}^3$.

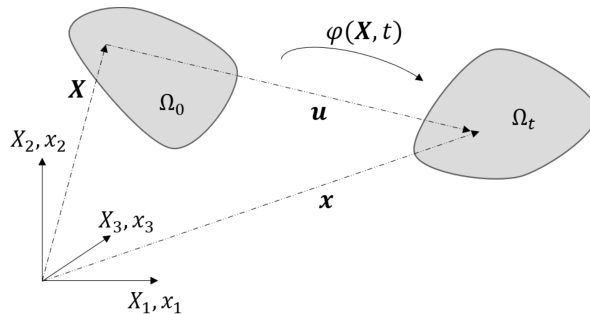


Figure 2.20: Description of the motion of a body from its reference configuration Ω_0 to current configuration Ω_t .

If one picks the point 0 in \mathbb{R}^3 as the origin of a fixed coordinate system and when the physical body \mathcal{B} occupies at time $t = 0$ (not always necessary), one can define $\varphi(\mathcal{B}) = \Omega_0$ as its *reference* configuration. A material point of the deformable body X consequently occupies a position, which can be characterized correspondingly by the vector $\mathbf{X} = X_i \mathbf{E}_i$, where \mathbf{E}_i are the orthonormal base vectors for $i = 1, 2, 3$ with respect to the reference configuration. The coordinates (X_1, X_2, X_3) of \mathbf{X} consequently identify the material point in the Euclidean space. The coordinate labels X_i are called *material* coordinates, see Fig. 2.20. Furthermore, when the body moves through \mathbb{R}^3 over a period of time and occupies a *current* configuration $\varphi(\mathcal{B}, t) = \Omega_t \subset \mathbb{R}^3$ at time t , the material points X in

$\bar{\Omega}_0$ (the closure of Ω_0) are mapped into positions \mathbf{x} in $\bar{\Omega}_t$:

$$\mathbf{x} = \varphi(\mathbf{X}, t), \quad (2.25)$$

where φ is defined as the motion of the body and is an injective mapping. The system (x_1, x_2, x_3) is called the *spatial* coordinates system with the vector $\mathbf{x} = x_i \mathbf{e}_i$, where \mathbf{e}_i are orthonormal vectors with respect to the spatial coordinates. The description of the motion in these configurations allows different perspective. The interpretation is that in the material description, one would see through the lens of the material point or particle, what happens to it as it moves. This is usually used in the *Lagrangian* framework to describe solid mechanic problems, whereas the *Eulerian* framework is preferred for fluid problems, referring to spatial coordinates and studying what happens at the point in the spatial space as time changes.

Displacement vector and strain tensors

The displacement vector field is defined as the distance between the reference and current configuration of the material point at \mathbf{X} with:

$$\mathbf{u} = \varphi(\mathbf{X}, t) - \mathbf{X} = \mathbf{x} - \mathbf{X}. \quad (2.26)$$

Further the *displacement gradient* by derivation with respect to material coordinates is introduced.

$$\mathbf{H} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} - \frac{\partial \mathbf{X}}{\partial \mathbf{X}} = \mathbf{F} - \mathbf{I}, \quad (2.27)$$

where $\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$ is the *deformation gradient* and \mathbf{I} the identity tensor. With \mathbf{F} , the infinitesimal elements between the configurations can be mapped:

$$d\mathbf{x} = \mathbf{F}d\mathbf{X}, \quad d\mathbf{a} = J\mathbf{F}^{-T}d\mathbf{A}, \quad d\mathbf{v} = Jd\mathbf{V} \quad (2.28)$$

with $J = \det(\mathbf{F})$. Another resenatation of \mathbf{F} can be the multiplicative split into a rotation and a streching part:

$$\mathbf{F} = \mathbf{R}\mathbf{U} = \mathbf{V}\mathbf{R}, \quad (2.29)$$

with \mathbf{R} as the pure rotation tensor, \mathbf{U} the right stretch tensor and \mathbf{V} the left stretch tensor. With the split of \mathbf{F} , the right *Cauchy-Green* tensor can be defined as:

$$\mathbf{C} = \mathbf{U}^2 = \mathbf{U}^T\mathbf{U} = \mathbf{F}^T\mathbf{F}. \quad (2.30)$$

Further, the nonlinear *Green-Lagrangian* strain tensor is introduced:

$$\mathbf{E} = \frac{1}{2} (\mathbf{C} - \mathbf{I}) = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}), \quad (2.31)$$

which is usually used as the strain measure for within the theory of large deformation. Combining 2.27 and 2.31 one can write:

$$\mathbf{E} = \frac{1}{2} (\mathbf{H} + \mathbf{H}^T + \mathbf{H}^T \mathbf{H}). \quad (2.32)$$

The components are given as:

$$E_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_{ij}}{\partial X_i} + \frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right). \quad (2.33)$$

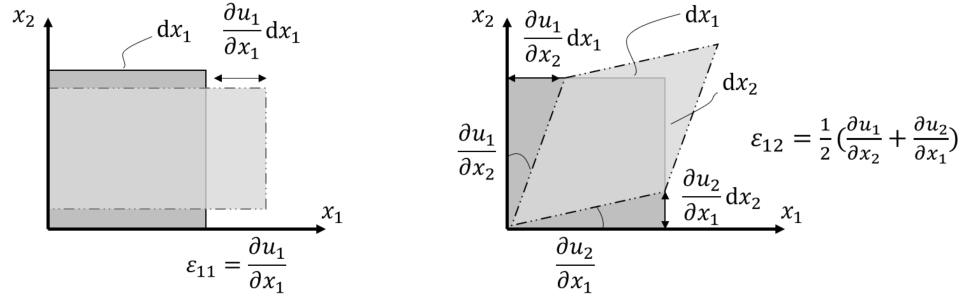


Figure 2.21: Strain components at an infinitesimal cuboid in 2D perspective. Recreated from [77].

Linearization of \mathbf{E} by assuming small strains $\frac{\partial u_k}{\partial X_i} \ll 1$ leads to the so-called *engineering strain tensor*:

$$\varepsilon_{ij} = E_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} \right). \quad (2.34)$$

Note that one does not distinguish between material frame and spatial frame, since the deformation is assumed to be small. The engineering strain tensor can be also written in the matrix form:

$$\varepsilon_{ij} = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{pmatrix}. \quad (2.35)$$

In this work, the linear elastic framework of small strains is assumed for all investigated problems. Damage or possible failure of the material will be discussed in the context of the problems, where necessary. To visualize the strain components ε_{ij} , one considers the example in 2D, see Fig. 2.21

$$\frac{\partial u_1}{\partial x_1} = \varepsilon_{11}, \quad \varepsilon_{12} = \frac{1}{2} \left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \right) = \varepsilon_{21}.$$

2.3.2 Concept of stress

Motion and deformation result in interactions between the material particle point and its neighboring particle point in the body. The result of the interaction is stress, which has the physical dimension force per unit area as $[\frac{N}{mm^2}]$. One can interpret that the stress is responsible for the deformation of materials.

Traction vector

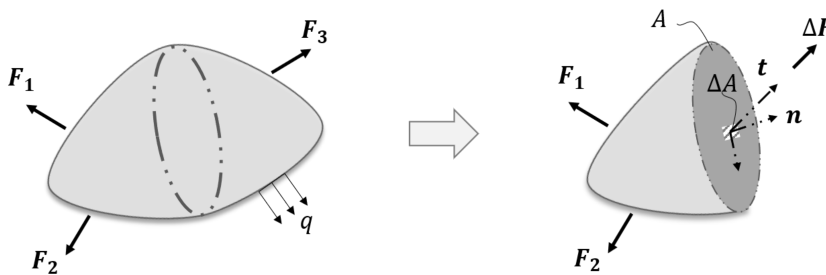


Figure 2.22: Definition of a traction vector for a deformable body under arbitrary load at the interior surface.

Suppose the body is arbitrarily loaded by any surface loads q or body loads F_i or any combination of both, denoted as external forces. This an (imaginary) surface, the

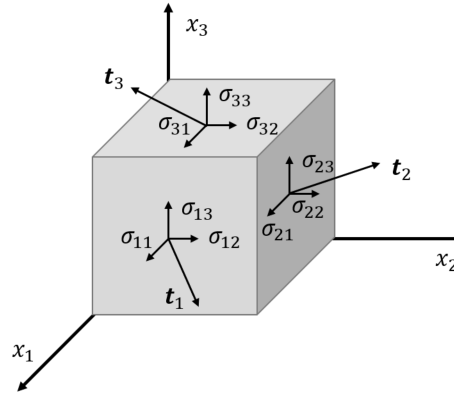


Figure 2.23: Stress components at an infinitesimal cuboid. Recreated from [78].

so-called reaction force inside the body. Thus, the limit value can be defined as the quotient of the reaction force to its surface element as the *traction vector*, which depends on the actual point P given by the external normal vector on the surface, see normal vector on the surface in Fig. 2.22:

$$\mathbf{t}(P, \mathbf{n}) = \lim_{\Delta A \rightarrow 0} \frac{\Delta \mathbf{F}}{\Delta A} = \frac{d\mathbf{F}}{dA}. \quad (2.36)$$

Cauchy's stress theorem and stress tensor

To uniquely compute the stress state for some given loads one uses the orthonormal set $\{e_i\}$ of basis vectors, and by projection on these vectors the stress state at a given point can be uniquely identified. The traction vectors on the surfaces of an infinitesimal cuboid are then given as, see Fig. 2.23:

$$\mathbf{t}_i = \begin{pmatrix} \sigma_{i1} \\ \sigma_{i2} \\ \sigma_{i3} \end{pmatrix} = \sigma_{ij} n_j, \quad (2.37)$$

where the components are used to construct the so-called *Cauchy stress tensor* σ_{ij} :

$$\boldsymbol{\sigma} = \sigma_{ij} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix}. \quad (2.38)$$

The components in the diagonal characterize the *normal* stresses and the off diagonal components the *shear* stresses, respectively. $t_i = \sigma_{ij}n_j$ is the so called *Cauchy formula*. Note that the Cauchy stress tensor is formulated in spatial coordinates and the equality holds

$$\boldsymbol{\sigma}(\mathbf{x}, t) \mathbf{n} \, d\mathbf{a} = \mathbf{P}(\mathbf{X}, t) N \, d\mathbf{A}, \quad (2.39)$$

where \mathbf{P} denotes the *First Piola-Kirchhoff* tensor or nominal stress tensor in the material frame and the following transformation holds:

$$\mathbf{P} = \mathbf{J} \boldsymbol{\sigma} \mathbf{F}^{-T}. \quad (2.40)$$

Throughout this thesis, linearized theory assuming small strain is applied, thus no differentiation with respect to the configurations is made.

2.3.3 Constitutive equation

The constitutive equations or material laws are generally used to describe the relationship between stresses and strains. This must be determined separately for the material of interest, apart from the mechanical balance equations. In this section, only linear elasticity is introduced. More material-specific constitutive models will be introduced later in the corresponding section. In elasticity, the stress tensor can be described by a unique mapping:

$$\boldsymbol{\sigma} = (\mathbf{F}), \quad (2.41)$$

this means that $\boldsymbol{\sigma}$ is time-independent, loading-history-independent and uniquely characterized by the deformation gradient \mathbf{F} . The unloading of the material causes no remaining deformation. For the small deformation theory it applies:

$$\boldsymbol{\sigma} = (\boldsymbol{\varepsilon}). \quad (2.42)$$

In general, this mapping is determined experimentally. The elasticity of the material can be classified as *homogeneous* or *inhomogeneous*. If the elastic material is not direction dependent, it is called *isotropic*, otherwise *anisotropic*. For the most general three-dimensional case, the above mapping can be described by *Hooke's law*:

$$\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon} \quad \text{or} \quad \sigma_{ij} = \mathbb{C}_{ijkl}\varepsilon_{kl}, \quad (2.43)$$

where \mathbb{C}_{ijkl} is the fourth order elastic stiffness tensor. In general, the number of elastic constants is $3^4 = 81$, and can be reduced by a number of symmetry conditions. From the symmetry of the stress and strain tensors $\sigma_{ij} = \sigma_{ji}$ and $\varepsilon_{ij} = \varepsilon_{ji}$ applies:

$$\mathbb{C}_{ijkl} = \mathbb{C}_{jikl}, \quad \mathbb{C}_{ijkl} = \mathbb{C}_{ijlk}, \quad (2.44)$$

which reduces the number of independent elastic constants to 36. Due to the quadratic form of elastic strain energy [77], it can be shown that:

$$\mathbb{C}_{ijkl} = \mathbb{C}_{klij}, \quad (2.45)$$

which further reduces the number of independent elastic constant to 21. The crystal symmetry of the material further reduces the number of independent constants down to only two for an isotropic material [79].

2.3.4 Momentum balance principle

The linear momentum L by considering the Body \mathcal{B} with set of particles in Ω with boundary $\partial\Omega$ at time t :

$$\mathbf{L}(t) = \int_{\Omega_t} \rho(\mathbf{x}, t) \mathbf{v}(\mathbf{x}, t) d\mathbf{v} = \int_{\Omega_0} \rho_0(\mathbf{X}) \mathbf{V}(\mathbf{X}, t) d\mathbf{V}. \quad (2.46)$$

The balance of linear momentum is

$$\dot{\mathbf{L}}(t) = \frac{D}{Dt} \int_{\Omega_t} \rho \mathbf{v} d\mathbf{v} = \frac{D}{Dt} \int_{\Omega_0} \rho_0 \mathbf{V} d\mathbf{V} = \tilde{\mathbf{F}}(t), \quad (2.47)$$

where

$$\tilde{\mathbf{F}}(t) = \int_{\partial\Omega_t} \mathbf{t} d\mathbf{a} + \int_{\Omega_t} \mathbf{f} d\mathbf{v}. \quad (2.48)$$

Making use of the Cauchy formula and the divergence theorem, it yields:

$$\int_{\partial\Omega_t} \mathbf{t} d\mathbf{a} = \int_{\partial\Omega_t} \boldsymbol{\sigma} \mathbf{n} d\mathbf{a} = \int_{\Omega_t} \text{div}(\boldsymbol{\sigma}) d\mathbf{v}. \quad (2.49)$$

With the equations above, the balance equations in material and spatial coordinates are given as:

$$\int_{\Omega_t} (\operatorname{div}(\boldsymbol{\sigma}) + \mathbf{f} - \rho \ddot{\mathbf{u}}) \, d\Omega_t = 0, \quad (2.50)$$

$$\int_{\Omega_0} (\operatorname{Div}(\mathbf{P}) + \mathbf{f}_R - \rho_0 \ddot{\mathbf{u}}) \, d\Omega_0 = 0, \quad (2.51)$$

where $\ddot{\mathbf{u}} = \dot{\mathbf{v}}$ is the acceleration term and together with ρ the mass density that forms the inertia forces. $(\dot{})$ represents the material time derivative. For reasons of completeness, the material time derivative is introduced in the following. One denotes a scalar, a vector or a tensor field in spatial coordinates as $\Phi = \Phi(\mathbf{x}, t)$. The *material time derivative* is $\frac{D(\dots)}{Dt} = (\dot{})$ for a fixed \mathbf{X} , and that is:

$$\frac{D\Phi(\mathbf{x}, t)}{Dt} = \frac{\partial\Phi(\mathbf{x}, t)}{\partial t} + \frac{\partial\Phi(\mathbf{x}, t)}{\partial\mathbf{x}} \frac{\partial\mathbf{x}}{\partial t}.$$

$\frac{\partial\Phi(\mathbf{x}, t)}{\partial t}$ denotes the time derivative of the corresponding field variable, $\mathbf{v} = \frac{\partial\mathbf{x}}{\partial t}$ the velocity and $\frac{\partial\Phi(\mathbf{x})}{\partial\mathbf{x}}$ the convective rate of change of Φ , respectively. Thus, this can be written as

$$\frac{D(\dots)}{Dt} = \frac{\partial(\dots)}{\partial t} + \operatorname{grad}(\dots) \mathbf{v},$$

with $\operatorname{grad} = \frac{\partial(\dots)}{\partial\mathbf{x}}$ for any field w.r.t the spatial coordinates.

The balance equation 2.50 in integral form is valid for each material point and generally given as partial differential equation (PDE) in local form:

$$\operatorname{div}(\boldsymbol{\sigma}) + \mathbf{f} = 0 \quad \text{or} \quad \sigma_{ij,j} + f_i = 0. \quad (2.52)$$

Note that the theory of small deformation does not distinguish between material and space coordinates, and that inertial forces vanish for static problems with generally two types of boundary conditions. For $\Gamma = \Gamma_u \cup \Gamma_t$, the *Dirichlet* boundary condition defines a prescribed set of displacements at its boundary:

$$u_i = u_i^0 \quad \text{on} \quad \Gamma_u, \quad (2.53)$$

and *Neumann* boundary condition defines a prescribed set of traction at its boundary:

$$t_i = t_i^0 \quad \text{on} \quad \Gamma_t. \quad (2.54)$$

In summary, the governing equations for the linear elasticity problem are given in Table 2.3. More specific equations, such as interface conditions, as in the case of cohesive simulation for the fiber network used in Chapter 4, and elastic-plastic constitutive relations for the simulation of dual-phase microstructure in Chapter 5, are given in the respective sections of the chapter.

Name	Equation (Eq.)	Number of Eq.
Stress equilibrium	$\sigma_{ij,j} + f_i = 0$	3
Kinematics	$\varepsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i})$	6
Constitutive equations	$\sigma_{ij} = \mathbb{C}_{ijkl} \varepsilon_{kl}$	6

Table 2.3: Governing equations for linear elastic problems of small deformation with in total 15 equations for 3 unknowns of u_i , each 6 unknowns for σ_{ij} and ε_{ij} with appropriate boundary conditions depending on the actual problem setting.

2.3.5 Variational (weak) form of governing equation

A variational form is obtained by multiplying the preceding Equation 2.50 by an appropriate arbitrary function, in the present case, a virtual displacement function δu_i , and integrating over the domain:

$$\delta W_{eq} = \int_{\Omega} \delta u_i [\sigma_{ij,j} + f_i] d\Omega = 0, \quad (2.55)$$

where δW_{eq} is defined as the virtual work related to the virtual displacement. The virtual work can be defined as the Galerkin method and is given as:

$$\delta W = G. \quad (2.56)$$

Integrating the stress term by parts leads to:

$$\int_{\Omega} \delta u_i \sigma_{ij,j} = - \int_{\Omega} \delta \varepsilon_{ij} \sigma_{ij} d\Omega + \delta u_i t_i \Big|_{\Gamma}, \quad (2.57)$$

where the virtual strain are denoted as:

$$\delta \varepsilon_{ij} = \frac{1}{2} (\delta u_{i,j} + \delta u_{j,i}). \quad (2.58)$$

For $\Gamma = \Gamma_u \cup \Gamma_t$, imposing the constraint that δu_i vanishes on Γ_u , one obtains the mechanical equilibrium in weak form as:

$$G_{eq} = \int_{\Omega} \delta \varepsilon_{ij} \sigma_{ij} d\Omega - \int_{\Omega} \delta u_i f_i d\Omega - \int_{\Gamma_t} \delta u_i t_i^0 d\Gamma = 0. \quad (2.59)$$

The above equation can be rewritten as:

$$G_{eq} = G_{int} - G_{ext}, \quad (2.60)$$

with

$$G_{int} = \int_{\Omega} \delta \varepsilon_{ij} \sigma_{ij} d\Omega, \quad (2.61)$$

represents the internal force and

$$\int_{\Omega} \delta u_i f_i d\Omega + \int_{\Gamma_t} \delta u_i t_i^0 d\Gamma, \quad (2.62)$$

the external loading of the virtual work, respectively.

2.3.6 Finite element approximation

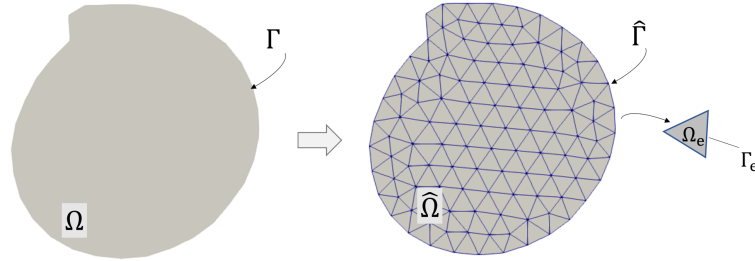


Figure 2.24: Finite element discretization of an exemplary 2D problem domain to subdomains of triangle elements.

In the finite element approximation, the true problem domain Ω is approximated by $\hat{\Omega}$ and divided into a set of elements or subdomains Ω_e , such that:

$$\Omega \approx \hat{\Omega} = \sum_e \Omega_e. \quad (2.63)$$

This subdivision also applies to the domain boundary:

$$\Gamma \approx \hat{\Gamma} = \sum_e \Gamma_e = \sum_{e_t} \Gamma_{e_t} + \sum_{e_u} \Gamma_{e_u}, \quad (2.64)$$

where Γ_{e_t} is a specified traction sub-boundary and Γ_{e_u} a specified displacement sub-boundary, respectively. A discretized form of Equation 2.59 takes the following form:

$$\delta G_{eq} \approx \delta \hat{G}_{eq} = \sum_e \left[\int_{\Omega_e} \delta \varepsilon_{ij} \sigma_{ij} d\Omega_e - \int_{\Omega_e} \delta u_i f_i d\Omega_e \right] - \sum_{e_t} \left[\int_{\Gamma_{e_t}} \delta u_i t_i^0 d\Gamma_{e_t} \right] = 0 \quad (2.65)$$

In addition, the solution vector u_i is approximated by:

$$u_i(x_j) \approx \hat{u}_i(x_j) = \sum_I N_I(x_j) v_{iI}(x_j), \quad (2.66)$$

with \hat{u}_i as the numerical solution of u_i , which is the so-called *shape functions* N_I multiplied by the nodal displacement vector v_{iI} , depending on the nodal coordinates x_j , for

$i, j = 1, 2, 3$ in the general three-dimensional case, and the subscript I is the number of nodes in the specified element. In the context of *isoparametric form*, the approximations are given by:

$$u_i(\xi^k) \approx \hat{u}_i(\xi^k) = \sum_I N_I(\xi^k) v_{iI}(\xi^k) \quad (2.67)$$

$$x_j(\xi^k) = \sum_I N_I(\xi^k) x_{jI}, \quad (2.68)$$

with ξ^k as the parametric coordinates and $k = 1, 2, 3$. In the *Galerkin* method, the virtual displacement function is approximated by the same shape function, denoted as:

$$\delta u_i(\xi^k) \approx \delta \hat{u}_i(\xi^k) = \sum_I N_I(\xi^k) \delta v_{iI}. \quad (2.69)$$

The computation of equation 2.59 requires the first-order derivation of u_i , thus the first derivatives of the shape functions w.r.t. x_j using the chain rules can be computed as:

$$\frac{\partial N_I}{\partial \xi^k} = \frac{\partial x_j}{\partial \xi^k} \frac{\partial N_I}{\partial x_j}, \quad (2.70)$$

or in the matrix form:

$$\frac{\partial N_I}{\partial \xi} = \mathbf{J} \frac{\partial N_I}{\partial \mathbf{x}}, \quad (2.71)$$

where

$$\frac{\partial N_I}{\partial \xi} = \begin{bmatrix} \frac{\partial N_I}{\partial \xi^1} \\ \frac{\partial N_I}{\partial \xi^2} \\ \frac{\partial N_I}{\partial \xi^3} \end{bmatrix}, \quad \frac{\partial N_I}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial N_I}{\partial x_1} \\ \frac{\partial N_I}{\partial x_2} \\ \frac{\partial N_I}{\partial x_3} \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} \frac{\partial x_1}{\partial \xi^1} & \frac{\partial x_2}{\partial \xi^1} & \frac{\partial x_3}{\partial \xi^1} \\ \frac{\partial x_1}{\partial \xi^2} & \frac{\partial x_2}{\partial \xi^2} & \frac{\partial x_3}{\partial \xi^2} \\ \frac{\partial x_1}{\partial \xi^3} & \frac{\partial x_2}{\partial \xi^3} & \frac{\partial x_3}{\partial \xi^3} \end{bmatrix}. \quad (2.72)$$

\mathbf{J} is the transformation between coordinates \mathbf{x} and ξ . With that, the shape function derivative w.r.t. \mathbf{x} can be written as:

$$\frac{\partial N_I}{\partial \mathbf{x}} = \mathbf{J}^{-1} \frac{\partial N_I}{\partial \xi}. \quad (2.73)$$

In addition, the virtual strain-displacement equation 2.58, and the virtual displacement can be rewritten in matrix form:

$$\delta \varepsilon = \underline{\mathbf{B}} \delta \mathbf{v} = \sum_I \mathbf{B}_I \delta v_I, \quad \delta \mathbf{u} = \underline{\mathbf{N}} \delta \mathbf{v} = \sum_I N_I \delta v_I, \quad (2.74)$$

with $\underline{\mathbf{B}}$ as the element shape matrix, and \mathbf{B}_I at each element node as:

$$\mathbf{B}_I^T = \begin{bmatrix} \frac{\partial N_I}{\partial x_1} & 0 & 0 & \frac{\partial N_I}{\partial x_2} & 0 & \frac{\partial N_I}{\partial x_3} \\ 0 & \frac{\partial N_I}{\partial x_2} & 0 & \frac{\partial N_I}{\partial x_1} & \frac{\partial N_I}{\partial x_3} & 0 \\ 0 & 0 & \frac{\partial N_I}{\partial x_3} & 0 & \frac{\partial N_I}{\partial x_2} & \frac{\partial N_I}{\partial x_1} \end{bmatrix}. \quad (2.75)$$

With the above equations, the virtual work in Eq. 2.59 for a single element can be written as

$$\hat{G}_{eq}^e = \delta \mathbf{v}^T \left[\int_{\Omega_e} \underline{\mathbf{B}}^T \boldsymbol{\sigma} \, d\Omega_e - \int_{\Omega_e} \underline{\mathbf{N}}^T \mathbf{f} \, d\Omega_e - \int_{\Gamma_{e_t}} \underline{\mathbf{N}}^T \mathbf{t} \, d\Gamma \right] = 0. \quad (2.76)$$

In case of linear elasticity, utilizing equation 2.43, and the same approximation for $\boldsymbol{\varepsilon}$ it follows:

$$\hat{G}_{eq}^e = \delta \mathbf{v}^T \left[\int_{\Omega_e} \underline{\mathbf{B}}^T \underline{\mathbf{C}} \underline{\mathbf{B}} \, d\Omega_e \mathbf{v}^e - \int_{\Omega_e} \underline{\mathbf{N}}^T \mathbf{f} \, d\Omega_e - \int_{\Gamma_{e_t}} \underline{\mathbf{N}}^T \mathbf{t} \, d\Gamma_{e_t} \right] = 0. \quad (2.77)$$

Summing over all the elements in the domain, it leads to the algebraic equation:

$$\mathbf{R} = \sum_e \delta \mathbf{v}^T (\mathbf{k}^e \mathbf{v}^e - \mathbf{f}^e) = 0, \quad (2.78)$$

with

$$\mathbf{k}^e = \int_{\Omega_e} \underline{\mathbf{B}}^T \underline{\mathbf{C}} \underline{\mathbf{B}} \, d\Omega_e, \quad \mathbf{f}^e = \int_{\Omega_e} \underline{\mathbf{N}}^T \mathbf{f} \, d\Omega_e + \int_{\Gamma_{e_t}} \underline{\mathbf{N}}^T \mathbf{t} \, d\Gamma_{e_t}, \quad (2.79)$$

as the element stiffness matrix and element force vector, respectively. The summation over all elements and proper assembling of the element matrices provides for arbitrary $\delta \mathbf{v}$, the global algebraic equation:

$$\mathbf{R} = \mathbf{K} \mathbf{V} - \mathbf{F} = 0, \quad (2.80)$$

where

$$\mathbf{K} = \sum_e \mathbf{k}^e, \quad \mathbf{V} = \sum_e \mathbf{v}^e \quad \mathbf{F} = \sum_e \mathbf{f}^e. \quad (2.81)$$

Here, \mathbf{K} represents the global linear stiffness matrix, \mathbf{V} the global solution vector containing the displacement vectors of all nodes, \mathbf{F} the global force vector, and \mathbf{R} the residual vector. In practice, the numerical solution of the above algebraic equation involves minimizing the residual vector, which requires direct or iterative solvers, and an error tolerance for convergence must be specified depending on the actual problem. Note that the integral in the equation 2.79 is determined numerically by quadrature rules. Throughout this work, the finite element framework *Multiphysics Object Oriented Simulation Environment (MOOSE)* [80] has been used.

3 Deep Learning-enabled Geometry-Lithiation Correlation in Battery Electrode Material

This chapter presents the research results of a long-term collaborative research project on battery materials with my collaborators at Texas A&M University under the supervision of Prof. Sarbajit Banerjee. I was provided with the raw image material and spectromicroscopic lithiation pattern data. Thus, much of the detailed experimental guidance, such as synthesis of the V_2O_5 material, instrumental instruction of the imaging techniques, and electrochemical characterization of the lithiation pattern, has been reported in our previous collaborative works [18, 81, 82, 83] and will be presented below. My main contribution includes the development of the deep learning model as an image analysis tool to characterize the morphological features of the particle ensemble, the derivation of the geometric descriptors, and the detailed workflow and analysis for the correlation of the extracted particle descriptors with the lithiation phase pattern.

3.1 V_2O_5 nanowires as battery cathode material and electrochemical properties

Understanding the principles that govern the chemistry of materials is essential for the rational design of LIBs. In particular, it is essential to be able to link measurements at the level of single particles and at the level of battery cells to understand how the structure and function of materials change at different length scales and how this affects the performance of hierarchical systems in LIBs. With this understanding, material researchers can develop new materials and design strategies that can improve the performance of energy storage systems [84, 85]. The combination of chemistry and mechanics has significant implications for the performance of cathode materials. Vanadium pentoxide (V_2O_5) is a prime example of an electrode dilemma, where many

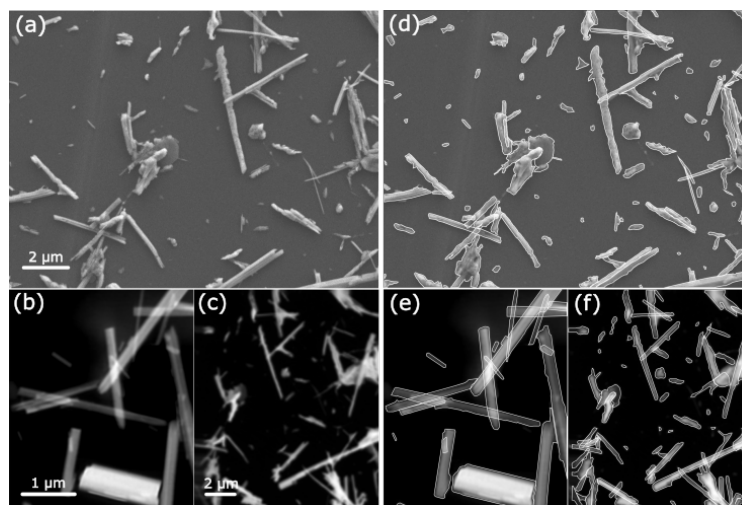


Figure 3.1: Overview of the microscopy images. **a**: SEM; **b**: X-ray ptychography; **c**: STXM; **d-f**: Corresponding manual annotation

beneficial properties are limited by material degradation due to the ongoing stress build-up from heterogeneous intercalation. V_2O_5 holds great promise as a cathode material because of its high theoretical capacity, potential for use in "beyond lithium" batteries, and the widespread availability of vanadium deposits, which reduce concerns about the criticality of cobalt-based battery chemistries. These advantages make V_2O_5 a potentially attractive option for a number of energy storage applications [86, 87, 88]. A major obstacle to the widespread adoption of the thermodynamically stable α - V_2O_5 phase is the occurrence of intercalation-induced structural transformations, which can create significant inhomogeneities and strain gradients within the material. These effects have been shown to degrade performance and durability, limiting the usefulness of this phase for practical applications [89, 90].

To enable a more profound understanding in this regard, hyperspectral X-ray spectromicroscopy techniques have been used to enable researchers to map compositional variations and phase separations over a wide range of length scales with high spatial and energy resolution. These techniques provide detailed information about the chemical and physical properties of a sample and can be used to study such phenomena [91, 92, 93, 94]. It has been shown that the patterns of lithiation in these systems are strongly modified by dimensional and morphological features such as particle size, particle geom-

etry, curvature, and interconnections. The coupling of high-dimensional X-ray imaging methods, ensemble measurements, and advanced image analysis tools holds great promise, and advanced image analysis tools holds great promise for identifying optimal crystalline sizes and geometries to overcome inherent material bottlenecks that lead to lithiation heterogeneities and ultimately stress accumulation [92, 93, 95]. However, the challenge of extracting shape and chemical features from large and complex datasets has limited the use of detailed measurements of particle networks. To overcome this challenge, techniques and tools must be developed that can help simplify and analyze these datasets, allowing for more detailed and accurate studies of particle networks and their properties.

In recent years, image segmentation algorithms that exploit the parallel processing capabilities of neural networks have received considerable attention for their potential to enable automated image analysis [66, 96]. For example, the well-received mask regional convolutional neural network (Mask R-CNN) algorithm [97] is now routinely used for general segmentation tasks. Common Object in Context (COCO) [98] and PASCAL Visual Object Classes (VOC) [99] have been developed together to train and benchmark the performance of algorithms for computer vision studies. However, the requirement for large datasets to train deep learning algorithms has been challenging to meet with experimental microscopy data in the material science community due to the inherent complexity in generation and the time-consuming nature of human annotation. Nevertheless, deep learning algorithms based on empirical data and human annotation have been developed for several classes of materials, such as graphene flakes [100] imaged by optical microscopy, carbon nanofibers [101] imaged by SEM, and another collection of electron microscopy images for various classes of materials [102]. Similarly, attempts to augment real image datasets of polycrystalline grains [103], or the use of image rendering techniques on nanoparticle images [104, 105] to counteract the prohibitive data acquisition step, have proven successful.

In this chapter, a deep learning model based on the Mask R-CNN algorithm is developed to overcome the challenges associated with limited training data, especially with an emphasis on instance segmentation and subsequent geometry-chemistry coupling. The model leverages the concept of transfer learning and has been retrained on synthetically generated microstructures. Three different types of microscopy images were collected by X-ray ptychography, scanning transmission X-ray microscopy (STXM), SEM techniques to test the robustness of the trained segmentation model. From an instance segmentation perspective, the V_2O_5 nanoparticle dispersions featured in this work, combined with their non-spherical geometry, lead to considerable irregularity in thickness, length-to-width aspect ratios, and edge profiles, which collectively result in significant variability in particle shape despite the all-encompassing nanowire classification. This shape variability

combined with particle agglomeration makes for an ambitious instance segmentation task that closely resembles the needs of many applications today [106, 107]. Through the lens of chemistry-mechanics coupling, V_2O_5 appears as a fascinating case study for the geometry-lithiation pattern correlation analysis [92, 93, 108]. After developing a robust instance segmentation model, the SVD technique in conjunction with a curated spectral database is applied to the STXM data to obtain highly accurate quantitative compositional lithiation phase maps of particle ensembles. These compositional phase maps are then used as the output property for correlation analysis of the particle descriptors segmented from the same dataset. The correlation analysis reveals the relationships between the geometrical features and the actual phase maps, and identifies how each geometrical feature affects the property of interest.

Outline of the chapter:

The actual chapter is organized as follows: In Section 3.2.1, the synthesis of the material and imaging techniques are introduced. This is followed by a brief introduction to the methodology of how to obtain the chemical composition maps using the SVD with the curated reference spectra database in Section 3.2.2. Afterward, details on the deep learning segmentation model and the synthetic data generation step are introduced in Section 3.2.3. In the results Section 3.3, the validation of the segmentation model with several images of different imaging methods to demonstrate the robustness model is presented. Then in the next Section 3.4, based on the developed deep learning model for nanowire characterization, the correlation of the geometrical descriptors obtained from segmentation results to lithiation patterns is showcased for the STXM data. The demonstrated method and results show the strong dependency of particle geometry on the lithiation pattern or phase composition of the nanowire material and implicate further potential for an automated, efficient structure-property analysis, see Fig. 3.2 for an overview of the workflow.

3.2 Data generation and methodologies

3.2.1 V_2O_5 nanowires & experimental imaging

Synthesis of V_2O_5 nanowires It is important to note here that two polymorphs of V_2O_5 are featured in this work. The α - V_2O_5 polymorph is the thermodynamically favorable phase in the V_2O_5 system and crystallizes in a layered structure with orthorhombic symmetry. ζ - V_2O_5 is a metastable 1D tunnel-structured polymorph that

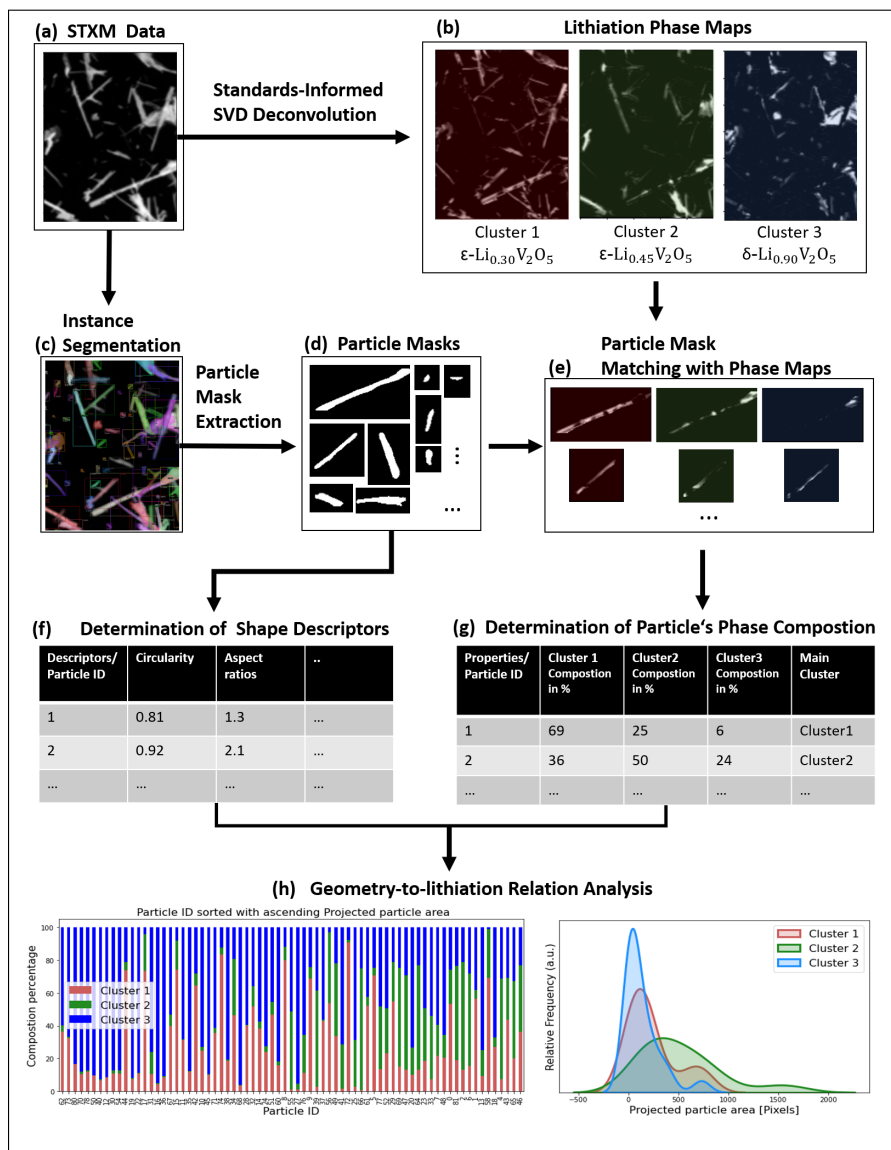


Figure 3.2: Overview of the workflow for geometry-lithiation correlation analysis.

preserves the composition of its thermodynamically stable counterpart while exhibiting drastically different structural motifs [90]. While distinct in terms of their crystal structure, both α -V₂O₅ and ζ -V₂O₅ produce nearly identical nanoparticle morphologies during hydrothermal synthesis. Nanowires of α -V₂O₅, the thermodynamic sink for binary vanadium oxides, were synthesized by a hydrothermal growth process. Briefly, V₃O₇·H₂O nanowires were initially prepared and calcined in air to obtain α -V₂O₅ nanowires crystallized in the orthorhombic phase, as reported previously [92]. Typical dimensions span from 50 to 400 nm in width and up to several microns in length. Chemical lithiation was achieved via submersion into a 0.01M *n*-butyllithium solution in heptane. Metastable ζ -V₂O₅ nanowires were prepared by a series of hydrothermal reactions as described in the previous work [109]. Briefly, bulk V₂O₅ and silver acetate were hydrothermally reacted to form an intermediate β -Ag_{0.33}V₂O₅ product. To create the tunnel-structured ζ -V₂O₅, β -Ag_{0.33}V₂O₅ was hydrothermally reacted with HCl in aqueous conditions to leach the Ag from the structure. For electrochemical sodiation, CR2032 coin cells were prepared under an inert argon environment. The working electrode was prepared by casting a mixture of the active material (ζ -V₂O₅, 70 wt.%), conductive carbon (Super C45, 20 wt.%), and binder [poly(vinylidene fluoride) 10 wt.%] dispersed in *N*-methyl-2-pyrrolidone onto an Al foil substrate. The counter electrode and separator used sodium metal and glass fiber, respectively. The 1M NaPF₆ solution was prepared using a solvent mixture of ethylene carbonate and diethyl carbonate (1:1 volumetric ratio) for the electrolyte. The extent of sodiation was controlled by galvanostatic discharging using a LANHE (CT2001A) battery testing system. Cells were disassembled, washed with dimethyl carbonate (DME), and dried for 24 h in an inert argon environment.

Scanning Electron Microscopy Before imaging, dispersions of α -V₂O₅ and ζ -V₂O₅ nanowires were created by drop-casting onto a silicon nitride substrate [92]. The SEM image shown in Fig. 3.1a was collected on a Tescan LYRA-3 instrument equipped with a Schottky field-emission source and a low aberration conical objective lens.

X-ray ptychography X-ray ptychography measurements were performed at the coherent scattering and microscopy beamline of the Advanced Light Source in Berkeley, CA. An optic with a 60 nm outer zone width and a 40 nm step size of the field of view was utilized. The image shown in Fig. 3.1b depicts the ratio between the X-ray absorption intensities at 527 eV and 529.8 eV, corresponding to known excitations to t_{2g} and e_g^* , respectively, which are indicative of the extent of intercalation. The symmetry labels herein indicate transitions to final vanadium 3d-O 2p hybrid states; the line shapes, peak positions, and relative intensities of these absorption features reflect the specifics

of vanadium reduction, electronic structure, and chemical bonding in these compounds and have been interpreted with the help of first-principles density functional theory calculations in past work [92, 93, 110].

Scanning transmission X-ray microscopy The STXM measurements were performed at the spectromicroscopy beamline 10D-1 of the Canadian Light Source in Saskatoon, SK utilizing a 7 mm generalized Apple II elliptically polarizing undulator source (EPU). Here, a focused beam spot was raster-scanned across the field of view with a 35 nm step size (thus determining the spatial resolution). A series of images were collected from 508 eV to 560 eV in 0.2 eV increments. The STXM image shown in Fig. 3.1c depicts the average absorption (optical density) contrast from 508 eV to 560 eV [92].

Manual annotation of the image data Manual annotation of the V_2O_5 datasets was facilitated by the web-based annotation tool Makesense.ai [111] in a polygon format, where points along a particle border are set to form the shape of the particles, see Fig. 3.1. This step was performed for every particle in the present images, and the annotated file was saved in JavaScript Object Notation (JSON) format to serve the validation purpose. It is worth noting that some limitations of the manual annotation process, such as sensitivity to human error and a dependence on the spatial resolution of the native images, naturally exist. Further sources of error stem from the dispersion's inherent complexity, resulting in many instances where particles are overlapped. The obtained data and its distribution are then used for the generation of synthetic data.

3.2.2 Lithiation property data

This section presents the workflow of transferring transmission information from large-scale STXM images to lithiation maps. The state of lithiation is captured by imaging at multiple X-ray energies chosen so that the X-ray transmission changes as the particles lithiate. However, it is unknown from the start how these states correlated with the X-ray transmission. To approach this relationship, two major steps are undertaken. The first step is to create a database of accurate spectra of pure lithiated phases. The second is a deconvolution scheme that converts the STXM data to lithiation maps using the reference database from step one. These two steps will be briefly introduced below.

Curation of spectroscopy reference data The first step of creating the reference data is to obtain lithiated phases by chemically reacting V_2O_5 particles with lithium in stoichiometric quantities, so their composition of $Li_xV_2O_5$ is precisely known. Upon

intercalation, V_2O_5 experiences several discontinuous changes to its crystal structure captured by X-ray diffraction patterns, resulting in α -, ϵ -, and δ -phases with increasing lithiation, corresponding to lithium stoichiometries of ca. $0.0 < x < 0.1$, $0.3 < x < 0.7$, and $0.9 < x < 1.0$, respectively [93]. Here, an experimental database of STXM measurements of phase-pure individual particles of α -, ϵ -, and δ - $Li_xV_2O_5$ is presented, see Fig. 3.3.

Deconvolution and phase mapping of STXM data Given the previous step's reference spectra data, STXM transmission data deconvolution is performed. In STXM, a focused beam spot was raster-scanned across the field of view at a series of energies $n = 1 \dots N$ to create transmission images of size $J \times K$ (length and width of the image), with their pixel positions, $p = 1 \dots P$. In a pre-processing step, the incident flux I_o is used to obtain the corresponding optical density image D related to the absorption coefficient μ , which depends on the state of lithiation and the path through the considered particles. A transmission image of heterogeneously lithiated particles enclose numbers of distinctive spectra $s = 1 \dots S$. Thus the optical density D_{NxP} is a spectrum at each pixel as:

$$D_{NxP} = \mu_{NxS} \cdot t_{SxP}, \quad (3.1)$$

with μ_{NxS} as the distinctive spectra set and their pixel-wise weighting t_{SxP} . Therefore, given the optical density from the measurements and the known absorption spectra, the spectral weighting maps can be obtained by matrix inversion using SVD:

$$t_{SxP} = \mu_{NxS}^{-1} \cdot D_{NxP}. \quad (3.2)$$

With the equation above, given the reference spectra from previous phase-pure lithiation states, the optical density maps can then be used to determine the lithium distribution within each particle of the STXM transmission images precisely. A schematic overview of the steps can be found in Fig. 3.3.

3.2.3 Mask R-CNN model & training

Numerous deep learning models for vision tasks can be found in the computer vision literature. For example, mask R-CNN and its variants (Mask scoring R-CNN [112], TensorMask [113]) have been reliable choices for the study of instance segmentation due to their high accuracy of instance mask predictions. Other algorithms, such as YOLO (You only look once) [114], have been found in the segmentation of nanowire-like materials [115]. In contrast, YOLO algorithms generally aim at higher training

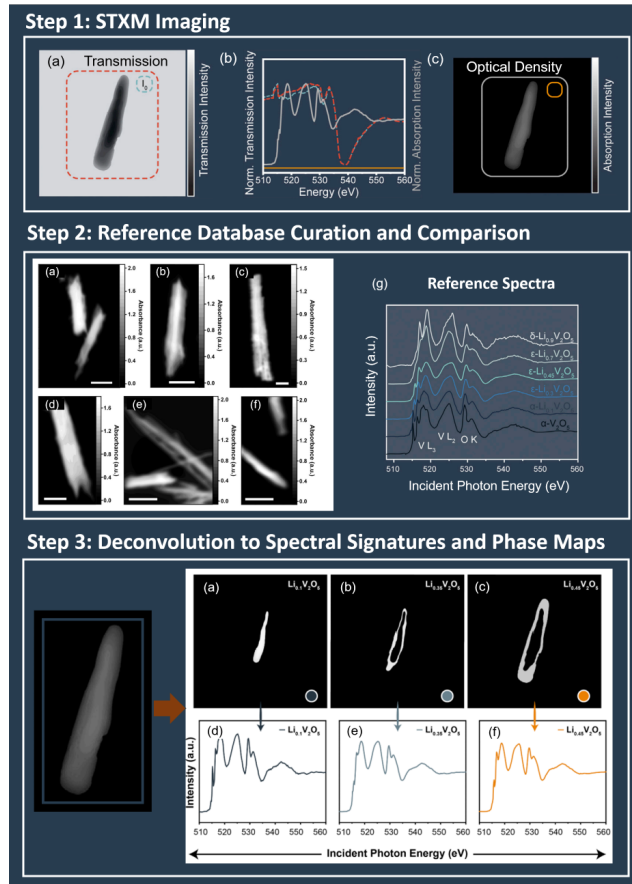


Figure 3.3: A schematic overview of obtaining lithiation phase maps. In Step 1 of data collection, STXM data shows a transmission image **a** of a nanowire particle and its intensity spectral data **b** w.r.t to different energy levels. The background spectrum I_0 is used to transfer the transmission data to the absorption profile of the sample **c**; Step 2 shows the STXM data for individual particles **a-f** of phase pure lithiation state and their intensity spectra **g**. The acceding order of curves in **g** corresponds to increasingly lithiated particles; In step 3, compared to the reference spectra data from step 2, direct deconvolution of the STXM image data is possible and leads to distinctively lithiated phase maps of the sample **a-c** and their intensity spectra **d-f**. Recreated with permission from [18].

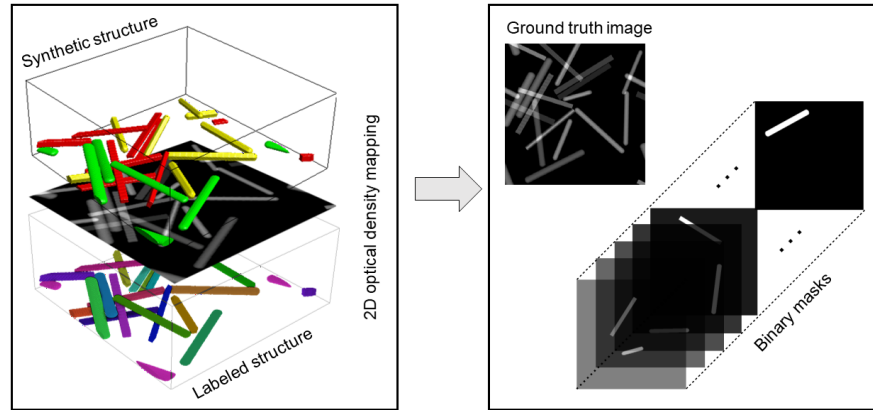


Figure 3.4: Synthetically generated dataset for training procedure. The 3D microstructure is compressed to create an optical density-based image as input data. The individually labeled nanowires in correlation with the optical density-based image are then used to create the binary masks for output data.

and prediction speed without providing prediction masks in the first place. To this end, recent extensions of YOLO, such as YOLACT [116], have been developed with mask predictions toward real-time application. Similar deep learning-based methods with specific modifications applied to fibrous structure segmentation can be found in the literature [117, 118]. While it is believed that different instance segmentation algorithms can also be applied for the present instance segmentation study, the decision is made to apply the well-established Mask R-CNN algorithm within the Detectron2 framework [119] due to its great built-in API and well-structured online documentation. Details on the essential components of the Mask R-CNN model can be found in the following section. It is important to note that while the synthetic dataset was modeled after experimental microscopy images, the training steps for the model developed in this work were performed solely on the synthetic dataset without any actual images, which is now introduced as follows.

Synthetic training data generation

A random nanowire generator using the software Geodict ® [120] to generate synthetic image datasets reminiscent of the V_2O_5 experimental datasets particles is developed. In the generation step, the number of particles, length, and shape distribution for each

training sample was specified to create 3D voxel-based structures, Fig. 3.4. The chosen size of the domain was 512x512x200 (WxBxH). For the present work, the number of particles, diversity of morphology, and resolution approximate the experimental information contained in the experimental X-ray ptychography data in Fig. 3.1b. Higher resolution, i.e., larger domain size, can be chosen at the cost of longer generation time and image file size. The height of the domain for the synthetic 3D microstructure was chosen to exceed the total height of the overlapped nanowires. Further, the nanowires were internally enumerated and deposited one after another. This workflow ensures that the labels are predetermined, thus bypassing the need for human annotation at a later phase. In order to mirror the transmission intensity generated by X-ray ptychography, an optical density compression step was applied to emulate thickness information. Here, voxels were first compressed along the out-of-plane direction, then summed and divided by the total thickness of the microstructure. The pixel values comprising the projected optical density image map in the in-plane directions were normalized and further transformed to a gray color scale, expressed as a value from 0 to 255. Accordingly, regions where two or more particles are overlapped can be distinguished by a sudden change in optical density (i.e., pixel intensity). In a subsequent step, a standard Gaussian filter (filter size of 2) was applied to the images to account for the blurring of the particle edges. As shown in Fig. 3.4, the ground truth sample was split into individual binary masks for each nanowire contained in the synthetic ground truth image. Therefore, for each training sample, the dataset consists of one input image and N output binary mask images with N as the number of particles in that image. These binary mask images can be further used to obtain statistical information about the morphology descriptors. Before evaluation, an initial dataset size of 250 images was generated for training purposes. After an initial assessment, additional images were generated in recursive steps involving a greater diversity of particle morphology to replicate the experimental data closely. A final dataset of 1250 synthetic images was obtained with an 80/20 split for the training and validation steps. In order to introduce further variability in the dataset, standard augmentation techniques have been applied as the dataset is fed forward to the data loader. It is noted that no additional dataset was created in the training process. To account for intrinsic variability in the contrast and brightness of experimental microscopy datasets, a random brightness and a random contrast filter ranging from 0.7 to 1.2 and a random flip in the horizontal and vertical directions with a probability of 0.5 were applied. Details on the training process and dataset can be found in the model training and data section. It is worth mentioning that there is potential to enhance the synthetic dataset by incorporating additional real images.

Model structure

The following section briefly explains the basic structure of the Mask R-CNN model and its workflow. The model architecture can be divided into 3 main parts [97], as illustrated in Fig. 3.5:

1. *Feature extraction.* This step is usually referred to as the backbone of the model and is constructed with multiple CNN layers. The input image is introduced and passed through the CNNs to extract representative features of the entire image. The CNN layers are usually deep and contain most model weights updated during the training steps. The backbone used here can easily be tailored to the desired segmentation task to improve speed and performance. The backbone used for the training in this work is a Feature Pyramid Network (FPN) with the ResNet-50 network [96] pre-trained on COCO-dataset [97]. FPN addresses one of the main challenges in object detection: detecting objects at different scales [121]. FPN is constructed of two pathways, bottom-up and top-down, so it can capture features at different scales. See [122] and [81] for more details on FPN structures.
2. *Region of interest proposal and alignment.* This step in the model is designed to identify and extract instances from feature maps produced by the backbone. The Region proposal network (RPN) achieves this by generating a series of regions of interest (ROIs), each encapsulating a single instance. The model generates hundreds of ROIs and an associated confidence score to quantify the probability of encompassing an object for the given ROI. After filtering and modifying the coordinates of each ROI, RPN advances portions of the feature map (corresponding to each ROI) with a fixed size to the model prediction head in order to determine the properties (e.g, bounding box, and instance mask, etc.) of each instance.
3. *Overhead for mask and bounding box prediction and classification.* The so-called prediction heads are functions that predict the characteristics of that proposed instance. For object detection purposes, most common R-CNN structures typically provide two instance heads: the bounding box regression head, which draws a bounding box around an instance, and the instance classification head to classify the object class. The typical prediction head of Mask R-CNN is therefore an extension of R-CNN models with the mask segmentation, in which a binary mask is generated to label the predicted instance.

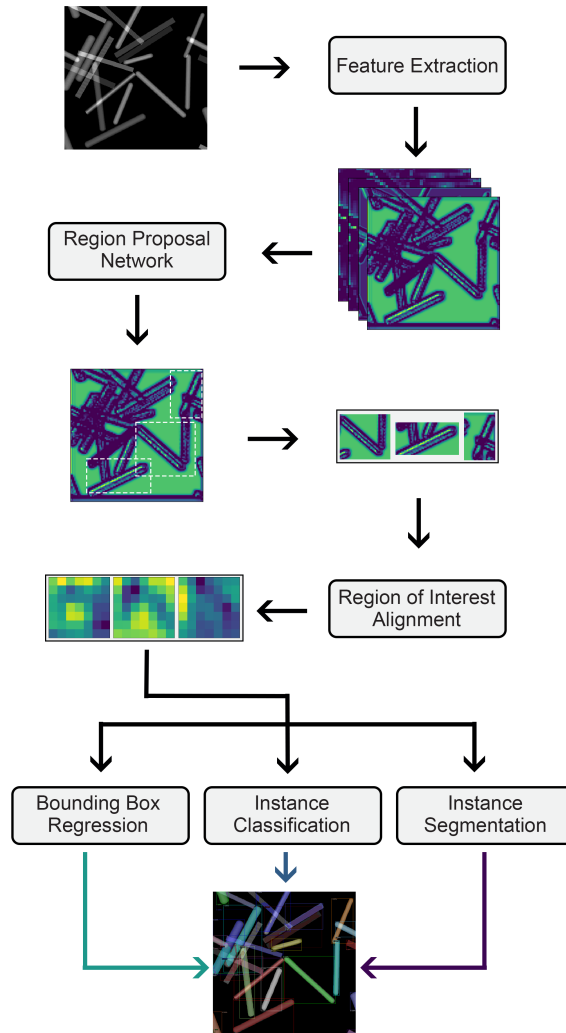


Figure 3.5: Basic workflow of the Mask R-CNN algorithm: The input image is processed by different network components to extract high-level features, and regions of interest for the prediction of bounding boxes, classes, and masks.

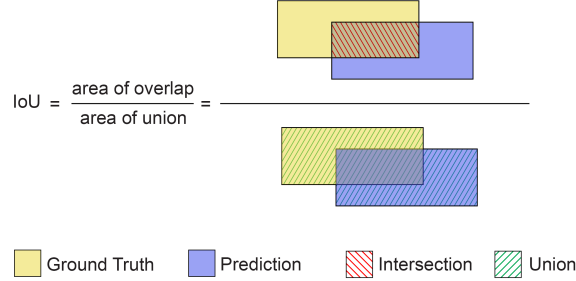


Figure 3.6: Definition of intersection over union: Area of object intersection over the area of object union. This definition is used to calculate precision and recall for a given IoU threshold.

Model loss functions

The difference between model prediction and ground truth should be minimized during training. This optimization procedure requires the definition of a function to perform this calculation, -usually referred to as a loss function or a cost function. Typically, the optimization goal in neural networks is to minimize this loss function. Different loss functions can be used for other tasks based on the input data and the desired output of the model. For example, in the Mask R-CNN model used in this work, the defined loss function is based on the summation of 3 individual loss functions [97]:

$$L = L_{\text{bbox}} + L_{\text{cls}} + L_{\text{mask}} \quad (3.3)$$

where L_{bbox} is a smooth L^1 loss function used for predicting bounding box coordinates, with the advantage of having steady gradients for large loss numbers and fewer oscillations during updates for smaller loss values. L_{cls} is a cross-entropy loss function to measure the classification of multiple classes and returns a probability between 0 and 1. In the case of a binary segmentation (in this case, the nanowire and the background), the cross-entropy loss function can be written as:

$$L_{\text{cls}} = -(y \log(p) + (1 - y) \log(1 - p)) \quad (3.4)$$

where the \log is the natural log, y is a binary value (0 or 1) indicating the observation class, and p is the predicted probability for the given observation. As for L_{mask} , it is a binary cross-entropy for the generated binary mask of size $m \times m$ for each ROI.

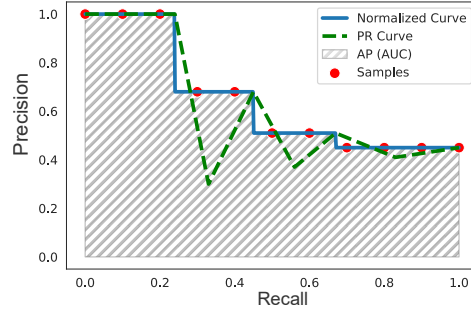


Figure 3.7: PR Curve: Exemplary precision values corresponding to given recall values, shown by red dots. According to VOC2010, to avoid the wiggly representation of the PR curve, shown by dashed green lines, precision values take the previous maximum values to the left to generate the normalized curve, shown by solid blue lines. The final AP is calculated as the area under the normalized curve.

Model evaluation metrics

To evaluate the segmentation results, three metrics were applied. The segmentation accuracy defined in Equation 3.5 is used here as simple pixel-wise correctness, which only examines the actually predicted pixels with predictions made from all the objects in the foreground. TP denotes the true positive, FP the false positive, and FN the false negative predictions. Note that this metric is only applied to the foreground, which differs from the common use considering the background pixels likewise.

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \quad (3.5)$$

The second evaluation scheme is according to COCO dataset [98] based on mean average precision (mAP). This is introduced briefly in the following section. Firstly, to confirm a correct prediction of the bounding box or mask, intersection over union (IoU) is used. It is defined by the area of intersection between bounding boxes divided by their union, as shown in Fig. 3.6. Predictions are true positive if IoU is higher than a given threshold and false negative if that is lower than that threshold. The most common thresholds used are $\text{IoU} > 50$ (AP_{50}) and $\text{IoU} > 75$ (AP_{75}). To further understand mAP, precision, and recall are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.6)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.7)$$

The recall is considered as a true positive prediction rate, i.e., the ratio between true positive predictions and all ground truths. Precision is defined as the ratio between true positive predictions and all predictions that are made. Further, the obtained precision and recall are plotted to obtain the so-called precision-recall (PR) curve, with the area under it referred to as average precision (AP). In VOC2010 [99], a modified PR curve was introduced, where precision for a given recall r is set to the maximum precision for any $\tilde{r} \geq r$. Afterward, the AP can be computed by numerical integration for the area under the curve (AUC) as shown in Fig. 3.7. In COCO, mAP is defined as the average of AP for all classes in each image. In this work, a single class of nanowires is segmented, but it is worth noting that additional classes can be accounted for with relative ease. For VOC, IoU > 50 is usually considered a true positive prediction, which results in true predictions, with any IoU higher than 0.5 contributing equally to the AP. To rectify this problem, COCO uses different thresholds for IoU ranging from 0.5 to 0.95 with a step size of 0.05 and then reports the average of all computed APs to vary thresholds as mAP, see Equation 3.8. In this work, we use AP for mAP_{coco} and assume the difference is clear from the context.

$$\text{mAP}_{\text{coco}} = \frac{\text{mAP}_{50} + \text{mAP}_{55} + \dots + \text{mAP}_{95}}{10} \quad (3.8)$$

Evaluation based on COCO also reports more detailed results based on the scale of the detected objects. With reported AP_{small} for objects with an area smaller than 32^2 pixels, $\text{AP}_{\text{medium}}$ for objects with an area between 32^2 and 69^2 pixels and AP_{large} for objects with an area greater than 69^2 pixels, one can evaluate the model performance on segmenting objects in different scales. Finally, given the fundamental motivation to extract particle statistics from image datasets, the performance of the model is further evaluated based on the accuracy of the predicted statistics. Computation is made based on the segmented masks of each particle. Statistical information obtained from the predictions is then compared to the manually annotated results.

Model training

The stochastic gradient descent method was used with the default setting provided by the Detectron2 implementation of Mask R-CNN algorithm [119]. The hyperparameters modified in the parameter study can be found in Tab. 3.1. For the present study, the effect of synthetic dataset size and hyperparameters such as learning rate, region of

interest batch size per image (ROI HEAD), and Non-Maximum-Suppression (NMS) were studied. ROI HEAD is a subset of the proposed ROIs bounding boxes from RPN, so the loss can be calculated on this subset rather than on all box proposals. IoU threshold (IoU THR) for RPN defines the ratio of overlaps between the ground truth boxes and proposed boxes during the training. NMS mainly acts as a filter to remove overlapping proposed boxes. More details and default hyperparameter settings can be found in [119]. Note that the current model leverages pre-trained weights on the COCO dataset and the chosen learning rates were, therefore, larger than training an entirely new network from scratch. Results on model prediction performances trained with smaller learning rates can be further found in [81]. The training was performed on four A100/V100 Nvidia GPUs at Lichtenberg Cluster, TU Darmstadt, for a total time of 5-10 hours with a batch size of 8 images per GPU. The training time here should only provide an approximation. Detailed study and optimization of the performance speed were not the objectives of this work.

3.3 Instance segmentation of experimental images

In this section, the segmentation results are evaluated considering the metrics defined in model evaluation metrics. It is important to re-emphasize that the deep learning model has been trained solely on synthetic image datasets modeled after the X-ray ptychography and scanning transmission X-ray microscopy data. Consequently, the SEM image, which is distinctive in terms of contrast generation is foreign to the trained model. All images are, obtained from mentioned microscopy techniques as they are, and were not filtered for evaluation purposes. The only pre-processing step applied to the ptychography and STXM images involves the conversion from transmission data to absorbance (optical density) [123]. The results obtained from the deep learning model are compared to manually annotated results, which are subject to uncertainty to certain level due to visual limitation. It is further noted that the current model leverages pre-trained model weights and can be re-trained and extended to multiple object classes of further morphologies, such as e.g., nano-spheres, see [81] for more details.

Before applying the model to real microscopy images, 20 models were trained with various hyperparameters to examine their influence on the synthetic images, see Tab. 3.1. Subsequently, all 20 trained models were applied to segment the three types of microscopy image. The best performer in mask segmentation AP for each image type was selected to visualize the segmentation masks in Fig. 3.8, 3.11, 3.13b. The evaluation of the model's segmentation accuracy begins by assessing its ability to accurately separate the nanowires from the background at a semantic level. The inherent strength of instance

No.	No. Epochs	Dataset Size	Learning Rate	ROI Head	IOU THR	NMS	AP BBOX	AP SEGM
1	250	250	0.02	256	0.6	0.7	89.442	87.392
2	250	500	0.02	256	0.6	0.7	91.843	88.878
3	250	750	0.02	256	0.6	0.7	92.219	89.568
4	250	1000	0.02	256	0.6	0.7	93.541	90.195
5	500	250	0.02	256	0.6	0.7	90.949	87.908
6	500	500	0.02	256	0.6	0.7	92.831	89.872
7	500	750	0.02	256	0.6	0.7	93.336	90.034
8	500	1000	0.02	256	0.6	0.7	93.878	90.254
9	750	250	0.02	256	0.6	0.7	92.219	89.568
10	750	500	0.02	256	0.6	0.7	92.820	90.095
11	750	750	0.02	256	0.6	0.7	93.454	89.958
12	750	1000	0.02	256	0.6	0.7	94.002	90.487
13	500	750	0.01	256	0.6	0.7	93.564	89.72
14	500	750	0.03	256	0.6	0.7	93.279	89.87
15	500	750	0.02	128	0.6	0.7	92.582	89.614
16	500	750	0.02	512	0.6	0.7	93.513	89.932
17	500	750	0.02	256	0.7	0.7	93.278	89.945
18	500	750	0.02	256	0.8	0.7	93.523	90.139
19	500	750	0.02	256	0.6	0.6	92.902	89.952
20	500	750	0.02	256	0.6	0.8	93.58	89.74

Table 3.1: Hyperparameter study in the training procedure

segmentation is that it includes the subordinate functionality of semantic segmentation, where the semantic results can be immediately extracted from the mask predictions. As shown in Fig. 3.8, 3.11, 3.13b, fairly good segmentation results were obtained. These object masks were then used to evaluate semantic segmentation accuracy presented in Fig. 3.8, 3.11, 3.13c. Blue denotes the true positive (TP) pixels, to which the model predicts correctly as given in the ground truth provided by human annotation. The green color indicates false positive (FP) pixels, which means the model has inaccurately predicted that these pixels belong to a particular nanowire. The red color denotes the false negatives (FN), which depict the pixels that belong to a nanowire (based on the ground truth) but were not identified as such by the model. The training model's performance is discussed in more detail in the following sections.

Furthermore, in order to enhance the accessibility of the model as a segmentation

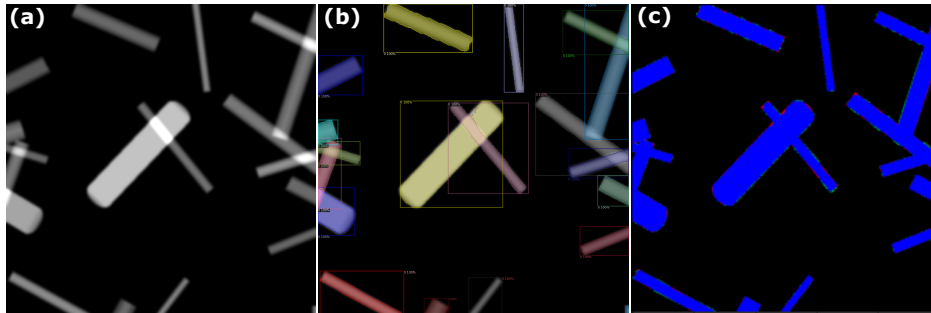


Figure 3.8: Model prediction on V_2O_5 nanowires within a synthetic image. (512x512 pixels) **a**: Test image; **b**: Predicted instance masks with lower opacity plotted on the test image; **c**: Semantic binary mask; Blue: TP, Red: FN, Green: FP; every nanowire instance in the present synthetic image has been successfully segmented with highest accuracy and negligible deviation in the overlapping regions and particle boundaries.

tool for nanorod-like structures, a web-based interactive application has been created. This application allows users to easily access and analyze their own image datasets, facilitating data mining and exploration. Upon uploading the data and initializing the prediction model, statistics on predicted masks can be obtained and visualized accordingly. Details and access on the web-based interactive application can be found in later section.

Synthetic nanowire image As for the synthetic nanowires shown in Fig. 3.8, not surprisingly, the deep learning model correctly segments particles contained within the synthetic nanowire datasets. The obtained AP of the tested 20 models reach a AP score of around 90 for bounding box regression and mask segmentation, respectively, see Tab. 3.1. As the synthetic image type is basically known to the model, good accuracy is expected. Tuning the hyperparameters, the test dataset would generally result in a better AP in bounding box prediction and segmentation masks, however, it does not influence the results on synthetic images significantly. To demonstrate the power of the current model in instance segmentation of overlapped, optical density-based nanowire images, the prediction capabilities of the current model is compared to a traditional routine algorithm -the distance map-based Watershed algorithm- for a simple exemplary synthetic image. It has been shown that the Watershed algorithm generally fails to isolate individual nanowires in the overlapped regions, while the current model is designed to perform such tasks [81].

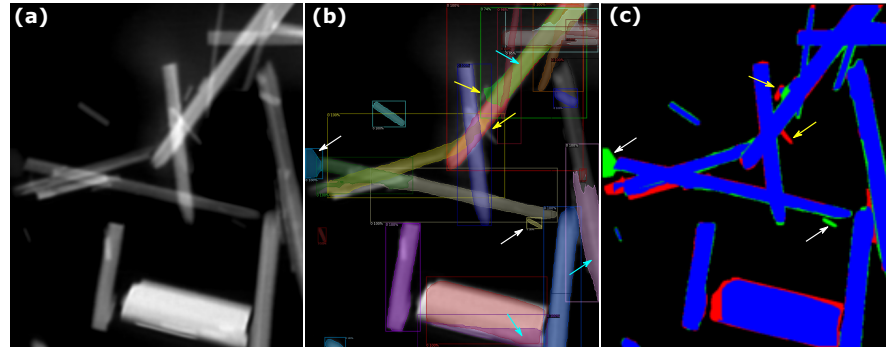


Figure 3.9: Model prediction on V_2O_5 nanowires within the X-ray ptychography image. (531x449 pixels) **a**: Test image; **b**: Predicted instance masks with lower opacity plotted on the test image; The best performer in AP segmentation mask in Tab. 3.2 was used to visualize the prediction masks in **b**. **c**: Semantic binary mask; Blue: TP, Red: FN, Green: FP; yellow arrows indicate the main FN pixels, i.e. nanowire pixels missed by the model. White arrows indicate the FP pixels, which were overlooked in the manual annotation but detected by the model. Cyan arrows indicate over-predicted pixels, which the model predicts as additional nanowires, but were only regions of higher intensity.

No.	Accuracy	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
13	86.6	39.145/42.327	64.638/62.519	43.965/39.964	25.248/22.442	52.753/58.03	72.525/85.05
19	85.5	41.809/41.719	74.603/67.042	43.25/40.438	28.34/19.785	52.772/52.774	62.624/70.099
14	86.2	38.584/41.662	63.618/62.803	42.159/42.159	25.248/26.931	44.998/50.644	75.05/70.099

Table 3.2: Model performance on X-ray ptychography image (Bounding box/Segmentation mask)

X-ray ptychography image Although the model has been trained solely by synthetic datasets, good segmentation results are observed on experimental datasets. For the presented X-ray ptychography image, the model predicts the overall binary mask with good accuracy and scores a segmentation score of 86.6, see Tab. 3.2. The AP, AP_{75} for both the bounding box and the segmentation mask are comparatively high and score around 40, respectively. AP_{50} reaches a score of around 62. From the metrics, it is mentionable that AP_l is greater than AP_s and AP_m , indicating that the image contains larger particles and they were segmented to a greater degree than smaller ones. At the instance level, two false-positive nanowires have been identified and are (shown in green) indicated by white arrows in Fig. 3.9c; the origin of this false-positive result

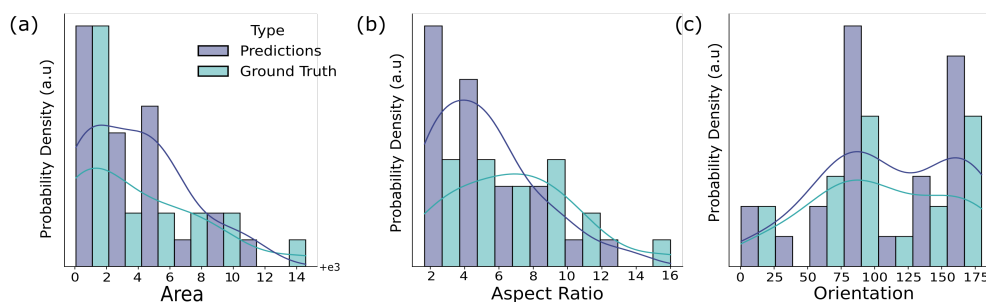


Figure 3.10: Particle statistics from the X-ray ptychography image are illustrated by histograms and their kernel-density estimates (KDE) as a function of area (summation of pixels corresponding to each particle mask), aspect ratio (particle length/width), and orientation (angle relative to the horizontal axis) in **a**, **b**, and **c**, respectively. The statistical results show qualitative agreement, however, the main discrepancy represented by over-predicted pixels causes the shift of the KDE curve in the respective number ranges.

is low pixel intensity near the threshold that separates particles from the background. For the same reason, these particles were not manually annotated but nevertheless were identified by the model thus demonstrating its performance, which is competitive with careful human annotation (while being much more accurate). The two particles shown in red (indicated by yellow arrows) are missed by the model, presumably by the noise in the corresponding region. But overall, considering the optical density input, the individual particles are extracted with good accuracy. In general, the mask predictions consistently perform extremely well when particles are well separated in space while overlapping regions (notoriously more difficult to segment) are still identified with good accuracy. Some issues arise when the optical density gradient is low with less clear transitions in the overlapped region. Further, the model tends to find smaller particles within larger instances as shown by the cyan arrow. The origin of this limitation stems from the broad range of particle aspect ratios and thicknesses in the experimental data (ca. 50 – 500nm), thus resulting in a less continuous optical density distribution with individual particles. In the training data, particle morphologies were generated assuming a prismatic structure with little-to-no variation in the cross-sectional shape. In contrast, the experimentally synthesized V_2O_5 are subject to defect formation, particle sintering, and intrinsic variations in the crystal growth during synthesis. This leads to a particle dispersion that is highly complex,

non-prismatic, and an ambiguous optical density mapping and further complicates the detection in the overlapped area and creates artifacts that would possibly mislead trained models. This also introduces additional challenges during synthetic data generation and segmentation tasks. Nevertheless, this complexity in particle size, shape, and extent of curvature has pronounced effects on the emergent properties of these cathode particles so their correct identification remains important [92, 93, 108]. Here, over-predicted particle masks inside the larger ones can be easily removed in a post-processing step. This step was not performed here to preserve the originality of the model prediction. However, to enhance the general prediction capability of the model and avoid post-processing procedures, the generation of non-prismatic structures for training datasets can be instrumental and will remain as future work. Lastly, statistical information on particle area size, aspect ratio, and orientation are compared in Fig. 3.10 in form of the histogram with kernel density estimation (KDE). The area size is determined by the summation of the pixels belonging to different particle masks. The aspect ratio is calculated as the ratio between the image coordinates of the longer edge to the shorter edge of the corresponding predicted mask. The orientation is considered as the angle between the particle alignment to the horizontal axis and ranges from 0 to 180 degrees. As can be found in Fig. 3.10, the statistical information from the X-ray ptychography data is in qualitatively good agreement. The main discrepancy is contributed by the number of additionally detected smaller particles inside the larger particles and as explained previously, leading to a higher density of the histogram for an area size of 2000-4000 pixels and aspect ratio 2-4. This perturbation can be also observed for orientation for particles with 75 to 100 degrees and 150 to 175 degrees. As the number of particles in the image is comparably small, the feature distribution becomes sensitive to the number of detected particles.

No.	Accuracy	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l
1	75.6	27.267/21.791	54.404/51.877	23.423/17.587	25.228/20.862	37.744/28.877	NaN
3	73.1	30.51/21.36	53.577/52.172	29.092/15.618	24.447/17.329	48.116/32.782	NaN
9	75.3	28.293/21.049	51.763/48.671	27.977/17.437	20.581/16.013	49.245/34.714	NaN

Table 3.3: Model performance on STXM image (Bounding box/Segmentation mask)

STXM image Of the two X-ray microscopy techniques considered in this work, X-ray ptychography offers the greatest spatial resolution (ca. 6 nm), thus, from a pure image segmentation perspective, it is expected that the performance of the model to be greatest for this class of images. Nevertheless, techniques such as scanning transmission X-ray

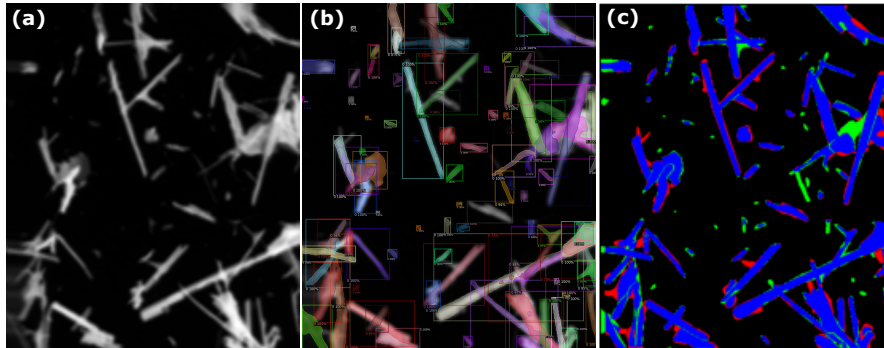


Figure 3.11: Model prediction on V_2O_5 nanowires image by STXM. (531x449 pixels - rescaled)
a: Test image; **b**: Predicted instance masks with lower opacity plotted on the test image; The best performer in AP segmentation mask in Tab. 3.3 was used to visualize the prediction masks in **b**. **c**: Semantic binary mask; Blue: TP, Red: FN, Green: FP; In **c**, many FP nanowires are present, which indicate the missing manual labels, but are detected by the model. The results demonstrate the robustness of the model in segmenting images of low-resolution and densely packed particles.

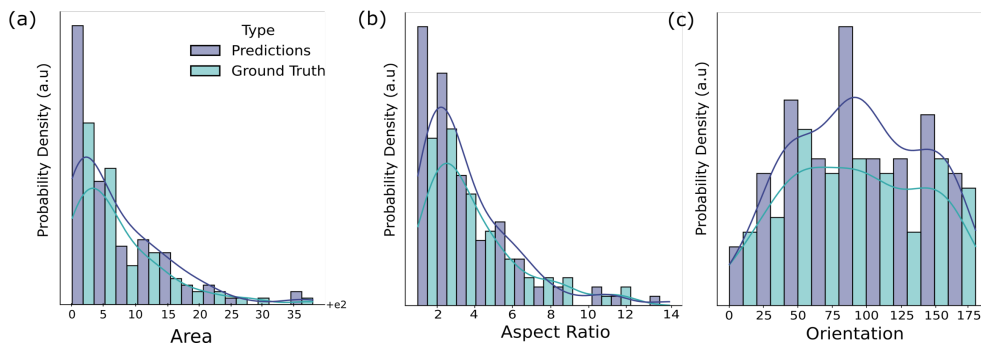


Figure 3.12: Histograms and KDE curves show particle statistics from the STXM image as a function of area, aspect ratio, and orientation in **a**, **b**, and **c**, respectively. The statistical results show good qualitative and quantitative agreement. The main reason for the shift of KDE curves is due to the FP pixels, leading to higher density estimates in the respective number range.

microscopy which offer slightly lower spatial resolution (ca. 25 nm) but enabled more detailed mapping of spectral features (i.e. have richer chemical information) are equally important for cheminformatics. In this work, the original resolution of the STXM image was 100x100 pixels and fewer than in the X-ray ptychography image. To enable a sharper manual annotation, the STXM image was rescaled to the size of ptychography image for easier visual access (it is important to note that this does not fundamentally change the resolution enabled by the experimentation). The number of particles, their variations in morphology, and the complexity of their dispersion are noticeably greater than the X-ray ptychography image making the segmentation task considerably more challenging. Nevertheless, the segmentation accuracy score is around 75, and the AP score is around 30/21 for bounding boxes and masks, respectively. Since there exist relatively no larger particles in the image, AP_l was not provided. The comparably lower but surprisingly good scores highlight the complexity of segmenting complex particle dispersions with several instances of overlap and agglomeration. This is especially noticeable for the false positive green particles in Fig. 3.11c, which were generally overlooked in the manual annotation process. Access of human annotation is strictly limited for images of such complexity. However, at the instance level, from a visual perspective, the model performs considerably well. Overlapped particles are consistently identified and agglomerations, while difficult to identify visually, are captured by the deep learning model. The statistical distribution of the features in Fig. 3.12 agree well both to a qualitative and quantitative extent. The shape of KDE agrees well with the manual ground truth distribution. As the model prediction captured smaller particles, which were not manually labeled, the statistics of the prediction are shown to have a higher density distribution in each feature characteristic, observable as a small peak shift of the KDE curve. The results suggest that for complicated particle networks contained in a relatively low-resolution and low-contrast image, deep learning models indeed deliver more comprehensive information on the statistical information than human annotations. For more comparison of manually annotated STXM images and their corresponding statistics, see [81].

SEM image The deep learning model shows success in segmenting the particles in the ptychography and STXM optical density images, in part, due to the mechanisms of contrast generation, which involves the transmission of an incident X-ray source through the bulk of the material. Here, the degree of transmission is related to the energy-specific elemental absorption cross-section, corresponding to the excitation of electrons from core levels to unoccupied or partially occupied states, giving rise to similar absorption contrast for compositionally homogeneous particles and allowing the discernment of overlapped

No.	Accuracy	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l
1	59.6	27.61/12.927	51.8/23.126	22.884/5.206	20.586/7.842	43.495/22.443	38.356/36.634
5	61.7	27.916/12.688	54.302/37.916	21.149/5.643	22.776/8.337	43.195/21.884	16.311/25.743
13	33.2	21.561/12.659	47.924/35.553	15.718/5.088	14.011/8.089	44.587/23.695	15.317/12.871

Table 3.4: Model performance on SEM image (Bounding box/Segmentation mask)

intersections (in form of increased optical density due to thickness effects). As a point of comparison, in scanning electron microscopy, the detection of secondary electrons or backscattered electrons from the surface is sensitive to surface morphology, edge effects, and charge build-up and is fundamentally different from the X-ray ptychography and STXM images shown in previous sections. To demonstrate the versatility of the model, a scanning electron micrograph was shown for the purposed of segmentation. Despite the fundamental differences in contrast generation between the data used to train the model and the SEM data utilized as an input here, the model performs sufficiently well in the overlapping regions despite the absence of optical density information. It is postulated that the deep learning model still captures the contrast gradients at the particle boundaries and utilizes it as a criterion to identify individual fibers in the overlapped regions, independent of the background. Nevertheless, residual errors in the segmentation persist; for instance, as in the previous dataset, agglomerations are not well separated, as shown by the cyan arrow in Fig. 3.13b. In addition to unidentified nanowires, (indicated by yellow arrows) minor discrepancies exist in the rightmost region of the image, where oversized masks (green, indicated by white arrows) were predicted for individual wires. Here, the absence of optical density information combined with the distinct mode of contrast relative to STXM and X-ray ptychography contributes to the lower AP shown for the SEM image. Nevertheless, a strong potential to improve the observed inconsistency can be observed. For example, an additional class for agglomerated phase can be introduced and generated in the microstructure generation step, to further differentiate between different foreground phases and, subsequently, nanowires instances. Despite the lower AP score of around 13, the statistical results seem to be less sensitive in the presence of more significant particle numbers. The main deviation lies in the number of undetected isolated particles of smaller size and therefore leading to an underestimation of statistical density w.r.t area size up to 1000 pixels, aspect ratio from 2.5 to 7.5, and orientation from 100 - 150 degrees, as shown in Fig. 3.14a-c.

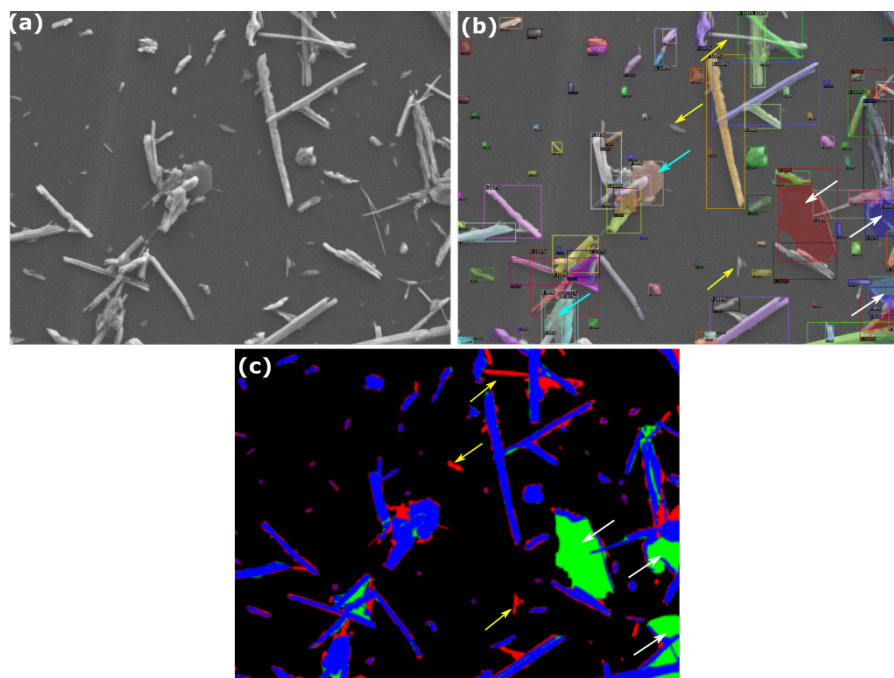


Figure 3.13: Model prediction on V_2O_5 nanowires within SEM image. (957x1280 pixels) **a**: Test image; **b**: Predicted instance masks with lower opacity plotted on the test image; The best performer in the AP segmentation mask in Tab. 3.4 was used to visualize the prediction masks in **b**. **c**: Semantic binary mask; While sufficiently good segmentation results are found for overlapping nanowires, several FN pixelated nanowires, indicated by yellow arrows in **b** were missed by the model. Further discrepancies can be found in the region of agglomeration denoted by FP pixels and white arrows. The absence of optical density information combined with the distinct mode of contrast relative to STXM and X-ray ptychography contributes to the comparably lower prediction performance.

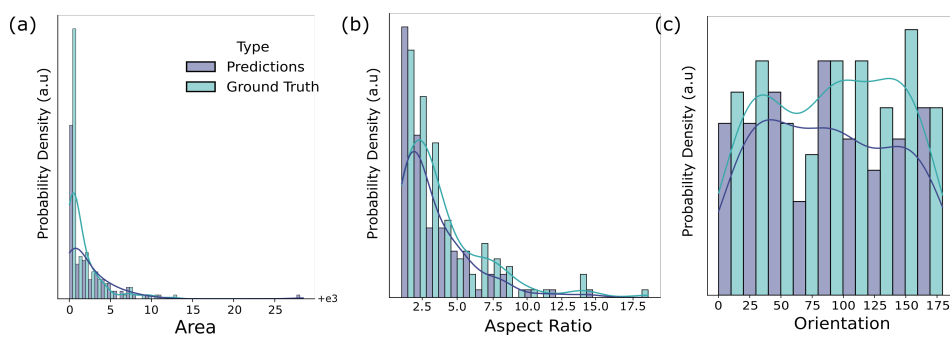


Figure 3.14: Histograms and KDE curves show particle statistics from the SEM image as a function of area, aspect ratio, and orientation in **a**, **b**, and **c**, respectively. The statistical results show good qualitative agreement. The main discrepancy exists for smaller FN nanowires missed by the model, thus leading to higher density estimates in the ground truth (manual annotation) in the respective number ranges, e.g., the peak in area size distribution.

3.4 Particle geometry-lithiation correlation analysis

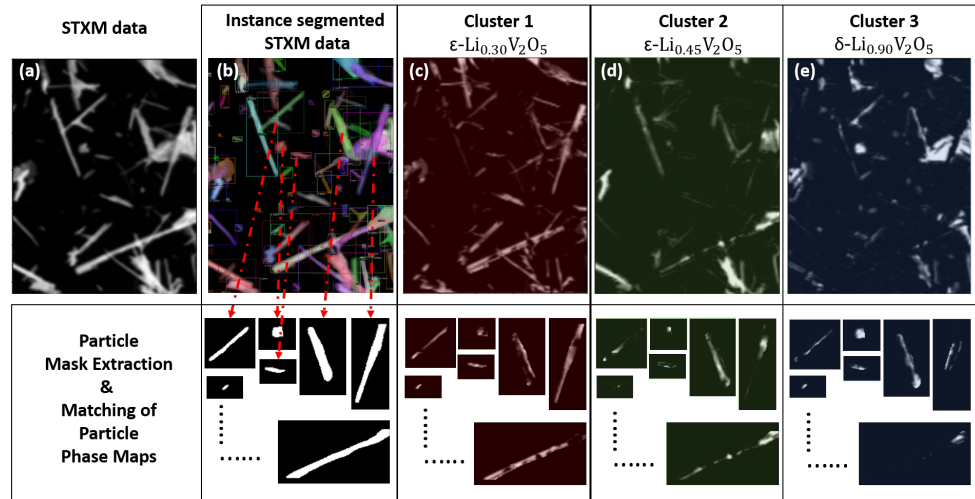


Figure 3.15: a: STXM image data; b: Instance segmented image dataset with particle masks as in Fig. 3.12; c-e: Lithiation phase maps.

Improvements in chemical imaging have facilitated advances in image acquisition speed, signal-to-noise, and computational, ML-based characterization algorithms have enabled more powerful tools to establish links between structure-property-function relationships. Based on the developed deep learning model for nanowire characterization, this section deals with the correlation of the geometrical descriptors obtained from the segmentation results with the lithiation patterns presented for the STXM data. The general workflow is shown in Fig. 3.2 and will now be introduced in more detail. The representative dataset is the previous STXM image, a network ensemble of >80 polydisperse $\text{Li}_x\text{V}_2\text{O}_5$, shown in the SEM figure in Fig. 3.15a. This dataset demonstrates the need to use advanced image analysis tools due to the significant number of particles and their existing overlapping structures at this mesoscale dimension. The dataset is first processed by SVD (as described in Section 3.2.2) to obtain distinctive lithiation phase maps. This resulted in three distinct patterns that accurately capture visualizations of the spatial distribution of lithiation pattern clusters 1-3, corresponding to $\epsilon\text{-Li}_{0.3}\text{V}_2\text{O}_5$, $\epsilon\text{-Li}_{0.45}\text{V}_2\text{O}_5$ and $\delta\text{-Li}_{0.9}\text{V}_2\text{O}_5$ phases, respectively. See Fig. 3.15c-e. At the same time, the previously trained instance segmentation model is applied to obtain the masks

and bounding boxes of individual nanowire particles in the STXM image Fig. 3.15a. The individual masks reflect the projected contour and shape of individual particles and form the working basis for further morphological feature extraction, shown in the second row of Fig. 3.15. They are stored in a multidimensional NumPy array of size $H \times W \times N$, where H, W are the height and width of the STXM image and the binary masks containing the segmented particle in binary form, and N is the number of segmented nanowires. In Fig. 3.15 the binary masks are cropped to the size of the particle for better visualisation. In a subsequent step, using the *findContours* function from the *OpenCV* library, the contour of the binary particle mask is extracted and used to compute the geometric descriptors for each particle. The considered descriptors and the statistical values are summarized in Tab. 3.5. There are 8 different descriptors that describe the morphological characteristics of the particle shapes defined in this work. The projected particle perimeter descriptor defines the length of the contour boundary. The projected particle area describes the area pixels enclosed by the contour boundary. Aspect ratio is calculated as the ratio of the long edge to the short edge of the detected bounding box. Circularity describes the closeness of the contour to a circle. For a circle, circularity is equal to 1. The closer an object is to a circle, the closer the circularity is to 1. Convexity is defined as the ratio of the convex perimeter to the perimeter. A convex perimeter can be determined by drawing a convex hull over the actual shape, i.e. the smallest convex polygon containing all points of the non-convex shape. Convexity is sometimes used to describe the irregularity in the shapes, such as surface roughness. Solidity is defined as the area divided by the convex hull area and describes the extent to which a shape is convex or concave [124]. The eccentricity is defined as the square root of the squared difference of the major axis L_M and the minor axis L_m divided by the major axis when fitting an ellipse through the particle shape, see Tab. 3.5. The difference between eccentricity and aspect ratio is the shape fitted by the actual particle. While aspect ratio uses the long and short edges of a fitted bounding box (rectangle), eccentricity uses the length of the minor and major axes of the ellipse. This is a direct measure when the particle shape is close to an ellipse rather than a rectangle. Orientation describes the general direction of the particle object relative to the horizontal axis. In particular, the descriptors projected particle perimeter and area are dependent on particle size and can be considered as absolute values, whereas the other six descriptors are relative, i.e. independent of particle size. The mean and variance numbers give a first insight into the variety in the morphology of the extracted dataset. As can be seen in 3.5, larger numbers in perimeter, area and aspect ratio deviation indicate a strong variation in particle size distribution and morphology in the particle ensembles. The smaller number in circularity indicates that the particles have a significant deviation from circular shapes, but with relatively smooth boundaries (high convexity), which

Descriptor	Definition	Mean	Std.
Projected particle perimeter	P [Pixels]	84.5	74.6
Projected particle area	A [Pixels]	222.4	275.9
Aspect ratio	L_s/L_l [-]	3.4	2.2
Circularity	$4\pi A/P^2$ [-]	0.42	0.21
Convexity	P_{convex}/P [-]	0.95	0.03
Eccentricity	$\sqrt{L_M^2 - L_m^2}/L_M$ [-]	0.89	0.16
Solidity	A/A_{Convex} [-]	0.83	0.11
Orientation	-	89	56

Table 3.5: Geometrical descriptors, their definition and statistical numbers for the presented dataset. L_s, L_l denote the short and the long edge of the detected bounding box around the particle and L_m, L_M denote the length of the minor and the major axis when an ellipse is fitted through the particle.

can be visually confirmed by the generally wire-shaped particles.

Next, to obtain the phase composition of each extracted particle, the individual binary masks are used to match the position of the particles in the derived phase maps by the so-called Hadamard product \odot or entry-wise product. More formally, let's denote the binary masks as \mathcal{M}_{hw}^N for each N -th mask, the deconvoluted phase maps as ϕ_{hw}^i , $i = 1, 2, 3$ for each i -th cluster. The composition map for each particle is the superposition of each phase map weighted by its corresponding stoichiometry coefficient [93, 18]:

$$C_{hw}^N = \sum_i x^i \mathcal{M}_{hw}^N \odot \phi_{hw}^i \quad (3.9)$$

with $x^1 = 0.3, x^2 = 0.45, x^3 = 0.9$ as the stoichiometry coefficients of the phase clusters and the lower subscript $()_{hw}$ as the pixel position in the image. Using equation 3.9, the compositional percentage of each contributing phase to the total composition can be determined as

$$CP^{N, i} = \sum_h \sum_w \frac{x^i \mathcal{M}_{hw}^N \odot \phi_{hw}^i}{C_{hw}^N} \quad (3.10)$$

and summing all the pixels of the mask area for the N -th mask and the i -th cluster.

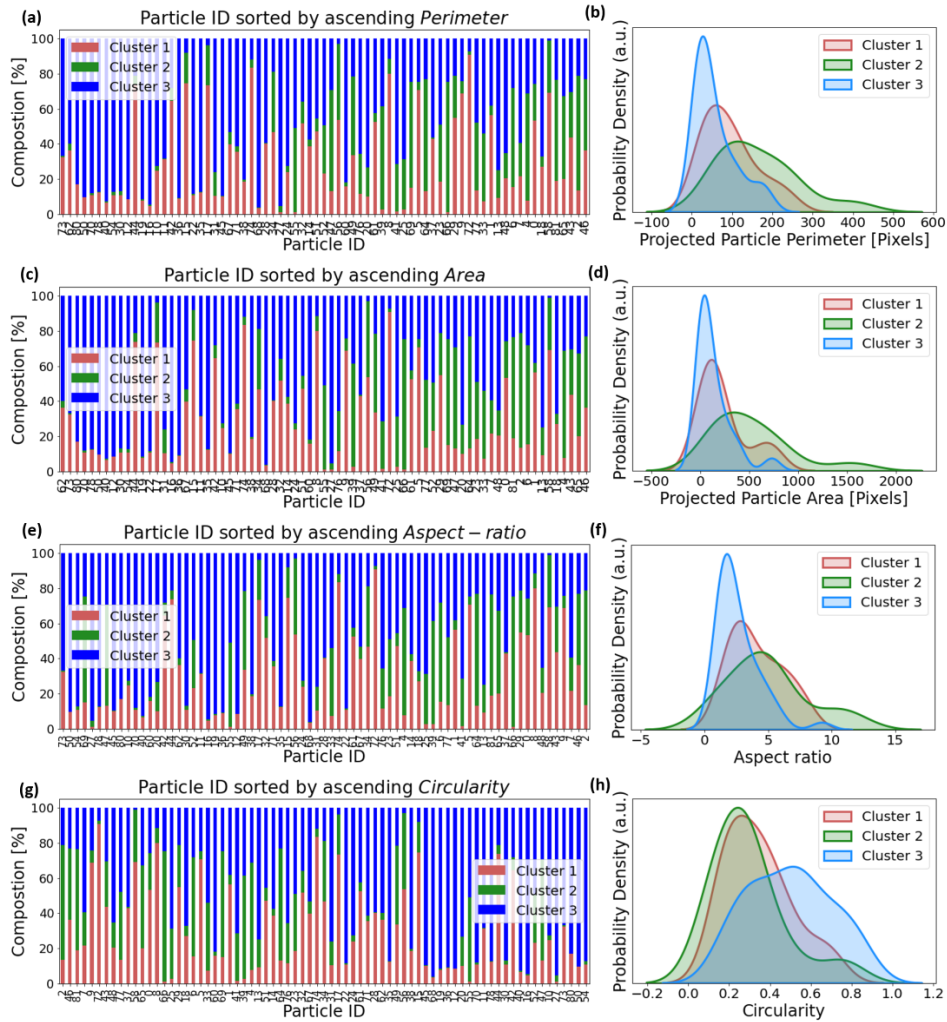


Figure 3.16: **Left column:** Stacked bar plot of particle composition, arranged with ascending descriptor numbers. **a:** Projected particle perimeter, **b:** Projected particle area, **c:** Aspect ratio, **d:** Circularity; **Right column:** KDE curves for each descriptor and their characteristic pattern clusters.

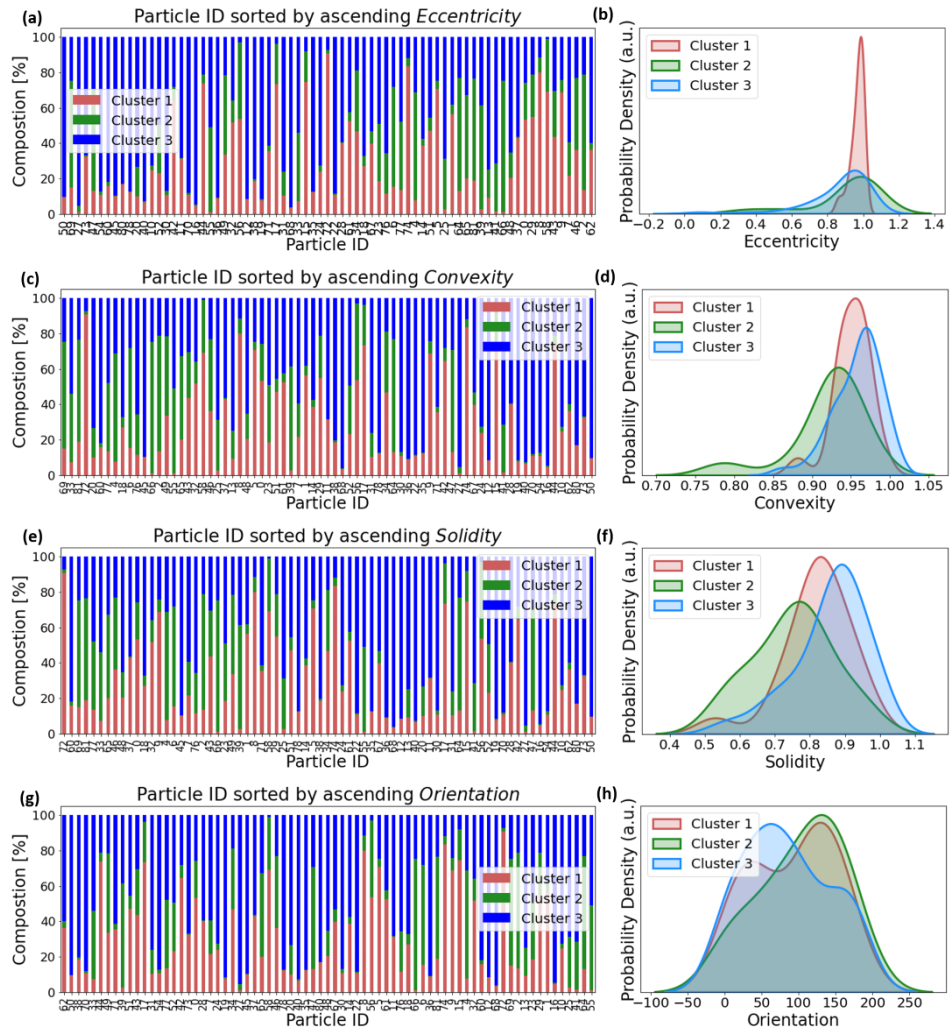


Figure 3.17: **Left column:** Stacked bar plot of particle composition, arranged with ascending descriptor numbers. **a:** Eccentricity, **b:** Convexity, **c:** Solidity, **d:** Orientation; **Right column:** KDE curves for each descriptor and their characteristic pattern clusters.

The above equations are evaluated for each particle correlated with each cluster in the STXM dataset. This is first visualized in the stacked bar plot shown in Fig. 3.16 and 3.17. The sub-figures in the left column show the different segmented particle IDs arranged by ascending descriptor values on the x-axis and the composition of each particle on the y-axis. The aim of this type of visualization is twofold. The first is to see how the descriptors correlate with the state of lithiation, i.e. whether a particle is high or low lithiated based on its shape descriptor (as cluster 1 indicates a low state of lithiation and cluster 3 indicates a high state of lithiation). The second is to visualize to what extent the phenomenon of phase segregation is present, i.e. the degree to which different phases coexist in the same particle. In order to determine more quantitatively how the lithiation states are related to the descriptors, statistical distribution plots for each cluster or descriptor are shown in the right column.

A first evaluation of the results shown in Fig. 3.16 indicates a strong correlation between the descriptors and the lithiation state. Here, clusters 1,2,3 correspond to low, medium and high lithiated domains within the particle (concentration distribution of lithium ions), respectively. The composition percentage is the quantitative measure of the lithiation state, denoting the ratio between the individual phase composition of each cluster and the total composition as introduced in Eq. 3.10. First, there is a clear inverse correlation between the projected 2D particle perimeter, area and lithiation pattern. This is observed in the shrinking blue bars with increasing perimeter and area in Fig. 3.16a,c, which implies that with increasing particle area, and hence perimeter, the overall lithiation decreases and the lithiation pattern shifts towards less lithiated clusters, namely clusters 1 (red) and 2 (green). A more detailed quantitative analysis by density distribution plot supports the same findings, as shown by the mode values (x-values to the peak of the curve) for each KDE curve. While the high density of cluster 3, denoted by the peak of the blue curve, can be found around 20-30 pixels for the perimeter (around 80-100 pixels for the area), the peak density of the other two clusters decreases more clearly towards the larger perimeter and area compared to the stacked bar plot. At the same time, as shown by the mixture of stacked bars to the right, phase segregation becomes more dominant as area and perimeter increase. The inverse correlation can also be found for increasing aspect ratio, again demonstrated by the decreasing percentage of blue bars and high peak density (mode value around 2) towards smaller aspect ratios in Fig. 3.16e, f. It can be concluded that particles with high aspect ratio tend to have less lithiation and a higher degree of phase mixing. The circularity describes the deviation of the shape of the particle from a perfect circle and is relatively low for the particles of nanowire morphology considered, denoted by 0.2-0.3 of clusters 1, 2 and between 0.4-0.6 for a relatively wide distribution of cluster 3 in the distribution plot in Fig. 3.16h. It can be interpreted that the more circular the

particles are, the better the lithiation for the set of particle shapes presented. However, it should be noted that the large size variation of the current particle ensemble may weaken this interpretation, as this shape metric is not evaluated on the same size basis. As described earlier, eccentricity describes the closeness of the actual particle shape to an elliptical shape. The results in Fig. 3.17a show a similar inverse pattern with the lithiation pattern decreasing towards higher eccentricities. Notably, while the mode values are high for all clusters, the high density is extremely pronounced for cluster 1 as shown in Fig. 3.17b. This suggests that highly eccentric particles tend to be less lithiated. This is consistent with the observation of Mistry et.al. [125], as they found that phase mixing or heterogeneity in the lithium distribution within the particle, or phase segregation, is at least in part driven by geometric asphericity (high eccentricity and aspect ratio) and extended defects within the particles. The density distributions of convexity and solidity have relatively small deviations with respect to the three clusters due to the fact that the morphology of the particles is similar, albeit with a large deviation in particle size. Direct correlations in these two descriptors can be found with the lithiation state, as shown by the tendency of the increasing amount of blue in the bars towards higher convexity and solidity numbers, corresponding to the mode value in the density distribution plot. Since no clear correlation can be observed for the descriptor orientation, it can be assumed that this descriptor has no influence on the actual property of interest.

Based on the results, it can be understood that reducing the particle size and perimeter (or surface area in 3D) can effectively lead to a higher lithiation of the particles, while increasing the particle circularity, convexity and solidity results can achieve the same effect and reduce phase segregation and consequently a higher lithiation state. However, the qualitative trend of decreasing perimeter with increasing lithiation state, see Fig. 3.16a, should be treated with caution. This observation is mainly driven by the pronounced particle size variation in the current particle ensembles. In fact, smaller particles tend to be highly lithiated, see Fig. 3.18b, and smaller particle areas logically lead to smaller perimeters. To mitigate this problem, another descriptor is introduced, namely the perimeter-to-area ratio (P/A), which can be interpreted as the ratio of surface area to volume or specific surface area in a 3D context. This descriptor is explained in many applications in science and engineering where diffusion or adsorption processes are expected by the transport of matter through the surface [126, 127]. In the present context, the larger the P/A ratio (there is more surface area per unit volume through which the lithium ions can diffuse), the faster the uptake of lithium ions and thus leading to a higher state of lithiation (given the same diffusion rate within the particle domain). This interesting observation can be confirmed for the data presented in Fig. 3.18b, as the increasing P/A ratio results in a larger percentage of composition of

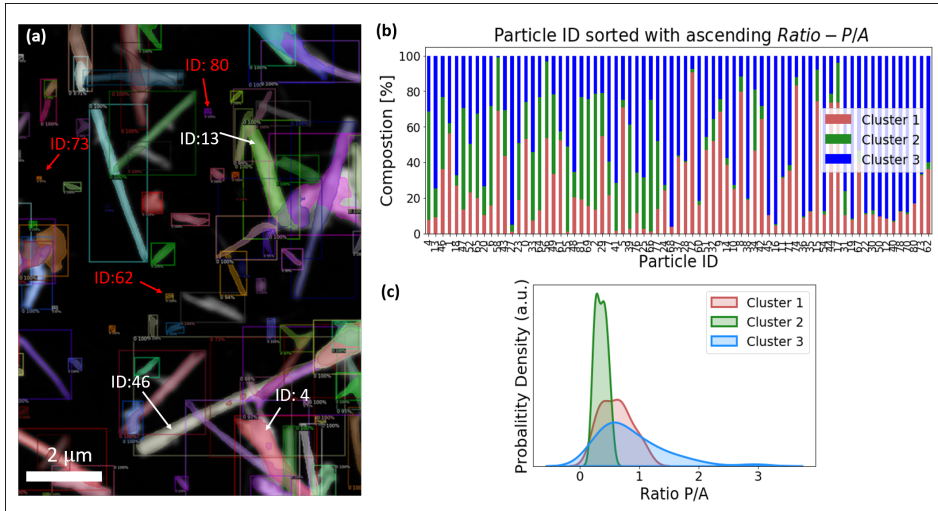


Figure 3.18: **a**: STXM image of instance segmented particles with particles of interest and their IDs highlighted. Red IDs 73, 80, 62 point to the particles with the largest P/R ratio. White IDs 4, 13, 46 point to the particles with the smallest P/R ratio; **b**: Stacked bar plot of particle composition arranged in ascending P/A ratio values. Particles 4, 13, 46 show a high degree of phase coexistence, while particles 73, 80, 62 have a more homogeneous concentration of lithium from cluster 3; **c**: KDE curve for the same descriptor.

cluster 3 (increasing blue portion to the right). Thus, it is clear that P/A ratio or specific area is the determinant geometrical descriptor, neither particle size nor perimeter should be considered alone.

In conclusion, the above characteristics can be explained by the fact that smaller particles with high specific surface area reduce the diffusion path length [18], and increase the reaction interface with electrolyte and lithium, and promote lithium transport in the host material. This further reduces the specific surface current rate and improves the stability and specific capacity [128]. The analysis of phase coexistence and its relationship with particle geometry provides valuable insights and also emphasises the importance of using well-defined standards as reference spectra. By taking into account the interplay between electrochemistry and mechanics, reducing particle dimensions reduces the effect of interfacial mismatch induced by phase boundaries. As a result, phase separation, which contributes to stress and battery degradation, can be suppressed [18].

As shown by Luo et. al. Micrometer-sized platelets of V_2O_5 exhibit significant variations in phase distribution across the electrode thickness, whereas electrodes composed of homogeneous nanometer-sized spherical V_2O_5 exhibit remarkably uniform lithiation characterised by extended solid solution behaviour. Furthermore, due to the isoperimetric inequality (the sphere has the smallest surface-to-volume ratio for a given volume), a design suggestion can be derived from the preceding analysis for the morphological property, e.g. by introducing acute-angled spikes to further increase the specific area and consequently the higher state of lithiation.

3.5 Perspectives on data extension

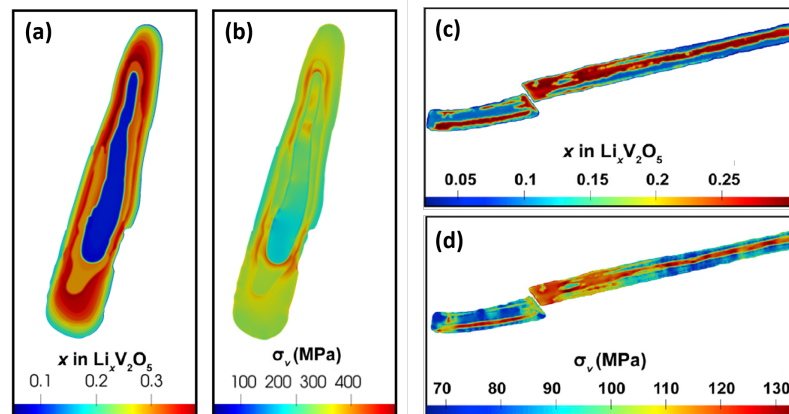


Figure 3.19: **a,c**: Spatially resolved composition maps of exemplary particles for subsequent FE simulation; **b,d** Calculated von Mises stress map. More details on the workflow of lithiation-to-stress mapping can be found in [18].

As previously mentioned, the interaction between mechanics and electrochemistry is crucial in the transformation of electrode materials. The intercalation of lithium ions into the host material leads to the formation of multi-phase coexistence domains, which in turn generates significant misfit strains at the boundaries between differently lithiated phases. Considering the strain mismatch observed in the composition map, the current dataset can be expanded by incorporating physically sound FE analysis. This involves creating a finite element mesh based on the particle contour to define the

particle domain. The composition map is then converted to lithium concentration using available literature data and mapped onto the nodes of the FE mesh within the problem domain. By considering the concentration distribution within the particle domain, it can be modeled as an eigenstrain field, which is incorporated into the constitutive equation in linear-elastic mechanics. By setting appropriate boundary conditions, a boundary value problem can be solved using the given heterogeneous eigenstrain field, resulting in a stress map for further analysis. These steps have been previously performed in our work [18], and the resulting von Mises stress maps for exemplary particles are shown in Figure 3.19 for illustrative purposes. Incorporating stress information as an additional output property can be seen as an extension of the previous geometry-lithiation correlation analysis and remains a potential direction for future research.

To expand the current dataset, there is a potential for further enrichment by incorporating additional particle shapes into the descriptor datasets. Currently, the data analysis primarily focuses on nanowire-shaped particles, which restricts the range of derived descriptors. However, as previously discussed, gathering experimental data can be a time-consuming and expensive process. One possible solution is to extend and enrich the dataset by generating synthetic particle morphologies, which would allow for a broader range of descriptors. This approach would enable the exploration of optimal particle shapes that may not be readily available or difficult to obtain in experimental setups. In line with this concept, a collection of artificially generated particles can be created computationally using a parametric formulation in polar coordinates [129].

$$r = [L_{\text{ratio}} |\cos(\frac{d}{2}(\theta + \phi))|^p + |\cos(\frac{d}{2}(\theta - \phi))|^p]^{1/p}. \quad (3.11)$$

The parametric formulation includes several parameters that define the particle shape. The ratio between the major and minor axis lengths is denoted as L_{ratio} , d represents the number of axes, ϕ corresponds to the rotation phase shift, and p indicates the surface curvature of the particle shape. By using this parametric formulation, the particle contour data is inherently provided. From the contour data points, the geometrical feature descriptor can be easily derived by fitting the points into the previously described workflow. By employing physically sound FE models, such as electro-mechanically coupled equations presented in works [130, 131, 132], the electro-chemical properties obtained through experimental procedures with STXM techniques can be further simulated. This approach involving synthetically data ensures the generation of a wide range of descriptor values, providing valuable insights into particle morphologies that are challenging or expensive to synthesize in experimental settings. As this methodology is currently under investigation, it can be viewed as an intriguing opportunity that complements the experimental analysis presented in this study.

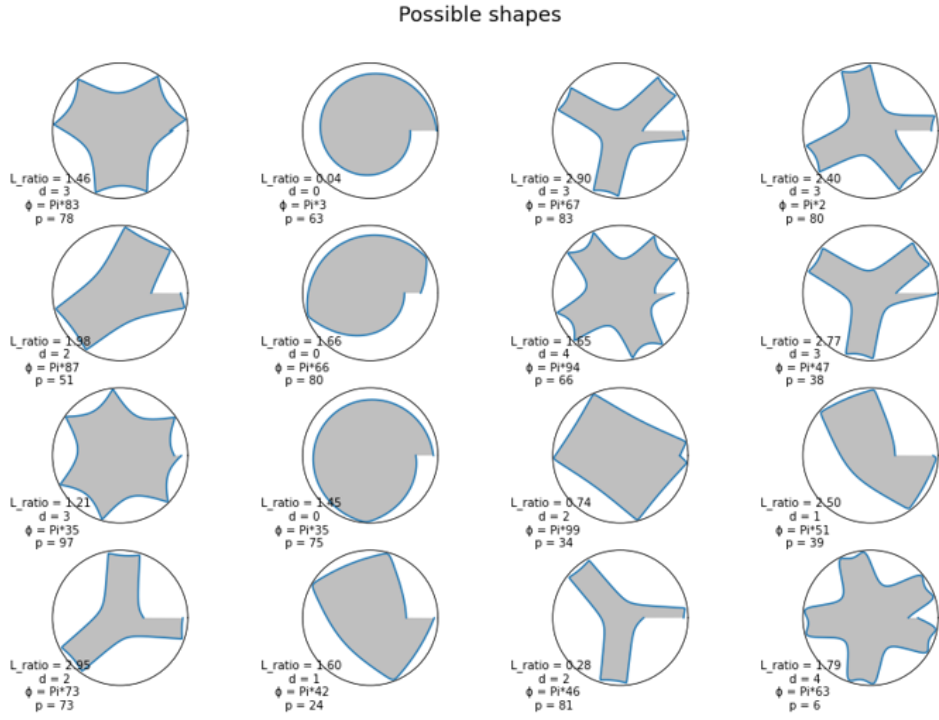


Figure 3.20: A set of synthetically generated particles by varying the parameters given in Eq. 3.11. The broken symmetry or shape defect in the particle shapes is purposely introduced and achieved by setting the L_{ratio} to be a decimal number. One can preserve the particle symmetry in the particle shapes by setting this parameter equals to an integer [Manuscript under preparation].

4 Data-driven Microstructure Sensitivity Study of Fibrous Paper Material

This chapter presents the results of my research within the Functional Paper Research Project (FiPRE) at the Technical University of Darmstadt. I collaborated with my colleagues from the Surface Physics Group and the Macromolecular and Paper Chemistry Group under the supervision of Prof. Robert Stark and Prof. Markus Biesalski. During the project, I was provided with raw images of fiber networks and some material parameters, as well as experimental data for the validation of FE models. Much of the content discussed in this chapter has been reported in our joint publications [133, 134, 135] and will be presented below. My main contribution includes the 3D fiber network generation, Cohesive Zone Model (CZM) development, FE simulation, ML model development, and the analysis shown in this chapter.

4.1 Fiber network structure of paper material and mechanical properties

Paper materials have immense potential in terms of their mechanical properties, coupled with their renewable and sustainable nature, making them the subject of extensive research for various applications. The processability and serviceability of paper are highly dependent on the properties of its constituent fibers, the bonds between these fibers, and the intricate structural details at the microscopic level [136]. To enhance the strength of paper materials at the macroscopic level, mechanical and chemical processes are used to increase the cross-linking contact areas and improve the interfacial strength. These processes contribute to the strengthening and densification of paper materials. Despite the inherent stochastic nature of the fiber structures in paper, they exhibit a consistent overall orientation, albeit with statistical variations caused by the nature of the fibers themselves and the papermaking process.

In order to explore the underlying mechanisms at a finer scale, many research efforts

have extensively investigated the micromechanical and microstructural properties of fibers, fiber networks, and the bonds formed between adjacent fibers through a combination of experimental and numerical approaches. Advanced imaging techniques such as Micro Computed Tomography (Micro CT) and X-ray synchrotron have emerged as valuable tools for characterizing the morphology of individual fibers as well as the interfiber contacts between them [137, 138, 139, 140]. These techniques allow detailed examination of fiber characteristics and the nature of the bonds formed at the microscopic level. CT images provide valuable 3D microstructure information; however, this information is primarily available in the form of grayscale images. Extracting and accurately identifying individual fibers and interfiber contacts from these 3D images poses a significant challenge in the field of image processing, primarily due to the intricate and disordered nature of such structures [141, 142, 143, 144]. The complexity of these disordered structures requires ongoing efforts within the image processing community to develop effective methods for accessing and labeling the individual fibers and interfiber contacts within the 3D images. Accurately determining the statistical properties of fiber geometry, including length, diameter, and orientation distributions, is a significant challenge for computational simulations. This limitation hinders the feasibility of simulating fibrous materials with transversely isotropic properties. As a result, performing mechanical simulations at the fiber network level using real image data remains a challenging task, making it difficult to obtain precise and accurate results.

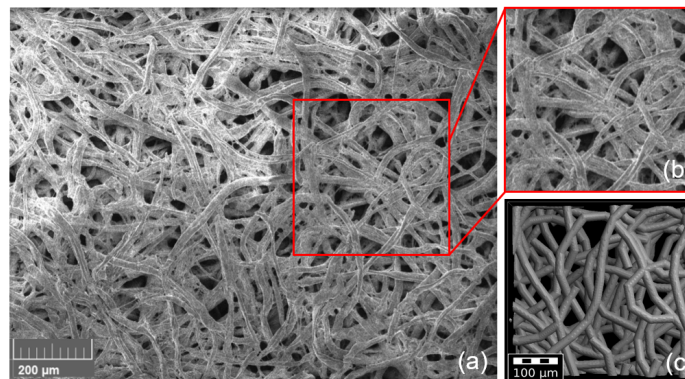


Figure 4.1: **a,b**: SEM image of Linters paper (Image courtesy by Julia Auernhammer) vs. **c**: Artificially generated fiber network sample.

At a smaller scale, such as the level of individual fiber/fiber intersections or interfiber

contacts, mechanical analysis has focused on studying the properties of these contacts, often referred to as surface charge. Several studies [139, 140, 145] have investigated the loading process by studying the mechanical behavior of these interfiber contacts. The stress-strain characteristics for single fiber/fiber intersections were analyzed using a combination of finite element simulations and experimental techniques. To address the challenges encountered at the fiber network level, computer-generated fiber network models have proven to be valuable tools for gaining insight into such issues. Synthetic fiber network models have been created using various deposition algorithms that incorporate statistical variations in fiber morphology and orientation. These models aim to establish representative systems for studying the mechanical behavior of fibrous materials [146, 147, 148, 149]. In order to explore the mechanical behavior of fiber networks, previous studies [150, 146, 147] have conducted extensive analyses. These analyses focused on various factors such as different grammages (basis weights), variations in fiber morphology distribution, and different loading conditions to understand the average mechanical response of the fiber network. At the scale of a paper sheet, or even at larger scales related to paper production, a stochastic continuum approach has been used by Mansour et al. [148] to analyze the mechanical response. This approach takes into account the inherent stochastic nature of the materials and provides insight into the overall mechanical behavior. At the nanoscale, where bridging nanofibrils and hydrogen bonding play an important role, studies have been conducted on cellulose nanopaper using cohesive zone-based crack-bridging models [151, 152]. The fracture toughness of the fiber network, taking into account its dependence on fiber orientation, has been studied in previous work [151] using randomly oriented and aligned nanofibers. In another study by Zhu et al. [153], the seemingly conflicting mechanical properties of strength and toughness were successfully improved simultaneously by changing the diameter of the constituent cellulose fibers from the microscale to the nanoscale. This finding strongly suggests the potential for optimizing mechanical properties through morphological changes in complex fibrous materials.

As motivated in the introduction section, the application of ML algorithms has found a wide presence in computational materials science and is applied to different material systems. In the context of establishing the relationship between microstructure and material properties, various machine learning models have been applied, including neural networks, support vector machines with different kernels, and other state-of-the-art ML models. These models have been used for image-based or feature-based microstructure analysis. The goal is to construct the MPR by using experimental data [154, 155, 156] or by using physics-based computer simulations [157, 158, 159]. Based on the available information, the application of ML techniques to fibrous paper materials has not been extensively explored in the existing literature. This is mainly due to the inherent

complexity and randomness of fibrous materials, which makes data-driven approaches highly promising for overcoming the challenges associated with establishing the MPR. However, in order to effectively utilize these powerful ML tools, it is crucial to have a clear understanding of the fundamental aspects of the paper material system and to identify relevant features and targets for analysis. To date, the majority of studies on paper materials, whether experimental or numerical, have focused primarily on investigating the influence of fiber morphology and bond strength on mechanical performance in terms of average values [160, 161, 162, 163, 146, 164]. However, due to the intrinsic characteristics of paper structures and the papermaking process, the sensitivity and variation of fiber network orientation, fiber morphology, contact area size, spatial contact, and contact orientation distribution, which are essential properties of the paper structure, have not been investigated. While various primary parameters (related to the definition of the fiber network) and secondary parameters (derived from the fiber network) have been studied separately, there is currently no approach available that sufficiently characterizes the mechanical response using sensitive parameters. As a result, the significance of these features in terms of mechanical performance remains unknown.

In this study, the aim is to simultaneously analyze both the primary parameters and the secondary contact properties based on the given geometry parameters. During the simulation process, individual fibers are assigned orientation-dependent transversely isotropic material properties and cohesive zone models are used to capture the fiber/fiber separation phenomenon. Subsequently, this thesis establishes statistical relationships between the input data and the mechanical response obtained from numerical simulations using ML techniques and regression analysis. A local sensitivity analysis is performed to evaluate the importance of each feature on the mechanical properties. Through this comprehensive analysis, the importance of the primary parameters and secondary contact properties in relation to the mechanical behavior of paper materials is revealed.

Outline of the chapter:

For the first, in Section 4.2.1, a database of 3D voxel-based fiber network microstructures is generated: The samples are generated using a random generator with a basic set of prescribed feature variations. The prescribed and derived feature descriptors characterizing the fiber network are briefly introduced. This database contains a sufficiently large amount of representative microstructure samples, where the number of samples is determined by evaluating the convergence behaviour of their corresponding mechanical responses. After generating the microstructure database, the input feature

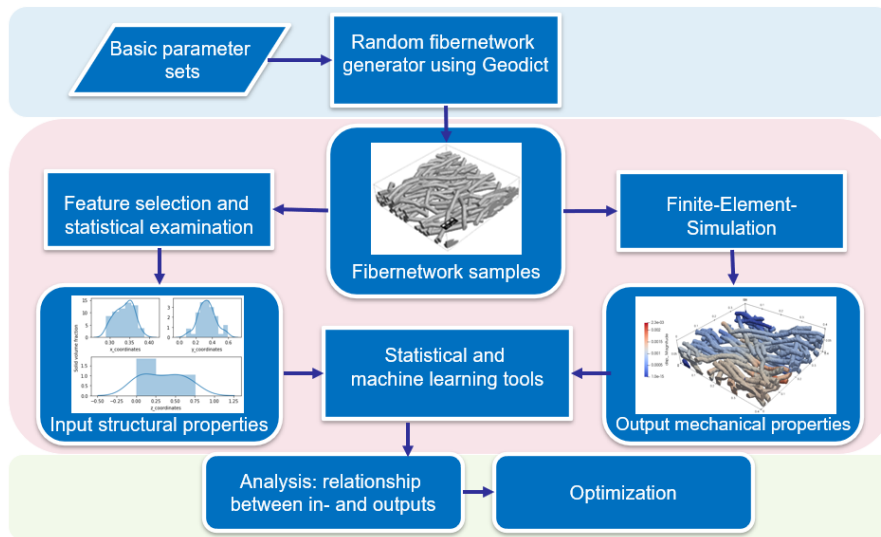


Figure 4.2: Workflow chart of the study

space is statistically evaluated. The evaluation involves examining the parameters and conducting feature selection using hierarchical clustering to obtain a reduced set of the input structural properties. In the subsequent Sec. 4.2.2, the mechanical behavior of the fiber network are simulated using the so-called cohesive zone model, which is able to capture the separation behaviour of fiber interconnects within the fiber network. The mechanical responses of these network samples are used as the corresponding output properties for each fiber network. In the ML-related Section 4.2.3, the database is randomly split into 70/30 percent sets and used to train ML-based surrogate models and to test the performance of the trained model. The ML-based model is capable of predicting the mechanical property of a new fiber network based on the defined input feature parameters. In the final section of the correlation analysis, the database and the selected ML model are used to evaluate the importance of the input parameters on each mechanical response. This analysis provides an insight into which of the parameters have a significant influence on the mechanical behavior of the fiber network. The workflow is outlined in Fig. 4.2.

4.2 Data generation and methodologies

4.2.1 Fiber network generation & feature determination

Fixed fiber network setting		Structural features as Input		
Domain size x, y	400 μm	<i>Basic structural distributions</i>		
Distribution in Orientation, Length, Diameter	Gaussian	Std orientation	[-90°, 90°]	F-O STD
		Std fiber length	[-100, 100] μm	F-L STD
		Std diameter	[-10, 10] μm	F-D STD
Grammage	30g/m ²	<i>Derived contact area features</i>		
Mean orientation	0° in x	Mean contact area size		C-AS Mean
Mean fiber length	300 μm	Std of contact area size		C-AS STD
Mean diameter	17 μm	Mean spatial projection of contacts (X,Y,Z)		C-ASD_X,Y,Z Mean
Mechanical target variables as Output		Std spatial projection of contacts (X,Y,Z)		C-ASD_X,Y,Z STD
Strain to failure		Mean contact area normal (X,Y,Z)		C-ANO_X,Y,Z Mean
Effective stiffness		Std contact area normal (X,Y,Z)		C-ANO_X,Y,Z STD
Maximal stress				

Table 4.1: Fiber network settings and ML input and output parameters

First, a fiber network random generator is developed using the commercial software package Geodict[®]. It is similar to the deposition procedure reported in the references [146, 147, 149]. Fibers are deposited for given Gaussian distributions of fiber orientation, fiber diameter, and fiber length. For simplicity, it is assumed that the cross sections of the fibers are circular and remain constant along the fiber axis. Fibers are generated in a flat plane, then dropped by gravitational forces, and finally deposited on the previously dropped fiber layers. The generation of a fiber network is stopped when the specified grammage is reached. The standard deviations of the Gaussian distributions are varied to investigate the structural sensitivity of the mechanical properties, in particular the failure strain, the maximal stress and the effective stiffness. In the generated samples, the variation over the basic parameters is chosen to be uniform. The fiber network settings and the associated input and output features are summarized in Table 4.1. The size of the fiber network and the fibers are chosen to be similar to [147]. The size dependence of the mechanical properties is a known problem and will be the subject of discussion later on. For the chosen sample size with variation of the structural features, the uncertainty in the random configurations is quantified by verifying the convergence of the considered mechanical properties vs. the realized number of samples [165, 50] in order to obtain reliable statistical results. As can be seen

in Fig. 4.3, the mean and standard deviation are evaluated for each target mechanical property (the calculation of mechanical properties will be introduced in the next section). With increasing number of samples, if the mean and standard deviation values of the considered mechanical property fall within the confidence interval (in this case, the 1% error bound is considered), then the number of samples is considered sufficient. In total, about 2000 samples were generated to ensure representativeness.

A few selected fiber network samples are demonstrated in Fig. 4.6a and Fig. 4.7. The numerical experiments are designed by changing the sensitivity or standard deviation rather than the mean. For example, the higher of the fiber network orientation, the greater the "disorder" of the fiber network, and vice versa. In other words, the sensitivity determines the alignment of the fiber network within the prescribed distribution. The distribution deviation of fiber diameter and length describes the degree of mixing of thick and thin, and long and short fibers. The mean value of the fiber network orientation is set constant to the machine direction due to the paper manufacturing process. The mean values of fiber diameter and length are also constant in the current study. They are usually descriptive for specific fiber materials. For example, eucalyptus fibers are generally shorter than linters and pine fibers [166]. The mean and standard deviation of the projection of the spatially distributed contact areas would indicate how the interfiber contacts are centered and whether they are homogeneously distributed throughout the fiber network. The size of the contact area and the normal orientation are important: A larger individual area has more total energy stored and is therefore more difficult to be fully separated. The orientation of the contact area determines whether and to what extent shear and normal separation will occur. Note that the tensile separation behavior of the interfiber contacts is different.

After collecting the database of input features, a feature selection step is performed. If the features are collinear or highly linearly dependent, permutating a feature variable will have little or no effect on model performance, since it can get the same information from a correlated one. For multicollinear features, hierarchical clustering is performed on the Spearman rank order correlations. This technique is based on [167] and implemented in the *SciPy* library [168]. The steps are as follows: First, a correlation heatmap is plotted and a threshold of 0.9 is manually set. After a visual inspection of the dendrogram, which groups the features into clusters, the features whose values are below 0.9 are selected, while keeping one of each feature from the clusters. The results comparing the features before and after the selection are shown in Fig. 4.4.

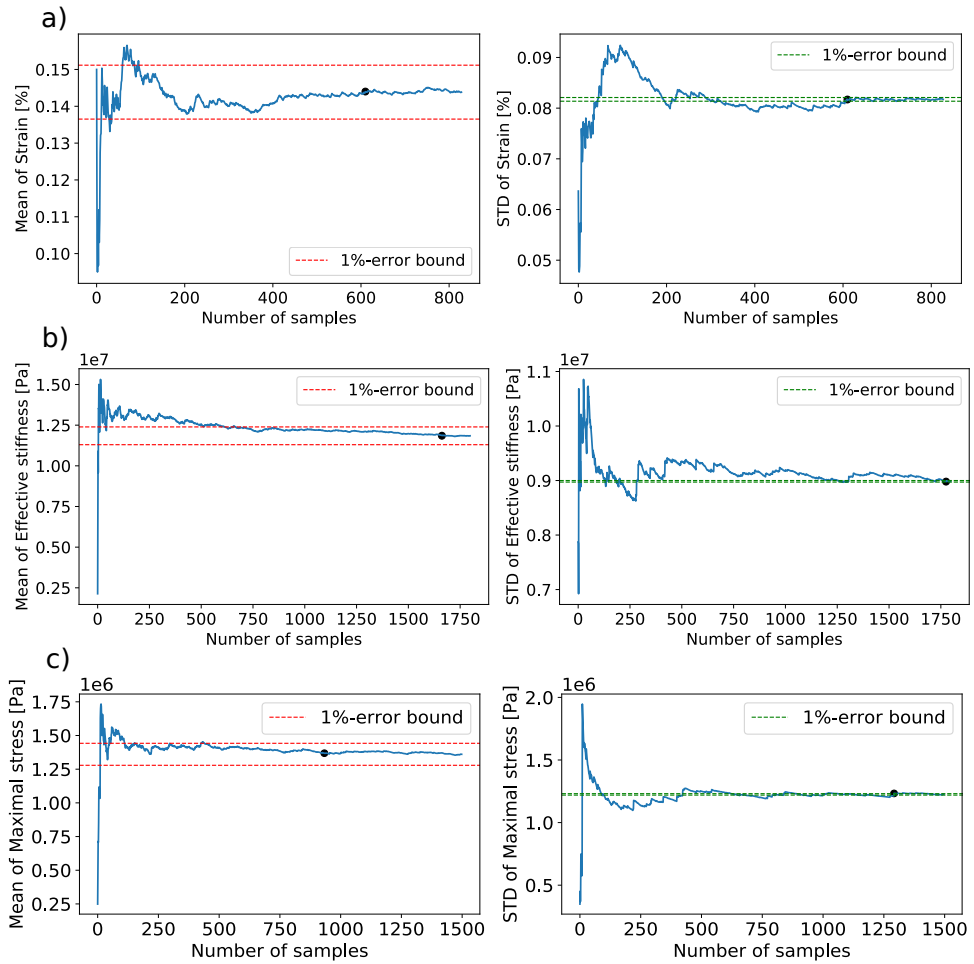


Figure 4.3: The property strain to failure converges with approx. 600 samples for mean and standard deviation (STD) values. Whereas for effective stiffness, 1662 samples for the mean value and 1775 for the STD value are needed. For maximal stress, 933 samples and 1292 samples are sufficient to draw a convergence, for the mean and STD, respectively.

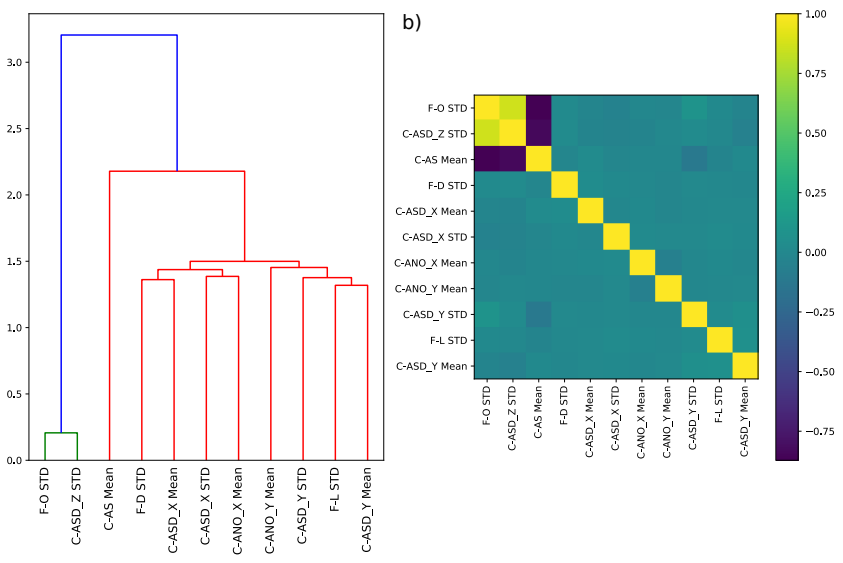
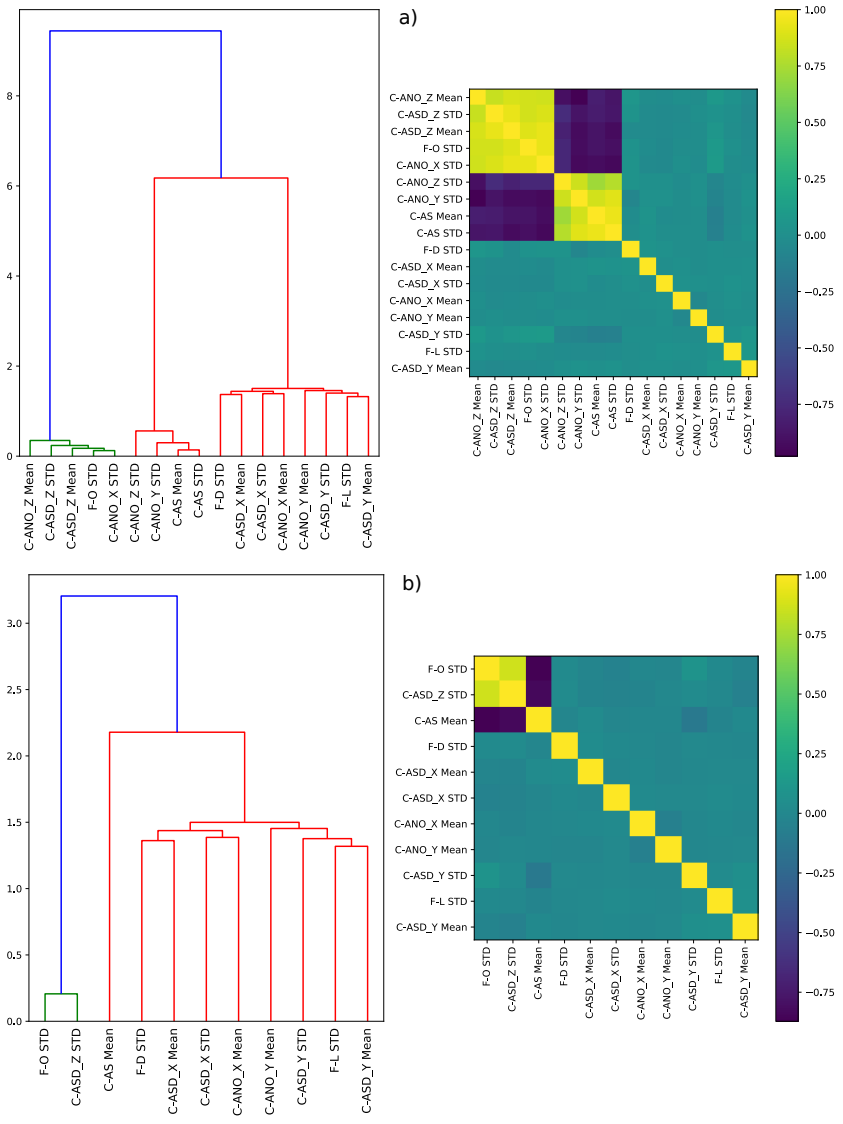


Figure 4.4: Dendrogram and heatmap **a**: Before feature selection; **b**: After feature selection.

4.2.2 Fiber network model & cohesive FE simulation

The governing equation describing the fiber network deformation with interfiber contact model can be characterized by the principle of virtual work in [169]:

$$\int_{\Omega} \delta \boldsymbol{\varepsilon} : \boldsymbol{\sigma} \, dV + \int_{\Gamma_{int}} \delta \boldsymbol{\Delta} \cdot \mathbf{T} \, dS = \int_{\Gamma} \delta \mathbf{u} \cdot \mathbf{T}_{ext} \, dS, \quad (4.1)$$

where $\delta \boldsymbol{\varepsilon}$, $\delta \mathbf{u}$, $\delta \boldsymbol{\Delta}$ are the virtual strain, displacement and separation along the interface, respectively. $\boldsymbol{\sigma}$ is the stress tensor, \mathbf{T} is the traction vector on interface Γ_{int} and \mathbf{T}_{ext} the external traction on outer boundary Γ . Equation 4.1 describes the energy balance between the strain energy or internal force in the domain Ω , plus the energy contributed by the interface elements and the energy done by external forces. Of note, the energy term by the interface extends the Equation 2.59 from the previous fundamental section.

Transversely isotropic fiber model For each fiber, the orientation tensor in the voxel-based geometry is assigned in the simulation step by use of Euler angles, such that the material property of each fiber was assured and assigned. We assumed that fibers are not pre-damaged and only interfiber contact failure occurs.

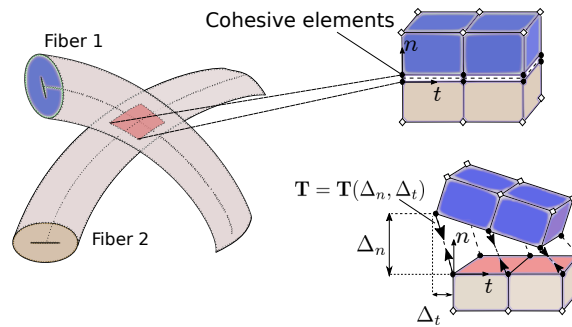


Figure 4.5: Illustration of the cohesive zone model in the interfiber contact.

Cohesive zone model for interfiber contact Similar to works [147, 170], a cohesive zone-based approach was utilized to characterize the fiber debonding behavior, see Fig. 4.5. In the current work, a non-potential based CZM [171] was exploited, which aims

Fiber material	Relation to E_1 [139]	Value	Interfiber material	Value
E_1	E_1	30 GPa [150]	δ_n^f	0.35 μm [150]
$E_2 = E_3$	$E_1/11$	2.72 GPa	δ_n^c	0.06 μm
$G_{12} = G_{13}$	$E_1/23$	1.30 GPa	σ_{max}	0.62MPa
G_{23}	$E_1/2(1 + \nu_{23})$	10.79 GPa	δ_t^f	1.56 μm [150]
$\nu_{12} = \nu_{13}$		0.022 [139]	δ_t^c	0.67 μm
ν_{23}		0.39 [139]	τ_{max}	1.94MPa

Table 4.2: Material parameters for the fiber network simulation

to give a proper behavior in mixed-mode loading scenario to avoid fiber penetration. The traction-separation laws are given as:

$$T_n(\Delta_n, \Delta_t) = \sigma_{max} \exp(1) \left(\frac{\Delta_n}{\delta_n^c} \right) \exp\left(\frac{-\Delta_n}{\delta_n^c} \right) \exp\left(-\frac{\Delta_t^2}{(\delta_t^c)^2} \right) \quad (4.2)$$

$$T_t(\Delta_n, \Delta_t) = \tau_{max} \sqrt{2 \exp(1)} \exp\left(\frac{\Delta_t}{\delta_t^c} \right) \exp\left(\frac{-\Delta_t}{\delta_t^c} \right) \exp\left(-\frac{\Delta_t^2}{(\delta_t^c)^2} \right), \quad (4.3)$$

where T_n, T_t are the traction components of \mathbf{T} in their normal and tangential loading state. Δ_n, Δ_t are the local contact displacement jump between two fibers. σ_{max} and τ_{max} are the maximal stresses in pure normal and pure shear separation. Respectively, δ_n^c and δ_t^c are the characteristic length values to the maximum stresses given by $\delta_n^c = \phi_n / (\sigma_{max} \exp(1))$ and $\delta_t^c = \phi_t / (\tau_{max} \sqrt{0.5 \exp(1)})$. Where ϕ_n, ϕ_t are the energy per surface for both separation modes. The damage variable is further defined as:

$$D = \frac{\Delta - \Delta_c}{\Delta^f - \Delta_c}, \quad (4.4)$$

where $\Delta = \sqrt{\Delta_n^2 + \Delta_t^2}$, $\Delta_c = \sqrt{(\delta_n^c)^2 + (\delta_t^c)^2}$ are the current mixed-mode separation and the separation corresponding to the maximum strength, and $\Delta^f = \sqrt{(\delta_n^f)^2 + (\delta_t^f)^2}$ the final separation at complete failure, respectively.

FE simulation and mechanical target properties To evaluate the mechanical behavior of the generated fiber networks, mechanical FE simulations were executed. A single fiber is a layered structure mainly consist of four different layers with different thickness in the cell wall. The elastic property of the whole composite at the microscale is rather complex and depends on factors as microfibril angle, lignin and hemicellulose content [172].

In this work, the fibers are assumed to be linearly elastic and transversely isotropic. The associated elastic constants are given in Table 4.2. The relationship of transversely isotropic material parameters depending on E_1 with uniform property related to the S2 layer is used [139], since this layer represents the main constituent of the fiber. Note that the constants are given with respect to the fiber orientation. For the mechanical simulations, they must be rotated to global coordinates for each fiber. A CZM [171] is applied to the contact areas between the fibers. It describes the fiber debonding behavior by two traction-separation laws, for the normal and tangential directions, respectively. The corresponding parameters can be found in Table 4.2. The final separation values are chosen from [150]. Reference [150] and [146] use a node based cohesive zone model for the fiber contact, in which cohesive forces were experimentally calibrated and provided. As the corresponding contact surface area is not provided, the cohesive strength are assumed relatively small in this work, such that interfiber contact failure can be assured. The critical separations are calculated by equating the cohesive energy terms $\phi_n = \frac{1}{2} \cdot \sigma_{max} \cdot \delta_n^f$, $\phi_t = \frac{1}{2} \cdot \tau_{max} \cdot \delta_t^f$ from the bilinear CZM [150] and exponential CZM in this work $\phi_n = \delta_n^c \cdot (\sigma_{max} \exp(1))$, $\phi_t = \delta_t^c \cdot (\tau_{max} \sqrt{0.5} \exp(1))$. As stated in [139], the CZM can capture the bridging mechanism between the interfaces, representing the effect of bridging fibrils on the large scale or chemical bonds on the smaller scales. For the FE simulation, the network sample is subjected to the displacement boundary with in-plane quasi-periodic condition in machine process direction, denoted as X-direction in Fig. 4.6 : $u_x = 0$ on the left boundary, $u_x = u$ on the right boundary and $u_y = 0$ on the upper and lower bound. The out-of-plane and other degrees of freedom on each side of the boundary are unspecified and free of stress.

Fig. 4.6a demonstrates the top-view and the cross-section perpendicular to the stretched longitudinal X-direction of one exemplary fiber network. Each fiber was labeled in different color and was discretized by 3D solid finite elements. Fig. 4.6b highlights the interfiber contact area in the undeformed state. The contact area was meshed with the planar cohesive zone elements. Fig. 4.6c highlights the deformed state. It can be seen that in the deformed state, certain contacts are fully separated, whereas others still remain bonded. The successive failure of contacts takes place and finally leads to the failure of the whole sample. The modeled fiber network sample involves approximately 0.6 to 1 million tetrahedron volume elements and around 40 thousand local cohesive 2D elements to resolve the contact area. A mesh size study was carried out in order to ensure the mesh independence of the results. The FE simulations were performed using the open source software MOOSE [80] on a high-performance computer with 24 cores for 5 - 10 hours per calculation depending on the actual microstructure.

From the FE results of the fiber network under the uniaxial stretch, the stress-strain

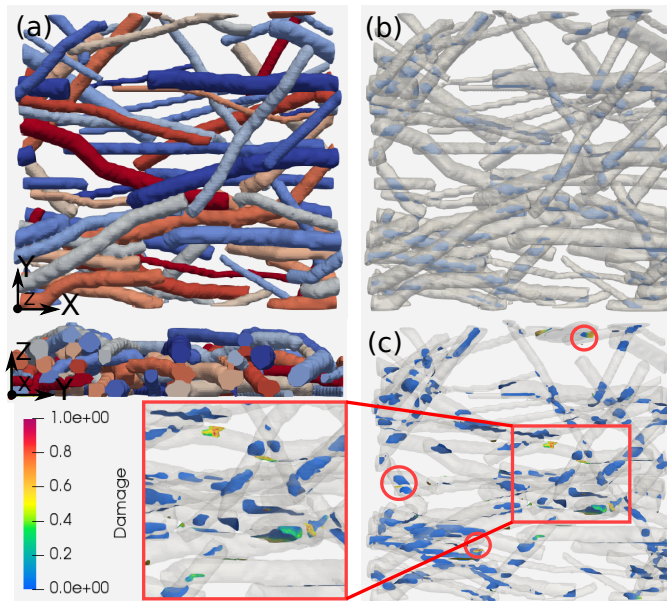


Figure 4.6: **a**: Coloured individual fibers; **b**: Interfiber contacts highlighted in undeformed state; **c**: Deformed state with damaged and separated fibers.

curve can be obtained. Fig. 4.7a shows the stress-strain curves of a number of fiber network samples, which share the same grammage of 30g/m^2 and the same fiber and contact properties. Apparently, the stress-strain curves vary largely with the fiber network structure. It emphasizes the importance to study the structure-property relationship and the potential of optimizing the property by manipulating the fiber network structure. Fig. 4.7b singles out two representative stress-strain behaviors with distinct features. Three mechanical target variables were considered: namely the failure strain, the maximal stress and the initial longitudinal effective stiffness, as denoted in the subfigure. The dashed blue line undergoes firstly a linear region and becomes flattened until it reaches the maximum stress. Afterwards, it enters the softening area and was assumed to be fully damaged when the sudden drop of stress occurs. The overall course is depressive. The dotted yellow curve is slightly progressive and almost linear until initiation of damage. In fact, the slight progressive stress-strain behaviors are more associated to fiber networks with small orientation deviation, see fiber network samples II and III in Fig. 4.7. These highly orientated fiber networks are considered damaged,

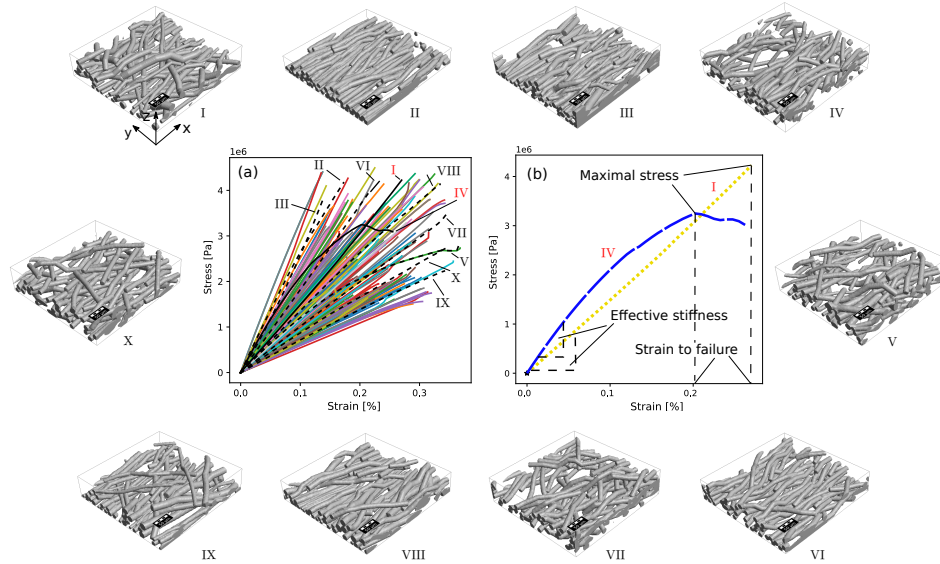


Figure 4.7: **a:** A fraction of stress-strain curves from the database; **b:** Illustration of the target properties for two typical curves in the database.

as the numerical structure gradient change becomes too large in the turning point. This leads to the termination of the numerical calculation. It must be mentioned that only the initiation of damage in the fiber network is considered. Analysis of the post-failure region in a more detailed fashion is out of current scope in the presented work. Further, the mechanical behavior of the fiber network has been reported to be strongly size-dependent. In general, loading in the sample length direction, average strength generally decreases [146]. After a critical length of 20 mm, strength follows a weakest-link scaling law form, as stated in [173]. The size dependency in stiffness mostly depends on the average length of the fibers and boundary conditions [150]. For the chosen sample size with variation in the structural features, the convergence of the considered mechanical properties w.r.t number of random samples is assured to obtain reliable statistical results, as can be found in the previous section. Worth mentioning, average area density or basis weight was fixed to be 30 g/m² in the presented study, making the mechanical behaviour calculated to be independent of the basis weight. This reasons the optimization of "performance-to-weight" ratio by finding critical design

parameters in paper materials. The basis weight can be changed to higher values, but the main limitation would arise in the computing capability of general direct simulation using 3D solid elements, as very fine mesh is needed to ensure the results to be mesh-independent. Current computing resources allow currently calculations of a sample size of 500x500 μm^2 with a maximal basis weight of 40 g/m^2 . Experiments [174] reported for a laboratory handsheets of size 10 x 4 mm^2 under tensile test, that increasing the basis weight did not change the tensile stiffness index and the tensile index significantly, but increased the strain to failure. Note that tensile index is defined as tensile strength [N/m] divided by basis weight [g/m^2] in experimental settings. Clearly, all else being the same, the tensile strength will be greater for paper of higher basis weight.

4.2.3 Shallow ML models & training

Before all fiber geometry features are input, multicollinear features are filtered out by hierarchical clustering on Spearman rank order correlations [167]. The initial 17 input parameters (3 basic parameters and 14 contact parameters) are reduced to 11 input parameters, see Fig. 4.4. Next, different types of machine learning models are trained using the fiber network feature parameters and the simulated mechanical features. Several shallow ML models are used: the linear regression model, the lasso model (a regularized linear model), K-nearest neighbor regression (a nonparametric model), and gradient boosting model (an ensemble model), representing different classes of machine learning models, are used to analyze the the dataset. These models are implemented in Python code using the open source machine learning library *scikit-learn* [64]. The default hyperparameter settings are retained for all models. It is worth noting that when training on one of the target mechanical features, the other two target features are included as complementary features in order to train the ML models accurately. Fig. 4.8a-c shows the ML predicted value as scatter points compared to the FE simulation results. The closer the data to the dotted diagonal line, the more accurate the predicted results are. The performance of the cross-validated ML models is evaluated by R^2 and Mean Absolute Error (MAE) on the test dataset, as defined:

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4.5)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4.6)$$

where \hat{y}_i is the predicted value, y_i is the test value, \bar{y} is the mean of the test set and n the number of samples, respectively. The variance in the data is adequately explained by the model, as can be seen in Fig. 4.8. For all the mechanical properties of interest, the Gradient Boosting (GB) method outperformed the other models, obtaining the best R-squared values of 0.88 for failure strain, 0.94 for effective stiffness, and 0.92 for maximum stress. The well-trained model can now calculate previously unseen feature combinations without running time-consuming FE simulations, and can accurately predict mechanical performance with the morphological features in a straightforward manner.

4.3 Correlation and sensitivity analysis

After evaluating the ML models, the features that have the most influence on each target variable are systematically investigated. In this work, the permutation importance based on the trained GB model and the Pearson Correlation Coefficient (PCC) are used. The PCC linearly quantifies the correlation between the input features and the targets. It's value ranges between -1 and 1, which indicates the extent to which two variables are positively or negatively linearly related. The formula is given as:

$$R_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4.7)$$

x_i, y_i are two features or outputs to be correlated and \bar{x}, \bar{y} their mean values. By using equation 4.7, one could rank the features according to their absolute value with respect to the target characteristics. Permutation importance is defined as the decrease in the score of an ML model when a single feature value is randomly shuffled [175]. This procedure breaks the relationship between the features and the targets, so the decrease in model score is an indication of how much the model depends on that feature. This technique benefits from being model agnostic and can be computed many times with different permutations of the feature [64]. The ML model used explores a non-linear relationship between the input features and the target variables. Thus, these two correlation analysis methods were used in a complementary manner to consistently reveal linear and higher order correlations.

In Fig. 4.9, both permutation importance and PCC analysis indicate that among all features, standard deviation in orientation (F-O STD), mean contact area size (C-AS Mean), and standard deviation of spatially distributed contact in thickness direction (C-ASD_Z STD) are the most important for the target variable failure strain. As for the

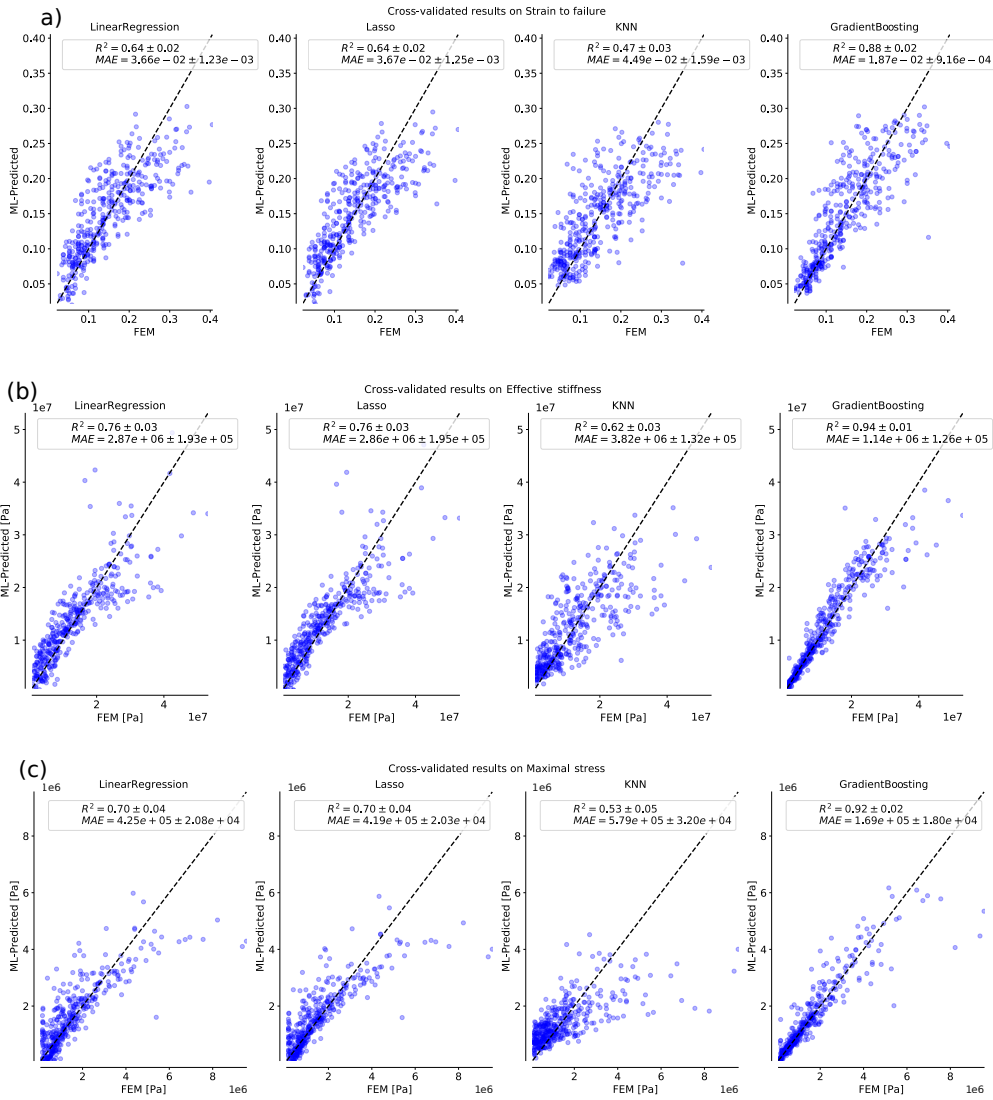


Figure 4.8: ML-prediction of different models as a function of the FE simulated mechanical features. a: Strain to failure; b: Effective stiffness; c: Maximal stress.

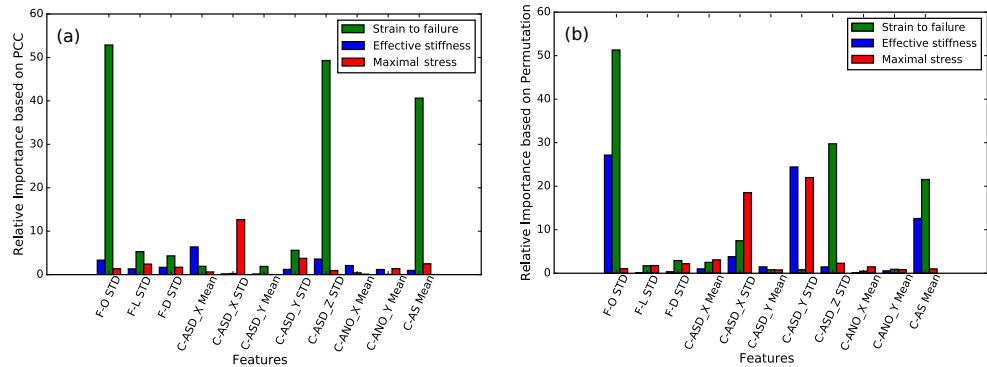


Figure 4.9: Barplot of relative importance based on **a**: Pearson correlation coefficient; **b**: Permutation.

effective stiffness, the nonlinear correlation based on permutation shows that F-O STD, C-ASD_Y STD, and C-AS Mean are more dominant. The orientation effect on the deformability and stiffness of the fiber network is well known and has been summarized in the review paper [176]. Regarding the maximum stress class, C-ASD_X STD appears to be the most influential structural feature according to the linear correlation of PCC, while permutation determined the feature C-ASD_Y STD to be the most important for the same objective. Both properties are contact properties and describe the homogeneity of the contact distributions. The fiber length deviation (F-L STD) and the fiber diameter deviation (F-D STD) turn out to have less influence on all three target variables among the considered parameters. It should be noted that the current study differs fundamentally from the literature [177, 178, 163, 160], as they reported the effect of fiber geometry on the tensile test of paper materials with different type of fibers, (e.g. fibers from softwood or hardwood) in an experimental context. The fiber types were characterized by their mean values of fiber length, diameter, wall thickness, and strength. It is reasonable that longer fibers (mean length) will influence the mechanical properties, especially the corresponding sheet strength [179, 163, 146]. However, the degree of variation in the fiber structures has generally not been considered as an inherent fiber characteristic due to the nature of the fibers and the papermaking process. It implies that for property optimization, controlling the overall orientation of the of the fiber network may be more effective than mixing fibers of different lengths or diameters. It also shows that all the mean values of the spatial contact distributions have less influence on all the targets. However, the standard deviation of the spatial contact (C-ASD_STD) features

dominate the contributions, indicating that the key to higher mechanical performance is a homogeneous distribution of contacts in the fiber network. This makes sense, since a homogeneous distribution of contacts is likely to prevent or suppress local failures. In general, both methods showed good agreement in determining the important features. After checking the significance of the features, the qualitative trend can be drawn by a simple linear fit curve through the data points, and it shows how the change in each structural feature variable affects each target variable, see Fig. 4.10. The target variables are shown in the figure as a function of F-O STD and C-AS Mean, since these two features are the most significant, along with C-ASD_X,Y,Z STD. As mentioned above, they provide information about the overall degree of homogeneity of the contact distribution, which also implies the homogeneity of the fiber network structure itself. The more uniform the spatial contact distribution becomes, the higher the mechanical performance can be obtained. The data analysis in Fig. 4.10 shows that a disordered structure, indicated by a high F-O STD value, tends to have a large failure strain. This can be inferred from the simple linear curve through the data points. Fiber networks with direction, indicated by small F-O STD values, have high effective stiffness but small failure strain. A small mean contact area size is preferred for large strain. On the other hand, for high effective stiffness, a large mean contact area is desirable. However, there is no clear tendency for maximum stress or ultimate strength. In fact, it implies a strong interplay between several factors, see Fig. 4.10 f,i. Three extreme cases for the highest target performance based on the current database are shown in Fig. 4.10 j,k,l. Similarity in fiber reinforced plastic materials is shown in Fig. 4.10 k. For specific or desired load bearing situations such as uniaxial tension, optimal mechanical performance can be achieved by orienting the fibers in the direction of the load. Nevertheless, for cases such as multi-axial loading, the shown structure may not be the optimal one. However, strain, strength, and stiffness are properties that compete to some extent in many materials. Materials with higher ultimate strains usually have lower stiffness. Strong and stiff materials tend to be brittle and have lower strain at failure. The data analysis and the trained model confirm that paper materials fall into a similar scheme. Thus, structural optimization for multiple target variables simultaneously is non-trivial. In other words, optimizing the structure features so that the targets are well balanced or have an optimal combination of desired features requires training the models to fit multiple targets simultaneously. As shown in [133], the task of fitting multiple targets simultaneously is challenging, and leading to low performance scores, thus complicating the inverse structure design and optimization. Finally, statistical analysis and machine learning seemed to be two essential elements to address such problems. While machine learning provided more practical regression models for prediction, statistical tests were useful for feature selection, multicollinearity detection, and providing statistical significance of the regression models.

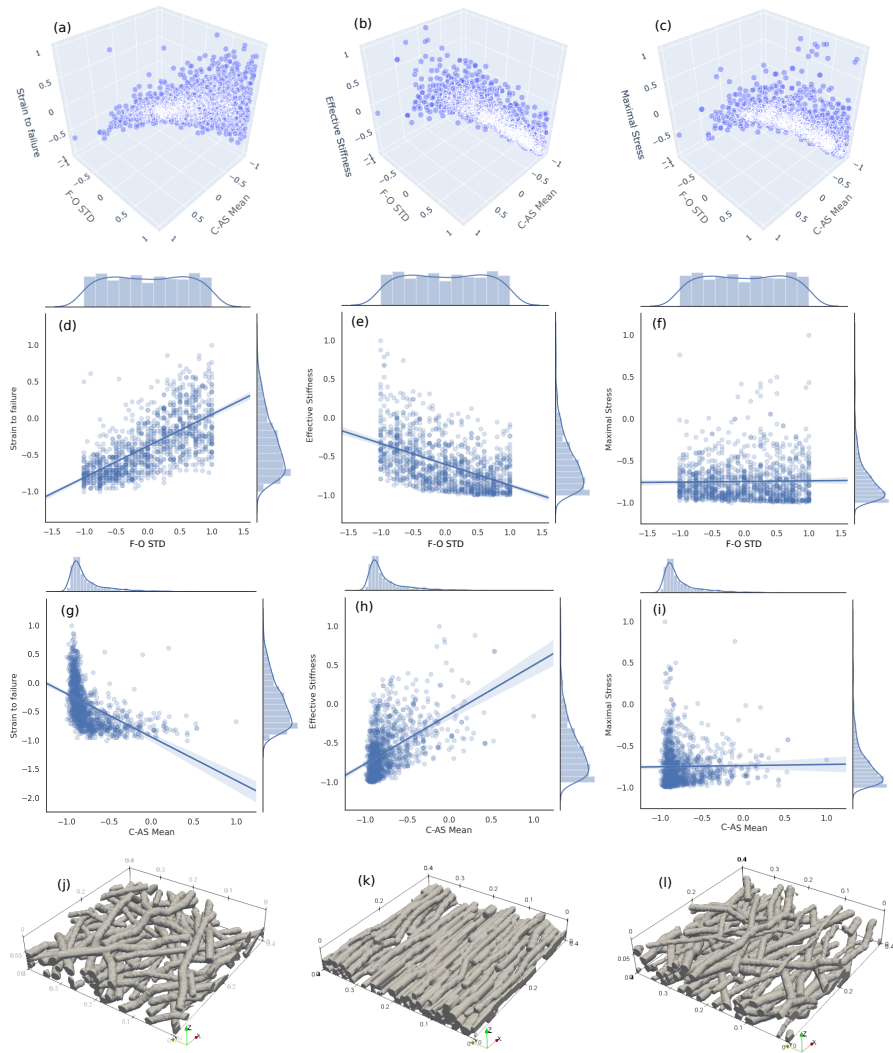
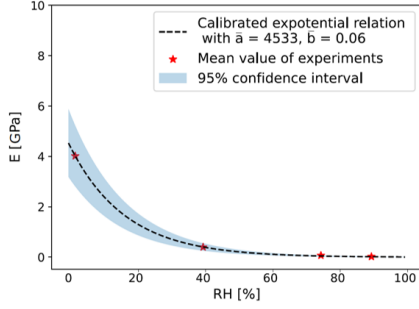
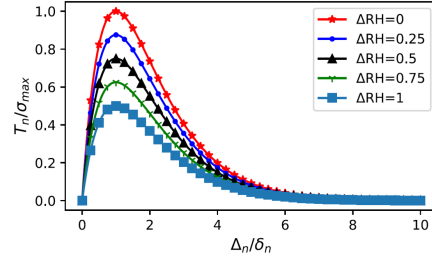


Figure 4.10: Scatter plots with linear fit and data distribution (The values are normalized for easier access). **d-f**: Depending on the Std. of fiber network orientation; **g-i**: Depending on the mean contact area size; **j-l**: Structures with highest target values with respect to strain, stiffness and stress.

4.4 Perspectives on data extension



(a) Calibrated exponential law showing the decay of the fiber elastic modulus w.r.t the RH levels.



(b) Normalized decay curve of traction-separation law w.r.t to RH levels with $K = 0.5$, exemplary shown for the normal cohesive behavior.

Figure 4.11: Modelling of humidity effects at single fiber and interfiber level [135].

Paper materials typically exhibit hydrophilic properties unless they are subjected to chemical and mechanical processing treatments. The presence of moisture, in the form of relative humidity, has a significant effect on several aspects of paper properties. These include the elasticity of the fibers, the behavior of the interfiber joints, and the failure mechanism. In the context of the current chapter, which mainly focuses on the microstructural features as the input data set, the discussion can be equally extended to the variation of material parameters, in particular with respect to the influence of moisture on mechanical properties. Our previous work [135, 134] has shown that at the single fiber level, exposure to different humidity levels leads to hygroscopic expansion and dimensional instability of the fibers. This morphological change, in turn, induces significant variations in the mechanical behavior of the fiber network. It also leads to a humidity induced exponential decay of the elastic property of a single fiber, see Eq. 4.8, a weakening of the interfiber strength, see Eq. 4.9 as shown in Fig. 4.11 and a change of the damage pattern in the fiber network Fig 4.12. The exponential decrease in elasticity of the fibers can be modeled as follows:

$$E(RH) = \bar{a} \cdot e^{-\bar{b}RH}, \quad (4.8)$$

where \bar{a} can be understood as E_{max} , which represents the averaged modulus at 0% RH. \bar{b} determines the degree of elastic decaying. The weakening of the interfiber strength can be viewed as the extension of previous traction-separation law in Eq. 4.2, considering

the change of RH as the variable:

$$\mathbf{T}(\Delta RH) = \mathbf{T}^{dry} (1 - K(\Delta RH)), \quad (4.9)$$

where K is a softening parameter and describes the degree of decrease and the bound of the allowable cohesive strength, and \mathbf{T}^{dry} the vector of previous traction-separation components in normal and tangential directions.

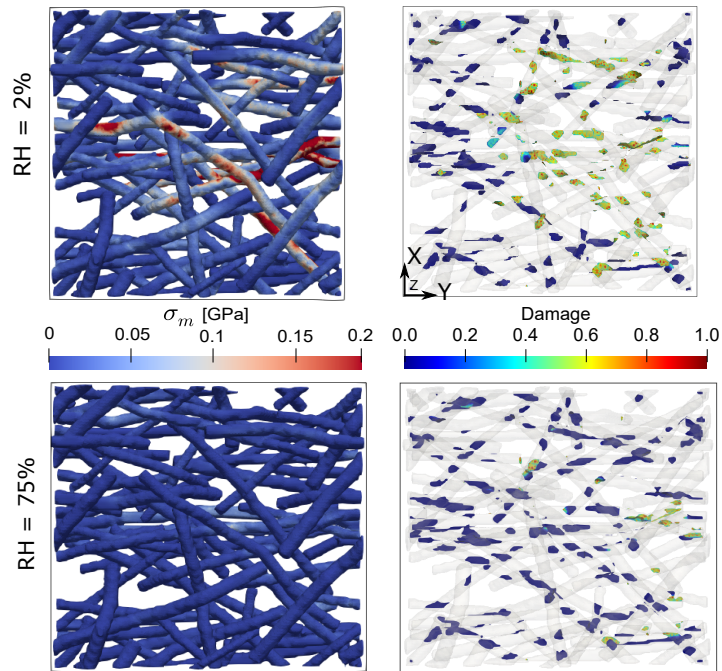


Figure 4.12: The von Mises stress map and damage pattern show variations at different relative humidities (RH). High stress levels are observed in the dry state, which are reduced as humidity increases. The presence of moisture allows greater flexibility in fiber deformation and loosening of strong interfiber bonds. Consequently, this supports greater deformations and reduces the stress intensity in the fibers, but pronounced interfiber failure and thus a reduction in the strength of the fiber network and the overall paper structure [135].

Although the fiber network examples presented in this chapter have been carefully

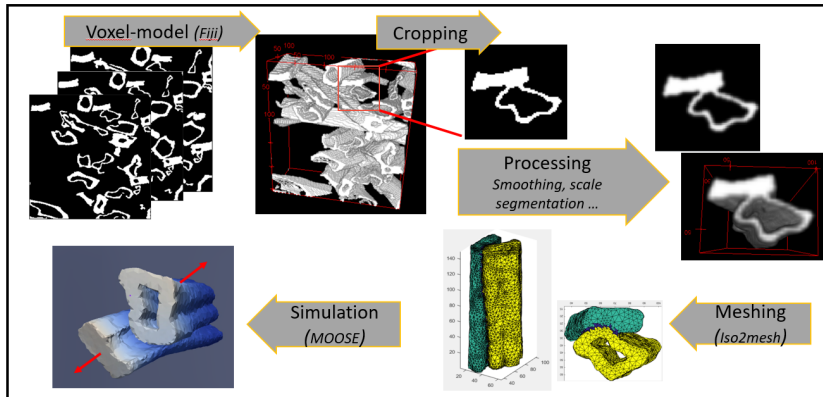


Figure 4.13: The simulation of a real CT dataset involves several steps such as cropping, smoothing, segmentation of fibers in the region of interest and subsequent 3D meshing of fibers, which are in contact with each other, and the assignment of cohesive zone elements up to boundary marking for the final mechanical FE simulation.

generated with geometrically plausible parameters, it is important to reemphasize that several simplifications have been made. For example, the modeled fibers do not fully represent the complex nature of real cellulose fibers, which typically have lumens and non-uniform cross-sectional shapes along their length. To address this limitation, a first attempt was made to perform FE simulations using micro-CT data of fiber networks. Fig. 4.13 shows a workflow diagram incorporating real image data. The process involves segmentation, preprocessing, and meshing of two fibers in contact based on the raw binary data. By incorporating cohesive zone elements to model interfiber contacts and using the methods developed in this study, simulations can be performed to investigate the mechanical behavior of real fibers with lumens at various levels, including the single fiber, fiber-fiber contact, and fiber network levels. The use of real data in these simulations allows for a more realistic interpretation of the computational studies, facilitates correlation analysis, and provides insight into the deformation behavior of the fiber network, ultimately allowing for further optimization of the fiber network design.

5 End-to-end Image-based Correlation Learning in Dual-phase Steel

This chapter presents the outcomes of the project during my time at the SDL Material Design Lab at the National High Performance Computing Center for Computational Engineering Science (NHR4CES), a collaboration between TU Darmstadt and RWTH Aachen University. Together with my co-worker from the Institute of Physical Metallurgy and Metal Physics, RWTH Aachen, under the supervision of Prof. Sandra Korte-Kerzel, the mechanical behavior of DP steel was investigated. During the project, I was provided with raw SEM images and their corresponding segmented phase maps. These materials served as the basis for conducting micromechanical modeling and FE simulations. The results presented in this chapter were initially reported in our work [180] and will be discussed below. My primary contributions in this research endeavor include performing microstructure analysis, developing an image-based meshing tool that seamlessly converts the segmented SEM phase maps into FE meshes for modeling and simulation purposes, and establishing the workflow for data generation and the implementation of the ML model. The analysis presented in this chapter stems from these contributions.

5.1 Dual-phase steel material and mechanical properties

DP steels are a special type of high performance steels consisting of a soft ferrite and a strong martensite phase in their microstructure. These steels are widely used in automotive chassis due to their exceptional mechanical properties. While the overall macroscopic mechanical properties of these steels are governed by the specific phases and defects within the material, predicting the local mechanical responses, such as stress concentration, strain distribution, and damage localization, is challenging due to the complex morphological variations and phase accumulation present in these materials [181]. Traditionally, numerical simulations can be performed for each individual microstructure sample based on its constitutive behavior. However, the large variation in microstructural features requires numerous simulations, resulting in high computational cost. In

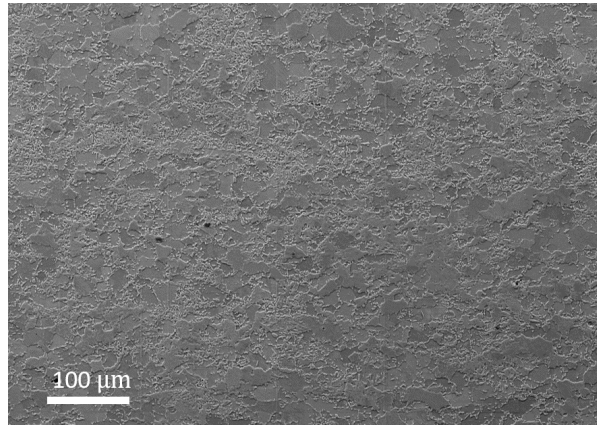


Figure 5.1: A high-resolution panoramic SEM graph showing the microstructural heterogeneity of the DP-steel. Image courtesy by Setareh Medghalchi.

addition, the transferability of individual simulation results to unseen microstructural data is limited, making it difficult to establish a direct microstructure-property model and its relationships. Therefore, the application of ML techniques serves as a valuable tool to address this issue. In particular, deep learning models have enabled the exploration of unseen microstructures and facilitated the development of MPR. Surrogate ML models have been developed based on user-defined or statistical microstructure feature descriptors [182, 133]. An alternative approach is to use the microstructure image itself as the input and using mechanical stress/strain field distributions throughout the microstructure as the output. This approach eliminates the need for simplifications and accounts for voids, defects, phase details, and spatial distributions in the most accurate manner. The prediction of field quantities allows the calculation of averaged quantities such as homogenized strain/stress and damage extent, including uncertainty measures. The extracted field properties can be further used to model macroscopic behavior and perform multiscale simulations. With sufficient and representative datasets, an end-to-end ML model offers high knowledge transferability. It can serve as a user-friendly tool for material scientists to understand and predict the mechanical behavior of the studied material system without requiring expertise in computational domains. In pursuit of these advantages, this chapter outlines the necessary steps to develop such a tool and provides technical details for its implementation.

Outline of the chapter:

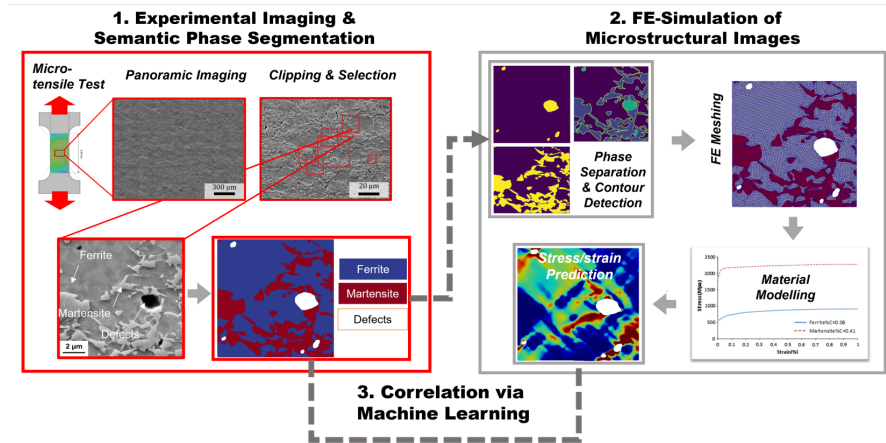


Figure 5.2: Data generation process flow. **Left:** Microstructural SEM micrographs are obtained from a DP steel sample via panoramic imaging and further post-processing steps. The micrographs are semantically segmented using machine learning based techniques [183]; **Right:** The segmented phase images are contour-meshed for elastic-plastic FE simulations under tensile loading to obtain the local von Mises stress and equivalent plastic strain fields. The segmented phase images (input data) and simulated stress/strain fields (output data) are correlated by an advanced CNN model.

This chapter is organized as follows: First, the input and output datasets for the machine learning model are presented, which are divided into two parts as shown in Figure 5.2. The first part focuses on the experimental input data and the segmentation of micrographs from the panoramic image of the DP steel sample. The second part presents detailed procedures of the image-to-mesh conversion step and the subsequent elastic-plastic FE simulations, which serve as the output of the ML model. Then, the machine learning model, specifically the U-net model, is introduced along with its training details. The trained models allow an end-to-end prediction of the stress/strain maps and are tested on another unseen dataset for evaluation. This section concludes with further discussion and interesting data extension prospects.

5.2 Data generation and methodologies

5.2.1 Experimental imaging & microstructure phase segmentation

The commercial DP800 Dual-Phase steel (ThyssenKrupp Steel Europe AG) was investigated as a model system in the present study. As indicated by its marketing label, this type of DP steel has a minimum tensile strength of 800 MPa. The test specimen was cut from a 1.5 mm thick plate in the shape of a dog's bone and elongated uniaxially in the rolling direction. An Electron Discharge Machine (EDM) cuts the specimen up to its necking area. High-resolution panoramic SEM images were obtained from 900x900 μm metallographically prepared areas using secondary electrons of field emission scanning electron microscopes (LEO 1530; Carl Zeiss Microscopy GmbH, Jena, Germany) and TESCAN CLARA, detailed experimental instructions can be found in [183]. The spatial resolution of the images obtained was 32.5 nm/pixel. The high resolution electron microscopy images of the microstructure were segmented using deep learning based convolutional neural networks [183] developed with *Tensorflow 2.0.0*. For this purpose, the panoramic images were cropped to smaller window sizes of 512x512 pixels. They were then manually labeled using the Image Labeler application of Matlab 2019 on randomly selected images. Three known phases in the microstructure, ferrite, martensite and defect phases, were segmented. Since the manual labeling of the images was labor-intensive, image augmentation algorithms were applied to the training data to increase the number of images and the accuracy of the network. The network was trained on a computer cluster at RWTH using an Nvidia V100 GPU card. The resulting network was used to segment and stitch the microstructure images.

5.2.2 Image-based meshing & elastic-plastic FE simulation

Image-based meshing To perform the FE simulation, the segmented microstructural phase images were first meshed using a customized workflow based on the open-source meshing tool Gmsh. Here, the defects (white phase) and the martensite phases (red phase) were separated and segmented by thresholding, which were considered as foreground objects according to their pixel values. The ferrite phase (blue phase) was considered as the background color. Then, contour detection was performed on the foreground objects using the *findContours* method with the OpenCV library in Python. These contour points were converted to Gmsh number formats and line loops were created for these contour objects. A large ferrite contour was added to the contours to serve as a background mesh for the material domain. Finally, material assignment was performed for different phases, skipping mesh generation only for defect phase contours

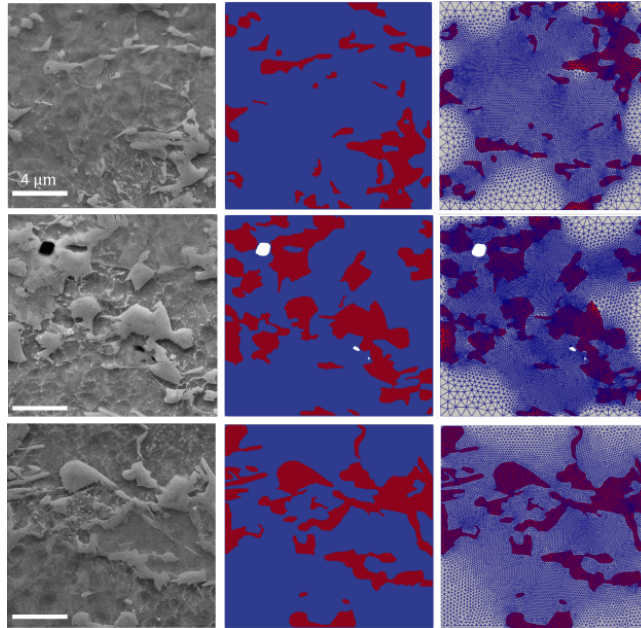


Figure 5.3: Exemplary microstructures and their corresponding segmentation and FE meshes.

to create "holes" in the simulation mesh domain. As shown for a few examples in Fig. 5.3, this procedure has been automated to process a large database of microstructure images randomly selected from the panoramic SEM image shown in Fig. 5.1.

Elastic-plastic FE model After setting up the image-based mesh, small strain FE simulations with uniaxial tensile boundary conditions were performed using the open-source FE code MOOSE. Extension of the model to finite deformation theory will be considered in future work. The governing equations for a classical rate-independent plasticity model are briefly summarized below [184]:

$$\operatorname{div} \boldsymbol{\sigma} = 0 \quad \text{in } \Omega, \quad \mathbf{u} = \mathbf{u}_0 \quad \text{on } \Gamma_u, \quad \mathbf{t} = \mathbf{t}_0 \quad \text{on } \Gamma_t. \quad (5.1)$$

where $\boldsymbol{\sigma}$ is the stress tensor and Ω is the image domain. The boundary conditions, either as displacement \mathbf{u} or traction \mathbf{t} boundary conditions, can be applied in general cases. The total strain tensor $\boldsymbol{\varepsilon}$ is defined as the symmetric part of the gradient of displacements within the small strain theory and classically decomposed by the split into the elastic

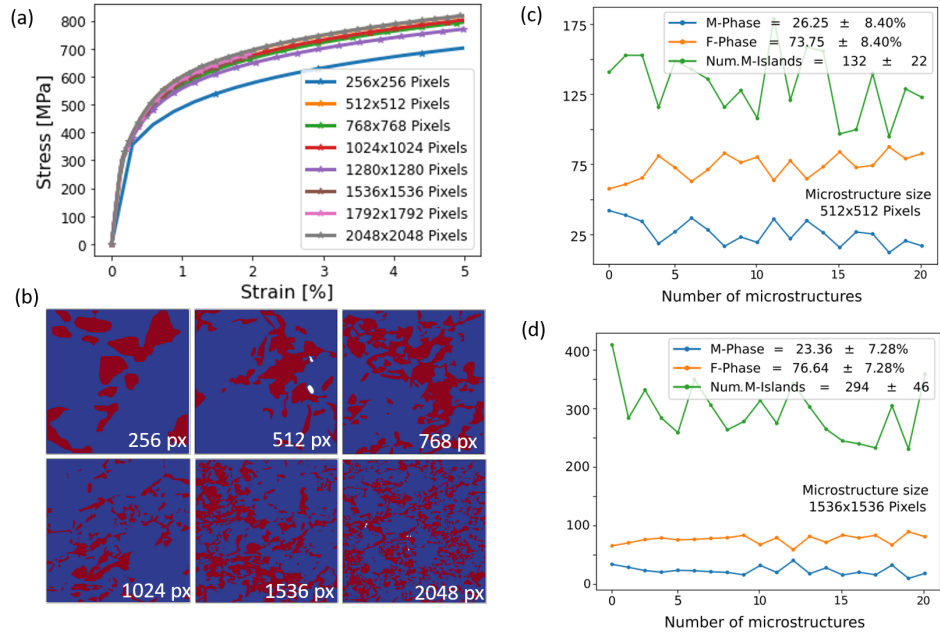


Figure 5.4: Influence of microstructure size on structural characteristics and mechanical strain-strain behaviour. **a**: Stress-strain curves for 8 different sizes of randomly cropped microstructures; the stress-strain response is similar, with a small deviation for the smallest size of 256x256 pixels. As the size increases, the stress-strain response converges accordingly; **b**: Microstructures shown for increasing size. Different levels of detail can be observed. As the domain size increases, the microstructure becomes more homogeneous; **c-d**: Statistical evaluation (Mean and Std.) of each phase fraction and number of martensite islands for the domain size of 512x512 and 1536x1536 pixels, for 20 random microstructure samples. The statistical variation in the phase fraction of each phase for the larger microstructure set in **d** is comparatively small compared to **c** and the size can be considered representative. Note that the phase fraction of possible defects is neglected in this case since it is less than 0.2 % of the total panorama image. The resolution of each pixel is 32.5 nm.

and the plastic part: ε_e and ε_p :

$$\varepsilon = \frac{1}{2}[\nabla \mathbf{u} + \nabla \mathbf{u}^T], \quad \varepsilon = \varepsilon_e + \varepsilon_p, \quad \boldsymbol{\sigma} = \mathbf{C} : [\varepsilon - \varepsilon_p], \quad (5.2)$$

with the corresponding constitutive equation and \mathbf{C} the fourth-order elasticity tensor. Isotropic hardening conditions were assumed in this case. For common metals, when the material is loaded beyond its yielding point, J_2 -plasticity models can be utilized, which is characterized by the second invariant of the deviatoric stress tensor $J_2 = \frac{1}{2} \mathbf{s} : \mathbf{s}$, with $\mathbf{s} = \boldsymbol{\sigma} - \frac{1}{3} \text{tr}(\boldsymbol{\sigma}) \mathbf{I}$. The von Mises stress is defined in the same regard as $\sigma_{v,M} = \sqrt{3J_2}$. The plastic flow rule, yield function, and evolution of the hardening parameter are given as follows:

$$\dot{\varepsilon}_p = \gamma \frac{\partial f}{\partial \boldsymbol{\sigma}}, \quad f = \|\mathbf{s}\| - \sqrt{\frac{2}{3}} K(\alpha), \quad \dot{\alpha} = \sqrt{\frac{2}{3}} \gamma, \quad (5.3)$$

with $\gamma = \|\dot{\varepsilon}_p\|$ and $\alpha = \int_0^t \dot{\alpha} dt$ commonly denoted as the equivalent plastic strain (PEEQ), that describes isotropic hardening of the von Mises yield surface. Further, the Kuhn-Tucker conditions fulfill the loading-unloading: $\gamma \geq 0$, $f \leq 0$, and $\gamma f = 0$. The hardening function K follows a linear hardening model for the martensite phase and a power law hardening model for the ferrite phase:

$$K(\alpha)^F = \sigma_F + \bar{K}_F \alpha^n, \quad K(\alpha)^M = \sigma_M + \bar{K}_M \alpha. \quad (5.4)$$

Here, σ_F and σ_M denote the yield strengths and \bar{K}_F and \bar{K}_M the hardening coefficients for the ferrite and martensite phases, respectively. n denotes the hardening exponent for the ferrite phase. Due to the lack of material data of individual phases, the material parameters from [185] for an exemplary DP980 steel, with $\sigma_F = 425$ MPa, $\bar{K}_F = 940$ MPa, $n = 0.2$ for the ferrite phase, $\sigma_M = 1180$ MPa and $\bar{K}_M = 1740$ MPa for the martensite phase is adapted, respectively. In particular, the exact material properties of the DP800 steel specimen may differ from these literature values. However, since the goal of this chapter is to train an end-to-end CNN model to predict the field quantities, it is considered appropriate to use these material parameter sets to perform FE simulations that generate output quantities for model training as illustrative examples. Determination of the material parameters of the DP800 specimen will be performed in future work. The calculation of the trial stress and the updating of the plastic strain and stress increment were solved numerically using the radial return methods implemented in the simulation software MOOSE.

FE simulation and target field maps In this section, the simulation results of the PEEQ field for an exemplary microstructure domain of the considered dual-phase steel

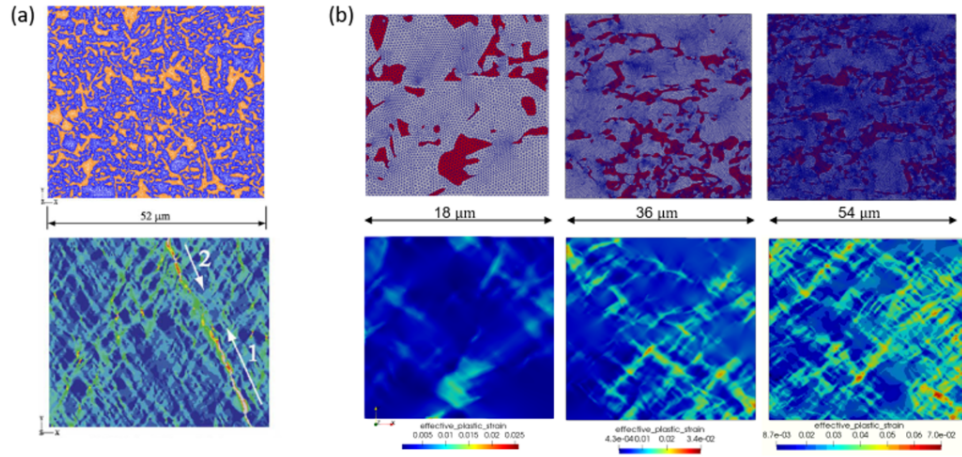


Figure 5.5: Qualitative comparison of PEEQ field with literature results. **a**: PEEQ map [185], adapted with permission; **b**: Our PEEQ map for selected size of microstructures.

are presented. Fig. 5.5a shows a similar simulation example under tensile loading by Sun et al. [185]. As indicated by the arrows, the failure mechanism follows a shear band type failure and propagates from the lower boundary until it meets another failure band propagating from the upper boundary. This observation can be confirmed by our simulations for similar microstructure size, as can be seen by the zigzag lines indicating the shear failure mode. More interestingly, the hotspots appear to be initially located in the largest martensite island and then propagate to the next larger martensite island connected by the ferrite phases. This is presumably due to the rather stiff nature of the martensite phase, which generates high levels of stress and thus leads to greater plastic strain. The direction of propagation of the fracture can be diverted as it is strongly controlled by microstructural details such as the location and size of the martensite island. Although a larger microstructure size may be more representative of the global failure mode, microstructure images with a window size of 256×256 were chosen and randomly selected for ML training of stress/strain field predictors. This is explained as follows: The size of the microstructure samples should provide a good trade-off between computational cost and microstructure variation that ensures representativeness. To validate this, a set of microstructures with different sizes was randomly selected. The set of microstructures was subjected to the same mechanical simulations and the stress-strain behavior was recorded and compared, see Fig. 5.4a. As can be seen, the mechanical

response is similar, with a small deviation for the smallest size of 256x256 pixels. As the size increases, the stress-strain response converges accordingly. Thus, in order to capture the randomness and variation in the microstructure with respect to the number of martensite phases and their morphologies contained in this image size, this particular input size was chosen to generate the database for the subsequent ML model. It should be noted that with a larger image size, the computational cost of resolving the corresponding details increases disproportionately for both meshing and FE calculations.

5.2.3 U-net model & training

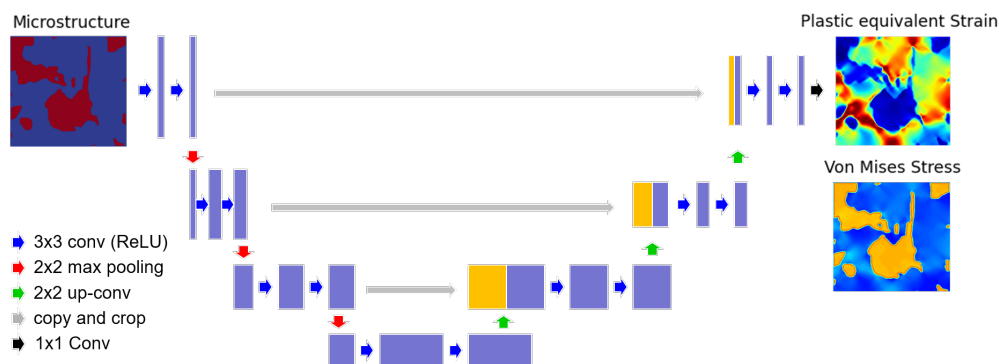


Figure 5.6: A schematic sketch for the U-net structure: Input data is the segmented micrograph phase maps, while the output is either the plastic equivalent strain map or the von Mises stress map.

The U-net is a convolutional network architecture initially aimed at fast and accurate image segmentation. The choice of U-net over other neural networks lies in its ability to extract local features through convolutional operations, since microstructural morphology and stress/strain field undoubtedly have important localized attributes. For the end-to-end field mapping in this work, the U-net model was used in its unmodified form as proposed by [186]. The U-net model, see Fig. 5.6, consists of a contraction path that learns to encode the features and follows the typical structure of a convolutional network, but with repeated application of convolution, ReLU activation, and max pooling operations. The expansion path decodes the contraction path by upsampling the feature map with further convolutions and concatenating the previously clipped feature map from the coding path. The reader is referred to [186] for more details. The U-net model

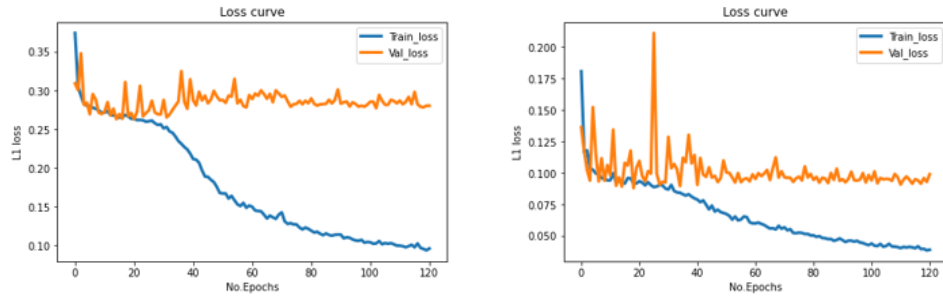


Figure 5.7: Learning curves for the train and the validation losses. **Left:** Model training for the PEEQ map; **Right:** Model training von Mises stress map. A clear trend of over-fitting can be found for both model training curves after approx. 30 epochs.

is implemented in our code using the deep learning framework *Pytorch* [187]. Further improvement and modification of the network hyperparameters, as similarly done in [9, 188], are sought in further studies. For the training of the field predictor, the input data of the model are the segmented microstructures with three phases, namely the ferrite, martensite and defect phases of only one channel, with pixel intensity values ranging from 0 to 255. The output data of the model is either the PEEQ or the von Mises stress field image obtained from the elastic-plastic FE simulations at the last loading step. The field images were mapped to their undeformed configuration to ensure image size compatibility. In total, 1200 samples of segmented phase maps with their corresponding PEEQ and von Mises maps were generated and subsequently split into a 75/20/5 % ratio, making 900 samples for training, 250 samples for validation, and 50 samples for testing, respectively. Two models were separately trained for the strain and the stress field for 120 epochs with 4 images per batch. The learning rate was set to $5e-4$. The training steps used the Adam optimizer with default settings from *Pytorch* and the L1 loss. Model weights were saved every 10 epochs. Each epoch took about 5 to 6 minutes, and the training and validation took 6 to 8 hours running on 4 Nvidia A/V100 GPUs. The loss curves for both models are shown in Fig. 5.7. As can be seen from the learning curves, there is a trend of overfitting for both model training curves after about 30 epochs. Therefore, we used the model weights at 30 epochs to test the unseen data set. The possible origin of this overfitting event will be discussed in the later section.

5.3 End-to-end microstructure-to-field prediction and correlation analysis

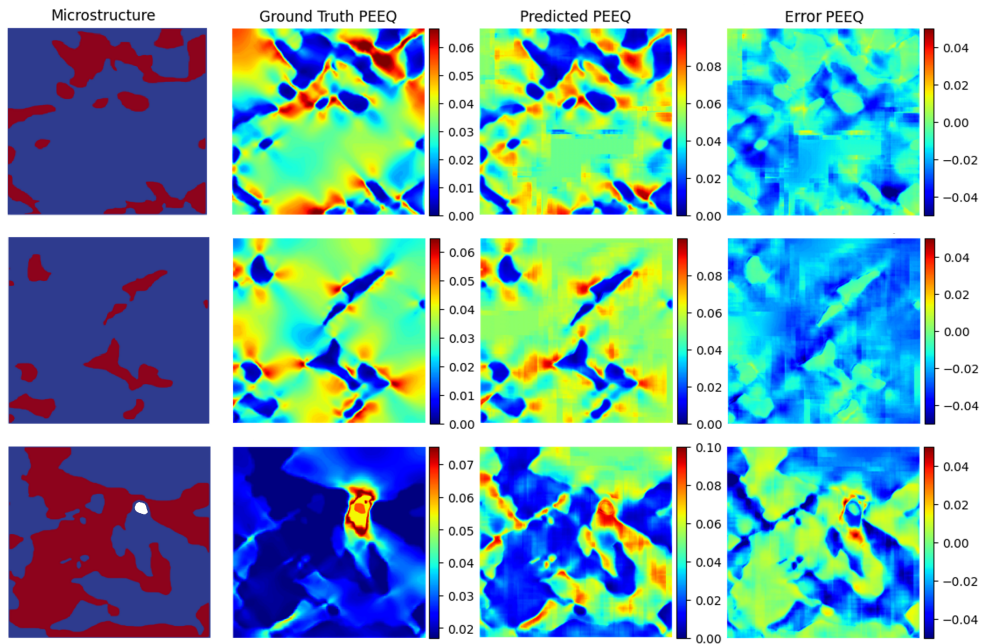


Figure 5.8: PEEQ prediction results by the U-net model for exemplary microstructures. **First column:** Exemplary microstructures; **Second column:** Ground truth PEEQ field calculated by FE simulation; **Third column:** Predicted field results by the U-net model; **Fourth column:** The pixel-wise error between the ground truth and the model prediction.

The model prediction results for exemplary unseen microstructure images are shown in Fig. 5.8, 5.9 for the PEEQ maps and von Mises stress maps, along with their ground truth data from FE simulation and the pixel-wise error, respectively. Here, the first column shows exemplary microstructures, the second column shows the ground truth data, and the third column shows the field prediction by the model and the pixel-wise error in the last column. The rows show the results for different microstructures. To measure the overall prediction performance, the Percentage Mean Absolute Error

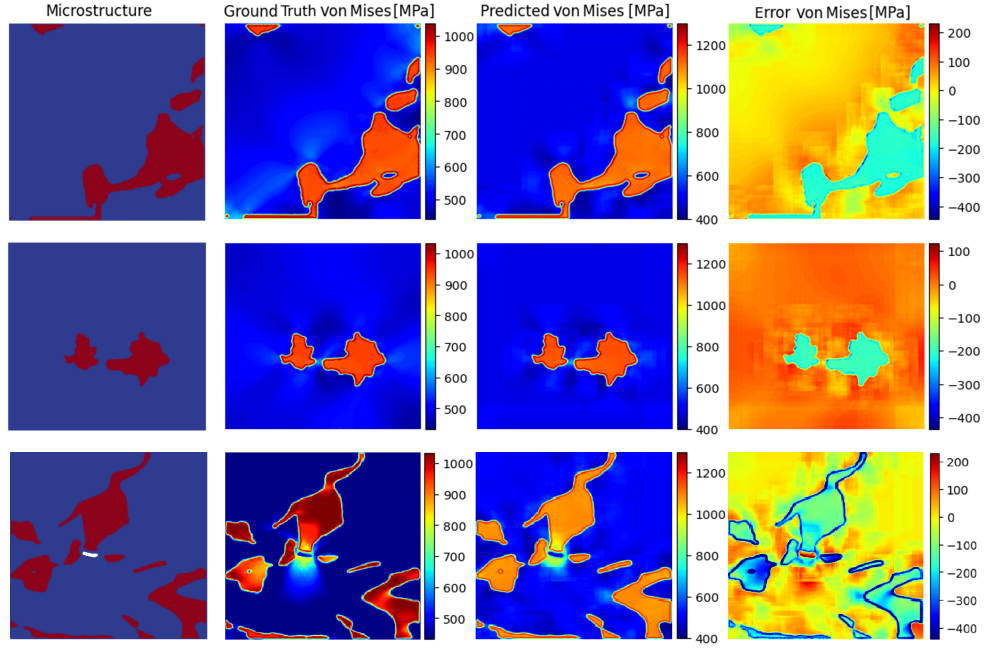


Figure 5.9: Von Mises stress prediction results by the U-net model for exemplary microstructures. **First column:** Exemplary microstructures; **Second column:** Ground truth stress field calculated by FE simulation; **Third column:** Predicted field results by the U-net model; **Fourth column:** The pixel-wise error between ground truth and the model prediction.

(PMAE) is introduced as in [189]:

$$\text{PMAE} = \frac{\text{MAE}}{\text{Max}(Y) - \text{Min}(Y)} \times 100\%, \quad \text{MAE} = \frac{1}{N} \sum |y_{gt}^i - y_{predict}^i| \quad (5.5)$$

with $\text{Max}(Y)$, $\text{Min}(Y)$ denote the maximum and minimum pixel value of the image Y , and $|y_{gt}^i - y_{predict}^i|$ the absolute error between the i^{th} ground truth pixel value and the model-predicted pixel value of the PEEQ or the von Mises stress field image. N is the number of pixels within one considered image. As can be seen in Fig. 5.8, for the first two microstructure samples with no defect phase, the plastic strain regions are primarily located in the ferrite phase due to its lower stiffness and earlier plastification, mainly around the sharp edges of the martensite phase. The ML prediction qualitatively captures these plastic strain localization patterns compared to the PEEQ ground truth

FE data. However, the error column still shows a small deviation in some sharp regions around the martensite phase and the pixelated nature of the model prediction. As for the third example microstructure with a slight defect inclusion, large deviations were observed for the model prediction compared to the ground truth data. Apparently, the model missed the plastic strain concentrations around the defect inclusion and predicted relatively homogeneous strain distributions following the previous pattern, resulting in larger overall deviations. The PMAE of the PEEQ field predictor is about $14 \pm 3.5 \%$ for the entire test data set of 50 microstructure samples. Similar results can be found for the von Mises stress prediction in Fig. 5.9. While the trained machine learning model predicts acceptable results for other similar microstructures without defects, there are increasing deviations and can be found for a microstructure with a similar defect inclusion. Nevertheless, the stress predictor performs slightly better and captures the stress concentration in the vicinity of the defect compared to the strain predictor. An even larger deviation at the martensite boundaries can be seen in the error map. The score for the von Mises stress predictor is $\text{PMAE} = 11 \pm 4.2 \%$. It is assumed that the lower prediction performance for the microstructure with defect phases of both strain/stress predictors is due to the significant data imbalance within the phase distributions between ferrite, martensite and defect pixels in the training data set. The defect phase currently represents less than 0.2 % of the total pixels from the entire panoramic image. Due to the uniform sampling and cropping of the panoramic image data, the total defect phase pixels correspond to this approximate fraction in the training dataset.

5.4 Perspectives on data extension

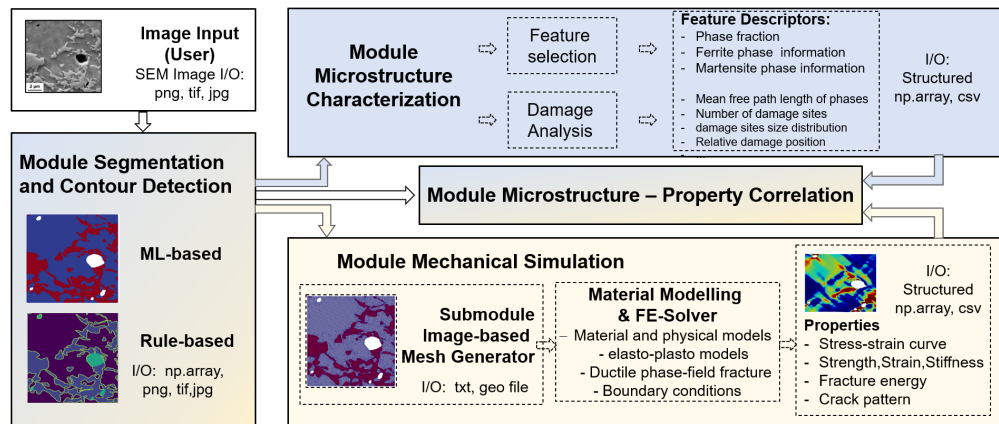


Figure 5.10: Flowchart of the web-based application under current development. This application should enable easy usage of models and techniques developed in this work. At the start, the user can upload a sample of DP-steel microstructure for segmentation. After the segmentation step, the user can select different features for characterization or mechanical stress/strain analysis. Alternatively, since the trained ML model is based on a specific simulation model under fixed loading conditions, as in this work, the user can choose to generate the FE mesh for his own simulation model. After obtaining the features and mechanical properties, the data can be visualized for a straight-forward correlation analysis.

As mentioned in the previous section, the image dataset has a significant class imbalance, especially with respect to the defect phase. As a result, the trained model shows poor prediction performance when it comes to microstructures containing defects. An effective approach to address this issue is to provide the algorithm with additional image data specifically focused on the defect phases. This can help to correct the problem of overfitting. The defect phase data set can be generated in a variety of ways, such as by creating a specific window around the defect phase within the panoramic image. In addition, data enhancement techniques such as rotation, flipping, and others can be applied to enhance the generated defect phase microstructures. Another technique is the creation of synthetic defect phases using parametric formulations, similar to the creation of artificial particle shapes in the case of battery electrode materials. While increasing the amount of data is often a straightforward solution, another avenue

to explore is manipulating the loss functions. Adaptive loss balancing techniques, such as those mentioned in the references [190, 191], can be explored to balance the training effect when a batch of images containing defects is fed into the model. This provides an alternative means of addressing the challenge posed by class imbalance in the dataset. Other aspects to be considered include the incorporation of variations in physical boundary conditions and material parameters. Furthermore, there is the potential to extend the current physical models to include crystal plasticity models, as discussed in the works of Roters et al. [192] and Raabe et al. [193]. This extension would allow for a more refined representation of the microstructure, taking into account individual grains with their unique orientations and material properties. To facilitate the use of the developed techniques, it is intended to develop a web-based application similar to the one described in Chapter 3, which can be accessed by the wider community. A schematic representation of the logic of the application is shown in Figure 5.10. Note that this application is currently under development and will be released as part of future work.

6 Conclusion and Outlook

Finally, a brief summary of the main chapters is given below, along with a discussion of potential limitations and suggestions for future improvements.

On the nanowire structure of the battery cathode material:

- *Segmentation tool development.* A deep learning model was developed specifically for instance segmentation of nanowire structures in spectromicroscopy images. The model was trained using synthetic datasets generated to emulate experimental microscopy data, overcoming data acquisition and labelling issues which are extremely expensive in terms of time and cost. Despite variations in spatial resolution, particle dispersion densities, and contrast generation among the microscopy data, the model achieved high prediction accuracy, demonstrating its versatility in segmentation tasks. Furthermore, this model was deployed as a web-based and interactive segmentation tool, accessible through https://share.streamlit.io/linbinbin92/V205_app/V205_app.py and the associated training data are publicly available for further use and exploration [194]
- *Geometry-lithiation correlation analysis* The developed image analysis tool enables extraction of morphological feature descriptors and statistical measurements for quantitative correlation analysis. It reveals strong connection between particle geometry and lithiation pattern. It demonstrates confirming other experimental observations that scaling down particle size increases perimeter/area ratio and reduces the diffusion path length and promotes the lithium-ion transport in cathode material. Eccentricity or high aspect-ratio introduces negative effect such as phase-coexistence, leading to increased misfit-strain and thus continues stress buildup. Further, it is emphasized that the ratio between perimeter and particle size and (surface area-to-volume surface) should be considered as an important descriptor rather than treating them separately.

Despite the success demonstrated here, there are limitations that must be mentioned. The first is inherent to the quality and resolution of the spectromicroscopy images and

the trade-off between image resolution and chemical information obtained. The second is the presence of instrumental noise, which biases the segmentation results. Another aspect is the extension of the correlation analysis to a 3D setting. Although the 3D information of the particles is implicitly available through the optical density values, extracting or reconstructing the 3D geometry information to derive descriptors is a major challenge. Currently, the interpretation of the correlation results can only be heuristically transformed from 2D to 3D, lacking robust and solid experimental or computational evidence. Further, possible crystal structure defects, impurities inside the particle were not considered, which may cause the variation in correlation observed in the results section earlier. Furthermore, possible defects in the crystal structure, impurities inside the particle were not considered, which may cause the variation in the correlation observed in the results above. The above mentioned issues can be partly addressed by computational studies, for example by performing chemomechanical simulations on 3D reconstructed particles enabled by CT [130, 115]. This has great potential for incorporating possible defects in the particle structure. Since the experimental particle data usually exhibit a certain type of shape with little to medium variation, synthetically generated particles by advanced modeling software can fill the gap, using sophisticated physical models to simultaneously investigate the MPR, as done for the fiber network structure for paper materials in the second main chapter. This also facilitates simplified access to the microstructural descriptors and the corresponding property of interest, allowing accurate, rapid construction of MPR models using machine learning techniques and their subsequent sensitivity analysis and optimization.

On the fibrous structure of the paper material:

- *Fiber network generation and FE simulation.* The sensitivity parameters were used as descriptors of morphological features to assess the randomness of paper fiber network and a large dataset of fiber networks was generated. A physically-based cohesive zone model in finite element simulations was used to generate mechanical properties, which includes the description of interfiber separation process during fibrous network deformation.
- *Feature engineering, ML model training and correlation analysis.* Morphological properties related to fiber contacts were extracted from generated fiber network samples and aggregated with sensitivity parameters defining the fiber network as input parameters for training ML models. A dimension reduction technique was then applied to this set of inputs to reduce co-linear features in the dataset. Together with the computed mechanical output properties, a surrogate model based

on gradient boosting was obtained, allowing a rapid screening of the possible input parameter space for mechanical property prediction. A feature importance study based on the ML model shows that "disorder", represented by the standard deviation of the fiber network orientation, and mean contact area size are the most influential factors on fracture strain and effective stiffness, while homogeneous distribution of contact areas has the greatest impact on fracture strength in our simulation setting, respectively. Finally, the code, input data, and output data from this study are archived and publicly available for further exploration, see https://github.com/linbinbin92/data_driven_microstructure_paper_material

While physically based FE simulation combined with synthetic fiber network generation has many benefits, its limitations should also be noted. These include the aforementioned simplification of fiber properties such as fiber lumen, prismatic cross sections, and fiber deposition techniques that do not exactly resemble the reality found in the fiber network after the papermaking process. In addition, the material models used in the current work are rather homogenized models that do not fully capture the behavior of the multilayer wall structure and the microfibril dependence of the material behavior of the fiber. The constitutive model for interfiber separation has a strong influence on the overall fiber network mechanics, an accurate determination is extremely difficult due to the inherent variation in fiber/fiber contacts and limited experimental access. The dataset presented is also limited to the simulation boundary conditions. Although the presented work already considers 3D fiber networks, future work is sought in the incorporation of image-based fiber structures that realistically represent the structures captured in the paper materials with large variations in their microstructural characteristics, that can be then further enrich the dataset for more precise prediction of MPR models. From the material modeling perspective, advanced model consideration, finite deformation, plastification, damage of single fibers should be developed to obtain more accurate output properties. In terms of MPR, image-based feature input as performed in the next main chapter might help to overcome the derivation of necessary input descriptors in the presence of complex fiber, interfiber contact distributions, nevertheless, computational cost arises with 3D image input in model training, model evaluation increases significantly.

On the phase structure of the DP steel:

- *Segmented microstructure map to FE simulation.* Based on the segmented micrograph of DP steel microstructures, a fast, efficient image to FE mesh procedure was developed. This technique extracts the contour points of different phases present in the microstructure and transformed the data structure for subsequent meshing.

This procedure was repeated to generate a database of various microstructures randomly selected from the paratomic image. The obtained microstructure meshes were then transferred to FE simulation jointly with elastic-plastic material models to simulate the mechanical behaviour of DP steel microstructure tensile loading. The output mechanical properties are stress/strain field maps, resolving a high fidelity view of local concentration of stresses/strains depending on the actual microstructure.

- *CNN-based surrogate MPR model.* Two CNN models were trained based on the FE simulation field data, and allows an end-to-end workflow from experimental SEM micrographs to segmented phase maps, and then to stress and strain field predictions. The models demonstrated good agreement with test data and performed well on unseen microstructural datasets. They can capture the complexities of the microstructure, including voids, defects, different phases, and spatial distributions, without the need for simplifications or descriptor derivations. This enabled the prediction of detailed field quantities, which can be used to calculate averaged quantities such as homogenized strain/stress and damage extent, including possible uncertainty measures for MPR model. Further, the developed approach allows non-experts in computational domains to gain insights into material behavior in a straightforward manner, for instance, by just providing an experimental microstructure image, enabling easier transferability of knowledge in an end-to-end manner.

Possible drawbacks can be seen in the same way as in the previous chapter, as the simulation models are based on certain simplifying assumptions. While the improvement of the physical models can be further refined, such as extending the current model to include crystal plasticity, future attempts can be made to extend the current framework to a 3D setting, as the damage is likely to appear in the microstructure and propagate into the third dimension under various loading scenarios. As stated earlier, there is also a notable imbalance in the image dataset, particularly concerning the defect phase. As a result, the predictive performance of the model is compromised when dealing with microstructures that contain defects. To effectively address this issue, a viable solution is to supplement the algorithm with additional image data that specifically highlights the defect phases. In terms of MPR, descriptor-based representation methods can be explored, since image-based representation suffers from the interpretability of microstructure features. Future perspectives can be explored to develop neural network structures that jointly aggregate both representation methods to facilitate better interpretation, representation of detailed microstructure and property information, and subsequent property prediction and material design.

Conclusive comments: In conclusion, this research emphasizes the significant potential of employing machine learning models, as well as physically sound simulation data to characterize microstructures, extract microstructural features, predict material responses, and establish MPR for improved material design. The ML models enable the automation of data processing and inference, making it possible to handle large volumes of data and achieve fast predictions (e.g. End-to-end field prediction). They capture not only the underlying physics present in experimental and simulated data but also establish the overall correlation between microstructure features or images and properties such as stress and strain field responses. The adaptability and learning capacity of these models are substantial, as they continuously improve and make better predictions with the availability of new data. However, it is crucial to realize that machine learning models heavily rely on high-quality data for training and testing. Acquiring such data in material science contexts can be demanding, time-consuming, and expensive. If the available data is incomplete, biased, or unbalanced, it can negatively impact the performance and accuracy of the models. For instance, models may overfit the training data, becoming too specialized and failing to generalize well. Challenges such as predicting stress/strain fields around defects or dealing with segmentation inaccuracies in the presence of instrumental noise in microscopy data can be limiting factors. Additionally, the computational cost associated with certain models, particularly image-based models trained with CNNs is very high, requiring several hours of training on modern GPU cards.

The data used for training ML models can come from experimental sources, simulation data, or ideally, a combination of both, as they can complement each other. On the one hand, physics-based simulation data incorporates fundamental laws and principles governing the system, providing a deep understanding of the underlying physical processes, such as in the case of modeling fiber networks or DP structures. This understanding can aid in the development of accurate and interpretable ML models. The trained models have the potential to generalize well to different scenarios and conditions, as the relationships learned between input features and output predictions can be transferable to experimental situations, allowing predictions beyond the specific simulation setup. Moreover, physics-based simulations enable controlled experiments with well-defined parameters and conditions. This controlled environment facilitates the generation of consistent and reproducible training data for ML models. Simulations can also complement experimental data by providing insights into scenarios that may be inaccessible or costly to explore experimentally. On the other hand, there are drawbacks to using physics-based simulation data. Simulations often involve simplifications and assumptions to model complex phenomena. ML models trained solely on simulation data may inherit these simplifications and assumptions, limiting their applicability to real-

world scenarios that deviate from the simulated conditions. Additionally, simulations introduce uncertainties and inaccuracies due to approximations, numerical methods, and model parameters. These uncertainties can propagate to the ML models, impacting their predictive capabilities and reliability. Furthermore, both ML models and physical simulations are computationally intensive and time-consuming. Training ML models on large-scale simulation data can be challenging due to the computational complexity and memory requirements involved.

As different representations of microstructure were employed, each comes with their strengths and weakness. Descriptor-based representations offer high compatibility with a wide range of machine learning models, particularly shallow models, making them applicable in various contexts. They also facilitate efficient mathematical computations, which is advantageous when dealing with large-scale datasets and complex models. Furthermore, descriptor-based representations provide flexibility for feature engineering, allowing domain-specific knowledge and transformations to enhance model performance. However, one potential limitation is the loss of information that can occur during the conversion of complex datasets into descriptor values. This process may not capture the full intricacies of the original data and heavily relies on the expertise of domain experts. On the other hand, image-based representations are designed at capturing visual information, making them well-suited for tasks such as microstructure characterization, segmentation and particle detection. Pre-trained convolutional neural networks offer powerful image representations learned from extensive datasets as shown for the nanowire segmentation, enabling efficient transfer learning and training. Moreover, image representations can be visually interpreted, providing valuable insights into the model's understanding of visual patterns and features. However, a common challenge with image representations is their high dimensionality, requiring substantial computational resources for training and inference as mentioned. Data augmentation techniques are often employed to increase the diversity and size of the training dataset, but this process can be time-consuming, as observed in particle segmentation tasks. Additionally, image representations may lack translation invariance, meaning slight shifts or rotations in the input image can affect the model's predictions, which may necessitate additional techniques to address this limitation.

Overall, machine learning and using physically sound simulation data presents great potential for various applications in the present MPR studies. However, it is crucial to consider the strength and weakness in order to effectively leverage its benefits while addressing its limitations. Ultimately, the choice of representation depends on the specific characteristics of the data and the requirements of the problem at hand. Trade-offs between interpretability, computational requirements, and compatibility with machine learning algorithms should be carefully considered when selecting a representation.

Bibliography

- [1] Lauri Himanen et al. “Data-driven materials science: status, challenges, and perspectives”. In: *Advanced Science* 6.21 (2019), p. 1900808.
- [2] and others. *The fourth paradigm: data-intensive scientific discovery*. Vol. 1.
- [3] Dana Bishara et al. “A state-of-the-art review on machine learning-based multiscale modeling, simulation, homogenization and design of materials”. In: *Archives of computational methods in engineering* 30.1 (2023), pp. 191–222.
- [4] Teng Long et al. “Constrained crystals deep convolutional generative adversarial network for the inverse design of crystal structures”. In: *npj Computational Materials* 7.1 (2021), p. 66.
- [5] Ziyuan Rao et al. “Machine learning-enabled high-entropy alloy discovery”. In: *Science* 378.6615 (2022), pp. 78–85.
- [6] Ankit Agrawal and Alok Choudhary. “Perspective: Materials informatics and big data: Realization of the “fourth paradigm” of science in materials science”. In: *Apl Materials* 4.5 (2016), p. 053208.
- [7] Yu-Chuan Hsu, Chi-Hua Yu, and Markus J Buehler. “Using deep learning to predict fracture patterns in crystalline solids”. In: *Matter* 3.1 (2020), pp. 197–211.
- [8] Linus C Erhard et al. “A machine-learned interatomic potential for silica and its relation to empirical models”. In: *npj Computational Materials* 8.1 (2022), p. 90.
- [9] Jaber Rezaei Mianroodi, Nima H. Siboni, and Dierk Raabe. “Teaching solid mechanics to artificial intelligence—a fast solver for heterogeneous materials”. In: *Npj Computational Materials* 7.1 (2021), p. 99.
- [10] Mauricio Fernández et al. “Application of artificial neural networks for the prediction of interface mechanics: a study on grain boundary constitutive behavior”. In: *Advanced Modeling and Simulation in Engineering Sciences* 7.1 (2020), pp. 1–27.

-
- [11] Michael E Deagen et al. “The materials tetrahedron has a “digital twin””. In: *MRS bulletin* 47.4 (2022), pp. 379–388.
- [12] William D Callister, David G Rethwisch, et al. *Materials science and engineering: an introduction*. Vol. 7. John Wiley & Sons New York, 2007.
- [13] TL Burnett and PJ Withers. “Completing the picture through correlative characterization”. In: *Nature materials* 18.10 (2019), pp. 1041–1049.
- [14] Siming Ma et al. “Microstructure-based numerical simulation of the mechanical properties and fracture of a Ti-Al₃Ti core-shell structured particulate reinforced A356 composite”. In: *Materials & Design* 191 (2020), p. 108685.
- [15] Mark D Wilkinson et al. “The FAIR Guiding Principles for scientific data management and stewardship”. In: *Scientific data* 3.1 (2016), pp. 1–9.
- [16] Swantje Bargmann et al. “Generation of 3D representative volume elements for heterogeneous materials: A review”. In: *Progress in Materials Science* 96 (2018), pp. 322–384.
- [17] Wei Chen, Akshay Iyer, and Ramin Bostanabad. “Data Centric Design: A New Approach to Design of Microstructural Material Systems”. In: *Engineering* 10 (2022), pp. 89–98.
- [18] David A Santos et al. “Multivariate hyperspectral data analytics across length scales to probe compositional, phase, and strain heterogeneities in electrode materials”. In: *Patterns* (2022), p. 100634.
- [19] Sen Li et al. “All-in-One Iontronic Sensing Paper”. In: *Advanced Functional Materials* 29.11 (2019), p. 1807343.
- [20] *Dualphasen-Stahl für Modernen Leichtbau: ThyssenKrupp Steel*. URL: <https://www.thyssenkrupp-steel.com>.
- [21] Hongyi Xu et al. “A descriptor-based design methodology for developing heterogeneous microstructural materials system”. In: *Journal of Mechanical Design* 136.5 (2014), p. 051007.
- [22] Joseph I Goldstein et al. *Scanning electron microscopy and X-ray microanalysis*. Springer, 2017.
- [23] Jill Chastain and Roger C King Jr. “Handbook of X-ray photoelectron spectroscopy”. In: *Perkin-Elmer Corporation* 40 (1992), p. 221.
- [24] J Epp. “X-ray diffraction (XRD) techniques for materials characterization”. In: *Materials characterization using nondestructive evaluation (NDE) methods*. Elsevier, 2016, pp. 81–124.

-
- [25] Franz Pfeiffer. “X-ray ptychography”. In: *Nature Photonics* 12.1 (2018), pp. 9–17.
- [26] Cynthia A Volkert and Andrew M Minor. “Focused ion beam microscopy and micromachining”. In: *MRS bulletin* 32.5 (2007), pp. 389–399.
- [27] David B Williams et al. *The transmission electron microscope*. Springer, 1996.
- [28] G Binnig et al. “Atomic resolution with atomic force microscope”. In: *Europhysics Letters* 3.12 (1987), p. 1281.
- [29] Donal P Finegan et al. “Machine-Learning-Driven Advanced Characterization of Battery Electrodes”. In: *ACS Energy Letters* 7.12 (2022), pp. 4368–4378.
- [30] Pingluo Zhao et al. “A new method for classifying and segmenting material microstructure based on machine learning”. In: *Materials & Design* 227 (2023), p. 111775.
- [31] Seyed Majid Azimi et al. “Advanced steel microstructural classification by deep learning methods”. In: *Scientific reports* 8.1 (2018), p. 2128.
- [32] Dirk Raabe. *Materials multiscale modeling, DFT, Continuum Modeling, plasticity, deformation*. URL: <https://www.dierk-raabe.com/multiscale-modeling/>.
- [33] Laura M Hillscher et al. “Functional paper-based materials for diagnostics”. In: *ChemTexts* 7 (2021), pp. 1–22.
- [34] Teng Long. “Accelerating the discovery of crystalline materials with desired intrinsic properties by machine learning”. In: (2022).
- [35] Yibin Xu, Masayoshi Yamazaki, and Pierre Villars. “Inorganic materials database for exploring the nature of material”. In: *Japanese Journal of Applied Physics* 50.11S (2011), 11RH02.
- [36] FJ Humphreys. “Characterisation of fine-scale microstructures by electron backscatter diffraction (EBSD)”. In: *Scripta materialia* 51.8 (2004), pp. 771–776.
- [37] Greg Johnson et al. “X-ray diffraction contrast tomography: a novel technique for three-dimensional grain mapping of polycrystals. II. The combined case”. In: *Journal of Applied Crystallography* 41.2 (2008), pp. 310–318.
- [38] Hongyi Xu et al. “A machine learning-based design representation method for designing heterogeneous microstructures”. In: *Journal of Mechanical Design* 137.5 (2015), p. 051403.
- [39] Frederick N Rhines. “Microstructure-property relationships in materials”. In: *Metallurgical Transactions A* 8.1 (1977), pp. 127–133.

-
- [40] Salvatore Torquato. *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Vol. 16. Springer Science & Business Media, 2005.
- [41] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [42] *Intro to autoencoders; Tensorflow Core*. URL: <https://www.tensorflow.org/tutorials/generative/autoencoder>.
- [43] Diederik P Kingma, Max Welling, et al. “An introduction to variational autoencoders”. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.
- [44] Ian Goodfellow et al. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.
- [45] Toshio Mura. *Micromechanics of defects in solids*. Springer Science & Business Media, 2013.
- [46] Bernard Budiansky. “Micromechanics”. In: *Computers & Structures* 16.1-4 (1983), pp. 3–12.
- [47] André Zaoui. “Continuum micromechanics: survey”. In: *Journal of Engineering Mechanics* 128.8 (2002), pp. 808–816.
- [48] Lukas M Keller et al. “Pore space relevant for gas permeability in Opalinus clay: Statistical analysis of homogeneity, percolation, and representative volume element”. In: *Journal of Geophysical Research: Solid Earth* 118.6 (2013), pp. 2799–2812.
- [49] Toufik Kanit et al. “Determination of the size of the representative volume element for random composites: statistical and numerical approach”. In: *International Journal of Solids and Structures* 40.13-14 (2003), pp. 3647–3679.
- [50] Xiaolei Yin et al. “Statistical volume element method for predicting microstructure-constitutive property relations”. In: *Computer methods in applied mechanics and engineering* 197.43-44 (2008), pp. 3516–3529.
- [51] Martin Ostoja-Starzewski. “Material spatial randomness: From statistical to representative volume element”. In: *Probabilistic engineering mechanics* 21.2 (2006), pp. 112–132.
- [52] Tom M Mitchell. *The need for biases in learning generalizations*. Citeseer, 1980.
- [53] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [54] Ahmad EL Sallab et al. “Deep reinforcement learning framework for autonomous driving”. In: *arXiv preprint arXiv:1704.02532* (2017).

-
- [55] Jens Kober, J Andrew Bagnell, and Jan Peters. “Reinforcement learning in robotics: A survey”. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1238–1274.
- [56] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *nature* 529.7587 (2016), pp. 484–489.
- [57] Rama K Vasudevan, Erick Orozco, and Sergei V Kalinin. “Discovering mechanisms for materials microstructure optimization via reinforcement learning of a generative model”. In: *Machine Learning: Science and Technology* 3.4 (2022), 04LT03.
- [58] Fanping Sui et al. “Deep reinforcement learning for digital materials design”. In: *ACS Materials Letters* 3.10 (2021), pp. 1433–1439.
- [59] Juan J De Pablo et al. “The materials genome initiative, the interplay of experiment, theory and computation”. In: *Current Opinion in Solid State and Materials Science* 18.2 (2014), pp. 99–117.
- [60] Deborah McGuinness et al. “NanoMine: Ontology-Enabled Polymer Nanocomposite Open Community Data Resource”. In: (2022).
- [61] Deborah McGuinness et al. “MaterialsMine: An open-source, user-friendly materials data resource guided by FAIR principles”. In: (2022).
- [62] Nico Brandt et al. “Kadi4Mat: A research data infrastructure for materials science”. In: *Data science journal* 20.1 (2021).
- [63] Robert H Schmitt et al. “NFDI4Ing-the National Research Data Infrastructure for Engineering Sciences”. In: *Zenodo* (2020).
- [64] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [65] Sucheta Chauhan et al. “A Comparison of Shallow and Deep Learning Methods for Predicting Cognitive Performance of Stroke Patients From MRI Lesion Images”. In: *Frontiers in Neuroinformatics* 13.July (2019), pp. 1–12. issn: 16625196. doi: 10.3389/fninf.2019.00053.
- [66] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.
- [67] Jianxin Wu. “Introduction to convolutional neural networks”. In: *National Key Lab for Novel Software Technology. Nanjing University. China* 5.23 (2017), p. 495.

-
- [68] Zijie J. Wang et al. “CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* (2020).
- [69] Amir Abbas Kazemzadeh Farizhandi and Mahmood Mamivand. “Spatiotemporal prediction of microstructure evolution with predictive recurrent neural network”. In: *Computational Materials Science* 223 (2023), p. 112110.
- [70] Daria Fokina et al. “Microstructure synthesis using style-based generative adversarial networks”. In: *Physical Review E* 101.4 (2020), p. 043308.
- [71] Andrea Gayon-Lombardo et al. “Pores for thought: generative adversarial networks for stochastic reconstruction of 3D multi-phase electrode microstructures with periodic boundaries”. In: *npj Computational Materials* 6.1 (2020), p. 82.
- [72] Brady D Lund and Ting Wang. “Chatting about ChatGPT: how may AI and GPT impact academia and libraries?” In: *Library Hi Tech News* 40.3 (2023), pp. 26–29.
- [73] Markus J Buehler. “FieldPerceiver: Domain agnostic transformer model to predict multiscale physical fields and nonlinear material properties through neural ologs”. In: *Materials Today* 57 (2022), pp. 9–25.
- [74] Zhenze Yang and Markus J Buehler. “Words to matter: De novo architected materials design using transformer neural networks”. In: *Frontiers in Materials* 8 (2021), p. 740754.
- [75] Nihang Fu et al. “Material transformers: deep learning language models for generative materials design”. In: *Machine Learning: Science and Technology* 4.1 (2023), p. 015001.
- [76] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.
- [77] Wilfried Becker and Dietmar Gross. *Mechanik elastischer Körper und Strukturen*. Springer-Verlag, 2013.
- [78] Dietmar Gross et al. *Technische Mechanik: Band 2: Elastostatik*. Springer, 2007.
- [79] Rong Yu, J Zhu, and HQ Ye. “Calculations of single-crystal elastic constants made simple”. In: *Computer physics communications* 181.3 (2010), pp. 671–675.
- [80] Cody J. Permann et al. “MOOSE: Enabling massively parallel multiphysics simulation”. In: *SoftwareX* 11 (2020), p. 100430. ISSN: 2352-7110. DOI: <https://doi.org/10.1016/j.softx.2020.100430>.

-
- [81] Binbin Lin et al. “A deep learned nanowire segmentation model using synthetic data augmentation”. In: *npj Computational Materials* 8.1 (2022), p. 88.
- [82] David A Santos et al. “Chemistry–mechanics–geometry coupling in positive electrode materials: a scale-bridging perspective for mitigating degradation in lithium-ion batteries through materials design”. In: *Chemical Science* 14.3 (2023), pp. 458–484.
- [83] David A Santos, Binbin Lin, and Justin L Andrews. “Data science enables X-ray vision”. In: *Patterns* 3.12 (2022), p. 100660.
- [84] Lane A Baker. “Perspective and prospectus on single-entity electrochemistry”. In: *Journal of the American Chemical Society* 140.46 (2018), pp. 15549–15559.
- [85] Wenzao Li et al. “Peering into batteries: electrochemical insight through in situ and operando methods over multiple length scales”. In: *Joule* (2020).
- [86] Luis R De Jesus et al. “Defining diffusion pathways in intercalation cathode materials: some lessons from V2O5 on directing cation traffic”. In: *ACS Energy Letters* 3.4 (2018), pp. 915–931.
- [87] David A Santos et al. “Assessing the role of vanadium technologies in decarbonizing hard-to-abate sectors and enabling the energy transition”. In: *Isience* 24.11 (2021), p. 103277.
- [88] M Stanley Whittingham. “The role of ternary phases in cathode reactions”. In: *Journal of The Electrochemical Society* 123.3 (1976), p. 315.
- [89] Luis R De Jesus et al. “Mapping polaronic states and lithiation gradients in individual V₂O₅ nanowires”. In: *Nature communications* 7.1 (2016), pp. 1–9.
- [90] Yuting Luo et al. “Cation reordering instead of phase transitions: Origins and implications of contrasting lithiation mechanisms in 1D ζ - and 2D α -V₂O₅”. In: *Proceedings of the National Academy of Sciences* 119.4 (2022).
- [91] Young-Sang Yu et al. “Three-dimensional localization of nanoscale battery reactions using soft X-ray tomography”. In: *Nature communications* 9.1 (2018), pp. 1–7.
- [92] David A Santos et al. “Bending good beats breaking bad: phase separation patterns in individual cathode particles upon lithiation and delithiation”. In: *Materials Horizons* 7.12 (2020), pp. 3275–3290.
- [93] Justin L Andrews et al. “Curvature-Induced Modification of Mechano-Electrochemical Coupling and Nucleation Kinetics in a Cathode Material”. In: *Matter* 3.5 (2020), pp. 1754–1773.

-
- [94] Milos Mandic et al. “Effects of catalyst activity, particle size and shape, and process conditions on catalyst effectiveness and methane selectivity for Fischer–Tropsch reaction: a modeling study”. In: *Industrial & Engineering Chemistry Research* 56.10 (2017), pp. 2733–2745.
- [95] Yuting Luo et al. “Effect of crystallite geometries on electrochemical performance of porous intercalation electrodes by multiscale operando investigation”. In: *Nature materials* (2021), pp. 1–11.
- [96] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [97] Kaiming He et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.
- [98] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.
- [99] Mark Everingham et al. “The pascal visual object classes (voc) challenge”. In: *International journal of computer vision* 88.2 (2010), pp. 303–338.
- [100] Satoru Masubuchi et al. “Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials”. In: *npj 2D Materials and Applications* 4.1 (2020), pp. 1–9.
- [101] Max Frei and Frank Einar Kruijs. “FibeR-CNN: Expanding Mask R-CNN to improve image-based fiber analysis”. In: *Powder Technology* 377 (2021), pp. 974–991.
- [102] Batuhan Yildirim and Jacqueline M Cole. “Bayesian Particle Instance Segmentation for Electron Microscopy Image Quantification”. In: *Journal of Chemical Information and Modeling* 61.3 (2021), pp. 1136–1149.
- [103] Boyuan Ma et al. “Data augmentation in microscopic images for material data mining”. In: *npj Computational Materials* 6.1 (2020), pp. 1–9.
- [104] Brian L DeCost and Elizabeth A Holm. “Characterizing powder materials using keypoint-based computer vision methods”. In: *Computational Materials Science* 126 (2017), pp. 438–445.
- [105] Leonid Mill et al. “Synthetic Image Rendering Solves Annotation Problem in Deep Learning Nanoparticle Segmentation”. In: *Small Methods* (2020), p. 2100223.
- [106] Paul Monchot et al. “Deep Learning Based Instance Segmentation of Titanium Dioxide Particles in the Form of Agglomerates in Scanning Electron Microscopy”. In: *Nanomaterials* 11.4 (2021), p. 968.

-
- [107] Ulrich Mansfeld et al. “Towards accurate analysis of particle size distribution for non-spherically shaped nanoparticles as quality control materials”. In: *Microscopy and Microanalysis* 25.S2 (2019), pp. 2328–2329.
- [108] Gregory A Horrocks et al. “Finite size effects on the structural progression induced by lithiation of V₂O₅: a combined diffraction and Raman spectroscopy study”. In: *Journal of Materials Chemistry A* 1.48 (2013), pp. 15265–15277.
- [109] Justin L Andrews et al. “Reversible Mg-ion insertion in a metastable one-dimensional polymorph of V₂O₅”. In: *Chem* 4.3 (2018), pp. 564–585.
- [110] Dimitrios Maganas et al. “First principles calculations of the structure and V L-edge X-ray absorption spectra of V₂O₅ using local pair natural orbital coupled cluster theory and spin-orbit coupled configuration interaction approaches”. In: *Physical Chemistry Chemical Physics* 15.19 (2013), pp. 7260–7276.
- [111] Piotr Skalski. <https://www.makesense.ai/>. 2021. URL: <https://www.makesense.ai/> (visited on 03/20/2021).
- [112] Zhaojin Huang et al. “Mask scoring r-cnn”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6409–6418.
- [113] Xinlei Chen et al. “Tensormask: A foundation for dense object segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2061–2069.
- [114] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.
- [115] Huitian Bai and Sen Wu. “Nanowire Detection in AFM Images Using Deep Learning”. In: *Microscopy and Microanalysis* 27.1 (2021), pp. 54–64.
- [116] Daniel Bolya et al. “Yolact: Real-time instance segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9157–9166.
- [117] Tomasz Konopczyński et al. “Instance segmentation of fibers from low resolution ct scans via 3d deep embedding learning”. In: *arXiv preprint arXiv:1901.01034* (2019).
- [118] Camilo Aguilar et al. “3D Fiber Segmentation With Deep Center Regression and Geometric Clustering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3746–3754.

-
- [119] Yuxin Wu et al. *Detectron2*. <https://github.com/facebookresearch/detectron2>. 2019.
- [120] GeoDict 2021 von Math2Market GmbH. “GrainGeo Handbuch”. In: (2021). URL: <https://doi.org/10.30423/userguide.geodict2021-graingeo>.
- [121] Edward H Adelson et al. “Pyramid methods in image processing”. In: *RCA engineer* 29.6 (1984), pp. 33–41.
- [122] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.
- [123] Mirna Lerotic et al. “MANTiS: a program for the analysis of X-ray spectromicroscopy data”. In: *Journal of synchrotron radiation* 21.5 (2014), pp. 1206–1212.
- [124] Matthew J Zdilla et al. “Circularity, solidity, axes of a best fit ellipse, aspect ratio, and roundness of the foramen ovale: a morphometric analysis with neurosurgical considerations”. In: *The Journal of craniofacial surgery* 27.1 (2016), p. 222.
- [125] Aashutosh Mistry et al. “Asphericity can cause nonuniform lithium intercalation in battery active particles”. In: *ACS Energy Letters* 7.5 (2022), pp. 1871–1879.
- [126] William J Krause. *Krause’s essential human histology for medical students*. Universal-Publishers, 2005.
- [127] Gorazd Planinšič and Michael Vollmer. “The surface-to-volume ratio in thermal physics: From cheese cube physics to animal metabolism”. In: *European journal of physics* 29.2 (2008), p. 369.
- [128] Chunhai Jiang, Eiji Hosono, and Haoshen Zhou. “Nanomaterials for lithium ion batteries”. In: *Nano today* 1.4 (2006), pp. 28–33.
- [129] Emily M Williamson et al. “Creating ground truth for nanocrystal morphology: a fully automated pipeline for unbiased transmission electron microscopy analysis”. In: *Nanoscale* 14.41 (2022), pp. 15327–15339.
- [130] Yang Bai et al. “A chemo-mechanical damage model at large deformation: numerical and experimental studies on polycrystalline energy materials”. In: *International Journal of Solids and Structures* 228 (2021), p. 111099.
- [131] Armin Asheri et al. “Data-driven multiscale simulation of solid-state batteries via machine learning”. In: *Computational Materials Science* 226 (2023), p. 112186.

-
- [132] Shahed Rezaei, Armin Asheri, and Bai-Xiang Xu. “A consistent framework for chemo-mechanical cohesive fracture and its application in solid-state batteries”. In: *Journal of the Mechanics and Physics of Solids* 157 (2021), p. 104612.
- [133] Binbin Lin, Yang Bai, and Bai-Xiang Xu. “Data-driven microstructure sensitivity study of fibrous paper materials”. In: *Materials & Design* 197 (2021), p. 109193.
- [134] Julia Auernhammer et al. “Mapping humidity-dependent mechanical properties of a single cellulose fibre”. In: *Cellulose* 28 (2021), pp. 8313–8332.
- [135] Binbin Lin et al. “Humidity influence on mechanics of paper materials: joint numerical and experimental study on fiber and fiber network scale”. In: *Cellulose* 29.2 (2022), pp. 1129–1148.
- [136] Xu Bai-Xiang. *Micromechanical modelling and simulation of functional paper materials*. 2018. URL: <https://gepris.dfg.de/gepris/projekt/405422877?language=en>.
- [137] Mikael S Magnusson, Xiaobo Zhang, and Sören Östlund. “Experimental evaluation of the interfibre joint strength of papermaking fibres in terms of manufacturing parameters and in two different loading directions”. In: *Experimental mechanics* 53.9 (2013), pp. 1621–1634.
- [138] Ulrich Hirn et al. “The area in molecular contact in fiber-fiber bonds”. In: *Transactions of the 15th Fundamental Research Symposium*. 2013, pp. 8–13.
- [139] Mikael S Magnusson. “Investigation of interfibre joint failure and how to tailor their properties for paper strength”. In: *Nordic Pulp & Paper Research Journal* 31.1 (2016), pp. 109–122.
- [140] Annsofie Torgnysdotter et al. “Fiber/fiber crosses: finite element modeling and comparison with experiment”. In: *Journal of composite materials* 41.13 (2007), pp. 1603–1618.
- [141] Maria Axelsson. “Estimating 3D fibre orientation in volume images”. In: *2008 19th International Conference on Pattern Recognition*. IEEE. 2008, pp. 1–4.
- [142] Stina Svensson and Mattias Aronsson. “Using distance transform based algorithms for extracting measures of the fiber network in volume images of paper”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 33.4 (2003), pp. 562–571.
- [143] Erik LG Wernersson, Anders Brun, and Cris L Luengo Hendriks. “Segmentation of wood fibres in 3D CT images using graph cuts”. In: *International Conference on Image Analysis and Processing*. Springer. 2009, pp. 92–102.

-
- [144] J Vigié et al. “Finding fibres and their contacts within 3D images of disordered fibrous media”. In: *Composites Science and Technology* 89 (2013), pp. 202–210.
- [145] Andrew Marais et al. “New insights into the mechanisms behind the strengthening of lignocellulosic fibrous networks with polyamines”. In: *Cellulose* 21.6 (2014), pp. 3941–3950.
- [146] Artem Kulachenko and Tetsu Uesaka. “Direct simulations of fiber network deformation and failure”. In: *Mechanics of Materials* 51 (2012), pp. 1–14.
- [147] Yujun Li et al. “Evaluation of the out-of-plane response of fiber networks with a representative volume element model”. In: *Tappi journal* 17.6 (2018), pp. 329–339. ISSN: 0734-1415. URL: <https://publications.rwth-aachen.de/record/730091>.
- [148] Rami Mansour et al. “Stochastic constitutive model of isotropic thin fiber networks based on stochastic volume elements”. In: *Materials* 12.3 (2019), p. 538.
- [149] Sebastian Domaschke et al. “A 3D computational model of electrospun networks and its application to inform a reduced modelling approach”. In: *International Journal of Solids and Structures* 158 (2019), pp. 76–89.
- [150] Svetlana Borodulina, Hamid Reza Motamedian, and Artem Kulachenko. “Effect of fiber and bond strength variations on the tensile stiffness and strength of fiber networks”. In: *International Journal of Solids and Structures* 154 (2018), pp. 19–32.
- [151] Qinghua Meng et al. “Effects of nanofiber orientations on the fracture toughness of cellulose nanopaper”. In: *Engineering Fracture Mechanics* 194 (2018), pp. 350–361.
- [152] Qinghua Meng et al. “A multiscale crack-bridging model of cellulose nanopaper”. In: *Journal of the Mechanics and Physics of Solids* 103 (2017), pp. 22–39.
- [153] Hongli Zhu et al. “Anomalous scaling law of strength and toughness of cellulose nanopaper”. In: *Proceedings of the National Academy of Sciences* 112.29 (2015), pp. 8971–8976.
- [154] YC Lin, Jun Zhang, and Jue Zhong. “Application of neural networks to predict the elevated temperature flow behavior of a low alloy steel”. In: *Computational Materials Science* 43.4 (2008), pp. 752–758.
- [155] Ruihao Yuan et al. “Accelerated discovery of large electrostrains in BaTiO₃-based piezoelectrics using active learning”. In: *Advanced materials* 30.7 (2018), p. 1702884.

-
- [156] Changsheng Wang et al. “A property-oriented design strategy for high performance copper alloys via machine learning”. In: *npj Computational Materials* 5.1 (2019), pp. 1–8.
- [157] Zijiang Yang et al. “Establishing structure-property localization linkages for elastic deformation of three-dimensional high contrast composites using deep learning approaches”. In: *Acta Materialia* 166 (2019), pp. 335–345.
- [158] Jiaqi Wang et al. “Machine learning for thermal transport analysis of aluminum alloys with precipitate morphology”. In: *Advanced Theory and Simulations* 2.4 (2019), p. 1800196.
- [159] Chun-Teh Chen and Grace X Gu. “Effect of constituent materials on composite performance: exploring design strategies via machine learning”. In: *Advanced Theory and Simulations* 2.6 (2019), p. 1900056.
- [160] P Kärenlampi and H Suur-Hamari. “Classified wood raw materials for diversified softwood kraft pulps”. In: *Paperi ja puu* 79.6 (1997), pp. 404–410.
- [161] P Kärenlampi and Y Yu. “Fiber properties and paper fracture-Fiber length and fiber strength”. In: *The Fundamentals of Papermaking Materials* (1997), pp. 521–545.
- [162] RS Seth and HF Jang. “Determining the mean values for fibre physical properties”. In: *Nordic Pulp & Paper Research Journal* 19.3 (2004), pp. 372–378.
- [163] Rolf Wathén et al. “Studies on fiber strength and its effect on paper properties”. In: (2006).
- [164] Stergios Goutianos, Rui Mao, and Ton Peijs. “Effect of inter-fibre bonding on the fracture of fibrous networks with strong interactions”. In: *International Journal of Solids and Structures* 136 (2018), pp. 271–278.
- [165] Ning Zhang and Rouzbeh Shahsavari. “Balancing strength and toughness of calcium-silicate-hydrate via random nanovoids and particle inclusions: Atomistic modeling and statistical analysis”. In: *Journal of the Mechanics and Physics of Solids* 96 (2016), pp. 204–222.
- [166] Jari Sirvio. “Fibres and bonds”. In: *Niskanen K (ed) Paper physics*. Ed. by Jan Fagerberg, David C. Mowery, and Richard R. Nelson. Finnish paper Engineers Association/Paperi ja Puu Oy, 2nd ed, 2008. Chap. 2, pp. 59–92.
- [167] Stephen Kokoska and Daniel Zwillinger. *CRC standard probability and statistics tables and formulae*. Crc Press, 2000.

-
- [168] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. doi: <https://doi.org/10.1038/s41592-019-0686-2>.
- [169] Kyoungsoo Park and Glaucio H Paulino. “Computational implementation of the PPR potential-based cohesive model in ABAQUS: Educational perspective”. In: *Engineering fracture mechanics* 93 (2012), pp. 239–262.
- [170] Mikael S Magnusson and Sören Östlund. “Numerical evaluation of interfibre joint strength measurements in terms of three-dimensional resultant forces and moments”. In: *Cellulose* 20.4 (2013), pp. 1691–1710.
- [171] J Patrick McGarry et al. “Potential-based and non-potential-based cohesive zone formulations under mixed-mode separation and over-closure. Part I: Theoretical analysis”. In: *Journal of the Mechanics and Physics of Solids* 63 (2014), pp. 336–362.
- [172] F El-Hosseiny and DH Page. “The mechanical properties of single wood pulp fibres: theories of strength”. In: *Fibre Science and Technology* 8.1 (1975), pp. 21–31.
- [173] Dionissios T Hristopulos and Tetsu Uesaka. “Structural disorder effects on the tensile strength distribution of heterogeneous brittle materials with emphasis on fiber networks”. In: *Physical Review B* 70.6 (2004), p. 064108.
- [174] Svetlana Borodulina et al. “Stress-strain curve of paper revisited”. In: *Nordic pulp & paper research journal* 27.2 (2012), pp. 318–328.
- [175] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.
- [176] Alexey Vishtal and Elias Retulainen. “Boosting the extensibility potential of fibre networks: A review”. In: *BioResources* 9.4 (2014), pp. 7951–8001.
- [177] N Gumagul and DH Page. “The difference between dry and rewetted zero-span tensile strength of paper”. In: (1989).
- [178] Mária Fišerová, Juraj Gigac, and Jozef Balberčák. “Relationship between fibre characteristics and tensile strength of hardwood and softwood kraft pulps”. In: *Cellulose Chemistry & Technology* 44.7 (2010), p. 249.
- [179] Per Tomas Larsson et al. “Fiber length and bonding effects on tensile strength and toughness of kraft paper”. In: *Journal of Materials Science* 53.4 (2018), pp. 3006–3015.
- [180] Binbin Lin et al. “A Machine Learning Enabled Image-data-driven End-to-end Mechanical Field Predictor For Dual-Phase Steel”. In: *PAMM* 22.1 (2023), e202200110.

-
- [181] Cemal Cem Tasan et al. “An overview of dual-phase steels: advances in microstructure-oriented processing and micromechanically guided design”. In: *Annual Review of Materials Research* 45 (2015), pp. 391–431.
- [182] Rampi Ramprasad et al. “Machine learning in materials informatics: recent applications and prospects”. In: *npj Computational Materials* 3.1 (2017), pp. 1–13.
- [183] Setareh Medghalchi et al. “Damage Analysis in Dual-Phase Steel Using Deep Learning: Transfer from Uniaxial to Biaxial Straining Conditions by Image Data Augmentation”. In: *JOM* 72.12 (2020), pp. 4420–4430.
- [184] Juan C Simo and Thomas JR Hughes. *Computational inelasticity*. Vol. 7. Springer Science & Business Media, 2006.
- [185] Xin Sun et al. “Predicting failure modes and ductility of dual phase steels using plastic strain localization”. In: *International Journal of Plasticity* 25.10 (2009), pp. 1888–1909.
- [186] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [187] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019).
- [188] Mohammad S Khorrami et al. “An artificial neural network for surrogate modeling of stress fields in viscoplastic polycrystalline materials”. In: *arXiv preprint arXiv:2208.13490* (2022).
- [189] Haoliang Jiang et al. “Stressgan: A generative deep learning model for two-dimensional stress distribution prediction”. In: *Journal of Applied Mechanics* 88.5 (2021).
- [190] Rafael Bischof and Michael Kraus. “Multi-objective loss balancing for physics-informed deep learning”. In: *arXiv preprint arXiv:2110.09813* (2021).
- [191] Zhao Chen et al. “Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks”. In: *International conference on machine learning*. PMLR, 2018, pp. 794–803.
- [192] Franz Roters et al. “Overview of constitutive laws, kinematics, homogenization and multiscale methods in crystal plasticity finite-element modeling: Theory, experiments, applications”. In: *Acta materialia* 58.4 (2010), pp. 1152–1211.

-
- [193] Dierk Raabe and Franz Roters. “Using texture components in crystal plasticity finite element simulations”. In: *International Journal of Plasticity* 20.3 (2004), pp. 339–361.
- [194] Binbin Lin et al. *Training dataset for "A deep learned nanowire segmentation model using synthetic data augmentation"*. Version 1.0.0. Apr. 2022. DOI: 10.5281/zenodo.6469773. URL: <https://doi.org/10.5281/zenodo.6469773>.



List of Figures

- 1.1 The evolution of scientific paradigms involves a profound transformation, shifting from conventional empirical science to complex computational approaches that enhance the understanding in scientific disciplines and create better research methodologies. The advent of artificial intelligence and machine learning opens up new avenues, enabling innovative possibilities that reshape the existing scientific framework. 1
- 1.2 Materials tetrahedron and its information tetrahedron twin - emphasizing the workflow from materials data generation and processing to information and knowledge extraction for materials development and design process. Adapted from [11] (CC BY 4.0). 2
- 1.3 Microstructural systems considered in this work and their exemplary material products. **Left:** Microstructure of a battery cathode material [18], reproduced with permission; **Middle:** Fibrous microstructure of a paper-based micro-fluidic device [19], reproduced with permission; **Right:** Microstructure of a dual-phase steel material used in automotive chassis [20]. 5
- 1.4 Overview of the chapters with considered microstructural systems. While different materials are subject to study, their MPR using machine learning techniques covering various aspects are the common ground of this study. **Left:** An ensemble of segmented nanowire particles of a battery cathode material and their lithiation phase pattern; **Middle:** Synthetically generated fiber network and its mechanical FE simulation showing the stress field; **Right:** An image-based microstructure model of a dual-phase steel and its FE simulation showing the plastic strain field. 7

- 2.1 Multiscale nature and structural characteristics of metals and alloys [32]. 12
- 2.2 Paper material at different scales [33], adapted with permission (CC BY 4.0). 13
- 2.3 Classification of microstructures into porous and non-porous solids and its subdivisions [16], adapted with permission (CC BY-NC-ND). 16



- 2.4 Hierarchical levels of descriptor-based representations: **A**: Composition level; **B**: Dispersion level; **C**: Particle geometry level. 18
- 2.5 Descriptor-based representations for exemplary microstructure classes. 20
- 2.6 Examples of statistical correlation functions 21
- 2.7 Examples of latent space representations. 22
- 2.8 Characteristic length and decoupling of the scales. Recreated from Micromechanics lecture notes by Bai-Xiang Xu. 23
- 2.9 Machine learning workflow: The main components of a machine learning workflow pipeline are: Task, Data, Model, Deployment. The learning algorithm "drives" the model to "learn" the data by iteratively optimizing the objective function. 26
- 2.10 Scatter plot of exemplary data points; **a**: Separated into train and test data; **b**: Linear regression fit with R^2 -score. 32
- 2.11 Illustrative example of the 2D gradient descent for the linear regression problem, where the model parameters m, b are the search space and the loss function to be minimized. $\nabla L(\mathbf{w}^{(t)})$ denotes the search direction and the learning rate γ the length of the search vector. 33
- 2.12 K-means clustering for a synthetic dataset with failure strain and ultimate strength as exemplary input parameters. **a**: Sample data points; **b-d** show the model prediction with $K = 2, 3, 4$ while 0, 1 are the cluster labels representing the global brittle and ductile material class, respectively. Cluster labels 2, 3 denote possible sub-classes. 35
- 2.13 Examples of shallow learning models 37
- 2.14 Network structure of a multilayer perceptron model, generated with TikZ package (CC BY 4.0). 40
- 2.15 Forward computation of the first layer in the MLP model. 41
- 2.16 A common CNN-Structure. 43
- 2.17 Exemplary computation of convolution and max pooling operation, visualized with CNN Explainer [68]. 44
- 2.18 Exemplary data points fitted with different degree of polynomial basis functions; **Left**: An underfitted dataset showing high bias between train and test data points; The bias is observable for the distance between the train, test data points and prediction curve; **Middle**: A good fit between the prediction curve and the train and test data; **Right**: The prediction curve overfits the train data, high variance can be observed for the test data. 46

2.19 Model evaluation schemes. Left: Illustrative example of the trade-off between underfitting and overfitting; while the training error decreases with increasing model complexity, the test error gradually increases, implying typical overfitting behavior; Right: Exemplary K-fold cross-validation scheme for K=3. The input data is first divided into training and test data. For cross-validation, the training data is again divided into k-fold data sets for training and validation. This process is repeated in k iterations to obtain a mean cross-validated performance score. The validated model is then applied to the test dataset to verify the generative power of the model.	47
2.20 Description of the motion of a body from its reference configuration Ω_0 to current configuration Ω_t	48
2.21 Strain components at an infinitesimal cuboid in 2D perspective. Recreated from [77].	50
2.22 Definition of a traction vector for a deformable body under arbitrary load at the interior surface.	51
2.23 Stress components at an infinitesimal cuboid. Recreated from [78].	52
2.24 Finite element discretization of an exemplary 2D problem domain to subdomains of triangle elements.	58
3.1 Overview of the microscopy images. a: SEM; b: X-ray ptychography; c: STXM; d-f: Corresponding manual annotation	64
3.2 Overview of the workflow for geometry-lithiation correlation analysis.	67
3.3 A schematic overview of obtaining lithiation phase maps. In Step 1 of data collection, STXM data shows a transmission image a of a nanowire particle and its intensity spectral data b w.r.t to different energy levels. The background spectrum I_0 is used to transfer the transmission data to the absorption profile of the sample c ; Step 2 shows the STXM data for individual particles a-f of phase pure lithiation state and their intensity spectra g . The acceding order of curves in g corresponds to increasingly lithiated particles; In step 3, compared to the reference spectra data from step 2, direct deconvolution of the STXM image data is possible and leads to distinctively lithiated phase maps of the sample a-c and their intensity spectra d-f . Recreated with permission from [18].	71



- 3.4 Synthetically generated dataset for training procedure. The 3D microstructure is compressed to create an optical density-based image as input data. The individually labeled nanowires in correlation with the optical density-based image are then used to create the binary masks for output data. 72
- 3.5 Basic workflow of the Mask R-CNN algorithm: The input image is processed by different network components to extract high-level features, and regions of interest for the prediction of bounding boxes, classes, and masks. 75
- 3.6 Definition of intersection over union: Area of object intersection over the area of object union. This definition is used to calculate precision and recall for a given IoU threshold. 76
- 3.7 PR Curve: Exemplary precision values corresponding to given recall values, shown by red dots. According to VOC2010, to avoid the wiggly representation of the PR curve, shown by dashed green lines, precision values take the previous maximum values to the left to generate the normalized curve, shown by solid blue lines. The final AP is calculated as the area under the normalized curve. 77
- 3.8 Model prediction on V_2O_5 nanowires within a synthetic image. (512x512 pixels) **a**: Test image; **b**: Predicted instance masks with lower opacity plotted on the test image; **c**: Semantic binary mask; Blue: TP, Red: FN, Green: FP; every nanowire instance in the present synthetic image has been successfully segmented with highest accuracy and negligible deviation in the overlapping regions and particle boundaries. 81
- 3.9 Model prediction on V_2O_5 nanowires within the X-ray pythography image. (531x449 pixels) **a**: Test image; **b**: Predicted instance masks with lower opacity plotted on the test image; The best performer in AP segmentation mask in Tab. 3.2 was used to visualize the prediction masks in **b**. **c**: Semantic binary mask; Blue: TP, Red: FN, Green: FP; yellow arrows indicate the main FN pixels, i.e. nanowire pixels missed by the model. White arrows indicate the FP pixels, which were overlooked in the manual annotation but detected by the model. Cyan arrows indicate over-predicted pixels, which the model predicts as additional nanowires, but were only regions of higher intensity. 82

-
- 3.10 Particle statistics from the X-ray ptychography image are illustrated by histograms and their kernel-density estimates (KDE) as a function of area (summation of pixels corresponding to each particle mask), aspect ratio (particle length/width), and orientation (angle relative to the horizontal axis) in **a**, **b**, and **c**, respectively. The statistical results show qualitative agreement, however, the main discrepancy represented by over-predicted pixels causes the shift of the KDE curve in the respective number ranges. 83
- 3.11 Model prediction on V_2O_5 nanowires image by STXM. (531x449 pixels - rescaled) **a**: Test image; **b**: Predicted instance masks with lower opacity plotted on the test image; The best performer in AP segmentation mask in Tab. 3.3 was used to visualize the prediction masks in **b**. **c**: Semantic binary mask; Blue: TP, Red: FN, Green: FP; In **c**, many FP nanowires are present, which indicate the missing manual labels, but are detected by the model. The results demonstrate the robustness of the model in segmenting images of low-resolution and densely packed particles. 85
- 3.12 Histograms and KDE curves show particle statistics from the STXM image as a function of area, aspect ratio, and orientation in **a**, **b**, and **c**, respectively. The statistical results show good qualitative and quantitative agreement. The main reason for the shift of KDE curves is due to the FP pixels, leading to higher density estimates in the respective number range. 85
- 3.13 Model prediction on V_2O_5 nanowires within SEM image. (957x1280 pixels) **a**: Test image; **b**: Predicted instance masks with lower opacity plotted on the test image; The best performer in the AP segmentation mask in Tab. 3.4 was used to visualize the prediction masks in **b**. **c**: Semantic binary mask; While sufficiently good segmentation results are found for overlapping nanowires, several FN pixelated nanowires, indicated by yellow arrows in **b** were missed by the model. Further discrepancies can be found in the region of agglomeration denoted by FP pixels and white arrows. The absence of optical density information combined with the distinct mode of contrast relative to STXM and X-ray ptychography contributes to the comparably lower prediction performance. 88
- 3.14 Histograms and KDE curves show particle statistics from the SEM image as a function of area, aspect ratio, and orientation in **a**, **b**, and **c**, respectively. The statistical results show good qualitative agreement. The main discrepancy exists for smaller FN nanowires missed by the model, thus leading to higher density estimates in the ground truth (manual annotation) in the respective number ranges, e.g., the peak in area size distribution. 89



3.15 **a:** STXM image data; **b:** Instance segmented image dataset with particle masks as in Fig. 3.12; **c-e:** Lithiation phase maps. 90

3.16 **Left column:** Stacked bar plot of particle composition, arranged with ascending descriptor numbers. **a:** Projected particle perimeter, **b:** Projected particle area, **c:** Aspect ratio, **d:** Circularity; **Right column:** KDE curves for each descriptor and their characteristic pattern clusters. 93

3.17 **Left column:** Stacked bar plot of particle composition, arranged with ascending descriptor numbers. **a:** Eccentricity, **b:** Convexity, **c:** Solidity, **d:** Orientation; **Right column:** KDE curves for each descriptor and their characteristic pattern clusters. 94

3.18 **a:** STXM image of instance segmented particles with particles of interest and their IDs highlighted. Red IDs 73, 80, 62 point to the particles with the largest P/R ratio. White IDs 4, 13, 46 point to the particles with the smallest P/R ratio; **b:** Stacked bar plot of particle composition arranged in ascending P/A ratio values. Particles 4, 13, 46 show a high degree of phase coexistence, while particles 73, 80, 62 have a more homogeneous concentration of lithium from cluster 3; **c:** KDE curve for the same descriptor. 97

3.19 **a,c:** Spatially resolved composition maps of exemplary particles for subsequent FE simulation; **b,d** Calculated von Mises stress map. More details on the workflow of lithiation-to-stress mapping can be found in [18]. 98

3.20 A set of synthetically generated particles by varying the parameters given in Eq. 3.11. The broken symmetry or shape defect in the particle shapes is purposely introduced and achieved by setting the L_{ratio} to be a decimal number. One can preserve the particle symmetry in the particle shapes by setting this parameter equals to an integer [Manuscript under preparation].100

4.1 **a,b:** SEM image of Linters paper (Image courtesy by Julia Auernhammer) vs. **c:** Artificially generated fiber network sample. 102

4.2 Workflow chart of the study 105

4.3 The property strain to failure converges with approx. 600 samples for mean and standard deviation (STD) values. Whereas for effective stiffness, 1662 samples for the mean value and 1775 for the STD value are needed. For maximal stress, 933 samples and 1292 samples are sufficient to draw a convergence, for the mean and STD, respectively. 108

4.4 Dendrogram and heatmap **a:** Before feature selection; **b:** After feature selection. 109

4.5 Illustration of the cohesive zone model in the interfiber contact. 110

4.6	a: Coloured individual fibers; b: Interfiber contacts highlighted in undeformed state; c: Deformed state with damaged and separated fibers. . .	113
4.7	a: A fraction of stress-strain curves from the database; b: Illustration of the target properties for two typical curves in the database.	114
4.8	ML-prediction of different models as a function of the FE simulated mechanical features. a: Strain to failure; b: Effective stiffness; c: Maximal stress.	117
4.9	Barplot of relative importance based on a: Pearson correlation coefficient; b: Permutation.	118
4.10	Scatter plots with linear fit and data distribution (The values are normalized for easier access). d-f: Depending on the Std. of fiber network orientation; g-i: Depending on the mean contact area size; j-l: Structures with highest target values with respect to strain, stiffness and stress. . .	120
4.11	Modelling of humidity effects at single fiber and interfiber level [135]. .	121
4.12	The von Mises stress map and damage pattern show variations at different relative humidities (RH). High stress levels are observed in the dry state, which are reduced as humidity increases. The presence of moisture allows greater flexibility in fiber deformation and loosening of strong interfiber bonds. Consequently, this supports greater deformations and reduces the stress intensity in the fibers, but pronounced interfiber failure and thus a reduction in the strength of the fiber network and the overall paper structure [135].	122
4.13	The simulation of a real CT dataset involves several steps such as cropping, smoothing, segmentation of fibers in the region of interest and subsequent 3D meshing of fibers, which are in contact with each other, and the assignment of cohesive zone elements up to boundary marking for the final mechanical FE simulation.	123
5.1	A high-resolution panoramic SEM graph showing the microstructural heterogeneity of the DP-steel. Image courtesy by Setareh Medghalchi. . .	126
5.2	Data generation process flow. Left: Microstructural SEM micrographs are obtained from a DP steel sample via panoramic imaging and further post-processing steps. The micrographs are semantically segmented using machine learning based techniques [183]; Right: The segmented phase images are contour-meshed for elastic-plastic FE simulations under tensile loading to obtain the local von Mises stress and equivalent plastic strain fields. The segmented phase images (input data) and simulated stress/strain fields (output data) are correlated by an advanced CNN model.	127



5.3 Exemplary microstructures and their corresponding segmentation and FE meshes. 129

5.4 Influence of microstructure size on structural characteristics and mechanical strain-strain behaviour. **a:** Stress-strain curves for 8 different sizes of randomly cropped microstructures; the stress-strain response is similar, with a small deviation for the smallest size of 256x256 pixels. As the size increases, the stress-strain response converges accordingly; **b:** Microstructures shown for increasing size. Different levels of detail can be observed. As the domain size increases, the microstructure becomes more homogeneous; **c-d:** Statistical evaluation (Mean and Std.) of each phase fraction and number of martensite islands for the domain size of 512x512 and 1536x1536 pixels, for 20 random microstructure samples. The statistical variation in the phase fraction of each phase for the larger microstructure set in **d** is comparatively small compared to **c** and the size can be considered representative. Note that the phase fraction of possible defects is neglected in this case since it is less than 0.2 % of the total paranoma image. The resolution of each pixel is 32.5 nm. 130

5.5 Qualitative comparison of PEEQ field with literature results. **a:** PEEQ map [185], adapted with permission; **b:** Our PEEQ map for selected size of microstructures. 132

5.6 A schematic sketch for the U-net structure: Input data is the segmented micrograph phase maps, while the output is either the plastic equivalent strain map or the von Mises stress map. 133

5.7 Learning curves for the train and the validation losses. **Left:** Model training for the PEEQ map; **Right:** Model training von Mises stress map. A clear trend of over-fitting can be found for both model training curves after approx. 30 epochs. 134

5.8 PEEQ prediction results by the U-net model for exemplary microstructures. **First column:** Exemplary microstructures; **Second column:** Ground truth PEEQ field calculated by FE simulation; **Third column:** Predicted field results by the U-net model; **Forth column:** The pixel-wise error between the ground truth and the model prediction. 135

5.9 Von Mises stress prediction results by the U-net model for exemplary microstructures. **First column:** Exemplary microstructures; **Second column:** Ground truth stress field calculated by FE simulation; **Third column:** Predicted field results by the U-net model; **Forth column:** The pixel-wise error between ground truth and the model prediction. 136



5.10 Flowchart of the web-based application under current development.
This application should enable easy usage of models and techniques developed in this work. At the start, the user can upload a sample of DP-steel microstructure for segmentation. After the segmentation step, the user can select different features for characterization or mechanical stress/strain analysis. Alternatively, since the trained ML model is based on a specific simulation model under fixed loading conditions, as in this work, the user can choose to generate the FE mesh for his own simulation model. After obtaining the features and mechanical properties, the data can be visualized for a straight-forward correlation analysis. 138



List of Tables

- 2.1 Some common activation functions [41] 42
- 2.2 Some common regression metrics [64]; y_i : Ground truth value; \hat{y}_i : Model predicted value, n : Number of samples; \bar{y} : Mean of the ground truth values 45
- 2.3 Governing equations for linear elastic problems of small deformation with in total 15 equations for 3 unknowns of u_i , each 6 unknowns for σ_{ij} and ε_{ij} with appropriate boundary conditions depending on the actual problem setting. 56

- 3.1 Hyperparameter study in the training procedure 80
- 3.2 Model performance on X-ray pychography image (Bounding box/Segmentation mask) 82
- 3.3 Model performance on STXM image (Bounding box/Segmentation mask) 84
- 3.4 Model performance on SEM image (Bounding box/Segmentation mask) 87
- 3.5 Geometrical descriptors, their definition and statistical numbers for the presented dataset. L_s, L_l denote the short and the long edge of the detected bounding box around the particle and L_m, L_M denote the length of the minor and the major axis when an ellipse is fitted through the particle. 92

- 4.1 Fiber network settings and ML input and output parameters 106
- 4.2 Material parameters for the fiber network simulation 111

Acronyms

Acronym	Full name
AI	Artificial Intelligence
ML	Machine Learning
PSPP	Processing-Structure-Property-Performance
FAIR	Findability, Accessibility, Interoperability, Reusability
MPR	Microstructure-Property relation
SEM	Scanning Electron Microscopy
XPS	X-ray Photoelectron Spectroscopy
XRD	X-ray Diffraction
XRP	X-ray Ptychography
FIB	Focused Ion Beam
TEM	Transmission Electron Microscopy
AFM	Atomic Force Microscopy
LIBs	Lithium-Ion Batteries
Mask-RCNN	Mask Regional Convolutional Neural Networks
STXM	Scanning Transmission X-ray Microscopy
EBSD	Electron Backscatter Diffraction
DCT	X-ray Diffraction Contrast Tomography
PCA	Principal Component Analysis
SVD	Singular Value Decomposition

AE	Autoencoder
CNN	Convolutional Neural Networks
VAE	Variational Autoencoder
GAN	Generative Adversarial Networks
RVE	Representative Volume Element
SVE	Statistical Volume Element
NFDI4ING	National Research Infrastructure for Engineering Sciences
KNN	K-Nearest Neighbors
DT	Decision Trees
GB	Gradient Boosting
FCFFN	Fully-Connected Feedforward Neural Network
MLP	Multilayer Perceptron
MAE	Mean Absolute Error
MSE	Mean Squared Error
MAPE	Mean Absolute Percentage Error
MA	Max error
MOOSE	Multiphysics Object-Oriented Simulation Environment
V ₂ O ₅	Vanadium Pentoxide
COCO	Common Object in Context
VOC	Visual Object Classes
DME	Dimethyl Carbonate
EPU	Elliptically Polarizing Undulator Source
JSON	JavaScript Object Notation
YOLO	You Only Look Once
FPN	Feature Pyramid Network
RPN	Region Proposal Network

ROI	Regions Of Interest
mAP	mean Average Precision
IoU	Intersection over Union
PR	Precision Recall
AUC	Area Under Curve
NMS	Non-Maximum-Suppression
TP	True Positive
FP	False Positive
TN	True Negative
FN	Fale Negative
KDE	Kernel Density Estimation
FiPre	Functional Paper Research Project
Micro CT	Micro Computed Tomography
CZM	Cohesive Zone Model
STD	Standard Deviation
GB	Gradient Boosting
PCC	Pearson Correlation Coefficient
F-O	Fiber Network Orientation
F-L	Fiber Length
F-D	Fiber Diameter
C-AS	Contact Area Size
C-ASD	Contact Area Spatial Distribution
C-ANO	Contact Area Normal Orientation
EDM	Electron Discharge Machine
NHR4CES	National High Performance Computing Center for Computational Engineering Science
PEEQ	Equivalent Plastic Strain

Curriculum Vitae

Education

- 09.2018 - 07.2023 Dr.-Ing. in Computational Materials Science
Thesis: "Data-driven analysis of microstructure-property relation in functional materials", with Prof. Bai-Xiang Xu, Institute of Material Science, Technische Universität Darmstadt (TUD), Germany
- 04.2016 - 08.2018 M. Sc. in Mechanics
Thesis: "Modeling of arteries and FE approximation for fluid-structure interaction in hemodynamics", with Prof. Herbert Egger, Department of Mathematics, TUD, Germany
- 09.2015 - 03.2016 Erasmus Exchange Program
Insitut Polytechnique de Grenoble, France
- 10.2012 - 01.2016 B. Sc. in Applied Mechanics
Thesis: "Calculation and construction of the buckling rod for determining the characteristics of fiber-plastic composites", with Prof. Helmut Schürmann, Department of mechanical Engineering, TUD, Germany

List of Contributions

Journal Publications

- [1] **Binbin Lin**, Yangbai, Bai-Xiang Xu, "Data-driven microstructure sensitivity study of fibrous paper materials", In: *Materials & Design* 197, 1091933 (2021), <https://doi.org/10.1016/j.matdes.2020.109193>.
- [2] Christoph Reimuth, **Binbin Lin**, Yangyiwei Yang, Peter Stein, Xiandong Zhou, and Bai-Xiang Xu, "Chemo-mechanical study of dislocation mediated ion diffusion in lithium-ion battery materials", In: *Journal of Applied Physics* 130, 035103 (2021), <https://doi.org/10.1007/s10570-021-04058-4>.
- [3] Julia Auernhammer, Tom Keil, **Binbin Lin**, Jan-Lukas Schäfer, Bai-Xiang Xu, Markus Biesalski, Robert W Stark, "Mapping humidity-dependent mechanical properties of a single cellulose fibre", In: *Cellulose* volume 28, pages 8313–8332, (2021), <https://doi.org/10.1007/s10570-021-04058-4>.
- [4] Xiandong Zhou, Yangyiwei Yang, Somnath Bharech, **Binbin Lin**, Jörg Schröder, Bai-Xiang Xu. "3D-multilayer simulation of microstructure and mechanical properties of porous materials by selective sintering", In: *GAMM-Mitteilungen* 44 (4), e202100017, (2021), <https://doi.org/10.1002/gamm.202100017>.
- [5] **Binbin Lin**, Julia Auernhammer, Jan-Lukas Schäfer, Tobias Meckel, Robert Stark, Markus Biesalski, Bai-Xiang Xu, "Humidity influence on mechanics of paper materials: joint numerical and experimental study on fiber and fiber network scale", In: *Cellulose* 29, 1129–1148 (2022), <https://doi.org/10.1007/s10570-021-04355-y>.

-
- [6] **Binbin Lin**, Nima Emami, David A Santos, Yuting Luo, Sarbajit Banerjee, Bai-Xiang Xu, "A deep learned nanowire segmentation model using synthetic data augmentation", In: npj Comput Mater 8, 88 (2022), <https://doi.org/10.1038/s41524-022-00767-x>.
- [7] David A Santos, Justin L Andrews, **Binbin Lin**, Luis R De Jesus, Yuting Luo, Savannah Pas, Michelle A Gross, Luis Carillo, Peter Stein, Yu Ding, Bai-Xiang Xu, Sarbajit Banerjee, "Multivariate hyperspectral data analytics across length scales to probe compositional, phase, and strain heterogeneities in electrode materials", In: Patterns 3, 100634 (2022), <https://doi.org/10.1016/j.patter.2022.100634>.
- [8] David A Santos, **Binbin Lin**, Justin L Andrews, "Data science enables X-ray vision", In: Patterns 3, 100660 (2022), <https://doi.org/10.1016/j.patter.2022.100660>.
- [9] David A Santos, Shahed Rezaei, Delin Zhang, Yuting Luo, **Binbin Lin**, Ananya R Balakrishna, Bai-Xiang Xu, Sarbajit Banerjee, "Chemistry–mechanics–geometry coupling in positive electrode materials: a scale-bridging perspective for mitigating degradation in lithium-ion batteries through materials design", In: Chemical Science 14, 458-484 (2023), <https://doi.org/10.1016/j.patter.2022.100634>.
- [10] **Binbin Lin**, Setareh Medghalchi, Sandra Korte-Kerzel, Bai-Xiang Xu, "A Machine Learning Enabled Image-data-driven End-to-end Mechanical Field Predictor For Dual-Phase Steel", In: PAMM 22, e202200110 (2023), <https://doi.org/10.1002/pamm.202200110>.
- [11] Xing Chen, **Binbin Lin**, Miriam Schuster, Suwen Chen, Bai-Xiang Xu, Jens Schneider, "Effect of moisture on the delamination properties of fractured PVB-laminated glass: A joint experimental and numerical study", In: Composite Structures 322 (2023): 117381, <https://doi.org/10.1016/j.compstruct.2023.117381>.

Oral Presentations, Reports & Posters

- [1] **Binbin Lin**, Bai-Xiang Xu, "Micromechanical Modelling and Simulation of Functional Paper Materials", FiPRE Status Meeting (12-2018), Darmstadt, Germany.
- [2] **Binbin Lin**, Yang Bai, Peter Stein, Bai-Xiang Xu, "Hygro-micromechanical analysis of the interfiber bonding in a fiber-network", 90th GAMM Annual Meeting (02-2019), Vienna, Austria.
- [3] **Binbin Lin**, Bai-Xiang Xu, "Micromechanical Modelling and Simulation of Functional Paper Materials", FiPRE Status Meeting (05-2019), Rossdorf, Germany.
- [4] **Binbin Lin**, "Microstructure generation with Geodict", MFM Fall Workshop (09-2019), Hadamar, Germany.
- [5] **Binbin Lin**, Bai-Xiang Xu, "Influence of geometrical sensitivity on micromechanical properties of fibrous paper materials", FiPRE Status Meeting (02-2020), Darmstadt, Germany.
- [6] **Binbin Lin**, Bai-Xiang Xu, "Micromechanical Modelling and Simulation of Functional Paper Materials", FiPRE Status Meeting (10-2020), Darmstadt, Germany.
- [7] **Binbin Lin**, Bai-Xiang Xu, "Data-driven microstructure sensitivity study in paper materials", Salon on machine learning for material science (12-2020), Darmstadt, Germany.
- [8] **Binbin Lin**, Mozhdeh Fathidoost, Josquin Vandeputte, "Enhancing the thermal conductivity of composite materials - a Bayesian optimization approach", Salon on machine learning for material science (12-2020), Darmstadt, Germany.
- [9] Zhen liu, **Binbin Lin**, Mozhdeh Fathidoost, "Finding new antiferroelectric materials - a classification study with data augmentation", Salon on machine learning for material science (12-2020), Darmstadt, Germany.
- [10] **Binbin Lin**, "Particle segmentation using deep Learning", Salon on machine learning for material science (12-2020), Darmstadt, Germany.

-
- [11] **Binbin Lin**, Bai-Xiang Xu "Data-driven modelling of functional paper materials", 16th U.S. National Congress on Computational Mechanics (07-2021), Online.
 - [12] **Binbin Lin**, Bai-Xiang Xu, "Conceptions on ML-based web application for dual-phase steels", NHR4CES Status Report (10-2021), Online.
 - [13] **Binbin Lin**, "Chaining simulations steps using Python with subprocesses", MFM Spring Workshop (03-2022), Darmstadt, Germany.
 - [14] **Binbin Lin**, "Introduction to physics-informed neural networks", MFM Spring Workshop (03-2022), Darmstadt, Germany.
 - [15] **Binbin Lin**, Bai-Xiang Xu, "Image-based micromechanical modelling and simulation of dual-phase steels ", NHR4CES Meeting (04-2022), Online, Germany.
 - [16] **Binbin Lin**, Bai-Xiang Xu, "A ML-enabled image-data-driven end-to-end field predictor for dual-phase steels", 92th GAMM Annual Meeting (08-2022), Aachen, Germany.
 - [17] Sebastian Wissel, Mozhdeh Fatidoost, **Binbin Lin**, "Gmsh basics & Introduction to Geodict software", MFM Fall Workshop (09-2022), Grasellenbach, Germany.
 - [18] Jean Kunkler, **Binbin Lin**, "Tutorial for ML crash course", MFM Fall Workshop (09-2022), Grasellenbach, Germany.
 - [19] **Binbin Lin**, Bai-Xiang Xu, "A deep learned nanowire segmentation model using synthetic data augmentation", Poster, MSE Congress (09-2022), Darmstadt, Germany.
 - [20] **Binbin Lin**, Bai-Xiang Xu, "Data-driven Microstructure sensitivity study in fibrous paper materials", MSE Congress (09-2022), Darmstadt, Germany.
 - [21] **Binbin Lin**, "A ML-enabled image data-driven web application for dual-phase steels", NHR4CES Meeting (09-2022), Darmstadt, Germany.
 - [22] **Binbin Lin**, "Data-driven research in material mechanics", Mawi Day (10-2022), Darmstadt, Germany.
 - [23] **Binbin Lin**, "Science communication", MFM Spring Workshop (03-2023), Bingen, Germany.

-
- [24] Setareh Medghalchi, **Binbin Lin**, Josha Kortmann, Bai-Xiang Xu, Ulrich Kerzel, Sandra Korte-Kerzel, "Three-dimensional characterization of damage sites and strain dependence in dual-phase steel", DPG Spring Meetings (03-2023), Hannover, Germany.