

# **Beschleunigtes Materialdesign durch künstliche Intelligenz im Forschungsdatenmanagement**

Zur Erlangung des akademischen Grades eines  
**Doktors der Ingenieurwissenschaften (Dr.-Ing.)**  
von der KIT-Fakultät für Maschinenbau des  
Karlsruher Instituts für Technologie (KIT)

genehmigte  
**Dissertation**

von

Lars Griem, M.Sc.

Tag der mündlichen Prüfung:	10. November 2025
Referentin:	Prof. Dr. Britta Nestler
Korreferent:	Prof. Dr. Markus Reischl



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz (CC BY-SA 4.0): <https://creativecommons.org/licenses/by-sa/4.0/deed.de>





# Kurzfassung

Die Entwicklung neuer Materialien steht zunehmend unter dem Druck, schneller, effizienter und nachhaltiger zu erfolgen. Klassische, experimentell geprägte Ansätze sind hierfür oft zu zeit- und kostenintensiv. Digitale Methoden eröffnen die Möglichkeit, Materialentwicklungsprozesse systematisch zu beschleunigen, indem Daten, Simulationen und maschinelles Lernen in einem konsistenten Forschungsdatenmanagement zusammengeführt werden.

Diese Arbeit adressiert diese Herausforderung am Beispiel offenzelliger Polyurethanschaumstrukturen aus dem Fahrzeugleichtbau, für die Struktur-Eigenschaft-Beziehungen untersucht werden. Hierfür wird ein vollständig digitaler Entwicklungsprozess realisiert, der auf generischen, modularen Workflows basiert und in der virtuellen Forschungsumgebung Kadi4Mat verankert ist. Grundlage bildet ein Konzept zum FAIRen Materialdesign, das mit Kadi4Mat umgesetzt wird. Zentrales Rückgrat ist eine in Kadi4Mat geführte Materialdatenbank, die fortlaufend mit Rekonstruktionen, generierten Mikrostrukturen und Simulationsergebnissen befüllt wird und aus der die Struktur-Eigenschaft-Beziehungen extrahiert werden.

Als Ausgangspunkt der Untersuchung wird eine Methode zur schnellen und präzisen Rekonstruktion digitaler Zwillinge aus CT-Daten entwickelt. Ein Segmentierungsnetzwerk, das auf synthetisch erzeugten Trainingsdaten basiert, ermöglicht hochgenaue Ergebnisse, während die geometrische Charakterisierung der Mikrostrukturen in einem automatisierten Workflow erfolgt.

Zur Erweiterung des Strukturraums wird ein 3D-Diffusion-GAN vorgestellt, das realistische Mikrostrukturen mit gezielt einstellbaren Eigenschaften erzeugt. Die

Methode übertrifft etablierte, geometriebasierte Ansätze sowohl hinsichtlich Genauigkeit als auch Geschwindigkeit. Neben einer eigenschaftskonditionierten Variante wird eine bildkonditionierte Version implementiert, die ohne explizite Merkmalsextraktion auskommt.

Die makroskopischen mechanischen Eigenschaften werden in einem phasenfeldbasierten Simulationsworkflow ermittelt, der anhand experimenteller Daten realer Schaumproben validiert wurde. Dadurch wird eine hohe Übereinstimmung zwischen Simulation und Experiment erreicht.

Die erzeugten und simulierten Strukturen fließen in eine Materialdatenbank ein, die mit einem Analyseworkflow ausgewertet wird. Dieser kombiniert maschinelles Lernen mit Methoden der erklärbaren KI und identifiziert verständliche Struktur-Eigenschaft-Beziehungen. Das trainierte Modell sagt mechanische Eigenschaften mit hoher Genauigkeit auf Basis weniger Struktureigenschaften vorher. Die Ergebnisse werden in Form interpretierbarer Materialkarten dargestellt und können unmittelbar für das Materialdesign genutzt werden.

Insgesamt zeigt die Arbeit, dass sich der gesamte Materialentwicklungsprozess von der Datenerfassung über Modellierung und Simulation bis zur datengetriebenen Analyse in einer einheitlichen, reproduzierbaren und skalierbaren Struktur abbilden lässt. Die entwickelten Workflows sind generisch, modular und FAIR dokumentiert, sodass sie auf weitere Materialsysteme übertragbar sind. Der Einsatz maschineller Lernverfahren automatisiert zentrale Arbeitsschritte und steigert die Effizienz des Entwicklungsprozesses deutlich.

Am Anwendungsfall der Schaumstrukturen wird demonstriert, dass mechanische Eigenschaften wie Druckmodul und Stauchgrenze mit hoher Genauigkeit aus wenigen mikrostrukturellen Eigenschaften vorhergesagt werden können. Dies reduziert den Bedarf an zeit- und kostenintensiven Experimenten erheblich und ermöglicht die gezielte Auswahl vielversprechender Strukturen. Die Arbeit liefert damit sowohl einen methodischen Beitrag zum generischen, datengetriebenen Materialdesign als auch einen konkreten Mehrwert für die Entwicklung energieabsorbierender Leichtbauelemente.

# Abstract

The development of new materials is increasingly under pressure to proceed faster, more efficiently, and more sustainably. Traditional, experiment-driven approaches are often too time- and cost-intensive. Digital methods provide a promising pathway to systematically accelerate material development processes by integrating data, simulations, and machine learning within a consistent research data management framework.

This thesis addresses this challenge using the example of open-cell polyurethane foam structures for automotive lightweight design. A fully digital development process is developed, based on generic, modular workflows anchored in the virtual research environment Kadi4Mat. The foundation is a FAIR material design concept, realized through Kadi4Mat. At its core, a materials database within Kadi4Mat serves as the backbone, continuously populated with reconstructions, generated microstructures, and simulation results, from which the structure–property linkages are extracted.

As a starting point, a method is developed for the rapid and precise reconstruction of digital twins from CT data. A segmentation network trained on synthetically generated datasets enables highly accurate results, while the geometric characterization of the microstructures is carried out in an automated workflow.

To expand the accessible design space, a 3D diffusion GAN is introduced that generates realistic microstructures with controllable properties. The method outperforms established geometry-based approaches in both accuracy and computational efficiency. In addition to a property-conditioned version, an image-conditioned variant is implemented that does not require explicit feature extraction.

Macroscopic mechanical properties are determined using a phase-field-based simulation workflow, validated against experimental data from real foam specimens. This ensures strong agreement between simulations and experiments.

The generated and simulated structures are integrated into a curated material database, which is analyzed using a workflow that combines machine learning with explainable AI techniques. This approach enables the identification of comprehensive structure–property relationships. The trained model predicts mechanical properties with high accuracy based on only a few structural descriptors. The results are presented in the form of interpretable material maps, which can be directly applied to material design.

Overall, this work demonstrates that the entire material development process—from data acquisition through modeling and simulation to data-driven analysis—can be represented within a unified, reproducible, and scalable framework. The developed workflows are generic, modular, and FAIR-documented, making them transferable to other material systems. The use of machine learning automates key steps and significantly increases the efficiency of the development process.

The application to polyurethane foam structures shows that mechanical properties such as compressive modulus and plateau stress can be accurately predicted from a small set of microstructural properties. This substantially reduces the need for time- and cost-intensive experiments and enables the targeted selection of promising structures. Thus, the thesis contributes both a methodological foundation for generic, data-driven material design and a concrete added value for the development of energy-absorbing lightweight components.

# Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Angewandte Materialien – Mikrostruktur-Modellierung und Simulation (IAM-MMS) sowie am Institut für Nanotechnologie – Microstructure Simulation (INT-MSS) des Karlsruher Instituts für Technologie (KIT).

Mein besonderer Dank gilt Frau Prof. Dr. Britta Nestler für die wertvolle Betreuung meiner Arbeit sowie für die Möglichkeit, meine Dissertation an ihrem Institut mit großer wissenschaftlicher Freiheit anzufertigen. Ebenso danke ich Herrn Prof. Dr. Markus Reischl für die Bereitschaft, das Korreferat dieser Arbeit zu übernehmen.

Für die zahlreichen fachlichen Diskussionen, die mir immer wieder geholfen haben, den Blick auf das große Ganze zu richten und die unterschiedlichen Bereiche meiner Arbeit miteinander zu verbinden, danke ich herzlich Dr. Michael Selzer, Dr. Arnd Koeppe sowie meinem ehemaligen Gruppenleiter Dr. Patrick Altschuh. Mein Dank gilt außerdem meinen aktuellen und ehemaligen Kolleginnen und Kollegen, die durch ihre Unterstützung und das angenehme Arbeitsumfeld entscheidend dazu beigetragen haben, dass ich stets Spaß bei der Arbeit hatte. Besonders erwähnen möchte ich Willfried Kunz, dem ich für die vielen aufmunternden Gespräche und für seine Hilfe beim Korrekturlesen meiner Arbeit sehr dankbar bin. Ebenso danke ich meinem Bruder Sven Griem, der als fachfremder Leser durch seine Korrekturen wesentlich dazu beigetragen hat, die Arbeit sprachlich ansprechend zu gestalten.

Mein tief empfundener Dank gilt meiner Frau Beliz, die mir während meiner gesamten Promotionszeit und insbesondere in den letzten Monaten vor der Abgabe

den Rücken freigehalten hat, mir gleichzeitig aber auch die nötige Motivation und den entscheidenden Anstoß gegeben hat, um meine Arbeit erfolgreich abzuschließen.

Abschließend danke ich meiner Familie, die mir stets ein verlässlicher Rückhalt in allen Lebenslagen ist.

Karlsruhe, im September 2025

*Lars Griem*

# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Danksagung</b>	<b>v</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Einordnung und Motivation	1
1.2 Zielsetzung und Beitrag der Arbeit	4
1.3 Stand der Forschung	5
1.4 Gliederung der Arbeit	11
<b>2 Theoretische Grundlagen</b>	<b>13</b>
2.1 Maschinelles Lernen	13
2.1.1 Neuronale Netzwerke	14
2.1.2 Trainingsprozess neuronaler Netze	20
2.1.3 Regularisierung und Trainingsstrategien	26
2.1.4 Normalisierung	29
2.1.5 Convolutional Neural Networks	30
2.2 Generative Modelle	32
2.2.1 Generative Adversarial Networks	32
2.2.2 Denoising Diffusion Probabilistic Model	34
2.3 Phasenfeldmethode	36
2.4 Repräsentatives Volumenelement	38
2.5 Forschungsdatenmanagement	40
2.5.1 Forschungsdaten	41



2.5.2	Metadaten . . . . .	42
2.5.3	Ontologien . . . . .	43
2.5.4	FAIR Prinzipien . . . . .	44
2.5.5	Repositorium . . . . .	45
2.5.6	Elektronisches Laborbuch . . . . .	46
2.5.7	Workflowsysteme . . . . .	47
<b>3</b>	<b>Einheitliches Konzept für FAIRes Materialdesign . . . . .</b>	<b>49</b>
3.1	FAIRes Materialdesign . . . . .	49
3.2	Prozessstrukturierung . . . . .	52
3.2.1	Konzept . . . . .	52
3.2.2	KadiStudio . . . . .	58
3.3	Datendokumentation . . . . .	68
3.3.1	Konzept . . . . .	68
3.3.2	KadiWeb . . . . .	71
3.4	Problemspezifisches Konzept . . . . .	75
<b>4</b>	<b>Rekonstruktion und Charakterisierung digitaler Zwillinge . . . . .</b>	<b>79</b>
4.1	Stand der Forschung . . . . .	80
4.1.1	Rekonstruktion . . . . .	80
4.1.2	Segmentierung . . . . .	82
4.1.3	Geometrische Charakterisierung . . . . .	84
4.2	Methoden . . . . .	85
4.2.1	Rekonstruktion . . . . .	85
4.2.2	Segmentierung der Bilddaten . . . . .	87
4.2.3	Geometrische Charakterisierung . . . . .	91
4.2.4	Datengetriebene Charakterisierung . . . . .	96
4.3	Verwendete Datensätze . . . . .	100
4.3.1	Polyurethanschaum . . . . .	100
4.3.2	Experimentelle Charakterisierung . . . . .	101
4.3.3	CT Bildgebung . . . . .	104
4.4	Rekonstruktion . . . . .	105
4.4.1	Datengenerierung . . . . .	105

4.4.2	U-Net Finetuning . . . . .	116
4.4.3	Experimentelle Untersuchung . . . . .	119
4.5	Geometrische Charakterisierung . . . . .	121
4.6	Einbindung in Kadi4Mat . . . . .	124
4.6.1	Rekonstruktion . . . . .	124
4.6.2	Charakterisierung . . . . .	128
4.7	Diskussion . . . . .	130
<b>5</b>	<b>ML-gestützte Generierung digitaler Zwillinge . . . . .</b>	<b>133</b>
5.1	Stand der Forschung . . . . .	134
5.1.1	Geometriebasierte Generierung . . . . .	134
5.1.2	Physikbasierte Generierung . . . . .	136
5.1.3	ML-basierte Generierung . . . . .	137
5.2	Validierung digitaler Zwillinge . . . . .	140
5.2.1	Einstellbarkeit der Struktureigenschaften . . . . .	140
5.2.2	Realismus der Strukturen . . . . .	141
5.3	Geometriebasierte Generierung . . . . .	141
5.3.1	Schaumgenerierung . . . . .	142
5.3.2	Nachbearbeitung . . . . .	145
5.4	Datengetriebene Generierung mittels Diffusionsmodellen . . . . .	154
5.4.1	DD-GAN . . . . .	155
5.4.2	Trainingsdaten . . . . .	165
5.4.3	Validierung . . . . .	172
5.4.4	Effizienz des Generators . . . . .	182
5.4.5	Validierung des Diskriminators . . . . .	184
5.4.6	Bildkonditioniertes DD-GAN . . . . .	187
5.4.7	Evaluation des bildkonditionierten Generators . . . . .	192
5.4.8	Finetuning . . . . .	197
5.5	Einbindung in Kadi4Mat . . . . .	200
5.5.1	DD-GAN Training . . . . .	201
5.5.2	Generierung von Strukturen . . . . .	202
5.6	Diskussion . . . . .	204

<b>6</b>	<b>Bestimmung der physikalischen Schaumeigenschaften . . .</b>	<b>207</b>
6.1	Stand der Forschung . . . . .	208
6.2	Phasenfeldmethode für Mechanikmodellierung . . . . .	209
6.2.1	Modell . . . . .	210
6.2.2	Simulationssetup . . . . .	212
6.2.3	RVE Bestimmung . . . . .	212
6.2.4	Parametrisierung der Simulation . . . . .	214
6.3	Experimentelle Untersuchungen . . . . .	214
6.3.1	Ziel und Versuchsaufbau . . . . .	215
6.3.2	Auswertung der Druckversuche . . . . .	215
6.3.3	Abgleich mit Simulationen . . . . .	217
6.4	Einbindung in Kadi4Mat . . . . .	218
6.5	Diskussion . . . . .	221
<b>7</b>	<b>Identifikation von Struktur-Eigenschaft-Beziehungen . . . .</b>	<b>223</b>
7.1	Stand der Forschung . . . . .	224
7.2	Methoden zur Analyse neuronaler Netze . . . . .	226
7.3	Aufbau der Datenbank . . . . .	227
7.4	Analyse der Datenbank . . . . .	234
7.4.1	Modellarchitektur und Training . . . . .	234
7.4.2	Identifikation relevanter Strukturparameter . . . . .	235
7.4.3	Modellreduktion und Validierung . . . . .	237
7.4.4	Fehleranalyse . . . . .	239
7.4.5	Erstellung der Materialkarten . . . . .	241
7.5	Einbindung in Kadi4Mat . . . . .	242
7.5.1	Datenvorbereitung . . . . .	243
7.5.2	Training . . . . .	244
7.5.3	Erstellung der Materialkarten . . . . .	245
7.6	Diskussion . . . . .	247
<b>8</b>	<b>Zusammenfassung und Ausblick . . . . .</b>	<b>249</b>
8.1	Zusammenfassung . . . . .	249
8.2	Ausblick . . . . .	252

<b>Literaturverzeichnis . . . . .</b>	<b>257</b>
<b>Abbildungsverzeichnis . . . . .</b>	<b>283</b>
<b>Tabellenverzeichnis . . . . .</b>	<b>299</b>
<b>Abkürzungsverzeichnis . . . . .</b>	<b>301</b>
<b>A Anhang . . . . .</b>	<b>303</b>
<b>B Veröffentlichungen . . . . .</b>	<b>307</b>



# 1 Einleitung

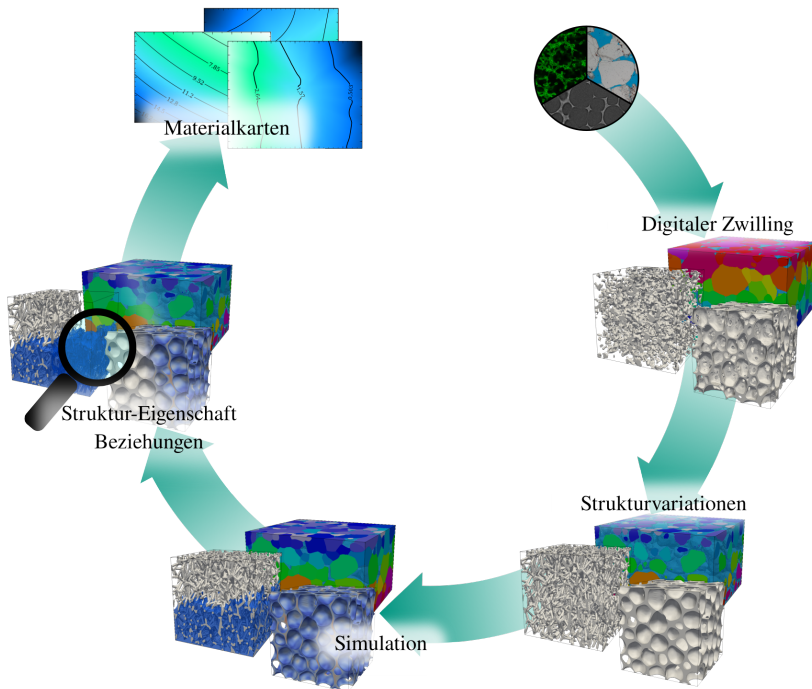
## 1.1 Einordnung und Motivation

Moderne Materialien bilden das Rückgrat technologischer Innovation. Die Leistungsfähigkeit von Windkraftanlagen, die Reichweite von Elektrofahrzeugen oder die Sicherheit medizinischer Implantate hängt maßgeblich von den eingesetzten Werkstoffen ab. So ermöglichen etwa neuartige Festkörperbatterien höhere Energiedichten bei gleichzeitig verbesserter Sicherheit. Allerdings scheitert ihr breiter Einsatz bislang an der Verfügbarkeit skalierbarer Elektrolyte. Ähnliche Herausforderungen zeigen sich in der Entwicklung wasserstoffbasierter Energiesysteme, biokompatibler Kunststoffe oder leistungsfähiger Leichtbaumaterialien.

Diese Beispiele zeigen: Der Fortschritt in Wissenschaft und Technik erfordert neue, maßgeschneiderte Materialien und das deutlich schneller als bisher. Angesichts globaler Herausforderungen wie der Klimakrise oder zukünftiger Gesundheitsrisiken steigt der Druck, Materialentwicklungsprozesse effizienter und systematischer zu gestalten.

Die digitale Materialforschung bietet hierfür einen vielversprechenden Ansatz. Durch die Kombination experimenteller Daten, physikalisch fundierter Simulationen und umfangreicher Materialdatenbanken entsteht ein leistungsfähiges Werkzeug für beschleunigtes Materialdesign. Im Zentrum steht der digitale Materialzwillings: ein konsistentes, virtuelles Modell, das zentrale Merkmale eines Werkstoffs wie Zusammensetzung, Kristall- und Mikrostruktur und oft auch seine Prozesskette abbildet.

Ausgehend von diesem Modell lassen sich gezielt generierte Varianten der Materialzusammensetzung oder -morphologie simulieren und deren Einfluss auf makroskopische Eigenschaften bewerten. So können systematisch Struktur-Eigenschaft-Beziehungen untersucht werden, die eine Brücke zwischen Mikrostruktur und Materialverhalten schlagen. Das Vorgehen ist in Abbildung 1.1 schematisch dargestellt. Es bildet die Grundlage für die gezielte Entwicklung neuer Werkstoffe, deren Eigenschaften im letzten Schritt experimentell validiert und zur Weiterentwicklung des digitalen Zwillings genutzt werden.



**Abbildung 1.1:** Allgemeiner Materialentwicklungskreislauf. Ausgehend von digitalen Zwillingen realer Strukturen werden Varianten mit gezielt veränderten Eigenschaften generiert. Ihre simulative Untersuchung liefert die zugehörigen makroskopischen Eigenschaften, aus deren Analyse sich Struktur–Eigenschaft–Beziehungen ableiten lassen.

Trotz dieser Fortschritte bestehen weiterhin zentrale Herausforderungen. Häufig fehlen standardisierte Methoden zur strukturierten Erfassung, Verwaltung und Analyse der entstehenden Daten. Stattdessen entstehen isolierte, projektspezifische Lösungen mit eingeschränkter Wiederverwendbarkeit. Dies erschwert den Aufbau fundierter Materialdatenbanken und behindert die Übertragbarkeit entwickelter Methoden, insbesondere bei der Analyse von Struktur-Eigenschaft-Beziehungen. Damit rückt die Etablierung des Forschungsdatenmanagements in den Vordergrund, das die FAIR-Prinzipien (Findable, Accessible, Interoperable, Reusable) berücksichtigt und so Transparenz, Nachvollziehbarkeit und Wiederverwendbarkeit gewährleistet.

Eine vielversprechende Methodik zur Bewältigung dieser Herausforderungen liegt in generischen, modular aufgebauten Workflows, die unabhängig vom Materialsystem einsetzbar sind. Basis hierfür ist ein aktives Forschungsdatenmanagement, das nicht nur Daten, sondern auch Modelle, Algorithmen und Auswertungsschritte systematisch erfasst und nachvollziehbar dokumentiert. Dadurch lassen sich vollständige Prozessketten reproduzierbar abbilden und flexibel weiterentwickeln.

Maschinelle Lernverfahren erweitern diesen Ansatz entlang der gesamten Entwicklungskette, von der automatisierten Merkmalsextraktion über die Vorhersage makroskopischer Eigenschaften bis hin zum inversen Design. In Kombination mit einem konsequenten Forschungsdatenmanagement entsteht so eine skalierbare und übertragbare Basis für FAIRes Materialdesign.

Im Rahmen dieser Arbeit wird der in Abbildung 1.1 dargestellte Entwicklungsprozess in eine strukturierte Methodik überführt, die diese Prinzipien integriert und durch generische Workflows realisiert. Die Anwendbarkeit des Ansatzes wird am Beispiel einer Materialentwicklung aus dem Fahrzeugleichtbau validiert, bei der offenzellige Polyurethanschäumstrukturen als energieabsorbierende Crashelemente eingesetzt werden. Die Auslegung dieser Strukturen erfolgt mithilfe makroskopischer Simulationen, in denen der Schaum als homogenes Materialmodell beschrieben wird. Für deren Parametrisierung sind mechanische Kennwerte wie Druckmoduls und Stauchgrenze erforderlich, die bislang meist experimentell bestimmt werden, ein zeit- und kostenintensiver Prozess.



Zur Beschleunigung der Entwicklung wird ein datengetriebener Ansatz entwickelt, der mithilfe digitaler Zwillinge und maschinellen Lernens Zusammenhänge zwischen Mikrostruktur und makroskopischen Eigenschaften erschließt. So lassen sich geeignete Kandidaten gezielter auswählen und unnötige Testreihen vermeiden. Eingebettet in eine virtuelle Forschungsumgebung entsteht damit eine transparente und reproduzierbare Methodik für das FAIRe digitale Materialdesign.

## 1.2 Zielsetzung und Beitrag der Arbeit

Ziel dieser Arbeit ist die Entwicklung einer Methodik für den in Abbildung 1.1 skizzierten Materialentwicklungsprozess. Sie wird am Beispiel offenzelliger Polyurethanschäumstrukturen formalisiert und durch generische, modulare Workflows abgebildet.

Der Fokus liegt darauf, alle relevanten Prozessschritte, von der Datenerfassung über die Modellbildung und Simulation bis zur Identifikation von Struktur-Eigenschaft-Beziehungen, so zu gestalten, dass sie reproduzierbar, skalierbar und auf andere Materialsysteme übertragbar sind. Gleichzeitig werden für den gewählten Anwendungsfall konkrete Struktur-Eigenschaft-Beziehungen offenzelliger Polyurethanschäumstrukturen untersucht, um deren Eignung als energieabsorbierende Leichtbauelemente charakterisieren.

Als methodisches Fundament wird ein Konzept für FAIRes Materialdesign entwickelt, das Daten, Modelle, Algorithmen und Auswertungsschritte konsistent strukturiert und nachvollziehbar dokumentiert. Auf dieser Basis werden die Workflows in einer virtuellen Forschungsumgebung reproduzierbar integriert.

Darüber hinaus werden maschinelle Lernverfahren gezielt eingesetzt, um zentrale Prozessschritte zu automatisieren, ihre Generalisierbarkeit zu erhöhen und die Entwicklungsgeschwindigkeit zu steigern.

Im Ergebnis entsteht eine Methodik, die durch durchgängige Digitalisierung, Modularisierung und Dokumentation, aufwändige experimentelle Testreihen reduziert und ein datengetriebenes, generisches Materialdesign ermöglicht. Im Anwendungsfall wird sie genutzt, um Struktur-Eigenschaft-Beziehungen offenzelliger Polyurethanschaumstrukturen zu analysieren und deren Potenzial als energieabsorbierende Leichtbauelemente aufzuzeigen. Damit leistet diese Arbeit sowohl einen Beitrag zur Etablierung einer übertragbaren Methodik für zukünftige Projekte im Bereich der digitalen Materialforschung als auch konkrete materialwissenschaftliche Einblicke in die Eigenschaften von Polyurethanschaumstrukturen.

## 1.3 Stand der Forschung

Ein zentrales Fundament für digitales und FAIRes Materialdesign ist ein leistungsfähiges Forschungsdatenmanagement (FDM). Es stellt sicher, dass Daten, Modelle und Auswertungsschritte nicht nur transparent dokumentiert, sondern auch reproduzierbar und nachnutzbar sind. Damit bildet FDM die methodische Grundlage für die in dieser Arbeit entwickelte Methodik.

In den vergangenen Jahren wurden zahlreiche Infrastrukturen etabliert, die jeweils auf unterschiedliche Aspekte des FDM fokussieren. Im Folgenden werden zentrale Entwicklungen in den Bereichen *Repositorien*, *Elektronische Laborbücher* und *Workflowsysteme* vorgestellt.

### Repositorien

Repositorien stellen den Kern jeder Forschungsdateninfrastruktur dar, da sie Daten langfristig sichern, versionieren und über persistente Identifikatoren zitierbar machen. Damit unterstützen sie die Umsetzung der FAIR-Prinzipien und schaffen die Basis für anschließende Analysen und Workflows.

Zenodo ist ein von CERN (Europäischen Organisation für Kernforschung) und OpenAIRE entwickeltes, disziplinübergreifendes Repositorium für Datensätze,

Software, Publikationen und andere Forschungsartefakte [1]. Forschende können alle gängigen Dateiformate ohne Beschränkungen bezüglich Größe, Lizenz oder Zugangshürden hochladen und erhalten für jedes Projekt einen persistenten DOI. Die Plattform erfüllt auf diese Weise die FAIR-Prinzipien. Sowohl eine Web-Oberfläche als auch eine REST-API erlauben manuelles und automatisiertes Management der Daten. Für die Beschreibung setzt Zenodo auf Metadaten-schemata, die sich stark an dem DataCite-Schema orientieren und so eine hohe Interoperabilität bieten.

Ein weiteres verbreitetes Repositorium ist Dataverse. Dataverse ist eine Open-Source-Webanwendung, die an der Harvard University entwickelt und weltweit betrieben wird (z. B. Harvard Dataverse, DataverseNL). Sie richtet sich an alle Disziplinen und verfolgt das Ziel, das Publizieren, Zitieren und Analysieren von Forschungsdaten zu vereinfachen [2]. Auch hier erhalten Datensätze persistente DOIs, begleitet von standardisierten Metadaten, um die FAIR-Prinzipien sicherzustellen. Mit granularen Zugriffsrechten, Versionierung und API-Schnittstellen lässt sich Dataverse nahtlos in Forschungsprozesse integrieren.

Im Gegensatz zu den vorstehend eingeführten, allgemeinen Repositorien ist die Materials Data Facility (MDF) speziell auf die Materialwissenschaften ausgerichtet und unterstützt gleichermaßen experimentelle und simulative Datensätze [3, 4]. MDF gliedert sich in zwei Dienste: (i) Der Data Publication Service (DPS) ermöglicht das strukturierte Hochladen von Daten beliebiger Größe, versieht sie automatisch mit projektspezifischen Metadaten und vergibt persistent DOIs wodurch die Daten zitierfähig sind. (ii) Der Data Discovery Service (DDS) erlaubt über eine detaillierte Suche, Volltext- und Bereichsabfragen gezielt auf alle veröffentlichten Datensätze sowie auf externe Datenquellen zuzugreifen. Auch hier kann der Zugriff auf das Repositorium über APIs automatisiert werden.

Neben den bereits vorgestellten Repositorien existiert eine Vielzahl weiterer fachspezifischer und disziplinübergreifender Plattformen, die je nach Forschungsgebiet oder Datentyp eingesetzt werden können. Um innerhalb der umfangreichen Auswahl das passende Archiv zu identifizieren, bieten spezialisierte Suchportale eine wertvolle Orientierung. So bildet re3data die derzeit weltweit umfangreichste

Datenbank für Forschungsrepositorien und erlaubt es, anhand von Kriterien wie Fachbereich, Datentyp oder Metadatenschema gezielt nach geeigneten Diensten zu suchen [5].

Die von der Deutschen Forschungsgemeinschaft (DFG) betriebene Plattform RISource konzentriert sich hingegen auf deutsche Forschungsinfrastrukturen und stellt detaillierte Informationen zu Repositorien, Netzwerken und Diensten bereit, die von der DFG gefördert werden [6]. Beide Portale erleichtern durch strukturierte Übersichten und vergleichende Filterfunktionen die Auswahl des optimalen Repositoriums für ein spezifisches Forschungsprojekt.

## **Elektronische Laborbücher**

Elektronische Laborbücher (ELNs) bilden das operative Rückgrat moderner Forschungsprozesse. Sie ermöglichen die strukturierte Erfassung von Experimenten, Simulationen und Ressourcen und stellen damit sicher, dass Daten bereits bei ihrer Erzeugung mit den notwendigen Metadaten verknüpft werden. Damit leisten sie einen entscheidenden Beitrag zur Nachvollziehbarkeit und Reproduzierbarkeit. Somit ergänzen sie Repositorien, die vor allem die Publikation und Langzeitarchivierung adressieren.

ElabFTW [7] ist eine quelloffene, webbasierte Lösung für elektronische Laborbücher und gehört zum ELN Consortium, das Standards für ELN-Systeme entwickelt. Mit elabFTW lässt sich die gesamte Laborarbeit digital abbilden. Neben dem klassischen Protokollieren von Experimenten und deren Ergebnissen bietet die Software zusätzlich Managementfunktionen für Equipment und Verbrauchsmaterialien. Über den integrierten Scheduler können Geräte gebucht sowie deren Belegungsstatus transparent verwaltet werden. Parallel dazu kann der Bestand von Verbrauchsgütern wie chemischen Reagenzien, Antikörpern oder Plasmiden erfasst und nachverfolgt werden. ElabFTW ist plattformunabhängig und kann sowohl über einen Desktop-Browser als auch mobil genutzt werden. Für eine Einbindung in automatisierte Workflows stellt es zudem eine REST-API bereit. Durch

diese Kombination aus Protokollierung, Ressourcenmanagement und Schnittstellen hat sich elabFTW als weltweit genutztes elektronisches Laborbuch etabliert.

Eine weitere quelloffene ELN-Lösung ist Chemotion, die speziell auf die Anforderungen der Chemiewissenschaft ausgerichtet ist [8]. Im Unterschied zu generalistischen Systemen wie elabFTW bietet Chemotion neben den üblichen Funktionen zum Planen, Dokumentieren und Teilen von Laborabläufen insbesondere chemiespezifische Werkzeuge. Eine zentrale Funktion sind die integrierten Molekül- und Reaktionseditoren, mit denen Nutzer Strukturen zeichnen, Reaktionsschemata anlegen und experimentelle Bedingungen festhalten können. Die Ergebnisse der chemischen Reaktionen lassen sich anschließend direkt im System analysieren und in der Datenbank ablegen. Dank leistungsfähiger Suchfunktionen können dabei sowohl die intern gespeicherten Datensätze als auch angebundene externe Quellen gezielt durchsucht werden. Darüber hinaus unterstützt Chemotion die Automatisierung der Datendokumentation durch direkte Anbindung an Laborgeräte [9]. Messwerte und Protokolle können dadurch automatisch erfasst und mit Metadaten verknüpft werden. Durch diese Kombination aus fachspezifischen Editoren, flexiblen Suchmöglichkeiten und Geräteintegration stellt Chemotion eine umfassende Plattform für das digitale Chemielabor dar.

## Workflowsysteme

Zusammen mit Repositorien und elektronischen Laborbüchern bilden Workflowsysteme einen zentralen Baustein moderner Forschungsdateninfrastrukturen. Während Repositorien vor allem der langfristigen Sicherung und Veröffentlichung von Daten dienen und ELNs deren strukturierte Erfassung im Forschungsalltag unterstützen, adressieren Workflowsysteme die automatisierte Verknüpfung und Ausführung komplexer Analysen. Damit sind sie besonders relevant für datenintensive Disziplinen wie die Materialwissenschaften.

Infrastrukturen, welche die Erstellung, den Austausch und die Ausführung von Workflows integrieren, sind bislang in verschiedenen Implementierungen realisiert,

beispielsweise Jupyter Notebooks [10], Galaxy [11] und AiiDA [12, 13]. Diese unterscheiden sich teilweise erheblich in ihrem Bedienkonzept, ihrem Zielpublikum und ihren Automatisierungsfähigkeiten. Jupyter Notebooks beispielsweise ist eine quelloffene, domänenunspezifische Umgebung, die Code, erläuternden Text und Visualisierungen in einem interaktiven Dokument kombiniert [10]. Forschende können Analyseschritte schrittweise ausführen und direkt nachvollziehen, wie sich Änderung im Code oder den Parametern auf die Ergebnisse auswirken. Diese enge Verknüpfung von Programmcode und beschreibenden Kommentaren fördert nicht nur ein intuitives Nachvollziehen, sondern auch die Reproduzierbarkeit von Experimenten und Simulationen. Standardmäßig erfordert Jupyter eine manuelle Bedienung, lässt sich jedoch beispielsweise mit Tools wie Papermill [14] automatisiert einsetzen.

Galaxy wurde ursprünglich im Bereich der Biowissenschaften entwickelt und verfolgt ein graphisches Bedienprinzip [15]. Bei diesem können über eine grafische Oberfläche mittels Drag&Drop komplexe Analysepipelines aus vorkonfigurierten Modulen zusammengestellt werden. Mit dem *Tool Shed* stellt Galaxy dabei ein breites Portfolio bioinformatischer Analysen wie Statistik- und Visualisierungsmodule zur Verfügung. Indem Galaxy die Notwendigkeit von Programmierkenntnissen weitgehend eliminiert, ermöglicht es auch Forschenden ohne entsprechende Kenntnisse, reproduzierbare Workflows zu erstellen und zu teilen.

AiiDA hingegen ist ein rein skriptbasiertes Workflow-System, das in der rechnergestützten Forschung zum Einsatz kommt [12, 13]. Dabei verzichtet es bewusst auf die Einbeziehung manueller Arbeitsschritte. Es organisiert alle Rechenschritte und deren Datenherkunft in einem gerichteten azyklischen Graphen und automatisiert das Scheduling an lokalen oder entfernten Rechenknoten. Dank eines modularen Plugin-Systems lässt sich AiiDA für verschiedene Forschungsdisziplinen anpassen und eignet sich besonders für groß angelegte Simulationen wie Parameterstudien.

Ein weiteres Workflowsystem ist CLS-Luigi [16]. Dabei handelt es sich um ein Python-basiertes, quelloffenes Workflowsystem, das speziell für die automatisierte Generierung und Ausführung komplexer Analyse- und Machine-Learning-Pipelines entwickelt wurde. Sein zentrales Ziel ist es, parametrische Studien und Hyperparametersuchen effizient und reproduzierbar zu gestalten, ohne jede mögliche Pipeline manuell definieren zu müssen. Dazu nutzt CLS-Luigi den Combinatory Logic Synthesizer (CLS) [17], der aus einer Auswahl typisierter Bausteine anhand eines strikten Typensystems automatisch alle zulässigen Workflow-Varianten erzeugt. Diese werden anschließend an Spotify's Luigi [18] übergeben, das durch intelligentes Scheduling und Caching redundante Ausführungen vermeidet. Durch eine modulare Plugin-Architektur lassen sich gängige ML-Bibliotheken wie Scikit-Learn, TensorFlow oder Optimierer wie Gurobi sowie eigene Datenlade- und Vorverarbeitungsroutinen direkt integrieren. Auf diese Weise übernimmt CLS-Luigi sowohl die Zusammenstellung als auch die Ausführung der Pipelines, wodurch groß angelegte Studien erst praktikabel werden. Seine Kombination aus generativer Flexibilität, robustem Job-Management und offener Entwicklungsumgebung macht CLS-Luigi besonders attraktiv für skalierbare ML-Workflows.

## Fazit

Die dargestellten Repositorien, elektronischen Laborbücher und Workflowsysteme zeigen, dass wesentliche Bausteine für ein modernes Forschungsdatenmanagement bereits verfügbar sind und die Umsetzung der FAIR-Prinzipien in vielen Disziplinen unterstützen. Allerdings existieren bislang nur wenige Ansätze, die diese Komponenten zu einer durchgängigen, domänenspezifischen Methodik für die Materialwissenschaft integrieren und zugleich die Analyse von Struktur-Eigenschaft-Beziehungen systematisch unterstützen. Hier setzt die vorliegende Arbeit an. Durch die Kombination generischer Workflows, eines aktiven Forschungsdatenmanagements und maschineller Lernverfahren wird ein Ansatz entwickelt, der die bestehenden Lücken schließt und eine übertragbare Grundlage für FAIRes digitales Materialdesign schafft.

## 1.4 Gliederung der Arbeit

Die vorliegende Arbeit ist entlang des digitalen Materialentwicklungsprozesses aufgebaut, wie er in Abbildung 1.1 dargestellt ist. Jedes Kapitel behandelt einen Schritt in diesem Entwicklungszyklus. Im Zentrum steht dabei die Entwicklung generischer, modularer Workflows, die unabhängig vom konkreten Materialsystem einsetzbar und in eine virtuelle Forschungsumgebung eingebunden sind. Da jedes Kapitel einen spezifischen Teil des Materialdesigns abdeckt, wird in jedem Kapitel zunächst ein Stand der Forschung zu diesem konkreten Bereich gegeben.

Kapitel 2 stellt die theoretischen Grundlagen bereit, die für das Verständnis dieser Arbeit notwendig sind. Dazu gehören neben den Grundlagen des maschinellen Lernens auch die Phasenfeldmethode, die zur Simulation mechanischer Eigenschaften eingesetzt wird. Zudem wird das Konzept des repräsentativen Volumenelements (RVE) eingeführt, das eine zentrale Rolle bei der Untersuchung digitaler Zwillinge spielt. Abschließend werden die Prinzipien und Konzepte des Forschungsdatenmanagements erläutert, die als methodische Basis dieser Arbeit dienen.

Kapitel 3 bildet den Kern und Ausgangspunkt der Arbeit. In diesem Kapitel werden zwei komplementäre Konzepte für FAIRes Materialdesign entwickelt. Zum einen ein Konzept zur Dokumentation wissenschaftlicher Prozesse, das in KadiStudio modelliert wird und zum anderen ein Konzept zur strukturierten Dokumentation wissenschaftlicher Daten, das in KadiWeb umgesetzt wird. Beide Konzepte greifen ineinander und schaffen gemeinsam eine konsistente Basis für das digitale Materialdesign. Abschließend werden die Konzepte auf den gewählten Anwendungsfall angewandt, wodurch die einzelnen Schritte des Materialentwicklungsprozesses identifiziert und zugleich der rote Faden für die folgenden Kapitel vorgegeben wird.

Kapitel 4 befasst sich mit dem ersten Schritt des digitalen Materialentwicklungsprozesses. Dieser behandelt die Erstellung eines digitalen Zwillings und dessen mikrostrukturelle Charakterisierung. Konkret wird eine Methode entwickelt, die die zuverlässige Rekonstruktion digitaler Zwillinge aus bildgebenden Verfahren



ermöglicht. Zudem werden geometrische Analysemethoden erarbeitet und in Form eines Workflows automatisiert. Die Ergebnisse des Kapitels in Form der Daten und Workflows werden auf Kadi4Mat strukturiert abgelegt.

Kapitel 5 beschäftigt sich ausgehend von Kapitel 4 mit der gezielten Erzeugung synthetischer digitaler Zwillinge, die reale Strukturen nachahmen und es ermöglichen eine diverse Strukturdatenbank gezielt aufzubauen. Dazu werden zunächst etablierte Methoden validiert und anschließend eine generische ML-basierte Methode entwickelt und ausgiebig validiert. Die Vorgehensweise zur Erstellung dieser Strukturgeneratoren und deren Anwendung wird abschließend in Form eines Workflows modelliert.

Kapitel 6 beschäftigt sich mit der simulativen Bestimmung der makroskopischen mechanischen Eigenschaften der in Kapitel 4 und Kapitel 5 erzeugten digitalen Zwillinge. Dazu kommt ein Phasenfeld-Mechaniksolver zum Einsatz, der auf Basis von experimentellen Untersuchungen realer Schaumproben parametrisiert ist. Abschließend wird ein generischer Workflow erstellt, der die Simulation unterschiedlicher Materialeigenschaften ermöglicht.

Kapitel 7 verknüpft die Methoden aus den vorangegangenen Kapiteln und erstellt gezielt eine umfangreiche Materialdatenbank auf Kadi4Mat. Diese wird anschließend mit Hilfe maschinellen Lernens untersucht, um Struktur-Eigenschaft-Beziehungen abzuleiten. Die identifizierten Struktur-Eigenschaft-Beziehungen werden anschließend in Form verständlicher Materialkarten visualisiert. Wie zuvor wird auch die Methode zur Analyse der Materialdatenbank in Form von generischen Workflows in KadiStudio modelliert und auf Kadi4Mat strukturiert abgelegt.

Kapitel 8 fasst die Ergebnisse der vorliegenden Arbeit zusammen. Zudem identifiziert es mögliche Weiterentwicklungen in den entwickelten Methoden und geht auf zukünftige Entwicklungspotenziale und Anwendungen ein.

## 2 Theoretische Grundlagen

Dieses Kapitel stellt die theoretischen Grundlagen bereit, die für die Entwicklung der in dieser Arbeit vorgestellten Methodik notwendig sind. Dazu gehören zentrale Konzepte des maschinellen Lernens, die in unterschiedlichen Schritten des Materialentwicklungsprozesses eingesetzt werden, und die Phasenfeldmethode, die zur Simulation makroskopischer Materialeigenschaften dient. Darüber hinaus wird das Konzept des repräsentativen Volumenelements (RVE) eingeführt, das die Generalisierbarkeit der Struktur-Eigenschaft-Beziehungen sichert. Abschließend werden die Prinzipien des Forschungsdatenmanagements erläutert, die die methodische Basis für ein FAIRes Materialdesign bilden.

### 2.1 Maschinelles Lernen

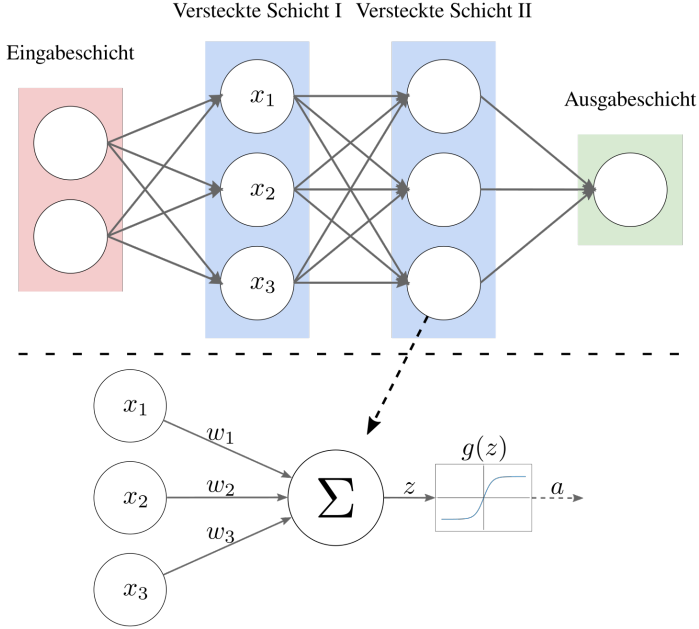
Künstliche Intelligenz ist nicht erst seit dem Erfolg großer Sprachmodelle wie ChatGPT und Bildgenerierungsnetzen in den Fokus der Forschung gerückt, sondern wird bereits seit Ende der 1940er Jahre erforscht. Der Terminus künstliche Intelligenz beschreibt hierbei Systeme, die darauf ausgelegt sind, Probleme zu lösen, ohne konkret dafür programmiert worden zu sein. Die aktuellen, vielversprechenden Entwicklungen in der KI entspringen dem Teilgebiet des maschinellen Lernens. Dieses umfasst alle Algorithmen, die selbstständig Informationen und Zusammenhänge aus Daten lernen, um eine Aufgabenstellung zu lösen. Eine spezielle Form des maschinellen Lernens ist das Deep Learning, das sich durch besonders tiefe und komplexe Netzwerke auszeichnet. Diese Netzwerke sind in der Lage, komplexe Zusammenhänge in Daten zu erkennen und dabei sogar menschliche Fähigkeiten zu übertreffen.

Da in dieser Arbeit Methoden des maschinellen Lernens verwendet werden, sollen nachfolgend deren Grundlagen dargelegt werden, die für das Verständnis notwendig sind. Zunächst soll der grundlegende Aufbau eines neuronalen Netzes gezeigt werden. Anschließend wird der Trainingsprozess detailliert dargelegt. Die dazu verwendeten Bewertungsmaße werden im darauffolgenden Kapitel präsentiert.

### 2.1.1 Neuronale Netzwerke

Neuronale Netze (NN) sind grundlegende Bestandteile des maschinellen Lernens und zeichnen sich durch ihre schichtartige Struktur aus. Die Funktionsweise neuronaler Netzwerke lässt sich besonders anschaulich anhand eines *Multilayer Perceptron* (MLP) erklären. Dieses Netzwerk besteht aus einer Eingabeschicht, mehreren versteckten Schichten und einer Ausgabeschicht [19]. Jede dieser Neuronenschichten ist mit allen Neuronen der vorangegangenen Schicht verbunden, wodurch sich grundlegende Konzepte wie die Weiterleitung von Daten, Gewichtsanpassungen und Aktivierungsfunktionen leicht veranschaulichen lassen. In MLPs fließen die Daten von der Eingabeschicht über die versteckten Schichten in Richtung der Ausgabeschicht.

Jede Schicht besteht aus Perzeptronen [20], die Daten empfangen, verarbeiten und weiterleiten. Der Aufbau eines Perzeptrons und seine Integration in ein Netzwerk sind in Abbildung 2.1 dargestellt. Ein Perzeptron empfängt einen Eingangsvektor  $\mathbf{x}$ , bestehend aus mindestens einem Eintrag. In einem MLP entspricht der Vektor  $\mathbf{x}$  dem Ausgangsvektor der vorangegangenen Schicht. Mittels einer Gewichtsmatrix  $\mathbf{W}$  wird aus dem Eingang  $\mathbf{x}$  eine gewichtete Summe gebildet, zu der ein Bias  $\mathbf{b}$  hinzuaddiert wird. Der Bias ist unabhängig von den verarbeiteten Daten und verschiebt die Werte der Ausgabe um definierte Beträge. So wird der Eingangsvektor  $\mathbf{x}$  in das Zwischenergebnis  $\mathbf{z}$  transformiert. Dieses Zwischenergebnis  $\mathbf{z}$  wird anschließend einer Aktivierungsfunktion  $g$  übergeben, die eine nichtlineare Transformation durchführt. So wird der Ausgangsvektor  $\mathbf{a}$  der aktuellen



**Abbildung 2.1:** Schematische Darstellung eines Multilayer Perceptron mit Eingabe-, versteckten und Ausgabeschichten. Jedes Perzeptron bildet eine gewichtete Summe und wendet eine Aktivierungsfunktion an, bevor die Ausgabe an die nächste Schicht weitergegeben wird.

Schicht berechnet. Dieser Vektor  $\mathbf{a}$  wird anschließend an die nächste Schicht weitergegeben:

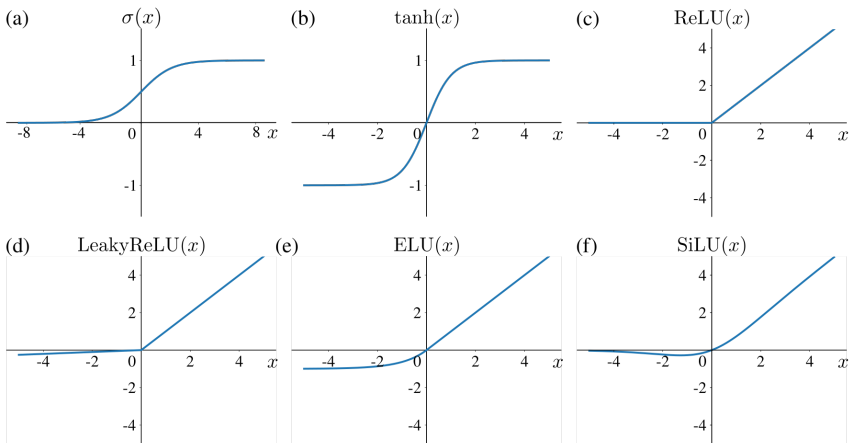
$$\mathbf{a} = g(\mathbf{z}) \quad \text{mit} \quad \mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}. \quad (2.1)$$

Ein neuronales Netz transformiert somit einen Eingangsvektor  $\mathbf{x}$  über mehrere Schichten zu einer Ausgabe  $\hat{\mathbf{y}}$ . Diese Ausgabe entspricht dabei dem Ausgangsvektor  $\mathbf{a}^L$  der letzten Schicht  $L$ . Die Funktion und das Wissen eines neuronalen Netzwerks sind in dessen Parametern gespeichert. Diese bestehen für ein MLP aus den Gewichten sowie den Biases und werden als  $\theta$  zusammengefasst:

$$NN: \mathbf{x}; \theta \mapsto \hat{\mathbf{y}} \quad \text{mit} \quad \theta = \{(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}), \dots, (\mathbf{W}^{(L)}, \mathbf{b}^{(L)})\} \quad (2.2)$$

### 2.1.1.1 Aktivierungsfunktionen

Die Aktivierungsfunktionen in neuronalen Netzen nehmen eine zentrale Rolle ein. Ohne Aktivierungsfunktionen wird der Input in jedem Layer linear transformiert und weitergegeben. Der Output des gesamten Netzes ist entsprechend eine lineare Transformation des Inputs. Theoretisch kann diese in einem Schritt implementiert werden. Es handelt sich bei solchen Netzen um lineare Regressoren, die zwar einfache Probleme lösen können, allerdings nicht in der Lage sind, komplexe Aufgabenstellungen zu bewältigen. Aus diesem Grund werden Aktivierungsfunktionen in neuronale Netze eingeführt, welche die Implementierung von nicht linearen Elementen ermöglichen. Abhängig vom Anwendungsfall und der betrachteten Schicht des Netzwerks, Eingang, versteckte Schicht oder Ausgabe, werden verschiedene Aktivierungen verwendet. In Abbildung 2.2 werden die für diese Arbeit relevanten Aktivierungsfunktionen gezeigt.



**Abbildung 2.2:** Vergleich der in dieser Arbeit verwendeten Aktivierungsfunktionen: (a) Sigmoid, (b) tanh, (c) ReLU, (d) Leaky ReLU, (e) ELU und (f) SiLU. Die unterschiedlichen Verläufe verdeutlichen ihre jeweiligen Eigenschaften, etwa Sättigungseffekte bei Sigmoid und tanh, die Robustheit von ReLU-Varianten sowie den glatten, differenzierbaren Verlauf der SiLU.

**Sigmoid** Die *Sigmoidfunktion* ist in Abbildung 2.2 (a) dargestellt und definiert als

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

Sie bildet reelle Zahlen auf den Wertebereich  $[0; 1]$  ab und eignet sich aufgrund dieser Eigenschaft besonders für probabilistische Interpretationen sowie binäre Klassifikationsaufgaben. Ein Nachteil liegt im sogenannten *Vanishing-Gradient-Problem*: Für große positive oder negative Eingabewerte wird die Ableitung sehr klein, wodurch sich Gewichte im Trainingsprozess nur langsam anpassen (siehe autorefsubsec:trainingsprozess). Zudem ist die Sigmoidfunktion nicht um Null zentriert, was die Konvergenzgeschwindigkeit weiter beeinträchtigen kann.

Aufgrund dieser Einschränkungen wird die Sigmoidfunktion heute vorwiegend in Ausgabeschichten neuronaler Netze eingesetzt, insbesondere für binäre Klassifikationen [21].

**Tangens Hyperbolicus** Die *Hyperbolische Tangensfunktion* ( $\tanh$ ), ist in Abbildung 2.2 (b) gezeigt und definiert als

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

Sie bildet reelle Zahlen auf den Wertebereich  $[-1; 1]$  ab und ist um Null zentriert. Dadurch werden positive und negative Eingabewerte symmetrisch behandelt, was den Gradientenfluss stabilisiert und eine schnellere Konvergenz fördern kann. Besonders vorteilhaft ist dies bei symmetrischen Datensätzen, deren Werte beispielsweise durch eine Standardisierung um Null verteilt sind (vgl. Unterabschnitt 2.1.4). Wie die Sigmoidfunktion ist jedoch auch  $\tanh$  vom Vanishing-Gradient-Problem betroffen, da ihre Ableitung für besonders große und kleine Eingabewerte gegen null geht. In der Praxis wird  $\tanh$  häufig in versteckten Schichten neuronaler Netze eingesetzt, insbesondere bei der Verarbeitung symmetrischer Datensätze [22].

**Rectified Linear Unit (ReLU)** Die *Rectified Linear Unit* (ReLU) [23], ist in Abbildung 2.2 (c) gezeigt und definiert als

$$\text{ReLU}(x) = \max(0, x).$$

Sie zeichnet sich durch ihre einfache Berechenbarkeit aus und hat sich aufgrund ihrer Effizienz und verbesserten Trainingsstabilität als Standard in modernen Deep-Learning-Architekturen etabliert. Durch ihren linearen Aufbau im positiven Bereich adressiert sie die Hauptnachteile von Sigmoid- und tanh-Funktionen, wie insbesondere das Vanishing-Gradient-Problem, und ermöglicht so eine schnelle und stabile Konvergenz.

ReLU wird häufig in den versteckten Schichten tiefer neuronaler Netze eingesetzt. Ein Nachteil besteht darin, dass negative Eingabewerte auf null gesetzt werden, was zum sogenannten *Dying ReLU*-Problem führen kann: Neuronen bleiben dauerhaft inaktiv und tragen somit nicht mehr zum Lernprozess bei. Um diesem Effekt entgegenzuwirken, wurden verschiedene ReLU-Varianten entwickelt [22].

**Leaky ReLU** Die Leaky Rectified Linear Unit (Leaky ReLU) ist in Abbildung 2.2 (d) dargestellt und erweitert die klassische ReLU-Funktion, indem für negative Eingabewerte  $x \leq 0$  eine kleine, konstante Steigung  $\alpha$  beibehalten wird:

$$\text{LeakyReLU}(x) = \begin{cases} x, & x > 0 \\ \alpha x, & x \leq 0 \end{cases}$$

Dadurch erhalten auch Neuronen mit negativen Eingaben einen kleinen Gradienten und bleiben aktiv, was das sogenannte Dying-ReLU-Problem mindert. Der Parameter  $\alpha$  muss dabei sinnvoll gewählt oder optimiert werden, was den Hyperparameterraum des Modells erweitert. Leaky ReLU wird insbesondere in tiefen neuronalen Netzen eingesetzt, wenn bei der Verwendung von ReLU eine hohe Anzahl inaktiver Neuronen beobachtet wird [24].

**Exponential Linear Unit (ELU)** Die Exponential Linear Unit (ELU) ist in Abbildung 2.2 (e) dargestellt und definiert als [25]:

$$\text{ELU}(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases}$$

Sie behält für negative Eingaben kleine Werte bei und zentriert die Ausgaben um Null, was den Gradientenfluss verbessert und die Stabilität des Lernprozesses erhöht. Der kontinuierliche Übergang von positiven zu negativen Werten kann das Training zusätzlich stabilisieren. Im Vergleich zu ReLU und Leaky ReLU ist ELU jedoch rechenintensiver. Sie wird vor allem in versteckten Schichten tiefer Netzwerke eingesetzt, wenn ein besonders robustes Training erforderlich ist [22, 24].

**Sigmoid Linear Unit (SiLU)** Die Sigmoid Linear Unit (SiLU) [26], dargestellt in Abbildung 2.2 (f), ist definiert als

$$\text{SiLU}(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}}.$$

Im Gegensatz zur ReLU ist die SiLU glatt und überall differenzierbar, was in tiefen neuronalen Netzen zu einer stabileren Optimierung und einer besseren Generalisierungsfähigkeit beitragen kann. Ihr nichtmonotoner Verlauf gibt kleine negative Eingabewerte teilweise weiter, was in bestimmten Architekturen zu Leistungssteigerungen gegenüber ReLU führt. Damit vereint die SiLU Vorteile der Sigmoidfunktion und der linearen Aktivierung, ohne die starke Sättigung negativer Werte wie bei der Sigmoidfunktion. In modernen Deep-Learning-Architekturen hat sich SiLU vielfach als leistungsfähiger als ReLU erwiesen. Ein Nachteil ist der höhere Rechenaufwand im Vergleich zu ReLU, da die Berechnung der Sigmoidfunktion erforderlich ist.



## 2.1.2 Trainingsprozess neuronaler Netze

Der Trainingsprozess eines neuronalen Netzes besteht darin, die Parameter  $\theta$  so zu optimieren, dass der Unterschied zwischen dem vorhergesagten Ausgabewert  $\hat{\mathbf{y}}$  und dem tatsächlichen Wert  $\mathbf{y}$  minimiert wird. Dieser Unterschied wird durch eine Verlustfunktion quantifiziert. Das Training eines MLPs erfolgt in mehreren Schritten, die im Folgenden erläutert werden. Ausführlichere Beschreibungen sind in [19, 27] zu finden.

### 2.1.2.1 Vorwärtspropagation

In der Vorwärtspropagation werden die Eingangsdaten  $\mathbf{x}$  schichtweise durch das Netzwerk propagiert. Jede Schicht  $l$  transformiert ihre Eingabe  $\mathbf{a}^{(l-1)}$  mit Hilfe einer Gewichtsmatrix  $\mathbf{W}^{(l)}$ , eines Bias-Vektors  $\mathbf{b}^{(l)}$  und einer Aktivierungsfunktion  $g^{(l)}$ . Die Ausgabe  $\mathbf{z}$  der  $l$ -ten Schicht ist gegeben durch:

$$\begin{aligned}\mathbf{z}^{(l)} &= \mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}, \\ \mathbf{a}^{(l)} &= g^{(l)}(\mathbf{z}^{(l)}).\end{aligned}\tag{2.3}$$

Im Falle der Ausgangsschicht  $L$  wird die Ausgabe  $\mathbf{a}^{(L)}$  als  $\hat{\mathbf{y}}$  interpretiert.

### 2.1.2.2 Kostenfunktion

Die Genauigkeit des Netzes wird durch eine Kostenfunktion  $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$  bewertet. Diese misst den Unterschied zwischen der gewünschten Ausgabe  $\mathbf{y}$  und der vorhergesagten Ausgabe  $\hat{\mathbf{y}}$ . Welche Kostenfunktion verwendet wird, ist abhängig von der Aufgabe des Netzes. Für Regressionen ist die *mittlere quadratische Abweichung* (MSE) ein häufig verwendetes Maß:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|^2.\tag{2.4}$$

Für Klassifikationsprobleme wird meist die *Kreuzentropie* verwendet:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C \mathbf{y}_{ij} \log(\hat{\mathbf{y}}_{ij}), \quad (2.5)$$

wobei  $N$  die Anzahl der Trainingsbeispiele und  $C$  die Zahl der Klassen repräsentiert.

### 2.1.2.3 Rückwärtspropagation

Mit der Kostenfunktion existiert ein Fehlermaß, auf dessen Grundlage die Parameter  $\theta = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^L$  des Netzes optimiert werden. Dies geschieht typischerweise mithilfe des *Gradientenabstiegsverfahrens* (engl. Gradient Descent), bei dem die Parameter schrittweise in Richtung des negativsten Gradienten der Kostenfunktion angepasst werden, um diese zu minimieren.

Die Anpassung der Parameter erfolgt über die *Rückwärtspropagation* (engl. Backpropagation). Dabei wird der in der Ausgabeschicht berechnete Fehler unter Anwendung der Kettenregel schrittweise zur Eingabeschicht zurückgeführt. So lassen sich die Gradienten der Kostenfunktion bezüglich aller Gewichte  $\mathbf{W}^{(l)}$  und Bias-Terme  $\mathbf{b}^{(l)}$  effizient bestimmen und direkt für das Parameterupdate verwenden. Für die  $l$ -te Schicht ergibt sich dies zu:

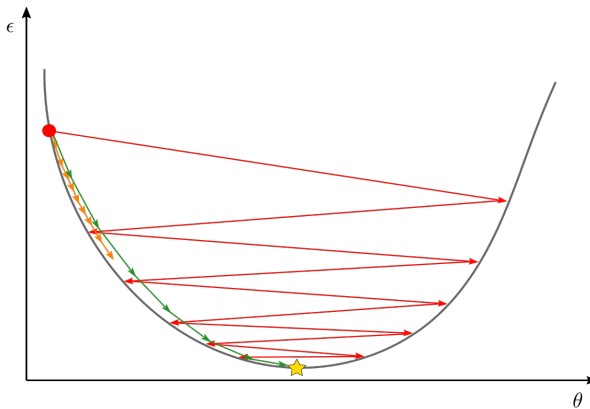
$$\Delta \theta^{(l)} = -\alpha \frac{\partial \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})}{\partial \theta^{(l)}}, \quad (2.6)$$

wobei  $\alpha$  die Lernrate bezeichnet. Diese bestimmt die Schrittweite des Gradientenabstiegs und beeinflusst maßgeblich, wie schnell und stabil das Netz während des Trainings konvergiert.

Abbildung 2.3 veranschaulicht den Einfluss der Lernrate  $\alpha$  auf das Training. Ist  $\alpha$  zu klein (grüne Pfeile), konvergiert das Netz nur langsam oder bleibt in lokalen Minima stecken. Ist  $\alpha$  zu groß (rote Pfeile), kann das Minimum überschritten werden, sodass die Parameter um die optimalen Werte oszillieren oder das Training

sogar divergiert. Eine geeignete Wahl von  $\alpha$  (orangene Pfeile) ermöglicht dagegen eine schnelle und stabile Konvergenz.

Die in diesem Abschnitt beschriebenen Grundlagen bilden die Basis für erweiterte Optimierungsverfahren wie Adam, die adaptive Lernraten und Momentum-Effekte nutzen, um die Konvergenz weiter zu verbessern.



**Abbildung 2.3:** Einfluss der Lernrate  $\alpha$  auf das Konvergenzverhalten eines neuronalen Netzes. Dargestellt ist die Optimierung entlang einer vereinfachten, konvexen Verlustfunktion. Kleine Lernraten (orange) führen zu langsamer Annäherung, große Lernraten (rot) zu Überspringen oder Divergenz. Eine optimal gewählte Lernrate (grün) ermöglicht schnelle und stabile Konvergenz zum Minimum.

### 2.1.2.4 Optimizer

Im vorherigen Abschnitt wurde beschrieben, wie mithilfe der Rückwärtspropagation die Gradienten der Kostenfunktion bezüglich aller Parameter  $\theta = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^L$  berechnet werden. Diese Gradienten bilden die Grundlage für das Parameterupdate im Training. Die Optimierung der Gewichte soll nachfolgend möglichst anschaulich beschrieben werden. Dazu wird eine Kugelanalogie verwendet. Diese Kugel rollt durch eine Verlustlandschaft, deren Höhe den Wert

der Kostenfunktion für eine bestimmte Parametrisierung  $\theta$  darstellt. Die Lernrate entspricht der Geschwindigkeit der Kugel.

Das einfachste Verfahren hierfür ist der *Stochastic Gradient Descent* (SGD), bei dem die Parameter schrittweise in Richtung des negativen Gradienten angepasst werden. Dies lässt sich mathematisch als

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})}{\partial \theta_t} \quad (2.7)$$

formulieren, wobei  $\alpha$  die konstante Lernrate und  $t$  den aktuellen Trainingsschritt bezeichnet. In der Kugelanalogie rollt die Kugel hier einfach bergab und ändert ihre Geschwindigkeit proportional zur Steigung. In flachen Regionen rollt sie entsprechend langsam, in steilen Bereichen schnell. SGD kann dadurch in Tälern (lokalen Minima) oder flachen Ebenen (Sattelpunkten) der Verlustlandschaft hängen bleiben.

**SGD mit Momentum** Um diese Schwächen zu verringern, ergänzt *SGD mit Momentum* die Aktualisierung um eine Trägheitskomponente, die frühere Gradienten berücksichtigt. Dazu wird eine Geschwindigkeit  $\mathbf{v}_t$  gespeichert, die sich in jedem Schritt nach

$$\mathbf{v}_{t+1} = \rho \mathbf{v}_t - \alpha \frac{\partial \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})}{\partial \theta_t} \quad (2.8)$$

ergibt, und aktualisiert anschließend die Parameter mit

$$\theta_{t+1} = \theta_t + \mathbf{v}_{t+1}. \quad (2.9)$$

Hierbei bezeichnet  $\rho \in [0, 1)$  die Momentum-Rate [19]. In der Analogie baut die Kugel eine immer größere Geschwindigkeit auf, wenn sie längere Zeit in dieselbe Richtung bergab rollt. So kann sie kleine Täler oder flache Abschnitte überwinden. Das stabilisiert den Weg zum Minimum und glättet den Pfad. Trotz dieser Vorteile nutzt SGD mit Momentum für alle Parameter die gleiche Lernrate und passt die Bewegungsgeschwindigkeit nicht an die jeweilige Form der Verlustlandschaft an. In Richtungen mit steilen Gradienten kann dies zu starkem Überspringen führen,

während in flacheren Richtungen nur langsame Fortschritte erzielt werden. Um diese Einschränkungen zu überwinden, führen Verfahren wie *Adam* eine adaptive, richtungsabhängige Anpassung der Geschwindigkeit ein.

**Adam** Wie beim Momentum-Ansatz speichert *Adam* (*Adaptive Moment Estimation*) [28] die bisherige Bewegungsgeschwindigkeit der Kugel und glättet dadurch den Weg ins Tal. Zusätzlich misst Adam jedoch, wie stark die Kugel in jeder einzelnen Richtung der Parameter bereits beschleunigt wurde. Richtungen, in denen die Kugel schon lange und stark beschleunigt wurde, werden gezielt abgebremst, um ein Überschießen des Minimums zu vermeiden. In flacheren Richtungen, in denen die Kugel bisher nur langsam Fahrt aufgenommen hat, wird sie hingegen beschleunigt, um schneller aus Plateaus herauszukommen.

Mathematisch geschieht dies, indem Adam einen gleitenden Mittelwert der Gradienten, den sogenannten ersten Moment  $\mathbf{m}_t$ , als Schätzer für die aktuelle Geschwindigkeit und einen gleitenden Mittelwert der quadrierten Gradienten, den zweiten Moment  $\mathbf{v}_t$ , als Maß für die bisherige Beschleunigung speichert. Beide Größen werden in den ersten Schritten des Trainings korrigiert, um Anfangsverzerrungen zu vermeiden.

Die Berechnung erfolgt schrittweise:

$$\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1 - \beta_1) \frac{\partial \mathcal{L}}{\partial \theta_t}, \quad \mathbf{v}_{t+1} = \beta_2 \mathbf{v}_t + (1 - \beta_2) \left( \frac{\partial \mathcal{L}}{\partial \theta_t} \right)^2, \quad (2.10)$$

wobei  $\beta_1$  und  $\beta_2$  die Zerfallsraten für den ersten bzw. zweiten Moment sind. Zur Korrektur der Verzerrung in den ersten Iterationen werden

$$\hat{\mathbf{m}}_{t+1} = \frac{\mathbf{m}_{t+1}}{1 - \beta_1^{t+1}}, \quad \hat{\mathbf{v}}_{t+1} = \frac{\mathbf{v}_{t+1}}{1 - \beta_2^{t+1}} \quad (2.11)$$

berechnet, bevor das Aktualisierung der Parameter erfolgt:

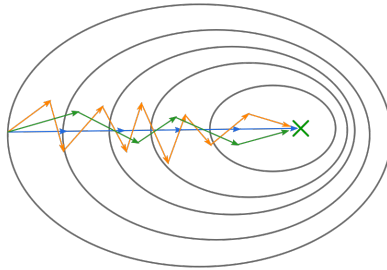
$$\theta_{t+1} = \theta_t - \frac{\alpha}{\epsilon + \sqrt{\hat{\mathbf{v}}_{t+1}}} \hat{\mathbf{m}}_{t+1}. \quad (2.12)$$

In der Kugel-Analogie entspricht  $\hat{\mathbf{m}}_{t+1}$  der korrigierten Geschwindigkeit der Kugel und  $\hat{\mathbf{v}}_{t+1}$  der korrigierten Messung ihrer bisherigen Beschleunigung in jeder Richtung. Die Division durch  $\sqrt{\hat{\mathbf{v}}_{t+1}}$  bewirkt, dass die Geschwindigkeit in stark beschleunigten Richtungen sanft gebremst und in wenig beschleunigten Richtungen verstärkt wird. Dadurch bewegt sich die Kugel gleichzeitig kontrolliert in steilen Tälern und effizient über flache Passagen, was zu einer schnellen und stabilen Konvergenz führt. Aufgrund dieser Eigenschaften ist Adam einer der am weitesten verbreiteten Optimizer und wird in dieser Arbeit primär verwendet.

### 2.1.2.5 Minibatch

In der Realität ist es oft unvorteilhaft, für jede Parameteraktualisierung den gesamten Trainingsdatensatz zu verwenden. Einerseits verlangsamen große Datensätze das Training und erfordern große Mengen an Speicherplatz. Andererseits werden die Parameter direkt in Richtung des gemeinsamen Minimums des gesamten Datensatzes optimiert. Dies erhöht die Gefahr, in lokalen Minima oder Sattelpunkten stecken zu bleiben, und begünstigt Overfitting (vgl. Unterabschnitt 2.1.3). Zudem werden schwierige Datenpunkte beim Aktualisieren auf allen Trainingsdaten einfach herausgemittelt. Das Netzwerk lernt entsprechend nicht, diese Eingabedaten adäquat zu verarbeiten.

Das andere Extrem ist die Verwendung nur eines einzelnen Datenpunkts für jede Aktualisierung. Zwar sind die einzelnen Schritte hier rechnerisch sehr günstig, jedoch wird die vorhandene Rechenkapazität nicht effizient genutzt und Parallelisierungsmöglichkeiten, etwa auf Grafikkarten (GPU), bleiben ungenutzt. Ein Vorteil der Verwendung nur eines Datenpunkts ist hingegen eine gewisse Regularisierung durch die Optimierung des Netzes in Richtung eines zufälligen Datenpunkts. Die Parameter bewegen sich dadurch nicht auf direktem Weg zu einem gemeinsamen Minimum, sondern oszillieren um den direkten Weg. Dadurch können lokale Minima und Sattelpunkte umgangen und ein Overfitting vermieden werden.

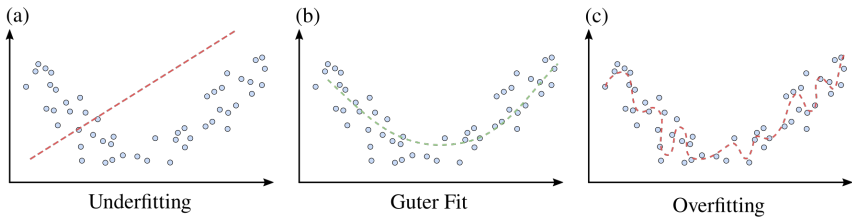


**Abbildung 2.4:** Einfluss der Batchgröße auf den Optimierungspfad: Training mit allen Daten (blau) kann in lokalen Minima stecken bleiben, einzelne Datenpunkte (orange) oszillieren stark, während Minibatches (grün) einen stabileren Weg finden.

Ein häufig verwendeter Kompromiss aus beiden Ansätzen ist der sogenannte *Minibatch* Ansatz. Dieser verwendet für jeden Trainingsschritt eine definierte Anzahl (*Batchgröße*) zufälliger Datenpunkte. Dadurch werden die Vorteile der beiden beschriebenen Ansätze kombiniert: Die Speicherkapazitäten der GPU werden effektiv genutzt und ein gewisses Maß der Regularisierung tritt auf [19]. Der Einfluss der Batchsize auf den Optimierungspfad ist in Abbildung 2.4 gezeigt.

### 2.1.3 Regularisierung und Trainingsstrategien

Das zentrale Ziel beim Training eines neuronalen Netzwerks ist die Generalisierbarkeit. Das Netzwerk soll nicht nur die Trainingsdaten korrekt erfassen, sondern auch neue, unbekannte Daten verlässlich verarbeiten. Zur Bewertung dieser Fähigkeit werden die verfügbaren Daten in Trainings-, Validierungs- und Testdaten aufgeteilt. Die Trainingsdaten dienen der Optimierung der Parameter  $\theta$ , während Validierungsdaten nicht in die Parameteraktualisierung einfließen, sondern während des Trainings regelmäßig zur Leistungsbewertung auf ungesehenen Beispielen eingesetzt werden. Die Generalisierbarkeit wird dabei anhand des Vergleichs der Verluste auf Trainings- und Validierungsdaten beurteilt. Abbildung 2.5 zeigt die drei typischen Szenarien:



**Abbildung 2.5:** Typische Trainingsszenarien: (a) Underfitting: Modell zu simpel, hohe Verluste. (b) Guter Fit: Niedrige, ähnliche Verluste auf Trainings- und Validierungsdaten. (c) Overfitting: Sehr niedriger Trainingsverlust, aber hoher Validierungsverlust.

- (a) Underfitting: Hohe Verluste auf Trainings- und Validierungsdaten. Das Modell ist zu simpel oder zu stark regularisiert.
- (b) Guter Fit: Vergleichbare, niedrige Verluste auf beiden Datensätzen. Gewünschter Modellzustand.
- (c) Overfitting: Sehr niedriger Trainingsverlust, aber hoher Validierungsverlust. Das Modell passt sich zu stark an Trainingsdaten an.

Ein Netzwerk, das komplex genug ist, die Trainingsdaten vollständig abzubilden, neigt schnell zum Overfitting. Durch gezielte Regularisierung, also durch Anpassungen an Architektur und Trainingsprozess, kann es jedoch zu besserer Generalisierungsfähigkeit gebracht werden, sodass Verlustwerte auf Trainings- und Validierungsdaten ähnlich ausfallen.

**Early Stopping** Eine einfache und oft wirksame Methode ist das *Early Stopping*, bei dem während des Trainings der Verlust auf Trainings- und Validierungsdaten überwacht wird. Beginnen beide zu divergieren, wird das Training beendet, um Overfitting zu vermeiden. Gleichzeitig verkürzt sich so die Trainingsdauer, was Early Stopping zu einer effizienten Regularisierungsmethode macht.



**Data Augmentation** Da Netzwerke generell bei zu kleinen Datenmengen zum Overfitting neigen, besteht in der *Data Augmentation* eine weitere Regularisierungsmethode. Diese erweitert die Trainingsdaten künstlich, z. B. durch Rotieren, Spiegeln oder Verzerren von Bildern oder durch Rauschen und Transformationen bei eindimensionalen Eingaben. So wird die Variabilität der Daten erhöht und die Generalisierung gefördert.

**L1- und L2-Regularisierung** Neben der Anpassung der Trainingsdaten und -dauer können auch die Gewichte des Netzes direkt regularisiert werden. Dazu wird die Verlustfunktion um Strafterme erweitert, sodass sie die Form

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}; \boldsymbol{\theta}) = \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) + \alpha_1 \|\boldsymbol{\theta}\|_1 + \alpha_2 \|\boldsymbol{\theta}\|_2^2. \quad (2.13)$$

annimmt. Beide Terme bestrafen hohe Gewichte. L1 ( $\alpha_1 \|\boldsymbol{\theta}\|_1$ , auch Lasso) verwendet die absoluten Werte der Gewichte und fördert dünnbesetzte Parametermatrizen für  $\boldsymbol{\theta}$  mit vielen Nullen. Hierdurch kommt es zu einer *Feature Selection*. Das Netz konzentriert sich nur auf die wichtigen Merkmale in den Trainingsdaten. L2 ( $\alpha_2 \|\boldsymbol{\theta}\|_2^2$ , auch Ridge) bestraft hingegen die quadrierten Gewichte. Dadurch hält sie Gewichte klein, setzt sie jedoch selten exakt auf null und zwingt das Netz, alle Neuronen gleichmäßiger zu nutzen.

**Dropout** Ein weiterer vielverwendeter Regularisierungsmechanismus ist *Dropout*. Dropout deaktiviert während des Trainings zufällig Neuronen, wodurch die Netzwerkkapazität reduziert wird. Dies verhindert, dass sich einzelne Neuronen zu stark spezialisieren, und kann als Ensemble-Ansatz interpretiert werden, bei dem viele leicht unterschiedliche Modelle parallel trainiert werden.

Alle genannten Methoden lassen sich kombinieren und stellen erprobte Ansätze dar, um die Generalisierungsfähigkeit neuronaler Netze zu erhöhen.

## 2.1.4 Normalisierung

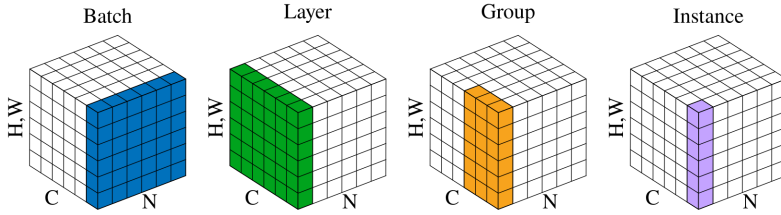
Zur Stabilisierung und Beschleunigung des Trainings neuronaler Netze werden die verwendeten Daten häufig normalisiert. Ziel ist es, bestimmte statistische Eigenschaften wie Mittelwert und Standardabweichung zu vereinheitlichen. Eine Normalisierung kann allgemein durch

$$\hat{x} = \frac{x - \mu}{\sigma}, \quad (2.14)$$

beschrieben werden, wobei  $x$  die Daten,  $\mu$  deren Mittelwert und  $\sigma$  die Standardabweichung bezeichnet. Auf diese Weise wird die Datenverteilung auf eine Standard-Normalverteilung mit  $\mu = 0$  und  $\sigma = 1$  gebracht.

Im maschinellen Lernen kann zwischen Daten- und Aktivierungsnormalisierung unterschieden werden. Die Datennormalisierung skaliert die Eingangsdaten so, dass alle Merkmale auf derselben Skala liegen. Unterschiede in Größenordnung oder physikalischen Einheiten beeinflussen das Training dadurch nicht mehr. Die Aktivierungsnormalisierung wird hingegen innerhalb des Netzes auf die Ausgaben einzelner Layer angewendet. Eine gängige Methode ist die *Batch Normalization*, die die Aktivierungen entlang der Batch-Achse normalisiert. Dadurch erhält jede Schicht stets gleich skalierte Eingaben, was den sogenannten *Covariate-Shift* reduziert und das Training beschleunigt [29].

Neben der Normalisierung entlang der Batch-Achse existieren weitere Verfahren: *Layer Normalization* normalisiert über alle Features eines Layers, während *Group Normalization* die Feature-Achse in Gruppen unterteilt und jede Gruppe separat normalisiert. Letzteres kann dazu führen, dass die Features einer Gruppe ähnliche Merkmale aus den Daten extrahieren. *Instance Normalization* normalisiert über ein Feature in einem Datenpunkt. Abbildung 2.6 veranschaulicht die Unterschiede dieser Normalisierungsmethoden.



**Abbildung 2.6:** Schematische Darstellung verschiedener Normalisierungsmethoden. Die farbigen Markierungen kennzeichnen jeweils die Datenbereiche, über die die Normalisierung durchgeführt wird. Als Eingabedaten dienen Bilddaten der Form  $(N, C, H, W)$ , wobei  $N$  die Batchgröße,  $C$  die Kanäle (Features) sowie  $H$  und  $W$  die Höhe und Breite des Bildes repräsentieren

## 2.1.5 Convolutional Neural Networks

Die bisher betrachteten Multilayer Perceptrons verarbeiten eindimensionale Eingangsvektoren, was für Bilddaten oder höherdimensionale Matrizen unpraktisch ist. Jeder Pixel würde einem eigenen Neuron in der Eingangsschicht entsprechen, wodurch die Parameterzahl mit der Bildgröße schnell untragbar ansteigt. Zudem ignorieren MLPs lokale Nachbarschaftsstrukturen, obwohl benachbarte Pixel oft stark korreliert sind und gemeinsam wichtige Muster bilden.

*Convolutional Neural Networks* (CNN) adressieren diese Probleme, indem sie lokale Filteroperationen einsetzen. Statt jeden Pixel isoliert zu verarbeiten, extrahieren sogenannte Filter (*Kernels*) lokale Merkmale aus dem Bild. Die Faltung eines Eingabebildes  $f$  mit einem Filter  $k$  wird durch

$$(f * k)(i, j) = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} k(m, n) f(i+m, j+n) \quad (2.15)$$

beschrieben, wobei  $k_h$  und  $k_w$  die Höhe bzw. Breite des Filters angeben. Die so erzeugten *Feature Maps* heben charakteristische Bildmerkmale wie Kanten, Texturen oder Muster hervor. Während des Trainings werden die Filtergewichte automatisch so angepasst, dass sie für die jeweilige Aufgabe optimale Merkmale erfassen.

Abbildung 2.7 veranschaulicht die Faltungsoperation: Jeder Pixelwert des Eingabebildes wird mit den entsprechenden Filterwerten multipliziert und die Ergebnisse werden aufsummiert, sodass ein neuer Pixelwert in der Feature Map entsteht. Um den Wertebereich konsistent zu halten, wird die berechnete Summe anschließend durch die Gesamtsumme aller Filtereinträge normalisiert.

$$\begin{array}{|c|c|c|c|} \hline 1 & 2 & 3 & 3 \\ \hline 5 & 4 & 4 & 7 \\ \hline 8 & 1 & 9 & 2 \\ \hline 5 & 0 & 4 & 3 \\ \hline \end{array} \quad * \quad \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1 & 8 \\ \hline 0 & 9 \\ \hline \end{array}$$

Eingabe  $f$                       Filter  $k$                       Feature Map

**Abbildung 2.7:** Schematische Darstellung der Faltung einer Eingabe  $f$  mit einem Filter  $k$  zur Erstellung einer Feature Map.

CNNs bestehen typischerweise aus mehreren Faltungsschichten, die ähnlich wie in MLPs Aktivierungsfunktionen verwenden, um nichtlineare Mustererkennung zu ermöglichen. Neben den trainierbaren Filtern kommen *Pooling Layer* zum Einsatz, um die Feature Maps zu verdichten. Diese reduzieren die räumliche Auflösung, verringern damit die Anzahl der Parameter und den Speicherbedarf und erhöhen zugleich die Robustheit des Netzes gegenüber leichten Verschiebungen im Bild. Beim *Max Pooling* wird eine Filtermaske mit einer definierten Größe (z. B.  $2 \times 2$  Pixel) über die Feature Maps geschoben und nur der größte Wert übernommen. *Average Pooling* berechnet hingegen den Mittelwert. Üblicherweise entsprechen Fenstergröße und Schrittweite (*Stride*) einander, sodass die Dimensionen gleichmäßig reduziert werden. Pooling wirkt damit wie eine gezielte Informationskompression, bei der wesentliche Strukturen erhalten bleiben und unwichtige Details verworfen werden.

Für Klassifikations- oder Regressionsaufgaben werden die finalen Feature Maps in einen Vektor überführt und von einem oder mehreren voll verbundenen Schichten weiterverarbeitet. Auf diese Weise können CNNs aus hochdimensionalen Eingaben wie Bildern kompakte, aussagekräftige Repräsentationen gewinnen und effizient weiterverarbeiten.

## 2.2 Generative Modelle

Die zuvor beschriebenen Faltungsschichten finden in zahlreichen Netzwerkarchitekturen Anwendung. Für diese Arbeit sind insbesondere generative Architekturen von Bedeutung, die neue Bilddaten erzeugen können. Da diese Modelle im weiteren Verlauf intensiv genutzt werden, werden ihr Aufbau und die mathematischen Grundlagen an dieser Stelle erläutert.

### 2.2.1 Generative Adversarial Networks

*Generative Adversarial Networks* (GANs), eingeführt von Goodfellow et al. [30], bestehen aus zwei gleichzeitig trainierten Netzwerken mit gegensätzlichen Zielen: *Generator*  $G$  und *Diskriminator*  $D$ . Das Training ist dabei *adversarial*, da beide Netzwerke in direktem Wettbewerb stehen. Der Generator versucht, den Diskriminator zu täuschen, während dieser seine Fähigkeit optimiert, echte von künstlichen Daten zu unterscheiden.

#### Generator

Der Generator erhält als Eingabe einen Zufallsvektor  $\mathbf{z}$ , der aus einer einfachen Verteilung (z. B. Normal- oder Gleichverteilung) stammt, und transformiert diesen in ein Bild  $G(\mathbf{z})$ . Ziel ist es, Bilder zu erzeugen, die statistisch nicht von echten Beispielen zu unterscheiden sind.

#### Diskriminator

Der Diskriminator erhält sowohl echte Bilder  $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$  als auch vom Generator erzeugte Bilder  $G(\mathbf{z})$  und gibt einen Wert zwischen 0 (gefälscht) und 1 (echt) aus. Er wird darauf trainiert, den Realismus eines Bildes möglichst korrekt zu bestimmen.

## Training

Das Zusammenspiel lässt sich als Minimaxspiel formulieren:

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (2.16)$$

wobei  $G$  der Generator und  $D$  der Diskriminator ist. Dieses Training entspricht einem Nullsummenspiel, bei dem der Generator sich verbessert, indem er den Diskriminator täuscht, während dieser gleichzeitig lernt, Fälschungen zu entlarven. Ein idealer Zustand entspricht einem Nash-Gleichgewicht, bei dem die vom Generator erzeugte Verteilung  $p_G$  der wahren Verteilung  $p_{\text{data}}$  entspricht. In diesem Fall kann der Diskriminator nicht besser als zufällig zwischen echt und gefälscht unterscheiden.

Das ursprüngliche Minimax-Ziel führt in der Praxis häufig zu nahezu verschwindenden Gradienten, sobald der Diskriminator deutlich besser als der Generator ist. In diesem Fall lernt der Generator kaum noch, da seine Gradienten gegen null gehen. Um diesem Problem zu begegnen, führten Goodfellow et al. [30] auch die sogenannte *non-saturating* bzw. nicht-sättigende Verlustfunktion ein, bei dem der Generator anstelle von

$$\min_G \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (2.17)$$

die modifizierte Zielfunktion

$$\max_G \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [\log D(G(\mathbf{z}))] \quad (2.18)$$

optimiert. Diese Formulierung ist im Nash-Gleichgewicht äquivalent, führt in der Praxis jedoch zu stabileren Gradienten und beschleunigt damit das Training.

## Eigenschaften

GANs sind in der Lage, hochrealistische und kontrastreiche Bilder zu erzeugen, weisen jedoch auch Probleme auf. Typisch sind die Instabilität des Trainings und der sogenannte *Mode Collapse*, bei dem der Generator nur einen kleinen Teil der Datenverteilung abdeckt. Diese Schwierigkeiten machen das Training notorisch anspruchsvoll und erfordern eine sorgfältige Wahl der Hyperparameter.

Ein alternatives Modell, das viele dieser Schwächen adressiert, ist das *Denoising Diffusion Probabilistic Model* (DDPM), das im nächsten Abschnitt detailliert vorgestellt wird.

### 2.2.2 Denoising Diffusion Probabilistic Model

Denoising Diffusion Probabilistic Models (DDPMs) sind eine Klasse probabilistischer generativer Modelle, die Daten durch ein sequentielles Hinzufügen und Entfernen von Rauschen modellieren [31, 32]. Das Modell basiert auf zwei Prozessen: *Vorwärtsdiffusion* und *Rückdiffusion*.

#### Vorwärtsdiffusion

In der Vorwärtsdiffusion werden Datenpunkte  $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x})$  schrittweise verrauscht, bis sie in eine Gauß-Verteilung überführt sind. Dieser Prozess ist eine *Markov-Kette* [33], da jeder Zustand  $\mathbf{x}_t$  nur vom unmittelbar vorherigen Zustand  $\mathbf{x}_{t-1}$  abhängt:

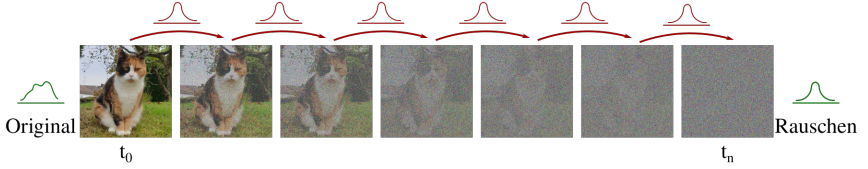
$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t} \mathbf{x}_{t-1}, (1 - \alpha_t) \mathbf{I}), \quad (2.19)$$

mit  $\alpha_t \in (0, 1)$  als Varianzplan-Parameter, der die Menge des aufgeprägten Rauschens für jeden Schritt definiert.

Mit Hilfe dieser Markov-Kette kann das verrauschte Bild  $\mathbf{x}_t$  zu jedem Zeitpunkt  $t$  durch

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}), \quad (2.20)$$

beschrieben werden, wobei  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$  das kumulative Produkt der Rauschstärken bezeichnet. Der Vorwärtsdiffusionsprozess stellt kein Training eines neuronalen Netzes dar. Es wird lediglich definiertes Rauschen auf die Eingangsbilder aufgebracht. Abbildung 2.8 visualisiert diesen Prozess.



**Abbildung 2.8:** Vorwärtsdiffusionsprozess. Ein klares Bild wird iterativ in reines Rauschen umgewandelt.

## Rückdiffusion

Die Rückdiffusion bildet das Gegenstück zur Vorwärtsdiffusion und wird während des Trainings erlernt. Ausgehend von reinem Rauschen  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$  entfernt das Modell schrittweise Rauschen, bis am Ende eine synthetische, realistische Struktur  $\mathbf{x}_0$  entsteht. Auch dieser Prozess ist als *Markov-Kette* formuliert, da jeder Schritt  $\mathbf{x}_{t-1}$  nur von  $\mathbf{x}_t$  abhängt:

$$p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\theta}(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I}), \quad (2.21)$$

wobei  $\mu_{\theta}$  durch ein neuronales Netz approximiert wird, das über die Parameter  $\theta$  trainiert wird. Der Varianzparameter  $\sigma_t^2$  ist dabei fest vorgegeben.

Das Ziel des Netzes ist es, das im Vorwärtsprozess hinzugefügte Rauschen  $\varepsilon$  zu rekonstruieren. Damit kann die Trainingsaufgabe des Netzwerks auf die Schätzung von  $\varepsilon$  reduziert werden, was in einer einfachen, aber effektiven Verlustfunktion resultiert:

$$\mathcal{L}_{\text{DDM}} = \mathbb{E}_{\mathbf{x}_0, \varepsilon, t} [\|\varepsilon - \varepsilon_{\theta}(\mathbf{x}_t, t)\|^2], \quad (2.22)$$



wobei  $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$ . Die Rückdiffusion unterscheidet sich damit wesentlich von GANs. Anstelle einer komplexen adversarialen Zielfunktion wird eine definierte, stabil optimierbare MSE-Verlustfunktion genutzt. Dies macht das Training robuster und vermeidet viele Instabilitäten, die bei adversarialen Verfahren auftreten können.

### Eigenschaften

Diffusionsmodelle sind stabil im Training und decken die Datenverteilung vollständig ab, erfordern jedoch viele Rückdiffusionsschritte und sind dadurch rechenintensiv. Eine Abhilfe bieten Hybridansätze wie *Denoising Diffusion GANs* (DD-GANs) [34], die Bildschärfe und Effizienz adversarialer Netze mit der Modelltreue von Diffusionsmodellen verbinden. Dies reduziert die Anzahl der Schritte, bringt jedoch durch zusätzliche Architekturkomplexität und Hyperparameter neue Herausforderungen mit sich. Eine detaillierte Beschreibung des in dieser Arbeit verwendeten DD-GANs findet sich in Kapitel 5.

## 2.3 Phasenfeldmethode

Die Simulation von Mehrphasenproblemen lässt sich grundsätzlich über zwei unterschiedliche Methoden realisieren. Zunächst besteht die Möglichkeit, den Übergang zwischen den vorliegenden Phasen als scharfe Grenze zu definieren, bei der die Phasen abrupt wechseln. Dies erfordert geometrieangepasste Gitter für die numerische Diskretisierung, die beispielsweise mit der Finite-Elemente-Methode (FEM) erfolgen kann. Für evolvierende Grenzflächen müssen die Gitter zudem im Laufe der Zeit immer wieder angepasst werden. Dies kann zu hohem Rechenaufwand führen, weshalb diese Methoden für dreidimensionale Anwendungen an ihre Grenzen stoßen können [35].

Neben einem scharfen Übergang kann die Phasengrenze auch als stetige Funktion beschrieben werden. Diese Annahme verwendet die Phasenfeldmethode zur Lösung von Mehrphasenproblemen. Grundidee dieser Modellierung ist die von Van der Waals beobachtete Diffusivität der Grenzfläche benachbarter Phasen [36]. Eine Phasengrenze kann demnach als kontinuierlicher Übergang von einer Phase in die nächste über mehrere Moleküldicken verstanden werden. Die Phasenfeldmethode beschreibt diesen Übergang über den Ordnungsparameter  $\phi$ . Für jeden Zeitpunkt  $t > 0$  weist dieser jeder Phase  $\alpha$  in jeder Zelle  $\mathbf{x}$  eines Gebiets  $\Omega$  einen Wert zu, sodass für die Summe aller Phasen  $N$

$$\forall \mathbf{x} \in \Omega, t > 0 : \quad \sum_{\alpha=1}^N \phi_{\alpha}(\mathbf{x}, t) = 1 \quad (2.23)$$

gilt. Im Gegensatz zum realen Phasenübergang, der im Nanometerbereich liegt, wird die Breite der Grenzfläche bei dieser Modellierung im Bereich der Diskretisierungsweite gewählt. Für ein genaues Simulationsergebnis sollte der Übergang allerdings möglichst klein im Verhältnis zum Gesamtvolumen der Phasen gewählt werden. Durch die Beschreibung des Phasenübergangs über den Ordnungsparameter kann die Phasenfeldsimulation auf zeitlich unveränderten und räumlich beliebigen Gittern durchgeführt werden [37].

Das Ziel der Phasenfeldsimulation ist die Bestimmung der Phasenfeldparameter abhängig vom Ort und der Zeit. Dazu dienen sogenannte Evolutionsgleichungen. Diese stellen Veränderungen des Ordnungsparameters abhängig von der Position der betrachteten Zelle sowie des Zeitpunkts dar und ermöglichen so die Berechnung der zeitlichen und räumlichen Veränderung des Ordnungsparameters. Grundlage der Evolutionsgleichungen sind Energiefunktionale, die die Energie eines Systems beschreiben. Die freie Energie in einem Volumen  $V$  setzt sich prinzipiell aus Energiebeiträgen aus Grenzflächen zwischen unterschiedlichen Phasen und aus Energiebeiträgen innerhalb einer Phase zusammen [38]. Ihre grundlegende Struktur lässt sich über ein Energiefunktional der Form:

$$\mathcal{F}(\phi) = \int_V W_{\text{intf}}(\phi, \nabla\phi) + \bar{W}_{\text{bulk}}(\dots) \, dV, \quad (2.24)$$

zusammenfassen. Hier beschreibt  $W_{\text{intf}}(\phi, \nabla\phi)$  den Energiebeitrag der Grenzfläche und  $\bar{W}_{\text{bulk}}(\phi, \dots)$  den der Bulkphasen. Als Bulkphase sind hier die Gebiete definiert, in denen nur eine Phase vorliegt. Durch das Anpassen der Energieterne oder das Hinzufügen weiterer Terme zum Energiefunktional kann das Phasenfeldmodell für verschiedene Anwendungsfälle wie die mechanische Rissausbreitung [39], die Phasenumwandlung von Metallen [38] oder die Fluidsimulation [40] verwendet werden.

Evolutionsgleichungen minimieren die Energie dieses Energiefunktionals und modellieren so die Entwicklung des Simulationsgebiets. Diese Evolutionsgleichungen lassen sich aus dem Energiefunktional mittels Variationsrechnung herleiten (siehe z. B. [41]). Je nach Anwendungsfall erfolgt die Minimierung der freien Energie nach unterschiedlichen Variablen, wie der Zeit oder der Verschiebung des Phasenfelds.

## 2.4 Repräsentatives Volumenelement

Zur Vermeidung hoher Rechenlaufzeiten ist es notwendig, den Simulationsaufwand von Materialien auf ein Minimum zu reduzieren. Das simulierte Subvolumen des Materials sollte entsprechend möglichst klein gewählt werden. Es muss allerdings groß genug sein, um alle Inhomogenitäten des Materials abzudecken und somit repräsentativ für das gesamte Material zu sein [42]. Nur so können aus der Simulation Erkenntnisse über das gesamte Material abgeleitet werden. Das Ziel einer RVE-Bestimmung ist es, die Größe dieses repräsentativen Volumenelements festzustellen. Die Größe eines solchen *repräsentativen Volumenelements* (RVE) ist für jede Materialeigenschaft unterschiedlich und muss folglich für jede einzeln bestimmt werden. Allgemein können zwei verschiedene RVE-Größen unterschieden werden, die sich in ihrer Bestimmung teilweise unterscheiden. Bei diesen handelt es sich um das deterministische und das statistische RVE [43].

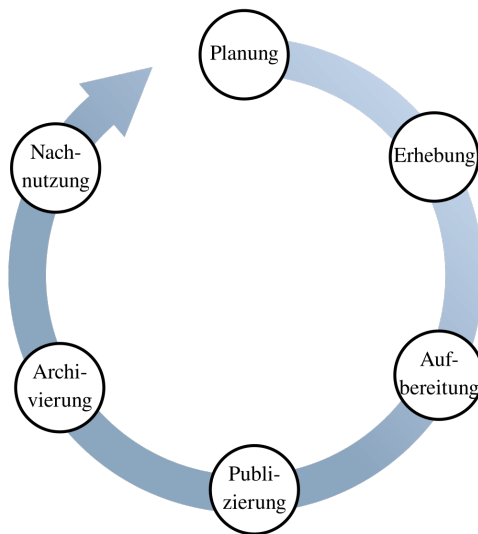
Das deterministische RVE beschreibt die Größe eines Subvolumens, ab dem sich die Abweichung, der im Subvolumen gemessenen Eigenschaften von einem Vergleichs-/Erwartungswert, auf einen zuvor definierten maximalen Fehler begrenzt [43]. Als Erwartungswert dienen häufig experimentell ermittelte Materialeigenschaften. Alternativ wird der Wert verwendet, dem sich die Simulationen mit steigender Subvolumengröße annähern. Die Messung eines deterministischen RVEs kann sowohl in einem einzelnen Subvolumen als auch in mehreren unabhängigen Subvolumina erfolgen. Bei der Messung mehrerer unabhängiger Subvolumina ist für den Vergleich mit dem Erwartungswert das arithmetische Mittel aller Messungen zu verwenden. Eine aus dem Mittelwert errechnete RVE-Größe verlangt bei späteren Eigenschaftsmessungen, den Mittelwert aus gleich vielen Messungen zu bilden, um valide Werte zu erhalten.

Das statistische RVE ist wie das deterministische über einen maximalen Fehler der Eigenschaftsmessung definiert. Dieser bezieht sich auf die Abweichung mehrerer Messungen unabhängiger Subvolumina voneinander. Dazu werden der Mittelwert und die Standardabweichung der Messungen errechnet. Liegt die Standardabweichung unter dem zuvor definierten Schwellenwert, kann von einem RVE gesprochen werden. Grundlegend unterscheiden sich statistische von deterministischen RVEs demnach dadurch, dass dem deterministischen ein Vergleichswert zugrundeliegt, wohingegen das statistische RVE die Abweichung mehrerer Messungen voneinander als Maß nimmt.

Sowohl die Bestimmung des statistischen als auch des deterministischen RVEs basiert auf der Analyse einer Vielzahl unterschiedlich großer Subvolumina. Die Auswahl und Simulation der Subvolumina erfolgen iterativ, wobei deren Größe sukzessiv erhöht wird [44]. Je nach Ansatz wird der Fehler gegenüber dem Erwartungswert oder die Standardabweichung der Messergebnisse ausgewertet. Sobald der jeweilige Grenzwert unterschritten ist, spricht man von einem RVE.

## 2.5 Forschungsdatenmanagement

Das Forschungsdatenmanagement (FDM) umfasst sämtliche Maßnahmen zur strukturierten Organisation, langfristigen Archivierung, Durchsuchbarkeit und Bereitstellung von Forschungsdaten aus unterschiedlichen Quellen. Ziel ist es, die Entstehung und Verarbeitung dieser Daten nachvollziehbar zu dokumentieren und somit eine reproduzierbare Forschung zu ermöglichen. Im Zentrum des FDM steht der vollständige Datenlebenszyklus, der in Abbildung 2.9 schematisch dargestellt ist. Dieser reicht von der Planung über die Datenerhebung und -nachbearbeitung bis hin zur Analyse, strukturierten Ablage sowie der Veröffentlichung und Nachnutzung der Daten durch Dritte.



**Abbildung 2.9:** Typischer Forschungsdatenlebenszyklus unterteilt in sechs Phasen.

Dabei gilt es, die Heterogenität wissenschaftlicher Datentypen und die Vielfalt potenzieller Nutzungsszenarien disziplinübergreifend zu berücksichtigen. FDM

dient damit nicht nur der Dokumentation, sondern auch der nachhaltigen Sicherung, Zugänglichkeit und Weitergabe von Forschungsdaten, auch fachübergreifend. Es bildet somit eine zentrale Grundlage des sogenannten „vierten Paradigmas“ der Wissenschaft, der datengetriebenen Forschung [45]. Darüber hinaus trägt es wesentlich dazu bei, wissenschaftliche Prozesse transparenter und nachvollziehbarer zu gestalten [46]. Vor diesem Hintergrund ist FDM bzw. der strukturierte Umgang mit Forschungsdaten ein integraler Bestandteil der Förderrichtlinien der Deutschen Forschungsgemeinschaft (DFG) und wird in zahlreichen fachübergreifenden Forschungsprojekten implementiert [47]. Zentrale Komponenten moderner FDM-Strategien sind dabei die Verwendung standardisierter Metadatenschemata für eine strukturierte Datenablage sowie die Nutzung von Workflowsystemen zur automatisierten und reproduzierbaren Datenverarbeitung.

## 2.5.1 Forschungsdaten

Forschungsdaten werden gemäß der Definition von Kindling und Schirmbacher als „[...] alle digital vorliegenden Daten, die während eines Forschungsprozesses entstehen oder ihr Ergebnis sind“ [48] definiert. Zu Forschungsdaten zählen entsprechend „Messdaten, Laborwerte, audiovisuelle Informationen, Texte, Surveydaten, Objekte aus Sammlungen oder Proben“ [47] und decken somit die Heterogenität unterschiedlicher Forschungsdisziplinen ab. Zudem können auch „methodische Testverfahren, wie Fragebögen, Software und Simulationen“ [47] in diesem Kontext als Forschungsdaten betrachtet werden, da sie im Zuge der Forschung entstehen und für die Reproduzierbarkeit der Forschungsergebnisse unabdingbar sind.

In dieser Arbeit werden unter Forschungsdaten entsprechend alle Primär- bzw. Rohdaten und die daraus durch Analysen oder Simulationen abgeleiteten Sekundärdaten verstanden. Neben den eigentlichen Daten besitzen die in dieser Arbeit erstellten Forschungsdaten zudem deskriptive Metadaten.

### 2.5.2 Metadaten

Metadaten sind, wie der Name bereits andeutet, „Daten über Daten“. Sie stellen eine besondere Form von Informationen dar, die nicht für sich allein stehen, sondern ergänzende Angaben über andere Ressourcen liefern. Durch diese Zusatzinformationen ermöglichen Metadaten das Auffinden, Verstehen, Verwenden und Verwalten von Daten. Je nach Zweck lassen sich Metadaten in drei Hauptkategorien unterteilen: beschreibende, administrative und strukturelle Metadaten [49]. Beschreibende Metadaten unterstützen die Identifikation und inhaltliche Erschließung einer Ressource etwa durch Titel, Autorenschaft oder Schlagworte. Administrative Metadaten enthalten Informationen zur technischen Handhabung und zur Langzeitarchivierung, wie etwa Dateiformate, verwendete Software oder Zugriffsrechte. Strukturelle Metadaten beschreiben die interne Organisation oder Beziehungen zwischen einzelnen Bestandteilen einer Ressource, etwa die Seitenzahl oder ein Inhaltsverzeichnis [50]. Ein besonderes Merkmal von Metadaten ist ihr oftmals längerer Lebenszyklus im Vergleich zur beschriebenen Ressource. Sie können bereits vor der Erzeugung der eigentlichen Daten erstellt und auch dann noch aufbewahrt werden, wenn die zugehörigen Daten schon gelöscht sind [49]. Im Gegensatz zu den häufig unstrukturierten Nutzdaten liegen Metadaten in der Regel in klar definierten, maschinenlesbaren Formaten wie *Extensible Markup Language* (XML) oder *Resource Description Framework* (RDF) vor. Durch die Verwendung standardisierter Metadatenschemata, die ein festgelegtes Vokabular und strukturierte Relationen definieren, lassen sich Metadaten automatisiert auslesen, analysieren und weiterverarbeiten. Diese Eigenschaften machen Metadaten nicht nur zu einem zentralen Bestandteil des Forschungsdatenmanagements, sondern auch zu einem wesentlichen Treiber für datenintensive Methoden wie maschinelles Lernen. Insbesondere bei der automatisierten Erschließung großer Datenbestände und der Generierung neuen Wissens aus strukturierten Datenquellen sind Metadaten essentiell.

### 2.5.3 Ontologien

Der Begriff Ontologie stammt ursprünglich aus der Philosophie und bezeichnet dort die „Lehre des Seins“. In der Informationswissenschaft beschreibt er heute den Versuch, Wissen formal, strukturiert und maschinenlesbar zu modellieren [51]. Ontologien dienen dazu, Konzepte sowie die Beziehungen zwischen ihnen innerhalb eines spezifischen fachlichen Kontexts explizit darzustellen. Im Unterschied zu Metadatenschemata, die primär die Syntax von Datenformaten definieren, bringen Ontologien eine Semantik in die Datenmodellierung ein. Sie ermöglichen es Maschinen, den Bedeutungszusammenhang von Informationen zu interpretieren und weiterzuverarbeiten.

Ontologien basieren auf grundlegenden Bausteinen wie Klassen, Instanzen, Eigenschaften und Relationen. Klassen beschreiben hierbei Konzepte wie beispielsweise einen Menschen, während eine Instanz der Klasse Mensch ein bestimmtes Individuum ist. Eigenschaften beschreiben Merkmale oder Attribute, und Relationen stellen Verbindungen zwischen Instanzen oder Klassen her. Die technische Umsetzung und Speicherung von Ontologien erfolgen typischerweise mithilfe von Sprachen des Semantic Web, insbesondere OWL (Web Ontology Language) und RDF (Resource Description Framework) [52]. Ziel des Semantic Web ist es, das World Wide Web um eine maschinenlesbare Bedeutungsschicht zu erweitern, sodass Informationen automatisiert verarbeitet, verknüpft und interpretiert werden können. In diesen Sprachen wird Wissen in Form sogenannter Tripel aus Subjekt–Prädikat–Objekt dargestellt. Jede Aussage besteht aus einem Subjekt (z.B. eine Person), einem Prädikat (z.B. „istAutorVon“) und einem Objekt (z.B. eine Publikation). Da Objekte auch Subjekte in anderen Tripeln sein können, lassen sich komplexe Zusammenhänge durch Verkettung solcher Aussagen modellieren [51]. Diese semantischen Verknüpfungen ermöglichen es, implizites Wissen aus den vorhandenen Daten zu erschließen und Zusammenhänge sichtbar zu machen, die nicht direkt in den Datenpunkten gespeichert sind. Zur Analyse und Nutzung dieses Wissens können Ontologien sowohl visualisiert als auch mit Sprachen wie SPARQL abgefragt werden, die semantische Graphen effizient durchsuchen und damit das modellierte Wissen wiederverwendbar und zugänglich machen [52].



## 2.5.4 FAIR Prinzipien

Ein zentrales Ziel des Forschungsdatenmanagements (FDM) besteht häufig in der Umsetzung der sogenannten *FAIR-Prinzipien*, die 2016 von Wilkinson et al. [53] formuliert wurden. Diese Prinzipien definieren grundlegende Anforderungen an die Auffindbarkeit, Zugänglichkeit, Interoperabilität und Wiederverwendbarkeit wissenschaftlicher Daten und fassen diese unter den Schlagworten **F**indable (auffindbar), **A**ccessible (zugreifbar), **I**nteroperable (interoperabel) und **R**eusable (wiederverwendbar) zusammen. Die konkreten Anforderungen lassen sich wie folgt zusammenfassen:

- **Findable:** Forschungsdaten müssen mit beschreibenden Metadaten versehen sein, die ihre gezielte Auffindbarkeit durch Mensch und Maschine ermöglichen. Zudem ist ein eindeutiger und dauerhafter Identifikator erforderlich.
- **Accessible:** Daten und Metadaten sollen über standardisierte Kommunikationsprotokolle abrufbar sein. Dabei kann, sofern erforderlich, eine Authentifizierung vorgesehen sein. Die Metadaten müssen zudem dauerhaft zugänglich bleiben, auch wenn die eigentlichen Daten nicht mehr verfügbar sind.
- **Interoperable:** Daten müssen in allgemein anwendbaren Formaten formuliert sein und über Vokabularien beschrieben werden, die wiederum den FAIR-Prinzipien folgen. Die Daten müssen in offenen, allgemein zugänglichen Formaten vorliegen und über Vokabularien beschrieben sein, die ihrerseits den FAIR-Prinzipien entsprechen.
- **Reusable:** Daten müssen mit reichhaltigen Metadaten und einer detaillierten Datenherkunft versehen sein. Zur Nachnutzung müssen sie außerdem mit einer eindeutigen Nutzungslizenz versehen sein.

Die FAIR-Prinzipien stellen damit fachübergreifende, technologieunabhängige Leitlinien dar, die Wissenschaftlerinnen und Wissenschaftler bei der transparenten und nachhaltigen Dokumentation ihrer Forschungsdaten unterstützen.

## 2.5.5 Repositorium

Der Begriff Repositorium, der als Lager übersetzt werden kann, bezeichnet im Kontext des FDMs eine digitale Infrastruktur zur strukturierten Speicherung, Verwaltung und Bereitstellung von Forschungsdaten. Repositorien ermöglichen es, Daten zugänglich zu machen, sowohl zum internen Austausch zwischen Forschenden als auch zur öffentlichen Nachnutzung durch die Forschungsgemeinschaft. Sie spielen insbesondere in den letzten Phasen des Forschungsdatenlebenszyklus (Publizieren, Archivieren und Nachnutzung) eine zentrale Rolle. Im Gegensatz zu klassischen Archiven, die in der Regel abgeschlossene und langfristig aufzubewahrende Daten speichern, können Repositorien auch aktuelle, noch nicht ausgewertete Forschungsdaten enthalten. Durch geeignete Authentifizierungs- und Zugriffskonzepte lassen sich diese Daten projekt- oder institutsintern teilen oder gezielt für eine spätere Veröffentlichung vorbereiten. In diesem Zusammenhang wird von sogenannten *dark data* [54] und *failed data* [55] gesprochen. *Dark data* bezieht sich auf Forschungsdaten, die zwar erzeugt, jedoch nie formal gespeichert oder veröffentlicht werden, insbesondere in kleineren Projekten ohne strukturierte Ablage. *Failed data* bezeichnet Daten, die in Forschungsprozessen entstanden sind, jedoch nicht zur erwarteten Erkenntnis führten und deshalb unpubliziert bleiben. Beide Datentypen bergen wertvolles Potenzial für die Wissenschaft, etwa zur Vermeidung von Redundanzen oder als Grundlage für großangelegte Analysen. Im Bereich der Repositorien lassen sich zwei grundlegende Ansätze zur Datenablage unterscheiden: tiefe und breite Repositorien [56]. Tiefe Repositorien spezialisieren sich auf wenige, stark standardisierte Datentypen und kommen häufig in disziplinspezifischen Großprojekten zum Einsatz. Die Standardisierung erleichtert die Entwicklung effizienter Such- und Analysewerkzeuge. Breite Repositorien hingegen nehmen eine Vielzahl unterschiedlicher Datentypen auf, häufig aus verschiedenen Fachdisziplinen. Diese Heterogenität stellt höhere Anforderungen an die Indexierung und Durchsuchbarkeit, bietet jedoch gleichzeitig eine Lösung für die strukturierte Ablage kleiner, vielfältiger Forschungsprojekte, vor allem in interdisziplinären Kontexten [54].

## 2.5.6 Elektronisches Laborbuch

Laborbücher im Papierformat, in denen Forschungsarbeiten unmittelbar während ihrer Durchführung dokumentiert werden, sind in vielen naturwissenschaftlichen Disziplinen wie der Chemie oder Biologie nach wie vor alltäglicher Standard und stellen einen bedeutenden Wissensspeicher innerhalb von Forschungseinrichtungen dar [57]. In ihnen werden unter anderem Versuchsaufbauten, Versuchsdurchführungen, Beobachtungen sowie Ergebnisse und daraus abgeleitete Erkenntnisse festgehalten. Papierbasierte Laborbücher haben jedoch klare Nachteile: Sie lassen sich nur schwer durchsuchen, und auch das Teilen der darin enthaltenen Informationen ist umständlich [57, 58]. Daher gibt es bereits seit längerer Zeit Bestrebungen, sogenannte elektronische Laborbücher – Electronic Lab Notebooks (ELNs) – zu etablieren [59]. ELNs ermöglichen die digitale Dokumentation jener Forschungsprozesse, die traditionell in papierbasierten Laborbüchern festgehalten wurden. Sie kommen insbesondere in den frühen Phasen des Forschungsdatenlebenszyklus, etwa bei der Datenakquise und -analyse, zum Einsatz. Durch ihre digitale Struktur lassen sich ELNs effizient durchsuchen und gezielt mit anderen Forschenden teilen. Darüber hinaus können sie große Datenmengen speichern und sicher archivieren. Ein wesentlicher Vorteil von ELNs ist die verbesserte Nachvollziehbarkeit und Reproduzierbarkeit wissenschaftlicher Ergebnisse, da eine einheitliche und strukturierte Dokumentation unterstützt wird [58]. Besonders relevant ist in diesem Zusammenhang die Möglichkeit, wissenschaftliche Geräte direkt an ein ELN anzubinden, sodass Messdaten automatisiert erfasst und innerhalb des Systems weiterverarbeitet werden können [57]. Wie klassische Laborbücher können auch ELNs sehr unterschiedlich ausgestaltet sein. Ihre Funktionalität hängt häufig stark von der jeweiligen Fachdisziplin und den spezifischen Anforderungen eines Forschungsumfelds ab. Ähnlich zu Repositorien lassen sich ELNs in zwei Kategorien einteilen: fachspezifische und allgemeine Systeme [59]. Beide Kategorien haben ihre jeweiligen Einsatzgebiete. Bei der Einführung eines ELNs sollte daher stets geprüft werden, welche Anforderungen an die Funktionalität bestehen, um eine geeignete Systemarchitektur auswählen zu können.

### 2.5.7 Workflowsysteme

Workflow-Management-Systeme (WfMS) sind Softwarelösungen zur Modellierung, Ausführung und Dokumentation strukturierter Arbeitsabläufe. Ein Workflow beschreibt dabei eine Abfolge einzelner Schritte, die sequentiell oder parallel ablaufen können [60]. Ziel von WfMS ist es, Prozesse planbar, reproduzierbar und automatisierbar zu machen.

Insbesondere in der Wissenschaft gewinnen WfMS an Bedeutung, da viele Forschungsprozesse wie Datenanalysen, Simulationen oder die Ablage von Ergebnissen repetitiv und datenintensiv sind [61]. Ein Workflow kann sowohl automatisierte Schritte wie Programmausführungen als auch manuelle Tätigkeiten umfassen, etwa die Bedienung von Laborgeräten oder das Eintragen von Messwerten [62]. WfMS helfen, diese Schritte zu formalisieren und in strukturierte Abläufe zu überführen. Sie schaffen dadurch eine klare Trennung zwischen methodischem Wissen und der technischen Umsetzung, was die Nachvollziehbarkeit und Wiederverwendung von Prozessen erleichtert. Zur Modellierung kommen skriptbasierte Sprachen (z.B. Nextflow [63], Snakemake [64]) oder grafische Oberflächen zum Einsatz. Häufig werden Workflows als gerichtete Graphen dargestellt, deren Knoten Prozessschritte und deren Kanten Daten- oder Steuerflüsse repräsentieren [65].

Ein verbreiteter Standard ist die *Business Process Model and Notation* (BPMN). Sie erlaubt eine visuell verständliche und zugleich formal exakte Darstellung komplexer Abläufe, inklusive Schleifen, Bedingungen und paralleler Ausführungen [60].

Ein zentraler Vorteil von WfMS ist die Wiederverwendbarkeit definierter Prozessmodule, die in neuen Kontexten angepasst oder erweitert werden können. Zusätzlich dokumentieren WfMS die Ausführungshistorie und erzeugen wertvolle Metadaten zur Datenherkunft. Dadurch werden sowohl Qualitätssicherung als auch die Zusammenarbeit in größeren Forschungsprojekten unterstützt. Sie lassen sich in nahezu allen Phasen des Forschungsdatenlebenszyklus einsetzen und tragen wesentlich zur Effizienz, Transparenz und Reproduzierbarkeit moderner Forschung bei.



## 3 Einheitliches Konzept für FAIRes Materialdesign

### 3.1 FAIRes Materialdesign

Die Entwicklung neuer Materialien ist ein komplexer und iterativer Prozess, der von Experimenten über Simulationen bis hin zu datengetriebenen Methoden reicht. Obwohl sich die grundlegenden Abläufe im Materialdesign häufig ähneln, werden sie in der Praxis meist individuell und ohne einheitliche Struktur dokumentiert. Während Forschungsdaten häufig archiviert werden, fehlt es oft an einer vergleichbaren systematischen Abbildung der zugrunde liegenden Methoden und Prozessschritte. Dies erschwert die Nachvollziehbarkeit, Wiederverwendung und Reproduzierbarkeit von Forschungsergebnissen erheblich.

Die FAIR-Prinzipien bieten hierfür einen Leitgedanken und stehen im Mittelpunkt zahlreicher Forschungsinitiativen wie beispielsweise *NFDI4Ing* [66]. Ziel ist es, Forschungsdaten nicht nur verfügbar zu machen, sondern auch so zu strukturieren, dass sie über Disziplinen hinweg nachnutzbar sind. Während zahlreiche technische Umsetzungen für ein FAIRes Forschungsdatenmanagement bereits existieren, konzentrieren sie sich überwiegend auf die Datenebene. Eine vergleichbare systematische Modellierung wissenschaftlicher Methoden und Prozesse bleibt bislang jedoch weitgehend aus.

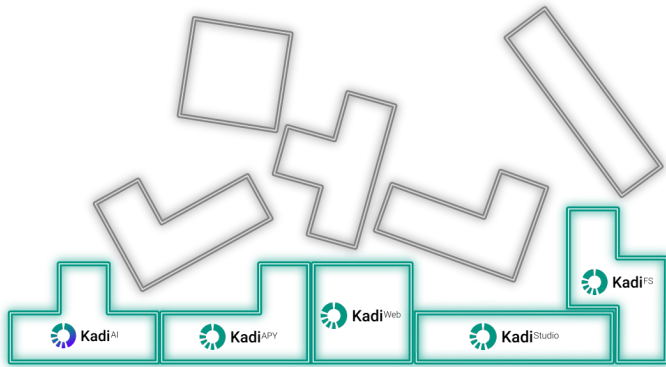
Dieses Kapitel entwickelt darauf aufbauend ein eigenes, generisches Konzept, das Forschungsprojekte in klar strukturierte Prozessschritte zerlegt und deren systematische Dokumentation ermöglicht. Ergänzend dazu wird ein Konzept zur

FAIRen Datendokumentation entwickelt, das die während der Prozesse entstehenden Forschungsdaten mitsamt Metadaten, Hierarchien und Relationen konsistent erfasst und den jeweiligen Prozessschritten zuordnet. Damit ergänzt die Datendokumentation die Prozessmodellierung komplementär und stellt ihre inhaltliche Entsprechung auf der Datenebene dar. Die beiden Konzepte werden anschließend auf die digitale Materialforschung angewendet, um grundlegende und wiederkehrende Abläufe des Materialdesigns zu identifizieren. Die so gewonnene einheitliche Struktur bildet den konzeptionellen Rahmen dieser Arbeit: Sie dient als Grundlage für ein *FAIRes Materialdesign*, das nicht nur Daten, sondern auch Prozesse transparent und reproduzierbar macht.

Zur praktischen Umsetzung dient die virtuelle Forschungsumgebung *Kadi4Mat*, eine quelloffene Plattform für Forschungsdatenmanagement, die maßgeblich am Karlsruher Institut für Technologie entwickelt wird [67]. Ziel von Kadi4Mat ist es, Forschungsdaten über ihren gesamten Lebenszyklus hinweg strukturiert zu erfassen, zu verwalten und wiederverwendbar zu machen. Damit adressiert die Plattform zwei zentrale Herausforderungen moderner Wissenschaft: die Gewährleistung von Reproduzierbarkeit sowie die systematische Speicherung großer Datenmengen als Grundlage datengetriebener Forschung, auch bekannt als das vierte wissenschaftliche Paradigma [45].

Kadi4Mat verfolgt hierzu einen modularen Ansatz, bei dem bestehende Forschungsprozesse nicht ersetzt, sondern durch flexibel kombinierbare Softwarekomponenten ergänzt werden. Dazu zählen unter anderem eine Weboberfläche (KadiWeb) mit Funktionen eines Community-Repositories und elektronischen Laborbuchs, Werkzeuge zur Integration lokaler Dateisysteme (KadiFS), eine Python-basierte Programmierschnittstelle (KadiAPY), Module für maschinelles Lernen (KadiAI) sowie ein generisches Workflow-System (KadiStudio). Über klar definierte Schnittstellen lassen sich zudem weitere fachspezifische Komponenten integrieren, sodass die Plattform domänenübergreifend anpassbar bleibt.

Abbildung 3.1 veranschaulicht dieses modulare Konzept. Die Rekombination der verfügbaren Bausteine erlaubt es, passgenaue Lösungen für unterschiedliche Anforderungen des Forschungsdatenmanagements zu konfigurieren.



**Abbildung 3.1:** Kadi4Mat Ökosystem. Kadi4Mat stellt verschiedene Module – KadiWeb, KadiStudio, KadiAPY, KadiAI, KadiFS – zur Verfügung, die unterschiedliche Funktionen implementieren. Durch definierte Schnittstellen können weitere Komponenten, angedeutet durch die grauen Formen, hinzugefügt werden.

Im Rahmen dieser Arbeit wird Kadi4Mat nicht nur als Infrastruktur eingesetzt, sondern auch im Hinblick auf die Anforderungen des digitalen Materialdesigns weiterentwickelt. Besonders im Fokus stehen dabei zwei zentrale Module: *KadiStudio* als generisches, visuelles System zur FAIRen Modellierung wissenschaftlicher Workflows, das die in Unterabschnitt 3.2.1 vorgestellte Prozessstrukturierung praktisch umsetzt und *KadiWeb* als zentrale Plattform zur strukturierten Datendokumentation, Verwaltung und Veröffentlichung von Forschungsdaten. KadiWeb ermöglicht es, die im Forschungsprozess erzeugten Daten mit Metadaten, Relationen und Hierarchien zu verknüpfen und damit das in Unterabschnitt 3.3.1 beschriebene Konzept der FAIRen Datendokumentation umzusetzen. Auf diese Weise konnten die entwickelten Konzepte nicht nur theoretisch beschrieben, sondern auch in einer realen Forschungsumgebung implementiert und evaluiert werden.



## 3.2 Prozessstrukturierung

Wie im vorangegangenen Abschnitt dargelegt, reicht ein FAIRes Materialdesign über die reine Speicherung von Forschungsdaten hinaus. Ebenso entscheidend ist die nachvollziehbare Abbildung der zugrunde liegenden Prozesse. Ziel dieser Arbeit ist es daher, den digitalen Materialentwicklungsprozess durch die Kombination von maschinellern Lernen und Forschungsdatenmanagement effizienter und reproduzierbarer zu gestalten. Dies umfasst nicht nur die Archivierung entstandener Daten, sondern auch die Modellierung wissenschaftlicher Abläufe in Form automatisierbarer und FAIRer Workflows.

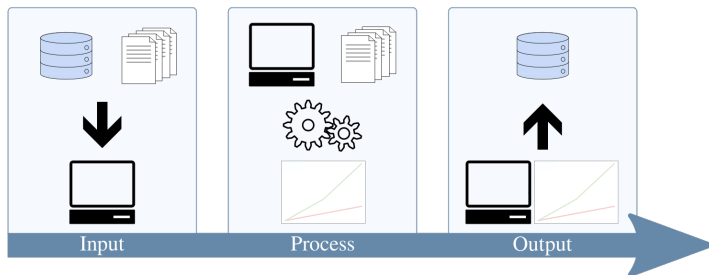
Um komplexe Prozesse in solcher Form abzubilden, wurde im Rahmen dieser Arbeit ein generisches Konzept zur Prozessstrukturierung entwickelt und in [68] publiziert. Es unterteilt ein Forschungsprojekt iterativ in atomistische, logisch abgegrenzte Schritte und ermöglicht so eine konsistente und wiederverwendbare Beschreibung experimenteller und digitaler Forschung. Die vorgestellte Modellierung bildet die methodische Basis für das in Unterabschnitt 3.2.2 beschriebene Workflowsystem *KadiStudio* und dient zugleich als konzeptioneller Rahmen für die in den folgenden Kapiteln entwickelten Methoden. Im Folgenden wird das Konzept in seinen Abstraktionsebenen eingeführt und systematisch hergeleitet.

### 3.2.1 Konzept

Die Entwicklung eines generischen Konzepts zur Modellierung von Forschungsprozessen setzt die Identifikation einer gemeinsamen Struktur voraus, die sich auf beliebige Prozesse übertragen lässt. Zu diesem Zweck werden wissenschaftliche Abläufe in dieser Arbeit schrittweise in atomistische Beschreibungen auf unterschiedlichen Detaillierungsebenen zerlegt. Der Begriff atomistische Beschreibung bezieht sich hier auf die Unterteilung eines Prozesses in unteilbare, kleinste Einheiten [65]. Diese Vorgehensweise erlaubt es, generische Elemente innerhalb von Abläufen zu identifizieren, die in anderen Anwendungsfällen wiederverwendet werden können.

## IPO Abstraktionsprinzip

Grundlage für die iterative Zerlegung bildet das in Abbildung 3.2 dargestellte *Input–Process–Output*-Prinzip (IPO), das dem im deutschsprachigen Raum gebräuchlichen EVA-Modell (Eingabe–Verarbeitung–Ausgabe) entspricht. Für die Konsistenz mit der in [68] veröffentlichten Methodik wird in dieser Arbeit jedoch die englischsprachige Bezeichnung IPO verwendet, die ein etabliertes Konzept aus der Systemanalyse [69, 70] beschreibt.



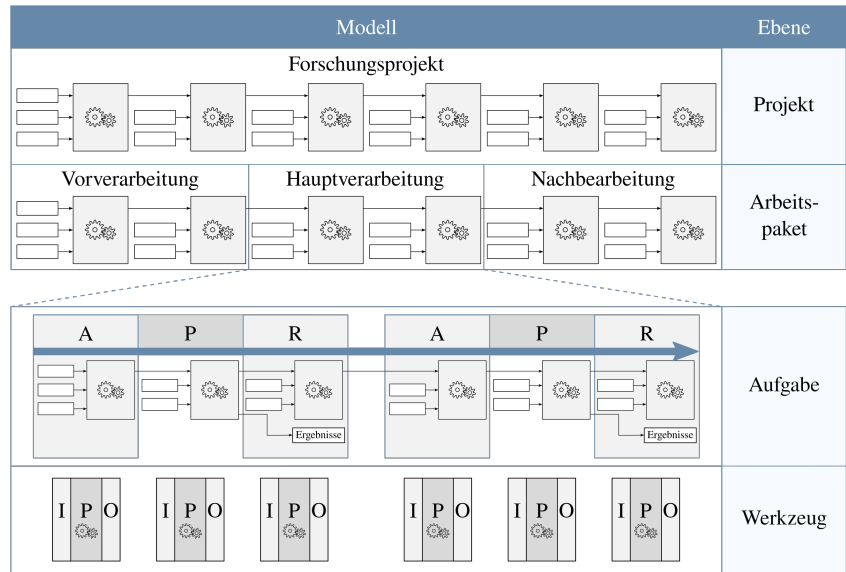
**Abbildung 3.2:** Schematische Visualisierung des IPO-Prinzips. Definierte Eingaben parametrisieren einen Prozess, der anschließend ausgeführt wird. Die erzeugten Ergebnisse sind über Ausgaben verfügbar. Adaptiert von [68].

Das IPO-Prinzip beschreibt Prozesse als Kombination aus Eingabe, Verarbeitung und Ausgabe. Ein Prozess beginnt demnach mit der Bereitstellung und Aufbereitung der erforderlichen Daten über definierte Eingaben. Anschließend erfolgt deren Verarbeitung gemäß einem festgelegten Prozess. Die hierbei generierten Ergebnisse werden abschließend über definierte Ausgaben bereitgestellt und stehen dadurch für nachfolgende Prozessschritte oder Auswertungen zur Verfügung.

## Hierarchisierung des Forschungsprozesses

Das iterative Anwenden des IPO-Prinzips auf einen Forschungsprozess erlaubt dessen Strukturierung auf unterschiedlichen Abstraktionsebenen. Die umfangreichste atomistische Beschreibung eines Forschungsprozesses findet auf Projektebene

statt und umfasst den vollständigen Ablauf, wie in Abbildung 3.3 dargestellt. Dies ist so zu verstehen, dass ein Forschungsprojekt nur durch den Gesamtprozess beschrieben werden kann. Die experimentelle Untersuchung einer Probe erfordert beispielsweise die Modellierung des gesamten Experiments. Eine weitergehende Unterteilung ist nicht möglich.



**Abbildung 3.3:** Abstraktion eines Forschungsprozesses auf unterschiedlichen Abstraktionsebenen. Die mit Zahnrädern markierten Felder stellen Arbeitsschritte dar, während die weißen Boxen deren Parametrisierung zeigt. APR bezieht sich auf die Beschreibung dieser Aufgaben als Datenakquise, -verarbeitung und -weiterleitung (englisch acquisition, processing, routing). Durch die iterative Strukturierung eines Forschungsprozesses gemäß des IPO-Prinzips wird dieser letztlich über mehrere generische Werkzeuge definiert. Adaptiert von [68].

Zur Beschreibung des Forschungsprozesses auf der weniger komplexen Ebene der Arbeitspakete wird das IPO-Prinzip herangezogen. Dieses unterteilt den gesamten Forschungsprozess in drei aufeinanderfolgende Phasen: Vorverarbeitung, Hauptverarbeitung und Nachbearbeitung. Diese entsprechen den Elementen

Eingabe (Input), Verarbeitung (Process) und Ausgabe (Output). Bezogen auf die oben erwähnte experimentelle Untersuchung einer Probe umfasst die Vorverarbeitung die Vorbereitung des Experiments, die Hauptverarbeitung die eigentliche Versuchsdurchführung und die Nachbearbeitung die abschließende Analyse und Interpretation der gewonnenen Daten. Jedes dieser Arbeitspakete kann eine beliebige Zahl an Einzelschritte enthalten. Die Vorverarbeitung könnte zum Beispiel das Schleifen und Polieren der Probe ebenso umfassen wie die Kalibrierung des Mikroskops. Die Aufteilung in Vor-, Haupt- und Nachbearbeitung ermöglicht es bereits generische Bausteine zu identifizieren, die in anderen Anwendungsfällen wiederverwendet werden können, wie zum Beispiel die Probenaufbereitung oder die Kalibrierung des Mikroskops. Für die generische Strukturierung von Forschungsprozessen ist die Detaillierung in Vor-, Haupt- und Nachbearbeitung noch nicht ausreichend. Stattdessen ist eine feingranularere Beschreibung der Arbeitspakete erforderlich. Diese Beschreibung wird im Folgenden dargelegt.

## **APR-Struktur**

Zur weiteren Detaillierung des Forschungsprozesses wird das IPO-Prinzip auf die zuvor identifizierten Arbeitspakete angewandt. Konkret wird jede zusammenhängende Aufgabe innerhalb der Arbeitspakete einzeln betrachtet. Eine zusammenhängende Aufgabe bezeichnet an dieser Stelle einen in sich geschlossenen, logisch abgrenzbaren Arbeitsschritt, wie beispielsweise das Schleifen einer Probe oder die Mikroskop-Kalibrierung in der Vorverarbeitung. Die Strukturierung dieser Aufgaben anhand des IPO-Prinzips führt zu einer Prozessbeschreibung auf Aufgabenebene, die als APR-Struktur bezeichnet wird und die schematisch in Abbildung 3.3 dargestellt ist.

Sie beschreibt den Datenfluss innerhalb und zwischen den einzelnen Schritten einer Aufgabe durch Datenakquise, -verarbeitung und -weiterleitung. Im Deutschen wäre die Entsprechung AVW, allerdings wird auch hier die englische Bezeichnung beibehalten, um die Konsistenz zur zugehörigen Publikation [68] zu wahren.

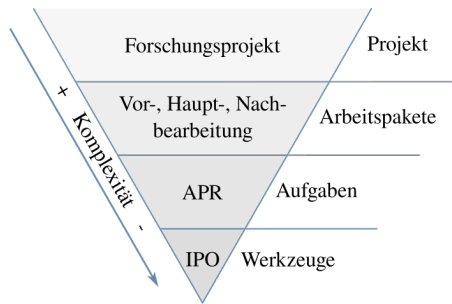
In dieser Abstraktion umfasst die Datenakquise alle Arbeitsschritte, die für die jeweilige Aufgabe die notwendigen Daten sammeln und aufbereiten, um sie anschließend an die Datenverarbeitung weiterzuleiten. Innerhalb der Datenverarbeitung werden diese Eingabedaten verarbeitet, um neue Daten zu erzeugen. Die so gewonnenen Daten, in Form von Ergebnissen oder Zwischenergebnissen, werden schließlich in der Datenweiterleitung weitergegeben. Die Weiterleitung der Ergebnisse kann dabei an beliebige Ziele erfolgen, etwa in eine Datei oder an die Datenakquise einer nachfolgenden Aufgabe. Die Arbeitspakete Vor-, Haupt- und Nachverarbeitung können jeweils aus beliebig vielen dieser APR-Prozesse bestehen und lassen sich daher als Aneinanderreihung dieser Elemente verstehen.

Das APR-Modell dient in erster Linie der Beschreibung des Datenflusses innerhalb jeder Aufgabe. Dadurch wird die Nachvollziehbarkeit des Datenflusses verbessert und das Verständnis des Prozesses gefördert. Außerdem lassen sich die identifizierten APR-Strukturen wiederverwenden und in unterschiedlichen Anwendungsfällen einsetzen. Da APR-Prozesse aus einer fest definierten Kombination mehrerer einzelner Arbeitsschritte bestehen, sind sie zu anwendungsspezifisch, um als generische Struktur für die Prozessstrukturierung zu dienen. Daher werden sie im Folgenden durch das erneute Anwenden des IPO-Prinzips weiter detailliert.

## **Atomistische Modellierung auf Werkzeugebene**

Zur Erstellung einer weiteren Abstraktionsebene wird das IPO-Prinzip ein weiteres Mal auf jeden Arbeitsschritt innerhalb der APR-Prozesse angewandt. Dies führt zu einer atomistischen Beschreibung des Forschungsprozesses auf Ebene der Werkzeuge. Die einzelnen Arbeitsschritte jedes APR-Prozesses werden nun als generische und wiederverwendbare Werkzeuge mit spezifischen Eingaben, Ausgaben und einem definierten Prozess beschrieben. Im Beispiel der Probenvorbereitung könnten dies beispielsweise die Arbeitsschritte Schleifen, Polieren und Ätzen sein. Da diese Strukturierung der ursprünglichen Definition des IPO-Prinzips entspricht, wird an dieser Stelle ebenfalls von IPO gesprochen.

Die eigentliche Datenerzeugung des Forschungsprojekts findet auf diesem Abstraktionslevel statt. Die Daten werden im Prozessschritt generiert und anschließend über den Output weitergegeben. Auf diese Weise lässt sich der Ursprung der generierten Daten präzise bestimmen und mit den entsprechenden Metadaten und Abhängigkeiten versehen, was eine FAIRe Speicherung ermöglicht. Die generischen Beschreibungen der Arbeitsschritte können nun beliebig kombiniert werden, um unterschiedlichste Prozessketten zu realisieren.



**Abbildung 3.4:** Reduktion der Prozesskomplexität durch das iterative Anwenden des IPO-Prinzips. Auf jedem Abstraktionslevel wird der Prozess über atomistische Schritte beschrieben. Das Funktionslevel stellt eine allgemein anwendbare Abstraktion zur Modellierung beliebiger Forschungsprozesse dar. Adaptiert von [68].

Zusammenfassend lässt sich festhalten, dass sich die in Abbildung 3.4 schematisch dargestellte Methode dazu eignet, die Komplexität beliebiger Forschungsprozesse bis hin zu atomistischen Beschreibungen der einzelnen Arbeitsschritte sprich der Werkzeuge zu reduzieren. Die umgekehrte Anwendung dieses Konzepts ermöglicht folglich den Aufbau eines generischen Systems zur Modellierung von Forschungsprozessen. Diese Modellierung basiert zunächst auf der allgemeinen Beschreibung der eingesetzten Werkzeuge nach dem IPO-Prinzip. Durch das anschließende Hinzufügen von Verknüpfungen, die den Datenfluss und die Parametrisierung der Werkzeuge im Sinne des APR-Modells beschreiben, wird eine weitergehende Spezifizierung des Prozesses erreicht. Die Aneinanderreihung mehrerer APR-Prozesse erlaubt schließlich die Abbildung von Vor-, Haupt-

und Nachverarbeitung, die zusammengekommen den vollständigen Forschungsprozess repräsentieren. Ähnliche Konzepte, bei denen atomistische Funktions-Beschreibungen nach dem IPO-Prinzip zu einem Workflow verknüpft werden, finden sich in zahlreichen etablierten Workflow-Management-Systemen. So setzen CWL [71], Snakemake [64] und Nextflow [63] auf skriptbasierte Ansätze, während Programme wie KNIME [72] oder Orange [73] dies in grafischen Oberflächen realisieren. Die weite Verbreitung dieser Konzepte in bewährten Systemen unterstreicht die Eignung des vorgestellten Ansatzes für die Formulierung von Workflows.

Die konkrete Umsetzung dieses Konzepts in einem generischen Workflow-System zur FAIRen Modellierung wissenschaftlicher Prozesse, eingebettet in die virtuelle Forschungsumgebung Kadi4Mat, wird im Folgenden vorgestellt.

## 3.2.2 KadiStudio

KadiStudio erweitert das Kadi4Mat-Ökosystem um die Möglichkeit, wissenschaftliche Prozesse in Form von Workflows abzubilden, zu automatisieren und reproduzierbar auszuführen. Im Mittelpunkt steht die Anwendung der FAIR-Prinzipien auf Forschungsprozesse selbst, also auf die Art und Weise, wie Daten erzeugt, verarbeitet und analysiert werden. Damit stellt KadiStudio die praktische Umsetzung des in Unterabschnitt 3.2.1 entwickelten Konzepts zur generischen Prozessstrukturierung dar. Die Software ermöglicht es, komplexe Abläufe in klar definierte, wiederverwendbare Einheiten zu zerlegen und diese systematisch zu dokumentieren. So wird die Nachvollziehbarkeit wissenschaftlicher Ergebnisse verbessert und die Grundlage für eine weitreichende Automatisierung datengetriebener Forschung geschaffen.

Im Folgenden wird die technische Umsetzung in der frei zugänglichen Software KadiStudio beschrieben [74]. Dabei wird gezeigt, wie das vorgestellte Konzept in einer benutzerfreundlichen und zugleich generisch erweiterbaren Umgebung realisiert ist. Die Inhalte dieses Kapitels sind bereits in der Arbeit von Griem et al. [68] veröffentlicht.

### 3.2.2.1 FAIRe Workflows

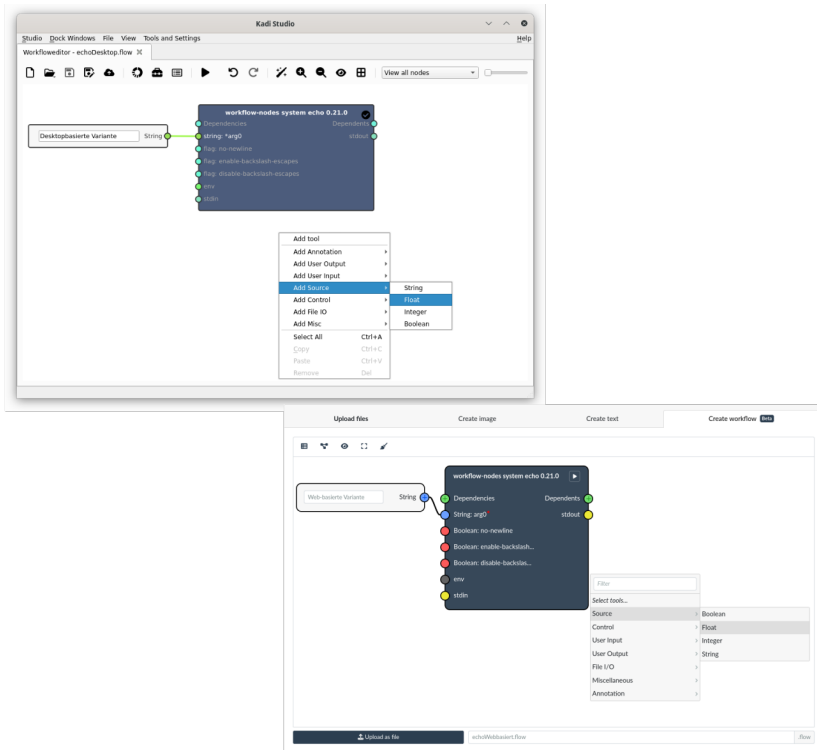
KadiStudio wurde mit dem Ziel entwickelt, eine benutzerfreundliche Softwarelösung für die FAIRe Modellierung wissenschaftlicher Prozesse bereitzustellen. Innerhalb des Kadi4Mat-Ökosystems ist es Teil der elektronischen Laborbuch-Funktionalitäten und bildet die technische Grundlage für die in Unterabschnitt 3.2.1 vorgestellte Prozessstrukturierung.

Technisch basiert der Editor auf einer quelloffenen Node-Editor-Bibliothek für das Qt-GUI-Framework [75]. In der virtuellen Forschungsumgebung Kadi4Mat existieren zwei Versionen dieses Editors (siehe Abbildung 3.5, die beide dasselbe JSON-basierte Datenformat zur Beschreibung von Workflows verwenden und so deren Interoperabilität sicherstellen. Zum einen gibt es eine desktopbasierte, eigenständige Softwarevariante namens KadiStudio, die ohne Internetverbindung oder laufenden Webserver genutzt werden kann, und zum anderen eine webbasierte Version, die in KadiWeb integriert ist. Da in dieser Arbeit stets die Desktopversion von KadiStudio zum Einsatz kommt, beziehen sich alle folgenden Erwähnungen KadiStudios auf diesen Editor.

Um zu veranschaulichen, wie KadiStudio eine FAIRe Modellierung von Forschungsprozessen ermöglicht, soll der Begriff FAIR zunächst auf Forschungsprozesse angewendet werden. Während Findable und Accessible unverändert übernommen werden können, müssen insbesondere die Elemente Interoperable und Reusable neu interpretiert werden. Interoperable bedeutet in diesem Kontext, dass modellierte Workflows auf allgemein anerkannten Datenformaten basieren und bei Bedarf flexibel an andere Formate angepasst werden können. Das Prinzip Reusable umfasst nach der Definition von Draxl et al. [45] zwei Aspekte: Zum einen muss ein Workflow die zuverlässige Reproduktion von Ergebnissen sicherstellen, zum anderen soll er ganz oder teilweise in unterschiedlichen Anwendungsfällen wiederverwendbar sein.

Die Bedingungen Findable und Accessible sind im vorliegenden Workflow-System bereits durch die direkte Integration in Kadi4Mat erfüllt. Workflows können im





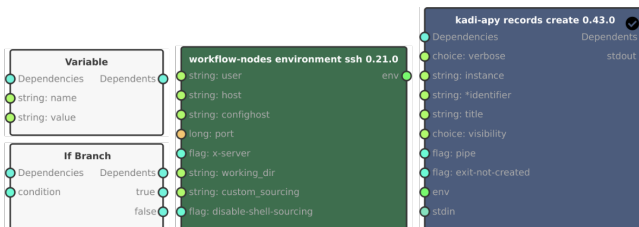
**Abbildung 3.5:** Übersicht über die verfügbaren Editoren: Oben ist die Desktopversion und unten die webbasierte Version gezeigt. Workflows werden durch das Hinzufügen und Verbinden von Knoten mittels eines Point-and-Click Mechanismus hinzugefügt.

Community Repository abgelegt und so mit der wissenschaftlichen Gemeinschaft geteilt werden. Wie bei gewöhnlichen Forschungsdaten können die in Kadi4Mat gespeicherten Workflows mit beschreibenden Metadaten versehen werden, die selektiv durchsuchbar sind. Daher stehen bei der technischen Umsetzung von KadiStudio insbesondere die Interoperabilität und Wiederverwendbarkeit im Mittelpunkt. Zudem werden die Anforderungen an den Editor, generisch und benutzerfreundlich zu sein, gemäß [76] berücksichtigt.

### 3.2.2.2 Workflow-Modellierung

Das in Unterabschnitt 3.2.1 beschriebene Schema zur Abstraktion wissenschaftlicher Prozesse definiert Werkzeuge als grundlegende Bausteine nach dem IPO-Modell. In KadiStudio sind diese als Knoten umgesetzt, die sich über eine grafische Benutzeroberfläche (GUI) hinzufügen und miteinander verknüpfen lassen, wie in Abbildung 3.5 gezeigt. KadiStudio verfolgt damit einen Bottom-Up-Ansatz, bei dem nur grundlegende Werkzeuge vorgegeben werden. Die Erstellung komplexer Zusammenhänge ist den Anwendern überlassen.

Jeder Knoten repräsentiert ein Werkzeug lässt sich einem Knotentyp zuordnen: Werkzeugknoten integrieren externe Programme oder Funktionen, Umgebungsknoten ergänzen und unterstützen die Ausführung der Werkzeugknoten, und eingebaute Knoten ermöglichen die Steuerung des Workflow-Ablaufs, fügen interaktive Optionen hinzu und stellen Variablen im Editor zur Verfügung. Die verfügbaren Knotentypen sind in Abbildung 3.6 dargestellt.



**Abbildung 3.6:** Verfügbare Knotentypen. Eingebaute Knoten sind grau, Umgebungsknoten grün und Werkzeugknoten blau.

Entsprechend dem IPO-Prinzip beschreibt jeder Knoten einen spezifischen Prozess, der sowohl Eingaben als auch Ausgaben aufweist, dargestellt durch Eingangs- und Ausgangsport. Je nach Funktion lassen sich die Ports in Parametrisierungs-, Abhängigkeits- (*Dependency*), Umgebungs- (*Environment*) sowie stdin/stdout-Ports unterteilen. Parametrisierungs-Ports übergeben zur Ausführung notwendige Argumente oder Optionen. Mittels Abhängigkeits-Ports können Ausführungsreihenfolgen definiert und Kontrollstrukturen wie if-Bedingungen oder for-Schleifen

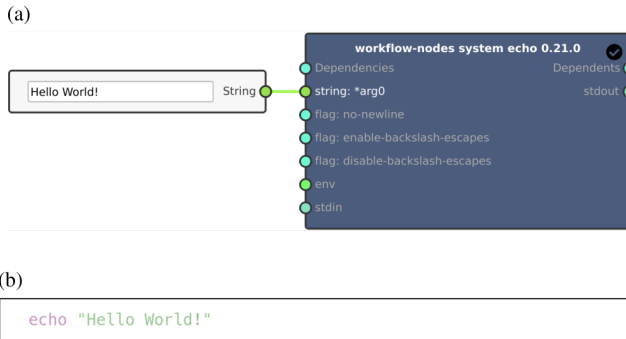
abgebildet werden. Umgebungs-Ports dienen dazu einen Werkzeugknoten in einer spezifischen Umgebung auszuführen, etwa über eine Secure Shell (SSH). Über stdin- und stdout-Ports kann zudem die Ausgabe eines Knotens an einen anderen weitergeleitet werden. Gerade die Definition dieser Ein- und Ausgabeschnittstellen bildet die Grundlage der Workflow-Modellierung in KadiStudio und entspricht dem in Abbildung 3.2 illustrierten IPO-Prinzip.

Durch das Verbinden und Parametrisieren der Knoten über die genannten Ports lässt sich der Datenfluss gemäß dem APR-Modell in den Workflow einfügen (siehe Unterabschnitt 3.2.1). So entstehen klar definierte Beziehungen, die auf einen Blick erkennen lassen, welche Eingaben ein Prozess benötigt und wohin seine Ausgaben weitergeleitet werden. Dadurch wird der Datenfluss übersichtlich und nachvollziehbar strukturiert. Bei der Ausführung eines Workflows übersetzt KadiStudio die Knoten und ihre Verbindungen in Befehle für die Kommandozeilenschnittstelle (CLI). Der Workflow-Editor kann somit als grafische Programmiersprache verstanden werden, deren intuitive Bedienung auch unerfahrenen Nutzern die Modellierung eigener Workflows ermöglicht.

#### 3.2.2.3 Parametrisierung von Workflows

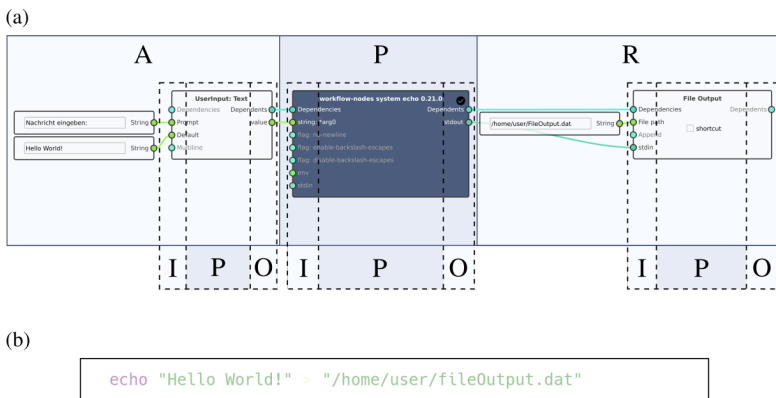
Die Parametrisierung in KadiStudio erfolgt über die Eingangsports der Werkzeugknoten, die den zugrunde liegenden CLI-Befehlen zugeordnet sind. Auf diese Weise können Argumente und Optionen direkt über die grafische Oberfläche definiert werden. Ein einfaches Beispiel für eine solche Parametrisierung ist in Abbildung 3.7 dargestellt. In diesen wird der *echo*-Knoten in (a) durch die Verbindung mit einem String-Quellknoten parametrisiert und so der in (b) gezeigte CLI-Befehl generiert.

Neben Quellknoten können Ausgaben auch über stdout-Ports an Eingaben anderer Knoten weitergeleitet werden. Dies ermöglicht die Abbildung komplexer Datenflüsse, wie sie im APR-Modell beschrieben sind. Die graphische Darstellung dieses Verbindens von stdout- und stdin-Ports ist in Abbildung 3.8 (a) zu sehen. Der



**Abbildung 3.7:** Übersicht zur Knotenkonfiguration: (a) Parametrisierung eines *echo*-Knotens durch einen String-Quellknoten. (b) Erzeugter Befehl.

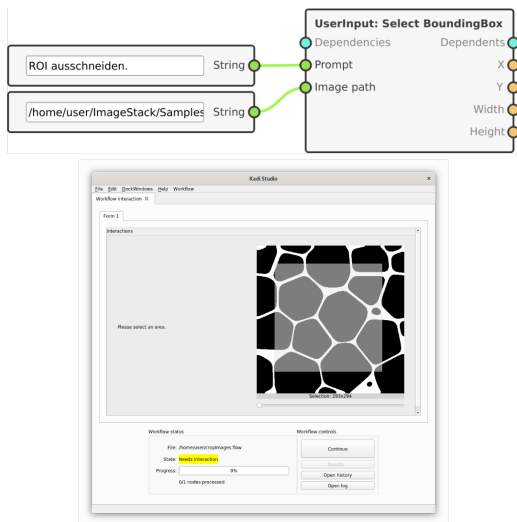
resultierende Befehl in Abbildung 3.8 (b) zeigt, dass die Standardausgabe des ersten Knotens direkt in die Standardeingabe des zweiten Knotens geleitet wird. Das Beispiel zeigt zudem, wie sich die Arbeitsschritte Datenakquise (*UserInput: Text*), Datenprozessierung (*echo*) und Datenweiterleitung (*File Output*) im APR-Modell gliedern lassen.



**Abbildung 3.8:** Workflow-Modellierung in KadiStudio: Mehrere Werkzeuge werden nach dem APR-Modell zu einem Workflow verknüpft. Das in (a) gezeigte Verbinden von stdout- und stdin-Ports leitet die Ausgabe des Werkzeugknotens an den File-Output-Knoten weiter und entspricht dem in (b) dargestellten CLI-Befehl. Adaptiert von [68].

Wie in Abbildung 3.8 gezeigt, lässt sich ein Workflow nicht nur über feste Quellknoten, sondern auch interaktiv parametrisieren. Dies geschieht mithilfe von *UserInput*-Knoten, die während der Ausführung Eingaben abfragen und so die weitere Prozessierung steuern. Im gezeigten Beispiel fragt ein *UserInput: Text*-Knoten eine Nutzereingabe ab, die anschließend an den *echo*-Knoten weitergeleitet wird. Zusätzlich sorgen verbundene Abhängigkeits-Ports dafür, dass der *File Output*-Knoten erst nach erfolgreicher Ausführung des *echo*-Knotens gestartet wird.

Neben der Abfrage grundlegender Eingaben wie Text oder Zahlen können über *UserInput*-Knoten auch Dateien ausgewählt oder Bildausschnitte zur weiteren Analyse beschnitten werden wie in Abbildung 3.9 gezeigt. Durch den Einsatz



**Abbildung 3.9:** Verwendung von *UserInput*-Knoten: Während der Ausführung wird der Nutzer zur Eingabe aufgefordert, etwa zur Auswahl eines Bildausschnitts [68].

solcher Nutzerinteraktionen erhält der Anwender Kontrolle über die Ausführung des Workflows und kann ihn flexibel an unterschiedliche Anwendungsfälle anpassen. Workflows sind damit nicht lediglich vordefinierte Skripte für fest definierte

Szenarien, sondern generische Werkzeuge, die sich während der Ausführung an den aktuellen Anwendungsfall anpassen lassen.

Gleichzeitig ermöglicht dieser Abfrage-Mechanismus auch die Abbildung manueller Arbeitsschritte in KadiStudio. Hierzu wird der Nutzer aufgefordert, einen definierten Arbeitsschritt mit bestimmten Eingaben durchzuführen und anschließend die erhaltenen Ergebnisse zurückzumelden, wie in Abbildung 3.10 dargestellt. Um die Reproduzierbarkeit von Workflows zu gewährleisten, in denen solche interaktiven Eingaben erfolgen, werden sämtliche Nutzereingaben in einer Log-Datei gespeichert. Beim Hochladen der generierten Daten in ein Repositorium wie Kadi4Mat können diese geloggten Eingaben als Metadaten verwendet werden.

Abgesehen von den interaktiven Knoten stellt KadiStudio verschiedene weitere eingebaute Knoten wie *Variable*, *Loop* oder *if Branch* bereit, mit denen sich die Workflow-Ausführung steuern und generische Forschungs-Workflows formulieren lassen. Durch das Angebot dieser eingebauten Knoten garantiert das Workflowsystem KadiStudio nicht nur die Reproduzierbarkeit sämtlicher manueller oder digitaler Forschungsprozesse, sondern ermöglicht auch die Wiederverwendbarkeit und den Transfer der erstellten Workflows auf neue Anwendungsfälle.

Form 1

Interactions

**Please investigate the samples using confocal microscopy**

**Device:**  
ZEISS LSM 980

**Settings:**  
**Emission filter:**  
BP380-430  
**Laser:**  
Enterprise  
**Excitation wavelength:**  
364nm  
**Detector**  
Normal

Select scanned images

**Abbildung 3.10:** Integration manueller Arbeitsschritte. Dem Nutzer werden alle notwendigen Parameter bereitgestellt und anschließend aufgefordert, die erzeugten Ergebnisse einzutragen. Adaptiert von [68].

### 3.2.2.4 Erweiterung von Workflows

Um die Interoperabilität modellierter Workflows zu gewährleisten und die heterogene Werkzeuglandschaft der Forschung abzubilden, muss sich das in KadiStudio verfügbare Knotenrepertoire leicht erweitern lassen. So können anwendungsspezifische Werkzeuge und Datenkonvertierungsknoten integriert werden, was etwa die Nutzung unterschiedlicher Dateiformate ermöglicht.

Die Schnittstelle hierfür ist bewusst einfach gehalten. Voraussetzung ist, dass der zugrunde liegende CLI-Befehl ausführbar ist und die Option `--xmlhelp` anbietet. Diese liefert eine maschinenlesbare Befehlsbeschreibung, die der Editor zur visuellen Darstellung nutzt. Falls eine Funktion diese Option nicht bereitstellt, kann sie mithilfe eines Wrapper-Skripts und der Python-Bibliothek *xmlhelpy* ergänzt werden [77].

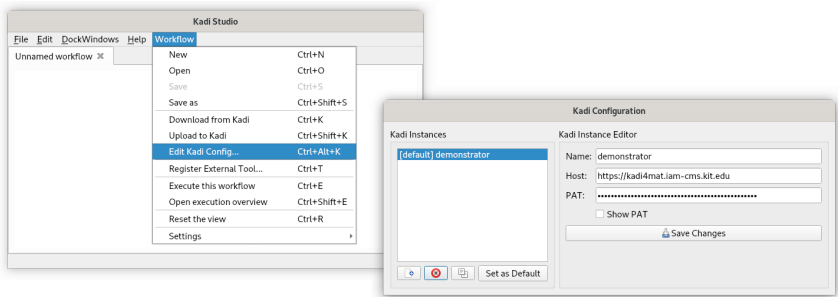
Die Bereitstellung dieser Schnittstelle ermöglicht nicht nur die Formulierung beliebiger Workflows, sondern auch die Anpassung bestehender Abläufe an verschiedene Formate. Sie stärkt zugleich die Reproduzierbarkeit, da Workflows durch Austausch oder Erweiterung von Werkzeugen flexibel auf neue Anwendungen übertragbar sind.

Das Hinzufügen neuer Knoten erfolgt über eine grafische Oberfläche, die alle im PATH verfügbaren Befehle auflistet und bei Bedarf um Suchpfade erweitert werden kann.

## Workflows und Repositorien

In KadiStudio erstellte Workflows können direkt in Kadi4Mat abgelegt, mit Metadaten versehen und so auffindbar gemacht werden. Da sich das FAIR-Prinzip nicht nur auf Workflows, sondern auf alle Forschungsdaten bezieht, unterstützt KadiStudio auch die FAIR-konforme Dokumentation der im Workflow erzeugten Daten.

Hierfür kann der Editor mit Repositorien verbunden werden. Als Referenz wurde diese Schnittstelle bereits für das Kadi4Mat-Repositorium über Werkzeugknoten der Bibliothek kadi-apy [78] umgesetzt. Diese greifen über die von Kadi4Mat bereitgestellte API auf das Repositorium zu und interagieren mit KadiWeb. Die Registrierung eines Repositoriums erfolgt über eine grafische Benutzeroberfläche, wie in Abbildung 3.11 gezeigt. Im Fall von Kadi4Mat müssen Host-Adresse sowie ein persönlicher Zugriffstoken (PAT) angegeben werden.



**Abbildung 3.11:** Dialog zur Registrierung von Kadi4Mat-Instanzen. Registrierte Instanzen können während eines Workflows genutzt werden, um die Funktionalitäten des Repositoriums zu verwenden und Daten auszutauschen.

Durch die Anbindung an Kadi4Mat können dort gespeicherte Workflows direkt in den Editor geladen und weiter bearbeitet werden. Gleichzeitig lassen sich im Repositorium abgelegte Daten importieren, verarbeiten und die Ergebnisse zurück nach Kadi4Mat hochladen und mit Metadaten versehen.

Auf diese Weise lassen sich Forschungsprozesse mitsamt den darin erzeugten Daten vollständig und nachvollziehbar modellieren. Die strukturierte Speicherung sowie die Abbildung von Abhängigkeiten ermöglichen eine lückenlose Verfolgung der Datenherkunft. Damit schafft KadiStudio die Grundlage für einen FAIRen Umgang mit Forschungsdaten. Für eine detaillierte Beschreibung der technischen Implementierung von KadiStudio wird verzichtet, da diese für die Zielsetzung der Arbeit nicht relevant ist. Für weiterführende Informationen sei auf Griem et al. [68] verwiesen.



## 3.3 Datendokumentation

Die bisher dargestellte Prozessmodellierung bildet die Grundlage für nachvollziehbare und wiederverwendbare Workflows. Gleichzeitig entstehen im Zuge dieser Prozesse vielfältige Forschungsdaten, deren langfristiger wissenschaftlicher Wert nur durch eine systematische und FAIRe Dokumentation gewährleistet werden kann. Im Folgenden wird ein Konzept entwickelt, das die FAIRe Datendokumentation realisiert. Besonderes Augenmerk liegt dabei auf der strukturierten Erfassung von Metadaten, der Nachvollziehbarkeit von Datenflüssen sowie der Möglichkeit, Daten mit der wissenschaftlichen Gemeinschaft zu teilen. Abschließend wird dargestellt, wie Kadi4Mat dieses Konzept umsetzt und somit die Anforderungen an die FAIRe Datenverwaltung unterstützt.

### 3.3.1 Konzept

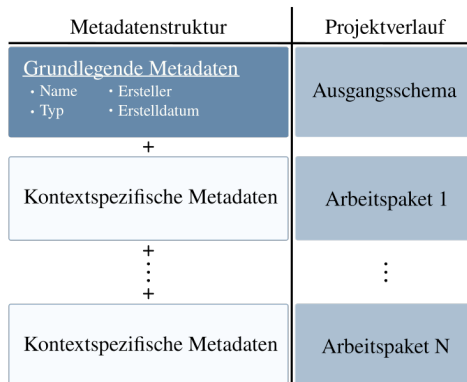
Ziel der FAIRen Datendokumentation ist es, wissenschaftliche Daten so zu organisieren und zu beschreiben, dass sie dauerhaft nachvollziehbar, strukturiert und maschinenlesbar abgelegt sind. Dies erfordert eine Datenstruktur, die sowohl die Beschreibung einzelner Datenpunkte als auch ihre Einbettung in den Gesamtzusammenhang wissenschaftlicher Prozesse ermöglicht. Um dies zu erreichen, soll eine möglichst vollständige Beschreibung aller im Forschungsprojekt entstandenen Daten mitsamt ihrer relevanten Metadaten erfolgen. Auf Ebene des gesamten Projekts ist es jedoch kaum möglich, ein *a-priori* festgelegtes, umfassendes Metadatenschema für sämtliche Daten zu definieren.

Aus diesem Grund wird die Datendokumentation analog zur in Unterabschnitt 3.2.1 vorgestellten Prozessmodellierung in unterschiedliche Abstraktionsebenen zerlegt. Die Metadatenschemata werden dabei nicht global, sondern auf Ebene einzelner Arbeitspakete definiert. Die Dokumentation der Metadaten erfolgt somit inkrementell auf Basis kleinerer, definierter Teilschemata, die jeweils einem Arbeitsschritt zugeordnet sind. Ein übergeordnetes Gesamtschema existiert zu Beginn nicht, sondern entsteht im Projektverlauf durch die Rekombination dieser Teilschemata.

Auf diese Weise wächst die Metadokumentation bottom-up zu einer konsistenten Gesamtstruktur zusammen, die dem in Unterabschnitt 3.2.1 beschriebenen Konzept zur Prozessmodellierung entspricht.

Grundlage bildet ein einfaches Ausgangsschema für bestimmte Datenobjekte, beispielsweise eine physische Materialprobe. Dieses Schema enthält lediglich elementare Informationen wie Erstellungsdatum, Format oder Ersteller. In jedem folgenden Arbeitsschritt werden neue, kontextspezifische Informationen ergänzt und an das bestehende Schema angehängt. Dadurch entsteht im Verlauf des Projekts ein inkrementell wachsender Metadatensatz [79]. Entscheidend ist, dass jeder Arbeitsschritt ein klar definiertes Schema verwendet. Auf diese Weise sind die erzeugten Metadaten konsistent und können in anderen Projekten wiederverwendet werden.

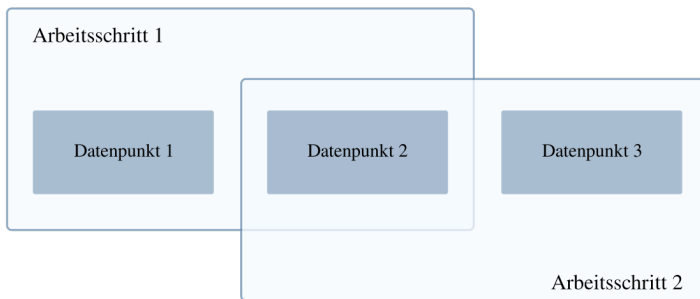
Abbildung 3.12 illustriert diese Idee wachsender Metadaten. Die iterative Erstellung reduziert den initialen Aufwand erheblich, da nicht für jedes Projekt ein umfassendes Schema entworfen werden muss. Stattdessen genügt die Entwicklung einzelner Arbeitsschritt-Schemata, die anschließend beliebig kombiniert und wiederverwendet werden können.



**Abbildung 3.12:** Konzept zur Strukturierung der Metadaten. Ausgehend von einem grundlegenden Metadatenschema werden mit jedem Arbeitspaket neue kontextspezifische Metadaten hinzugefügt.

Neben der Beschreibung einzelner Datenpunkte ist auch ihre Einbettung in den Gesamtzusammenhang entscheidend. Dafür sind zwei Konzepte zentral: (i) die hierarchische Gruppierung von Daten in logische Einheiten und (ii) die explizite Modellierung von Relationen zwischen Datenpunkten.

Erst durch diese Konzepte lässt sich die Datenherkunft um Projektverlauf nachvollziehbar darstellen. So können etwa alle Daten, die in einem Arbeitspaket entstehen, wie in Abbildung 3.13 zusammengefasst werden. Dadurch wird ersichtlich, wie die Daten logisch gruppiert sind, was die Nachvollziehbarkeit deutlich erhöht.

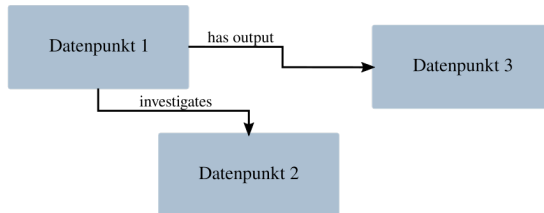


**Abbildung 3.13:** Hierarchische Strukturierung von Daten. Datenpunkte werden den Arbeitspaketen zugeordnet, so dass ihr Ursprung und ihre Abhängigkeiten nachvollziehbar sind. Ein Datenpunkt kann Teil mehrerer Arbeitsschritte sein.

Zusätzlich dokumentieren Relationen zwischen Datenpunkten, welche Ergebnisse aus welchen Daten hervorgegangen sind. Diese Relationen werden als gerichtete Graphen modelliert, in denen die Knoten Datenpunkten entsprechen und die Kanten deren Relation zueinander. Die Graphstruktur erlaubt es, Herkunft, Weiterverwendung und Transformation von Daten intuitiv nachzuvollziehen (Abbildung 3.14).

An dieser Stelle zeigt sich die enge Verbindung zwischen Daten- und Prozessmodellierung: Aus der zuvor eingeführten FAIRen Prozessmodellierung ist bekannt, welche Daten aus welchem Prozess entstehen. Dadurch lassen sich Relationen

zwischen erzeugten Datenpunkten direkt aus der Prozessbeschreibung ableiten, anstatt sie separat modellieren zu müssen.



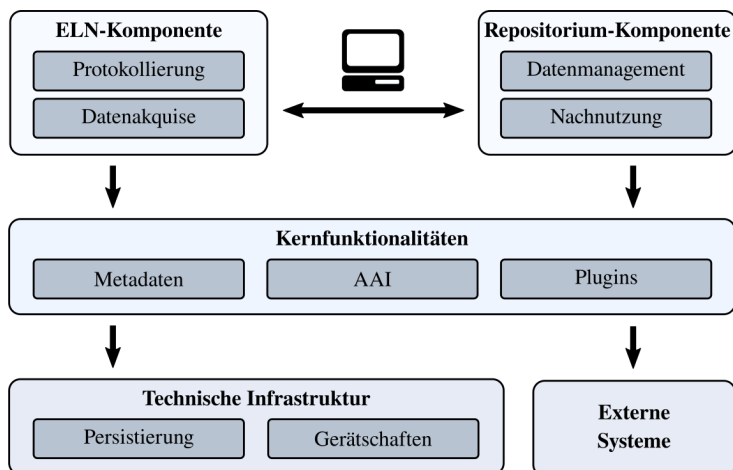
**Abbildung 3.14:** Konzept zur Darstellung von Datenrelationen

Zusammenfassend basiert das Konzept der FAIRen Datendokumentation auf drei Elementen: (i) elementare, rekombinierbare Metadatenschemata auf Ebene einzelner Arbeitsschritte, (ii) hierarchische Gruppierung von Datenobjekten auf unterschiedlichen Abstraktionsebenen und (iii) explizite Relationen zwischen Datenobjekten zur Abbildung von Datenflüssen. Die Kombination dieser Konzepte schafft eine flexible, modulare und wiederverwendbare Struktur, die eine FAIRe Dokumentation wissenschaftlicher Daten ermöglicht. Im Folgenden wird die technische Umsetzung in KadiWeb vorgestellt.

### 3.3.2 KadiWeb

KadiWeb bildet das zentrale Element von Kadi4Mat. Es handelt sich um eine quelloffene, webbasierte Forschungsumgebung, welche die Funktionen eines Community-Repositories mit denen eines elektronischen Laborbuchs (ELN) vereint. Das Repository stellt eine Infrastruktur zur gemeinsamen Nutzung, Versionierung und Verwaltung sogenannter „warmer“ Daten bereit. Als „warmer“ Daten werden Daten bezeichnet, die sich noch in Bearbeitung befinden und nicht archiviert sind. Im elektronischen Laborbuch können Experimente und Ergebnisse strukturiert dokumentiert, visualisiert, transformiert und analysiert werden.

Abbildung 3.15 veranschaulicht den konzeptionellen Aufbau von KadiWeb. Dabei wird deutlich, dass ELN und Repositorium nicht als getrennte Systeme, sondern als eng verzahnte Teile einer gemeinsamen Infrastruktur fungieren.

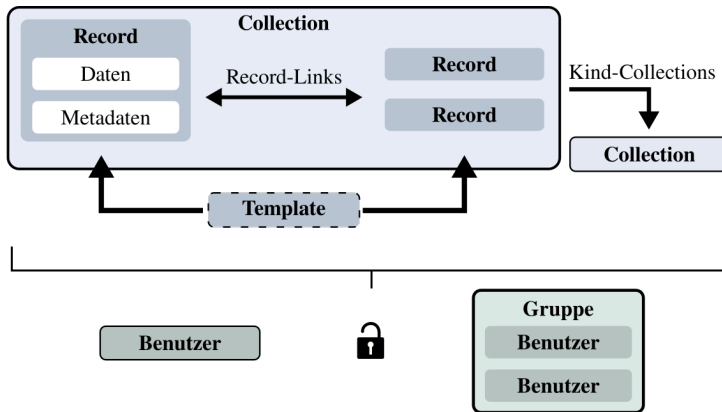


**Abbildung 3.15:** Konzeptioneller Aufbau von KadiWeb. KadiWeb stellt sowohl ein Repositorium zur strukturierten Datenablage, als auch ELN-typische Funktionalitäten zur Verfügung. Adaptiert von [80].

KadiWeb stellt den Nutzenden verschiedene Ressourcen zur Verfügung, mit denen Forschungsaktivitäten abgebildet und organisiert werden können. Bei diesen Ressourcen handelt es sich um Records, Collections, Templates und Gruppen.

Diese Ressourcen setzen die im vorangegangenen Abschnitt identifizierten Prinzipien der FAIRen Datendokumentation technisch um. Die kleinste logische Einheit bilden Records, die zur Ablage von Forschungsdaten dienen.

Ein Record kann als eine Art Schachtel verstanden werden, in die beliebige Dateien unabhängig von ihrem Format gelegt werden. Auf diese Schachtel werden anschließend Informationen, also Metadaten, geschrieben, die den Inhalt beschreiben. Diese Trennung von Daten und Metadaten ermöglicht es, Inhalte gezielt



**Abbildung 3.16:** Ressourcen KadiWeb. Daten werden in Records abgelegt und mit Metadaten versehen. Mehrere Records können in eine Collection organisiert werden. Templates helfen Metadaten zu vereinheitlichen. Über Benutzer und Gruppen können gezielt Zugangsrechte vergeben werden. Adaptiert von [80].

zu durchsuchen und zu analysieren. Die Metadaten eines Records setzen sich aus automatisch generierten Basis-Metadaten (Titel, Erstellungsdatum, Autor, Sichtbarkeit) und benutzerdefinierten Extra-Metadaten zusammen. Letztere folgen einem Key-Value-Schema mit typisierten Werten (String, Integer, Date, Boolean, Liste, Dictionary) und können durch Validierungsregeln weiter definiert werden.

Für die konsistente Erfassung dieser Metadaten kommen Templates zum Einsatz. Dabei unterscheidet KadiWeb zwei Typen: *Record-Templates* und *Extra-Templates*. Die Record-Templates ermöglichen die Erstellung neuer Records und die Definition ihrer grundlegenden Metadaten. Sie können entsprechend als Ausgangsschema für eine wachsende Metadatenbeschreibung verwendet werden. Extra-Templates hingegen erstellen keine neuen Records sondern ermöglichen das Hinzufügen neuer strukturierter Metadaten zu einem existierenden Record. Sie sind somit die technische Entsprechung der im Konzept beschriebenen Arbeitsschritt-Schemata.

Auf diese Weise entsteht eine inkrementelle Metadatenstruktur. Jeder Arbeitsschritt ergänzt bestehende Records um neue Informationen, sodass die Dokumentation im Projektverlauf Bottom-up wächst [79].

Neben der Beschreibung einzelner Datenpunkte erlaubt KadiWeb auch ihre hierarchische Organisation. Über *Collections* können mehrere Records gruppiert und verschachtelt werden, sodass ganze Arbeitspakete oder Projekte strukturiert abgebildet werden. Templates lassen sich auch auf Collections anwenden, wodurch übergeordnete Metadatenkonsistenz gewährleistet wird.

Darüber hinaus lassen sich Datenpunkte semantisch miteinander verknüpfen. KadiWeb bietet hierfür *Links*, die Relationen zwischen Records ausdrücken, z. B. „wurde erzeugt durch“ oder „dient als Eingabe für“. Diese Relationen bilden im Sinne des IPO-Modells gerichtete Graphen, in denen die Herkunft und Weiterverarbeitung von Daten nachvollziehbar dokumentiert wird.

Das Rechtemanagement in KadiWeb ist feingranular ausgelegt. Auf Ebene einzelner Records und Collections können Lese-, Schreib-, Verlinkungs- und Administrationsrechte vergeben werden. Zur Vereinfachung der Rechteverwaltung können Nutzer über Gruppen organisiert werden. Über diese können den Mitgliedern zentral Zugriffsrechte auf definierte Ressourcen zugewiesen werden. Auf diese Weise lassen sich institutionelle Strukturen wie Institute oder Arbeitsgruppen effektiv modellieren und es wird sichergestellt, dass alle berechtigten Personen Zugriff auf die jeweils relevanten Forschungsdaten haben.

Für die Veröffentlichung von Daten bietet KadiWeb sogenannte *Connected Services*. Über das Zenodo-Plugin lassen sich ausgewählte Records mitsamt Metadaten und zugehörigen Dateien automatisiert an Zenodo übertragen. Dort erhalten sie einen *Digital Object Identifier* (DOI) und werden dauerhaft öffentlich referenzierbar.

Außer den Repository-Funktionen beinhaltet KadiWeb auch Funktionen eines elektronischen Laborbuchs. Neben der Prozessdokumentation in Form automatisierbarer Workflows (siehe Unterabschnitt 3.2.2), stehen weitere Visualisierungs- und Dokumentationsfunktionen bereit. Ein Skizzenmodul erlaubt das Anfertigen von grafischen Annotationen, während eine integrierte Vorschaufunktion die Darstellung gängiger Dateiformate (z.B. PDF, Bilder, STL) ohne zusätzliche Software ermöglicht. Freitextfelder ermöglichen zudem die Einbindung von Beobachtungen

oder Interpretationen. Auf diese Weise können Notizen und Dokumentationen der Forschenden auf KadiWeb abgebildet und dokumentiert werden.

Zur Integration in bestehende Forschungsprozesse stellt KadiWeb eine REST-API bereit, die durch eine OpenAPI-Spezifikation dokumentiert ist. Die Python-Library *kadi-apy* bietet auf dieser Basis eine komfortable Schnittstelle, sodass Datenverwaltung direkt aus Jupyter-Notebooks, CI/CD-Pipelines oder Automatisierungsskripten heraus erfolgen kann.

KadiWeb setzt damit die in Unterabschnitt 3.3.1 beschriebenen Konzepte praktisch um: (i) elementare, wiederverwendbare Metadatenschemata über Templates, (ii) hierarchische Strukturierung von Daten über Collections, (iii) explizite Relationen über Links.

Durch diese modulare Architektur und die enge Integration von Repositorium, ELN und Automatisierungsfunktionen stellt KadiWeb eine leistungsfähige Infrastruktur für FAIRe Datendokumentation bereit, die flexibel an unterschiedliche wissenschaftliche Disziplinen angepasst werden kann.

## 3.4 Problemspezifisches Konzept

In den vorangegangenen Abschnitten wurden generische Konzepte zur FAIRen Prozessmodellierung und Datendokumentation entwickelt und ihre technische Umsetzung in Kadi4Mat vorgestellt. Diese Konzepte sind bewusst abstrakt formuliert, um eine breite Übertragbarkeit auf unterschiedliche wissenschaftliche Fragestellungen zu ermöglichen. Ein Bezug zu einem konkreten Forschungsprojekt wurde bislang jedoch nicht hergestellt.

Da die Konzepte in dieser Arbeit als methodisches Fundament für die Entwicklung neuer Ansätze dienen, sollen sie im Folgenden auf die zugrunde liegende materialwissenschaftliche Forschungsfrage angewendet werden. Diese befasst sich mit der Identifikation von Struktur-Eigenschaft-Beziehungen in Polyurethanschäumen, um deren Eigenschaften für Leichtbauanwendungen gezielt zu optimieren. Ziel



ist es, eine Brücke zwischen dem mikrostrukturellen Aufbau der Schäume und ihren makroskopischen mechanischen Eigenschaften zu schlagen.

Die Strukturierung des Forschungsprojekts erfolgt nach dem in Unterabschnitt 3.2.1 eingeführten Abstraktionsmodell. Ausgangspunkt ist das Kernziel der Forschungsfrage: Die Identifikation von Struktur-Eigenschaft-Beziehungen. Dafür müssen sowohl die Mikrostruktur als auch die mechanischen Eigenschaften der Schäume bestimmt werden. Entsprechend ergeben sich zunächst die beiden Arbeitspakete zur Charakterisierung der Mikrostruktur und zur Simulation der mechanischen Eigenschaften. Beide erfordern digitale Modelle der untersuchten Schäume, sodass als vorgelagertes Arbeitspaket die Rekonstruktion digitaler Zwillinge notwendig wird. Diese Rekonstruktionen basieren auf experimentellen CT-Daten realer Schäume. Rekonstruktion und Charakterisierung werden an dieser Stelle in einem Arbeitspaket zusammengefasst.

Zur Identifikation von robusten Struktur-Eigenschaft-Beziehungen ist eine möglichst umfangreiche Datenbasis notwendig. Die Erstellung einer solchen ist durch eine reine Rekonstruktion zu zeit- und kostenintensiv. Aus dieser Überlegung heraus folgt ein weiteres Arbeitspaket zur Generierung digitaler Schaumstrukturen, das auf den Arbeitspaketen Rekonstruktion und Charakterisierung aufbaut und logisch daran anschließt.

Schließlich werden die Ergebnisse aller vorangegangenen Schritte im übergeordneten Arbeitspaket zur Identifikation von Struktur-Eigenschaft-Beziehungen zusammengeführt. Der gesamte Forschungsprozess lässt sich damit in fünf miteinander verknüpfte Arbeitspakete gliedern: Rekonstruktion und Charakterisierung digitaler Schaumzwillinge, Generierung digitaler Schaumstrukturen, Simulation der mechanischen Eigenschaften und Identifikation von Struktur-Eigenschaft-Beziehungen. Abbildung 3.17 veranschaulicht diese Arbeitspakete in Form eines zyklischen Materialentwicklungsprozesses. In den folgenden Kapiteln werden sie detailliert analysiert, methodisch präzisiert und in Form generischer Workflows in Kadi4Mat dokumentiert.



Die Relationen zwischen den erzeugten Datenpunkten sowie ihre Verknüpfung mit den entwickelten Workflows werden über Recordlinks beschrieben, die an das Metadatenschema Metadata4Ing [81] angelehnt sind. Damit lassen sich Abhängigkeiten zwischen den Arbeitsschritten formal modellieren und transparent nachvollziehen.

Die hier identifizierten Arbeitspakete bilden den roten Faden der gesamten Arbeit. In den nachfolgenden Kapiteln werden sie nacheinander aufgegriffen, methodisch präzisiert und in Form generischer Workflows in Kadi4Mat dokumentiert. Gleichzeitig wächst die Metadatenbeschreibung Schritt für Schritt mit, sodass Prozesse und Daten konsistent miteinander verknüpft sind. Auf diese Weise werden die entwickelten Konzepte und Methoden nicht nur theoretisch beschrieben, sondern praktisch auf ein materialwissenschaftliches Anwendungsbeispiel übertragen und leisten damit einen Beitrag zur Weiterentwicklung digitaler Materialdesignprozesse.

## 4     **Rekonstruktion und Charakterisierung digitaler Zwillinge**

Die digitale Materialentwicklung fußt auf der Verfügbarkeit digitaler Zwillinge realer Materialsysteme. Diese bilden die relevanten physikalischen Eigenschaften ab und dienen als Grundlage für numerische Simulationen, datengetriebene Modellierungen und die Ableitung von Struktur-Eigenschaft-Beziehungen.

Ziel dieses Kapitels ist es, Methoden für die effiziente Rekonstruktion und Charakterisierung digitaler Zwillinge zu entwickeln und in reproduzierbaren Workflows zu automatisieren. Die Prozessschritte und Forschungsdaten werden dabei gemäß dem in Abschnitt 3.1 vorgestellten Konzept dokumentiert und in Kadi4Mat integriert. Maschinelles Lernen beschleunigt insbesondere die Segmentierung hochauflösender CT-Aufnahmen, während ein durchgängiges Forschungsdatenmanagement Wiederverwendbarkeit und Nachvollziehbarkeit sicherstellt.

Der Fokus liegt zunächst auf der Segmentierung dreidimensionaler CT-Aufnahmen der Mikrostruktur mittels maschineller Lernverfahren, unterstützt durch synthetische Trainingsdaten und validiert mit experimentell ermittelten Materialkennwerten. Anschließend erfolgt die geometrische Charakterisierung der rekonstruierten Mikrostrukturen anhand von Kennwerten wie Porosität, Porengrößenverteilung und Wandstärke. Die abschließend entwickelten Workflows bilden eine vollständig digitale Prozesskette von den Rohdaten bis zur strukturellen Analyse, beschleunigen die Materialentwicklung und schaffen die methodische Basis für die in den folgenden Kapiteln vorgestellten Generierungs- und Simulationsansätze.

## 4.1 Stand der Forschung

Die in diesem Kapitel vorgestellten Methoden basieren auf einer Vielzahl etablierter Verfahren aus der Bildgebung, der Bildverarbeitung und der geometrischen Charakterisierung. In diesem Abschnitt werden die relevanten Ansätze aus der Literatur zusammengefasst, die als Grundlage für die in dieser Arbeit entwickelten Verfahren dienen. Der Fokus liegt dabei auf bildgebenden Verfahren, Segmentierungsmethoden sowie Verfahren zur geometrischen Charakterisierung poröser Mikrostrukturen.

### 4.1.1 Rekonstruktion

Die Rekonstruktion digitaler Materialzwillinge poröser Strukturen basiert in der Regel auf zerstörungsfreien bildgebenden Verfahren, die eine hochauflösende dreidimensionale Erfassung des inneren Aufbaus ermöglichen. Für die Materialcharakterisierung stehen sowohl zweidimensionale als auch dreidimensionale Methoden zur Verfügung [82, 83].

Die *Rasterelektronenmikroskopie* (REM), *Transmissionselektronenmikroskopie* (TEM), *Focused Ion Beam* (FIB) oder die *Rasterkraftmikroskopie* (AFM) erreichen sehr hohe Auflösungen, sind jedoch auf 2D-Aufnahmen beschränkt. FIB kann zwar sehr kleine Volumina, wenige Kubikmikrometer, dreidimensional abbilden, ist dabei aber nicht zerstörungsfrei [82, 83]. Optische Verfahren wie die *konfokale Laserscanning-Mikroskopie* (CLSM) ermöglichen hingegen eine schnelle, kostengünstige Erfassung topologischer und volumetrischer Strukturen, sind jedoch in Auflösung und Probertiefe limitiert und erfordern offenzellige oder transparente Proben [84, 85].

Unter den verfügbaren Techniken hat sich die *Röntgen-Computertomographie* (CT) als weit verbreitete Methode in der Materialwissenschaft etabliert, da sie eine günstige Kombination aus Auflösung, Messvolumen und Verfügbarkeit bietet [83, 85–87]. Das Verfahren basiert auf einer Serie zweidimensionaler Projektionen

aus verschiedenen Rotationswinkeln, die die Wechselwirkung eines Röntgenstrahls mit einem Probenmaterial erfasst. Diese Wechselwirkung setzt sich aus Absorption und Phasenverschiebung zusammen und wird in den Projektionen als Grauwertinformation abgebildet. Durch mathematische Rekonstruktionsverfahren entsteht daraus ein dreidimensionaler Volumendatensatz, in dem jeder Voxel einen materialabhängigen Intensitätswert enthält [85, 88].

Die erreichbare Auflösung hängt wesentlich von Strahlungsquelle, Detektor, Messgeometrie und Probenpräparation ab. Industrielle Mikro-CT-Systeme bieten typischerweise Voxelgrößen von  $1\text{ }\mu\text{m/Voxel}$  bis  $100\text{ }\mu\text{m/Voxel}$ , während Synchrotron-CTs durch hohe Photonenflussdichte und parallele Strahlenführung Auflösungen bis unter  $50\text{ nm/Voxel}$  erreichen. Die Wahl der Messparameter stellt dabei stets einen Kompromiss zwischen Auflösung, Messvolumen und Scanzeit dar [85]. In der materialwissenschaftlichen Forschung kommen daher sowohl Mikro-CT-Systeme als auch Synchrotron-CTs zum Einsatz [83, 85, 86, 89].

Eine besondere Herausforderung stellen bei der Bildgebung durch CT Bildartefakte dar, die durch Strahlungscharakteristik, Detektorrauschen oder Rekonstruktionsfehler entstehen [90]. Diese äußern sich in lokalen Grauwertschwankungen und können die Genauigkeit nachfolgender Segmentierungen erheblich beeinflussen. Eine sorgfältige Bildvorverarbeitung sowie die Auswahl geeigneter Segmentierungsstrategien sind daher essenziell.

Die CT bildet damit die Grundlage für die dreidimensionale geometrische und topologische Analysen von Mikrostrukturen. Ihre Leistungsfähigkeit bestimmt maßgeblich die Qualität der rekonstruierten digitalen Zwillinge und damit die Aussagekraft aller darauf basierenden Analysen, Simulationen und datengetriebenen Modelle. Gleichzeitig macht sie eine präzise Segmentierung und anschließende quantitative Charakterisierung überhaupt erst möglich, die im nächsten Abschnitt näher betrachtet werden.

### 4.1.2 Segmentierung

Um die in CT-Aufnahmen enthaltenen Informationen nutzbar zu machen, ist eine Segmentierung der Grauwertbilder erforderlich. Ziel dieser Segmentierung ist es, das kontinuierliche Grauwertbild in diskrete Bereiche zu unterteilen, die die unterschiedlichen Materialphasen definieren. In der Regel werden dabei die beiden relevanten Phasen (Struktur- und Porenraum) voneinander getrennt, indem jedem Voxel eine Phase zugewiesen wird. Das Ergebnis ist eine binäre Repräsentation der Mikrostruktur, die als Grundlage für geometrische Charakterisierungen sowie für Simulationen dienen kann.

Zu den etablierten Ansätzen zur Segmentierung zählen globale Schwellwertverfahren, bei denen die Grauwertverteilung eines Bildes analysiert wird, um einen optimalen Trennwert zwischen Vorder- und Hintergrund zu bestimmen [91–94]. Diese Algorithmen zielen unter Anwendung verschiedener Metriken darauf ab, einen Schwellenwert in der Histogrammdarstellung des Bildes zu finden, um zu entscheiden, ob ein Pixel zum Vorder- oder Hintergrund gehört.

Eine weit verbreitete Methode dieser Art ist der *Otsu*-Algorithmus [95], der den globalen Schwellwert so bestimmt, dass die Varianz zwischen den identifizierten Klassen maximiert wird. Wie alle globalen Schwellwertverfahren ist auch die Otsu-Methode anfällig für Fehlsegmentierungen, insbesondere in verrauschten Bildern oder bei geringen Kontrasten zwischen den Phasen [96].

Neben der automatischen Bestimmung kann der Schwellwert auch manuell festgelegt werden. Hierbei erfolgt eine iterative Anpassung basierend auf experimentell bekannten Strukturparametern wie Porosität oder spezifischer Oberfläche [97]. Dieser Ansatz kann eine hohe Genauigkeit erreichen, setzt jedoch das Vorliegen entsprechender Referenzwerte voraus und ist nicht automatisierbar.

Um die Einschränkungen globaler Schwellwertmethoden zu überwinden, wurden erweiterte Segmentierungsverfahren entwickelt. Dazu zählen unter anderem *Variable Thresholding* [98–101], *Region-Growing* [102, 103] und *Level-Set*-Methoden [104, 105]. Diese Methoden berücksichtigen lokale Bildinformationen

und sind daher robuster gegenüber Bildrauschen und inhomogenen Grauwertverteilungen.

In letzter Zeit hat sich zudem der Einsatz maschineller Lernverfahren zur Segmentierung etabliert. Umfassende Übersichten zu ML-basierten Segmentierungsmethoden finden sich bei [106, 107]. Eine der am weitesten verbreiteten Architekturen ist *U-Net* [108]. U-Net wurde ursprünglich für die biomedizinische Bildsegmentierung entwickelt, hat sich jedoch auch in zahlreichen anderen Disziplinen als zuverlässige Methode etabliert, darunter in der Photogrammetrie [109, 110], der Satellitenbildanalyse [111] und der Vorhersage physikalischer Felder [112–116].

Der Aufbau des U-Nets ermöglicht sowohl kontextuelle Informationen großflächig zu erfassen als auch feine Strukturen präzise zu lokalisieren. Diese Eigenschaften machen das U-Net besonders geeignet für die Segmentierung komplexer und verrauschter CT-Daten [117, 118].

Zwar existieren erste unüberwachte U-Net Implementierungen zur Segmentierung von Bildern [119, 120], allerdings wird das U-Net meist überwacht trainiert. Für das Training eines U-Nets zur Segmentierung sind demnach annotierte Datensätze erforderlich, die aus Grauwertbildern und den zugehörigen, korrekt segmentierten Strukturen bestehen. Die manuelle Erstellung solcher Referenzdaten (*Ground-Truth-Daten*) ist jedoch zeitaufwendig, fehleranfällig und oft inkonsistent [96]. Um dies zu umgehen, kommen zunehmend synthetische Trainingsdaten zum Einsatz. Diese sind vor allem im medizinischen Bereich verbreitet [121, 122] und ermöglichen eine flexible und skalierbare Bereitstellung von Trainingsdaten, die gezielt an die Eigenschaften der Zielstruktur angepasst werden können. Die so erzeugten binären Materialrepräsentationen bilden die Grundlage für die geometrische Charakterisierung, wie sie im Folgenden beschrieben wird.



### 4.1.3 Geometrische Charakterisierung

Die segmentierten Bilddaten stellen die Grundlage für die geometrische Charakterisierung dar. Sie erlauben nicht nur die Untersuchung geometrischer und morphologischer Eigenschaften, sondern können auch zur Bestimmung der chemischen Zusammensetzung oder als Basis für simulationsgestützte Studien herangezogen werden. Da in diesem Kapitel die Geometrie der Strukturen im Vordergrund steht, konzentriert sich die folgende Darstellung auf den aktuellen Stand der Forschung zur Erfassung geometrischer und morphologischer Eigenschaften.

Grundsätzlich kann die Charakterisierung poröser Strukturen experimentell oder digital erfolgen. Für geschlossenzellige Strukturen sind etablierte experimentelle Methoden wie *Quecksilberporosimetrie* [123], *Kapillarflussporometrie* [124] oder *Gasadsorptionsmessungen* [125] nicht geeignet. Eine experimentell anwendbare Methode für solche Strukturen ist die *Pyknometrie*, mit der die Dichte der Strukturphase des Schaums bestimmt werden kann [126]. Aus dem Verhältnis von effektiver Schaumdichte zu Feststoffdichte lässt sich die Gesamtporosität berechnen. Für eine detaillierte Beschreibung der Zellgeometrie liefern experimentelle Verfahren für geschlossenzellige Strukturen jedoch nur eingeschränkte Informationen, sodass hierfür überwiegend digitale Ansätze eingesetzt werden [82, 127].

Die digitale Charakterisierung basiert in der Regel auf binarisierten Bilddaten der Mikrostruktur. Neben der direkten Bestimmung der Porosität ermöglichen diese Daten die Ableitung komplexerer morphologischer Kenngrößen. Eine etablierte Methode ist die Berechnung *euklidischer Distanzfelder* im Struktur- oder Porenraum [128, 129]. In Kombination mit *Skelettierungsverfahren*, bei denen Struktur- oder Porenraum auf eine *mediale Achse* reduziert werden, lassen sich präzise Aussagen zur mittleren Wandstärke ableiten [130, 131]. Die Verknüpfung von medialer Achse und Distanzfeld erlaubt zudem eine räumliche Zuordnung lokaler Abstände und damit eine realitätsnahe Erfassung der Porengrößenverteilung [132, 133].

Ergänzend zu diesen geometrischen Methoden kommen zunehmend statistische Verfahren zum Einsatz, um der hohen mikrostrukturellen Komplexität gerecht zu werden. Eine hierfür oftmals genutzte Methode ist die *Zweipunkt-Korrelationsfunktion*, mit der räumliche Zusammenhänge innerhalb der Mikrostruktur quantifiziert werden. Die resultierenden statistischen Beschreibungen erzeugen häufig hochdimensionale Merkmalsräume, die mittels *Hauptkomponentenanalyse* (PCA) [134] auf wesentliche Strukturmerkmale reduziert werden [135]. Ein wesentlicher Vorteil dieser Methodik liegt in der Möglichkeit der unüberwachten Klassifikation, die eine systematische Einordnung unterschiedlicher Schaumstrukturen erlaubt, ohne dass detaillierte Vorkenntnisse zu deren Eigenschaften erforderlich sind [136].

## 4.2 Methoden

Im Folgenden werden die Methoden zur Rekonstruktion und geometrischen Charakterisierung digitaler Materialzwillinge vorgestellt. Der Fokus liegt auf etablierten Verfahren, die für das untersuchte Materialsystem angepasst wurden. Das Kapitel umfasst die Rekonstruktion aus CT-Daten samt Segmentierung, die Bestimmung geometrischer Kennwerte sowie die datengetriebene Charakterisierung der Mikrostruktur. Diese Methoden bilden die Grundlage für die im weiteren Verlauf entwickelten Workflows.

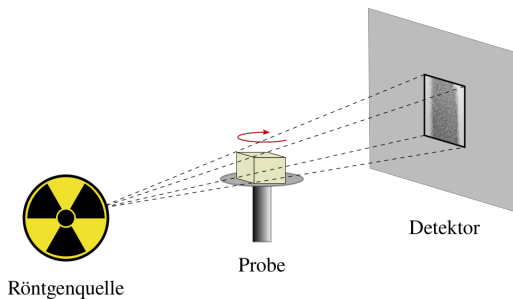
### 4.2.1 Rekonstruktion

Die Rekonstruktion digitaler Zwillinge beginnt in der Regel mit der Aufnahme hochauflösender Bilddaten. In dieser Arbeit wird hierfür die Computertomographie verwendet, da sie eine vollständige dreidimensionale Erfassung der inneren Schaumstruktur ermöglicht und somit eine geeignete Grundlage für die anschließende Segmentierung und Charakterisierung darstellt.

#### 4.2.1.1 Computertomographie

Die dreidimensionale Mikrostruktur der untersuchten Schaumproben wurde mit industrieller Cone-Beam-Computertomographie erfasst. Dieses zerstörungsfreie Verfahren liefert eine hochauflösende volumetrische Abbildung der inneren Struktur und bietet im vorliegenden Anwendungsfall eine ausreichende Auflösung bei deutlich geringerem logistischem Aufwand als Synchrotron-CT-Messungen.

Die Messungen selbst wurden mit einem *COMET YXLON FF20 CT* durchgeführt. Das Gerät nutzt ein Cone-Beam-Setup, bei dem die Probe zwischen einer stationären Röntgenquelle und einem Flachdetektor rotiert. Abbildung 4.1 zeigt den schematischen Aufbau. Während der Rotation wird eine Serie von Projektionsbildern aus verschiedenen Winkeln aufgenommen, die die Wechselwirkung des Röntgenstrahls mit dem Material visualisieren [87]. Diese Projektionen bilden die Grundlage für die anschließende 3D-Rekonstruktion. Für die hier verwendeten Aufnahmen wurde eine Auflösung von  $2,75\text{ }\mu\text{m/Voxel}$  erzielt.



**Abbildung 4.1:** Funktionsprinzip des verwendeten Computertomographen. Eine stationäre Röntgenquelle sendet kegelförmige Röntgenstrahlen aus, die die rotierende Probe durchdringen und von einem dahinterliegenden Detektor erfasst werden.

Die Rekonstruktion der Volumendaten aus den Projektionen erfolgte unter Anwendung einer automatischen Geometrieoptimierung (AutoAlignment) und einer Bad-Pixel-Korrektur. Weitere Korrekturen wie Beam-Hardening-, Ringartefakt- oder Rauschunterdrückungs-Filter wurden nicht angewendet, um die Rohdaten möglichst unverfälscht zu erhalten. Dies erhöht die Übertragbarkeit der in diesem

Kapitel entwickelten Segmentierungsmethoden auf andere Anwendungsfälle, da sie nicht auf optimalen Daten trainiert wurden. In den rekonstruierten Volumendaten sind daher typische CT-Artefakte erkennbar, die auf Röntgenstrahlcharakteristik, Detektorrauschen oder Rekonstruktionsalgorithmen zurückzuführen sind [90]. Diese äußern sich vor allem in lokalen Grauwertschwankungen (Bildrauschen) und können die Segmentierungsgenauigkeit beeinflussen.

## 4.2.2 Segmentierung der Bilddaten

Die aus bildgebenden Verfahren gewonnenen Bilder liegen in der Regel als kontinuierliche Grauwertbilder vor. Um daraus einen digitalen Zwilling der erfassten Struktur zu erzeugen, müssen die Bilder binarisiert werden. Hierbei werden die Bildpixel in Vordergrund- und Hintergrundpixel unterteilt, sodass jeweils ein binäres Bild entsteht, das lediglich die Pixelwerte 1 (Vordergrund) und 0 (Hintergrund) enthält. Zur Zuweisung der Bildpixel in Vorder- und Hintergrund stehen unterschiedliche Segmentierungsalgorithmen zur Verfügung. Die für diese Arbeit relevanten Algorithmen und Methoden werden nachfolgend dargelegt. Dabei handelt es sich um die globale Schwellwertmethode Otsu und die U-Net Architektur.

### Otsu

Eine gängige Methode zur Segmentierung von Bildern besteht darin, Histogrammetriken zu verwenden, um einen globalen Schwellwert zu bestimmen, der die Bildpixel in Vorder- und Hintergrund unterteilt [91–94]. Grundlage dafür ist das Vorliegen eines Graustufenhistogramms des zu segmentierenden Bildes, das die Verteilung der Pixelintensitäten im betrachteten Bild zusammenfasst und über

$$H(i) = \sum_{x=1}^M \sum_{y=1}^N [I(x,y) = i] \quad (4.1)$$

definiert ist. Hierbei entspricht  $M$  der Breite und  $N$  der Höhe des Bilds in Pixeln wohingegen  $[I(x,y) = i]$  eine Funktion darstellt, die den Wert 1 annimmt, wenn

der Pixelwert an der Stelle  $(x,y)$  den Wert  $i$  hat und ansonsten 0 ist. Basierend auf einem solchen Histogramm wird der Otsu-Algorithmus [95] zur Segmentierung verwendet. Dieser Algorithmus teilt ein Bild anhand der Graustufenintensitätswerte der Pixel in die Klassen Vorder- und Hintergrund auf. Das Ziel des Algorithmus besteht darin, einen globalen Schwellwert zu finden, der die Pixel optimal in diese beiden Klassen trennt. Dies kann entweder durch Maximierung der interklassischen Varianz oder Minimierung der intraklassischen Intensitätsvarianz erreicht werden. Zunächst errechnet der Algorithmus die Wahrscheinlichkeiten beider Klassen  $\omega_0$  und  $\omega_1$  für jeden Schwellwert  $t$  über das Graustufenhistogramm mit Hilfe der Formeln [95]

$$\omega_0(t) = \sum_{i=0}^{t-1} p_i \quad \text{und} \quad \omega_1(t) = \sum_{i=t}^{L-1} p_i. \quad (4.2)$$

Dabei ist  $L$  die Anzahl möglicher Grauwertstufen, 256 für 8-bit Bilder. In den Formeln gilt  $p_i = n_i/N$ . Das entspricht der Wahrscheinlichkeit, dass ein Pixel einen bestimmten Grauwert  $i$  aufweist. Anschließend wird die intraklassische Varianz als gewichtete Summe der Varianzen der beiden Klassen ermittelt. Der Schwellwert, der diese intraklassische Varianz minimiert, wird iterativ berechnet und teilt die Pixel im Bild in Vorder- und Hintergrund auf.

$$\sigma_W^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \quad (4.3)$$

Auf diese Weise kann schnell ein globaler Schwellwert zur Segmentierung eines Grauwertbildes ermittelt werden. Die Otsu-Methode nimmt jedoch normalverteilte und bimodale Grauwertintensitäten für Vorder- und Hintergrundpixel an. Dadurch ist sie anfällig für eine ungenaue Segmentierung beim Vorliegen von Bildrauschen. Die Anwendung dieses Segmentierungsalgorithmus wird in dieser Arbeit mit dem Open Source Bildbearbeitungsprogramm *Fiji* [137] durchgeführt, in dem die Otsu-Segmentierung integriert ist.

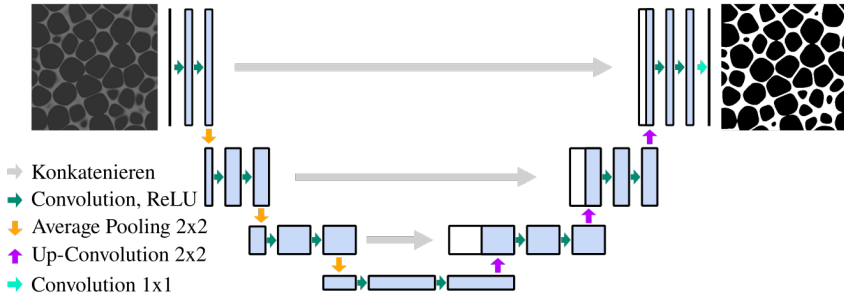
## U-Net

Neben der Verwendung globaler Schwellwerte stehen maschinelle Lernverfahren zur Verfügung, zu denen [106, 107] umfassende Übersichten geben. Angesichts der Schwächen klassischer Schwellwertmethoden, insbesondere bei verrauschten CT-Daten, wird in dieser Arbeit ein U-Net, ein speziell für die Bildsegmentierung entwickeltes Convolutional Neural Network (CNN), eingesetzt.

Das U-Net basiert auf einer sogenannten Encoder–Decoder-Architektur, bei der der Encoder schrittweise die räumliche Auflösung der Eingabebilder reduziert und gleichzeitig den Merkmalsraum erweitert, um kontextuelle Informationen auf unterschiedlichen Skalen zu erfassen. Der Decoder rekonstruiert daraus schrittweise die ursprüngliche Bildauflösung. Sogenannte *Skip-Connections* innerhalb der Encoder- und Decoder-Stufen sorgen dafür, dass hochaufgelöste Merkmalskarten (Feature Maps) aus dem Encoder direkt in den Rekonstruktionsprozess des Decoders einfließen, was insbesondere für die präzise Lokalisierung feiner Strukturen hilfreich ist. Die Skip-Connections können dazu entweder die Feature Maps aus dem Encoder an die Ausgabe des Decoders konkatenieren oder hinzuaddieren. Abbildung 4.2 stellt die U-Net Architektur schematisch dar.

Das Netzwerk nutzt lokale Bildfilter in Form von Convolutional Layers, die während des Trainings optimiert werden. Dadurch wird das U-Net automatisch an die geometrischen Eigenschaften und das Bildrauschen des betrachteten Anwendungsfalls angepasst. Die Architektur kann hinsichtlich Tiefe, Anzahl der Filter, Kernelgrößen sowie Aktivierungsfunktionen und so an unterschiedliche Datensätze und Anwendungsfälle angepasst werden.

Für die binäre Segmentierung wird in der Ausgabeschicht üblicherweise eine Sigmoid-Aktivierungsfunktion eingesetzt. Als Verlustfunktion kommen im U-Net häufig pixelweise Metriken wie die binäre Kreuzentropie (BCE) oder Kombinationen aus BCE und der Dice-Metrik zum Einsatz (siehe Abschnitt 2.1). Diese Auswahl trägt dazu bei, auch bei unausgeglichene Klassenverteilungen eine stabile Segmentierung zu gewährleisten.



**Abbildung 4.2:** U-Net Architektur. Der Encoder reduziert die Bildgröße und erweitert den Merkmalsraum. Der Decoder rekonstruiert daraus anschließend ein segmentiertes Bild. Skip connections verbinden den Encoder mit dem Decoder.

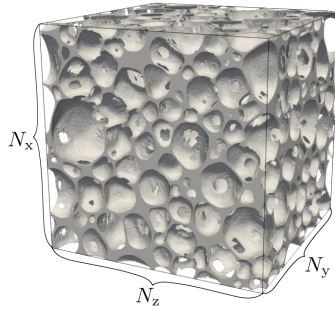
Für das Training als überwachter Lernalgorithmus benötigt der U-Net Ground-Truth-Daten in Form von Grauwertbild und korrekter binärer Segmentierung. Die manuelle Erstellung dieser Daten ist jedoch zeitaufwändig und fehleranfällig. Daher wird in dieser Arbeit eine Methode zur künstlichen Generierung realistischer, physikalisch konsistenter Trainingsdaten entwickelt und eingesetzt (siehe Abschnitt 4.4).

## Konvertierung zu 3D Modell

Nach der Segmentierung liegen die CT-Daten als Stapel zweidimensionaler binärer Bilder vor, in denen jeder Pixel entweder der Strukturphase oder dem Porenraum zugeordnet ist. Da die Orientierung der segmentierten Struktur relativ zu den kartesischen Bildkoordinaten nicht notwendigerweise übereinstimmt, müssen die Bilder zunächst mit geeigneten Bildbearbeitungsverfahren wie bilinearer oder bikubischer Interpolation ausgerichtet werden. Dies gewährleistet, dass die Strukturkanten mit den Achsen des kartesischen Koordinatensystems der Bilddaten übereinstimmen, sodass richtungsabhängige Struktureigenschaften zuverlässig bestimmt werden können. Anschließend werden die ausgerichteten Bilder übereinandergelegt und zu einem dreidimensionalen Modell verschmolzen, das den digitalen Zwilling bildet.

### 4.2.3 Geometrische Charakterisierung

Dieses Kapitel stellt die rechnergestützten Methoden dar, mit denen digitale Zwillinge poröser Mikrostrukturen quantifiziert werden können. Die digitalen Zwillinge stehen dazu in Form von Voxelgittern zur Verfügung. Diese erstrecken sich im kartesischen, dreidimensionalen Raum  $\Omega \in \mathbb{R}^3$  mit den Kantenlängen  $N_x$ ,  $N_y$  und  $N_z$ . Abbildung 4.3 stellt einen solchen digitalen Zwilling dar.



**Abbildung 4.3:** Exemplarische rekonstruierte Schaumstruktur mit den Kantenlängen  $N_x$ ,  $N_y$  und  $N_z$ .

Die Erstellung der digitalen Zwillinge kann entweder über die Rekonstruktion aus bildgebenden Verfahren (vgl. Abschnitt 4.4) oder die algorithmenbasierte Generierung (vgl. Unterabschnitt 5.3.1) erfolgen. Für die Analyse der Strukturgeometrie muss diese in binarisierter Form vorliegen. Während die algorithmenbasierten Methoden direkt binäre Strukturen erzeugen, erfolgt die Binarisierung von Bilddaten durch deren Segmentierung Abschnitt 4.4. In einer solchen binarisierten Struktur wird jedem Voxel  $\mathbf{x}$  ein Wert  $P(\mathbf{x})$  zugewiesen. Diese Zuweisung ist in Abbildung 4.4 visualisiert. Für diese gilt dabei

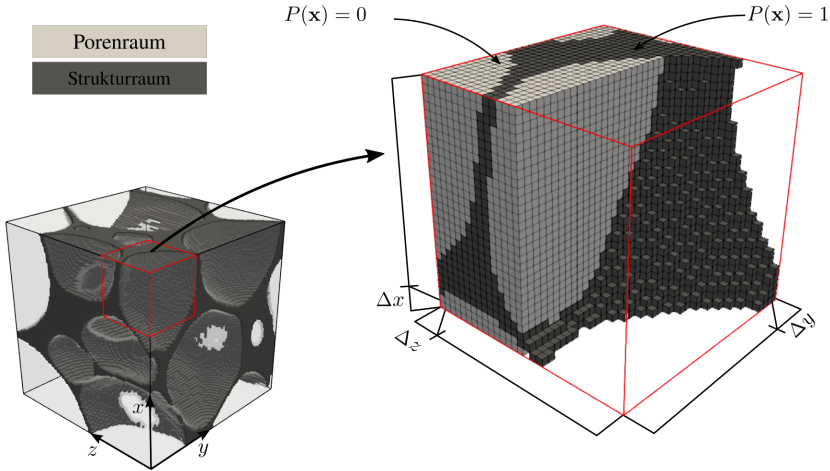
$$P(\mathbf{x}) = \begin{cases} P_S = 1, & \text{für } \mathbf{x} \in \Omega^+, \text{ Vordergrund,} \\ P_P = 0, & \text{für } \mathbf{x} \in \Omega^-, \text{ Hintergrund} \end{cases} \quad (4.4)$$

mit

$$\mathbf{x} = (i, j, k) \in \Omega | (i, j, k) \leq (N_x, N_y, N_z).$$



Die nachfolgend vorgestellten Algorithmen arbeiten stets auf der Strukturphase deren Wert 1 beträgt. Abhängig von der zu untersuchenden Strukturphase kann die Wertzuweisung aus Gleichung 4.4 invertiert werden.



**Abbildung 4.4:** Ausschnitt der rekonstruierten digitalen Schaumzwillinge als Voxelgitter, wobei der Porenraum als  $P_P(x) = 0$  und der Strukturraum als  $P_S(x) = 1$  abgebildet ist.

#### 4.2.3.1 Porosität

Die Porosität einer Struktur ist als das Verhältnis ihres Porenvolumens  $V_P$  zu ihrem Gesamtvolumen  $V$  definiert als

$$\varepsilon = \frac{V_P}{V}. \quad (4.5)$$

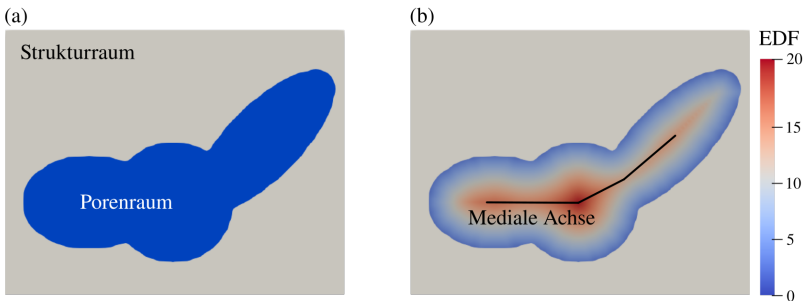
Für die Berechnung dieser Größe erfolgt zunächst die Invertierung des digitalen Zwillings, so dass  $P_P = 1$  und  $P_S = 0$  gilt. Das Aufsummieren aller Voxelwerte und

das Dividieren der Summe durch die Gesamtzahl aller Voxel ergibt anschließend die Porosität durch

$$\varepsilon = \frac{1}{N_x N_y N_z} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} P(\mathbf{x} = (i, j, k)). \quad (4.6)$$

#### 4.2.3.2 Struktur- und Porenraum

Neben der Beschreibung der Zusammensetzung der Gesamtstruktur über die Porosität, ist die Analyse der konstituierenden Phasen (Struktur und Pore) für eine ganzheitliche geometrische Charakterisierung der digitalen Zwillinge notwendig. Zur Beschreibung der Strukturphasen ist in dieser Arbeit die Wandstärke von großer Bedeutung, da diese einen großen Einfluss auf die mechanischen Schaumeigenschaften hat. Die Messung der mittleren Wanddicke des Schaums kann über einen Skelettierungs-Algorithmus [138] erfolgen. Dieser errechnet die mittlere Dicke der betrachteten Phase über einen dreistufigen Prozess: i) Berechnung eines Euklidischen Distanzfelds (EDF), ii) Bestimmung der medialen Achse, iii) Filterung des EDFs durch die mediale Achse. Die Errechnung des EDFs bestimmt für jeden Voxel der betrachteten Phase den minimalen Abstand zur jeweils anderen Phase. Das Ergebnis dieses Algorithmus ist ein Distanzfeld der betrachteten Phase. Abbildung 4.5 (b) zeigt das Distanzfeld der in Abbildung 4.5 (a) exemplarisch



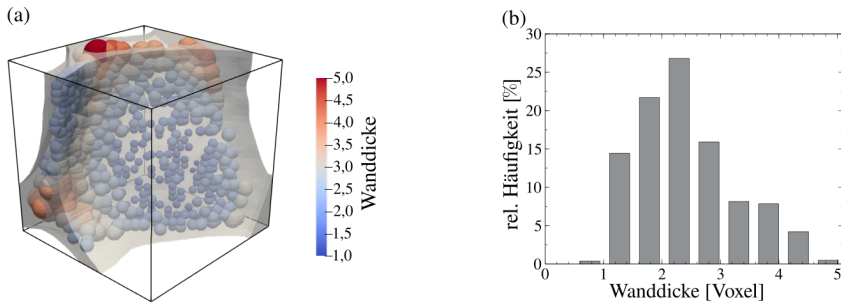
**Abbildung 4.5:** Euklidisches Distanzfeld und mediale Achse im Skelettierungs-Algorithmus. (a) zeigt eine exemplarische Schaumstruktur und (b) das EDF sowie die mediale Achse.

gezeigten Schaumstruktur. Auf diesem Distanzfeld wird im zweiten Schritt durch Skelettierung eine mediale Achse berechnet.

Die Berechnung der medialen Achse basiert auf dem von Palagyi et al. [139] vorgestellten parallelisierten Algorithmus. Das EDF wird dadurch auf die lokalen Minima reduziert, wodurch sich das Skelett der Struktur ergibt.

Der letzte Schritt des Skelettierungs-Algorithmus verwirft alle Werte des EDFs, die nicht auf der medialen Achse liegen. Entsprechend lässt sich aus dem resultierenden EDF über den Mittelwert und die Standardabweichung die Wanddickenverteilung beschreiben. Das Invertieren der Struktur- und Porenphase vor der Anwendung des Skelettierungs-Algorithmus, so dass  $0 \triangleq \text{Struktur}$  und  $1 \triangleq \text{Poren}$  gilt, ermöglicht die Berechnung der Porengrößenverteilung.

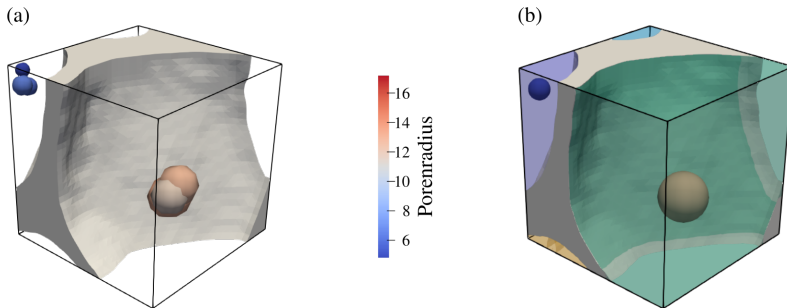
Abbildung 4.6 zeigt das Ergebnis der Strukturanalyse eines Schaumausschnitts durch den Skelettierungs-Algorithmus. Die Wanddicken werden entlang der medialen Achse durch Kugeln dargestellt, deren Farbe die Dicke kodiert. Die Wanddickenverteilung kann auf diese Weise zuverlässig bestimmt werden.



**Abbildung 4.6:** Ermittelte Wanddickenverteilung. In (a) ist die lokale Wanddicke entlang der medialen Achse als Kuglradien visualisiert; die Farbskala kodiert die Dicke. (b) zeigt die resultierende Wanddickenverteilung.

Bei der Berechnung der Porengrößenverteilung hingegen hat sich der verwendete Skelettierungs-Algorithmus als ungenau erwiesen, da die Porengröße entlang der

medialen Achse gemittelt wird. Die Porenradien sind allerdings nur für den Porenmittelpunkt definiert. Der Skelettierungs-Algorithmus unterschätzt die Porengrößen entsprechend. Abbildung 4.7 visualisiert diesen Nachteil des Algorithmus.



**Abbildung 4.7:** Unterschiede zwischen den ermittelten Poren (a) und den realen Poren (b). Jede Pore wird über mehrere Punkte beschrieben, was zu Ungenauigkeiten führt.

Zur Beschreibung des Porenraums wird in dieser Arbeit daher ein Algorithmus zur Extraktion eines Porennetzwerkmodells verwendet, der die Methode von Gostick [132] implementiert.

Der Algorithmus folgt einer zweistufigen Analyse des Porenraums: i) Unterteilung des Porenraums in einzelne Poren durch eine markerbasierte Wasserscheidentransformation. ii) Erstellung eines Graphs, der die Poren und Kanäle der Struktur beschreibt.

Im ersten Schritt wird zunächst das EDF des Porenraums berechnet. In diesem Distanzfeld werden lokale Maxima identifiziert, die als Mittelpunkte der jeweiligen Poren interpretiert werden. Diese Mittelpunkte dienen anschließend als Marker für die markerbasierte Wasserscheidentransformation, mit der der gesamte Porenraum in einzelne Porenbereiche unterteilt wird.

Im zweiten Schritt werden für jede identifizierte Pore geometrische Eigenschaften wie Volumen und Porenradius bestimmt. Der Porenradius ist dabei definiert als der Radius der größten Kugel, die vollständig in der jeweiligen Pore enthalten ist.

Eine ausführliche Beschreibung des Algorithmus findet sich in [133]. Das Ergebnis der Analyse durch den Algorithmus zur Extraktion des Porennetzwerks sind die isolierten Eigenschaften der identifizierten Poren. Der in dieser Arbeit relevante Porenradius wird ausschließlich über diesen Algorithmus gemessen, da er genauer als der vom Skelettierungs-Algorithmus ermittelte Radius ist.

### 4.2.4 Datengetriebene Charakterisierung

Neben geometrischen Kenngrößen wie Porosität, Porenradius und Wandstärke ist auch ein Maß erforderlich, das komplexe Strukturmerkmale und die mikrostrukturelle Morphologie erfasst. Dies ist insbesondere für den Vergleich digitaler Zwillinge relevant, die entweder aus bildgebenden Verfahren rekonstruiert oder mittels Generierungsalgorithmen synthetisch erzeugt wurden. Für die simulative Untersuchung der in dieser Arbeit verwendeten synthetischen digitalen Zwillinge spielt die Morphologie eine zentrale Rolle, da die Simulationen auf der vollständig aufgelösten Mikrostruktur basieren. Nur wenn diese derjenigen realer Strukturen entspricht, lassen sich die Simulationsergebnisse zuverlässig übertragen. Daher müssen Strukturen sowohl in den abgeleiteten geometrischen Kenngrößen als auch in ihrer räumlichen Morphologie mit den realen Daten übereinstimmen.

Zur objektiven Bewertung der Strukturähnlichkeit wird eine Kombination aus *Zweipunktkorrelation* (ZPK) und *Hauptkomponentenanalyse* (PCA) verwendet. Diese Vorgehensweise wurde bereits erfolgreich von Altschuh et al. [136] zum Vergleich generierter und realer Mikrostrukturen eingesetzt.

## Zweipunktkorrelation (ZPK)

Die n-Punktkorrelationen sind statistische Methoden, die räumliche Korrelationen in Strukturen in Form von Wahrscheinlichkeiten beschreiben. Die Zweipunktkorrelation ist eine spezielle Form, die die Wahrscheinlichkeit angibt, dass zwei lokale Zustände  $h$  und  $h'$  im Abstand eines Vektors  $\vec{r} = (r_x, r_y, r_z)^T$  gemeinsam auftreten. Es wird zwischen Autokorrelation ( $h = h'$ ) und Kreuzkorrelation ( $h \neq h'$ ) unterschieden. In dieser Arbeit wird ausschließlich die Autokorrelation der Strukturphase ( $h = h' = 1$ ) betrachtet.

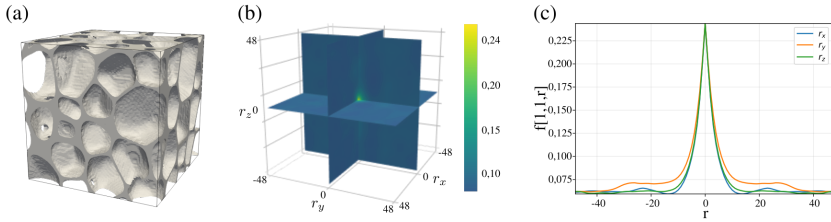
Die Berechnung erfolgt auf binarisierten 3D-Strukturen, wobei Poren den Wert 0 und die Struktur den Wert 1 erhalten. Damit lässt sich die Mikrostruktur als binäres Feld  $m[h, s]$  darstellen, wobei  $s$  der Voxelindex über alle  $S$  Voxel ist.

Die ZPK ist definiert als:

$$f[h, h', \vec{r}] = \frac{1}{S} \sum_{s=1}^S m[h, s] \cdot m[h', s + \vec{r}],$$

wobei  $\vec{r}$  alle diskreten Vektoren im kartesischen Raum durchläuft. Die ZPK ist zum Ursprung symmetrisch, sodass nur die Hälfte der Richtungen ausgewertet werden muss. Für eine Struktur mit einer Kantenlänge von  $N$  Voxel ergeben sich demnach  $N^3/2$  unabhängige Merkmale. Bei  $N = 200$  entspricht dies etwa  $4e6$  Merkmalen pro Struktur. Zur effizienten Berechnung wird die *Fast Fourier Transformation* (FFT) eingesetzt, wodurch der Rechenaufwand im Vergleich zur direkten Berechnung deutlich reduziert wird.

Die resultierende ZPK beschreibt die räumliche Wahrscheinlichkeitsverteilung der Strukturphase entlang der drei Raumrichtungen. Abbildung 4.8 zeigt in (a) beispielhaft eine Schaumstruktur und in (b) die zugehörige ZPK. In (c) sind die Vektorkomponenten  $r_x, r_y, r_z$  dargestellt. Ein besonders relevanter Punkt ist  $\vec{r} = (0, 0, 0)$ , da er die Volumenfraktion der Strukturphase angibt.



**Abbildung 4.8:** Zweipunktkorrelation (ZPK) einer exemplarischen Schaumstruktur: (a) binarisierte 3D-Mikrostruktur, (b) zugehörige ZPK, (c) ZPK-Verlauf entlang der Achsenrichtungen  $r_x$ ,  $r_y$  und  $r_z$ .

## Hauptkomponentenanalyse (PCA)

Die hohe Dimensionalität der ZPK-Daten erschwert eine direkte Analyse. Daher wird eine PCA durchgeführt, die die hochdimensionalen Korrelationsdaten in ein Koordinatensystem aus Hauptachsen (Principal Components) transformiert, das mit wenigen Dimensionen die maximale Varianz abbildet.

Formal lässt sich die PCA der ZPK für jede Struktur  $k$  als lineare Kombination der Basisfunktionen  $\varphi[1, 1, \vec{r}, i]$  darstellen:

$$f_k[1, 1, \vec{r}] = \sum_{i=1}^{\min(K-1, R)} \alpha_k[i] \cdot \varphi[1, 1, \vec{r}, i] + \bar{f}[1, 1, \vec{r}],$$

wobei  $K$  die Anzahl der Strukturen und  $R$  die ursprüngliche Merkmalsanzahl ist. Die Koeffizienten  $\alpha_k[i]$  sind die Projektionen der  $k$ -ten Struktur auf die  $i$ -te Hauptkomponente.  $\bar{f}[1, 1, \vec{r}]$  bezeichnet die gemittelte Autokorrelation über alle Strukturen. In dieser Arbeit werden vier Hauptkomponenten verwendet, da diese bereits den Großteil der Varianz erfassen.

## Bewertung im PCA-Raum

Für eine konsistente Vergleichbarkeit wird die PCA in dieser Arbeit ausschließlich auf den realen CT-Daten berechnet. Die generierten Strukturen werden anschließend mit der so bestimmten Transformation in denselben Hauptkomponentenraum projiziert. Liegen die Projektionen zweier Strukturen räumlich nahe beieinander, deutet dies auf ähnliche Morphologien der Mikrostrukturen hin. Damit ermöglicht diese Methodik eine systematische und objektive Bewertung des Realismus generierter Mikrostrukturen und bildet zugleich die Grundlage für sämtliche Bewertungen der Generierungsergebnisse in den folgenden Kapiteln.

## Messung von Bildähnlichkeiten

Zur Bewertung der Ähnlichkeit zwischen realen und synthetischen Mikrostrukturbildern stehen neben der Kombination aus Zweipunktkorrelation (ZPK) und Hauptkomponentenanalyse (PCA) weitere datengetriebene Verfahren zur Verfügung [140–143]. Zwei besonders etablierte Methoden sind die *Fréchet Inception Distance* (FID) [143, 144] sowie das *t-distributed Stochastic Neighbor Embedding* (t-SNE) [145].

Die FID kann, im Gegensatz zur ausschließlich auf binarisierten Strukturen arbeitenden ZPK, auch auf kontinuierliche Bilddaten wie Grauwertbilder angewendet werden. Dabei wird der Informationsgehalt zweier Bilddatensätze verglichen, ohne dass sie dieselben Strukturen zeigen müssen.

Zur Merkmalsextraktion werden vortrainierte neuronale Netze wie *Inception-v3* eingesetzt, welche die Bilder in niedrigdimensionale Tensoren überführen. Aus diesen werden multivariate Gauß-Verteilungen geschätzt, deren Fréchet-Distanz den FID-Wert ergibt. Ein niedriger FID deutet auf eine hohe visuelle Ähnlichkeit zwischen den betrachteten Datensätzen hin.

Da FID nicht auf domänenspezifischem Vorwissen beruht, ist es universell einsetzbar und unabhängig von der konkreten Mikrostruktur. Allerdings liefert es



primär qualitative Aussagen zur Bildähnlichkeit. Eine quantitative Bewertung struktureller Eigenschaften, wie sie mit ZPK und PCA möglich ist, erlaubt FID nicht.

Ein weiteres qualitatives Vergleichsmaß ist das *t-distributed Stochastic Neighbor Embedding* (t-SNE) [145]. Es reduziert hochdimensionale Bild-Embeddings auf zwei Dimensionen, wobei die lokalen Nachbarschaftsbeziehungen zwischen den Datenpunkten bestmöglich erhalten bleiben. Die Ähnlichkeit der Embeddings wird dabei über eine Distanzmetrik gemessen, typischerweise die Kosinus- oder euklidische Distanz. In der visualisierten Projektion lassen sich so Gruppierungen, Überschneidungen oder auch Abweichungen im Merkmalsraum intuitiv interpretieren.

### 4.3 Verwendete Datensätze

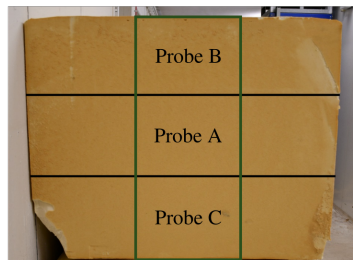
In diesem Kapitel werden die in dieser Arbeit untersuchten Materialien und die zugehörigen Messdaten vorgestellt. Ziel ist es, die physikalischen Eigenschaften der Proben sowie die Verfahren zu ihrer experimentellen und bildgebenden Erfassung zu beschreiben. Der Schwerpunkt liegt auf einem geschlossenzelligen Polyurethanschaum, der aufgrund seiner geringen Dichte und hohen Energieabsorption häufig in Leichtbauanwendungen, insbesondere als Crashelement in Fahrzeugen, eingesetzt wird [146]. Neben der Materialbeschreibung werden die experimentelle Bestimmung wesentlicher Kenngrößen und die tomographische Erfassung der Mikrostruktur erläutert.

#### 4.3.1 Polyurethanschaum

In dieser Arbeit wurde ein kommerziell erhältlicher, geschlossenzelliger Polyurethanschaum *RG200 PUR* [147] untersucht. Polyurethanschaum entsteht durch das Aufschäumen einer reaktiven Polymermischung, wobei Gasblasen in der aushärtenden Strukturmatrix eingeschlossen werden. Dadurch ergibt sich eine

homogene Schaumstruktur mit gleichmäßig verteilten Poren. Da die eigentliche Herstellung und die Auswirkung der Prozessparameter auf die finale Schaumstruktur nicht in dieser Arbeit betrachtet werden, soll nicht weiter darauf eingegangen werden. Relevant für die vorliegende Arbeit ist lediglich, dass es sich um einen geschlossenzelligen Schaum mit einer gleichmäßigen Porenverteilung handelt, der typischerweise in technischen Anwendungen eingesetzt wird.

Bekannt ist, dass sich durch den Herstellungsprozess entlang der Aufschäumrichtung teilweise unterschiedliche Mikrostrukturen ausbilden. Um diesen Effekt zu berücksichtigen, werden drei Proben aus demselben Schaumblock untersucht, die an unterschiedlichen Positionen entnommen wurden: *Probe A* in der Mitte, *Probe B* am oberen Rand und *Probe C* am unteren Rand des Schaums.



**Abbildung 4.9:** Entnahmepositionen der Schaumproben entlang der Aufschäumrichtung. Unterschiede in den Mikrostrukturen, die durch den Herstellungsprozess entstehen, werden durch die ausgewählten Proben abgedeckt.

### 4.3.2 Experimentelle Charakterisierung

Neben den CT-Daten ist über die verwendeten Schäume aus den Herstellerdatenblättern lediglich die effektive Dichte bekannt. Für den vorliegenden Schaum RG200 PUR [147] wird diese mit  $186 \text{ kg/m}^3$  bis  $205 \text{ kg/m}^3$  angegeben. Strukturgrößen wie die Porosität oder die Porengrößen sind nicht bekannt. Da es sich bei dem untersuchten Schaummaterial um geschlossenzellige Strukturen handelt,

konnte die Porengröße nicht wie in [97] bestimmt werden. Um dennoch Strukturinformationen zur Validierung der Segmentierung zu erhalten, wurde die Porosität der Schaumproben experimentell ermittelt.

Die Porosität  $\varepsilon$  eines porösen Materials lässt sich aus der effektiven Dichte  $\rho_{\text{eff}}$  und der Dichte der Polyurethanmatrix  $\rho_{\text{PUR}}$  über die Beziehung  $\varepsilon = \frac{\rho_{\text{eff}}}{\rho_{\text{PUR}}}$  berechnen. Voraussetzung dafür ist die Kenntnis der reinen Materialdichte  $\rho_{\text{PUR}}$ , die experimentell bestimmt werden muss, da die Dichte von Polyurethan aufgrund des variationsreichen Herstellungsprozesses nicht konstant ist [148].

Zur Bestimmung der Dichte des reinen Polyurethans wurde die Flüssigpyknometrie gemäß EN ISO 1183-1 [126] eingesetzt. Dieses Verfahren ermöglicht es, über die Volumenverdrängung in einer Referenzflüssigkeit die Dichte eines festen Stoffes zu berechnen. Dabei wird zunächst das leere Pyknometer, ein Glaskolben mit bekanntem Volumen, gewogen. Anschließend mit der Referenzflüssigkeit, in diesem Fall Ethanol mit einer Dichte von  $0,81 \text{ g/cm}^3$ , gefüllt und erneut gewogen. Danach wird eine definierte Menge des untersuchten Materials in das Pyknometer gegeben, das anschließend wieder mit Ethanol aufgefüllt und ein weiteres Mal gewogen wird. Aus diesen Messwerten kann die Dichte des Materials nach folgender Gleichung berechnet werden:

$$\rho_{\text{rein}} = \frac{m_{\text{Schaum}}}{m_{\text{Ethanol}} - m_{\text{Schaum+Ethanol}} + m_{\text{Schaum}}} \rho_{\text{Ethanol}}, \quad (4.7)$$

wobei  $m_s$  die Masse der Probe,  $\rho_{\text{IL}}$  die Dichte der Referenzflüssigkeit,  $m_1$  die Masse der Referenzflüssigkeit zum Füllen des leeren Pyknometers und  $m_2$  die Masse der Referenzflüssigkeit zum Füllen des Pyknometers mit Probe bezeichnet. Der Polyurethanschaum wurde vor der Messung mechanisch zerkleinert, um ihn in grobkörniger, porenfreier Form vorliegen zu haben und damit Lufteinschlüsse während der Messung zu vermeiden.

Zur Erhöhung der Genauigkeit wurden drei unabhängige Messungen durchgeführt, aus denen der Mittelwert der Dichte bestimmt wurde. Die so ermittelte Dichte des reinen Polyurethans bildet die Grundlage zur anschließenden Berechnung

**Tabelle 4.1:** Ergebnisse der Pyknometrie zur Ermittlung der Dichte des Polyurethans. Die römischen Ziffern, zeigen die Ergebnisse der drei Messungen.

	Einheit	I	II	III	Mittelwert
$m_{\text{Schaum}}$	[g]	0,2013	0,1964	0,2066	
$m_{\text{Ethanol}}$	[g]	7,0893	7,0933	7,0924	
$m_{\text{Ethanol+Schaum}}$	[g]	6,9758	6,9789	6,9734	
$\rho_{\text{rein}}$	[g/cm <sup>3</sup> ]	1,437	1,391	1,406	1,411

der Porosität der Schaumproben nach obiger Gleichung. Die Ergebnisse dieser Messungen sind in Tabelle Tabelle 4.1 zusammengefasst.

Anschließend wurde zur Berechnung der Porosität die effektive Dichte jeder Probe bestimmt. Hierzu wurden größere Proben aus den Bereichen geschnitten aus denen auch die untersuchten Schaumproben stammen und ihre Abmessungen genau erfasst. Anschließend wurden diese Proben auf einer Präzisionswaage mit einer Genauigkeit von 0,01 g gewogen. Aus Gewicht und Volumen wurde die effektive Dichte  $\rho_{\text{eff}}$  jeder Probe ermittelt. Die daraus berechneten Porositäten  $\varepsilon$  sowie die effektiven Dichten sind in Tabelle 4.2 zusammengefasst. Diese experimentell bestimmten Porositäten dienen als Referenzwerte zur Bewertung der späteren Segmentierung.

**Tabelle 4.2:** Effektive Dichte und Porosität jeder Probe. Durch Messung der effektiven Dichte kann die Porosität berechnet werden.

	Einheit	Probe A	Probe B	Probe C
$\rho_{\text{eff}}$	[kg/m <sup>3</sup> ]	154,93	197,72	248,34
$\varepsilon$	[%]	89,02	85,99	82,40

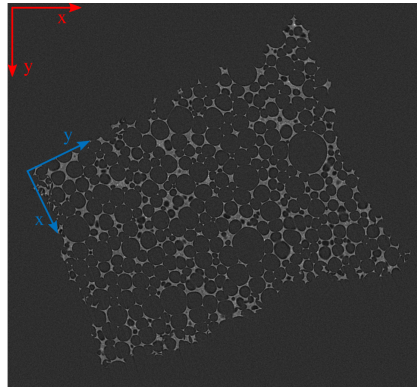
### 4.3.3 CT Bildgebung

Aufgrund der Verfügbarkeit und der zu erwartenden Dimensionen der Mikrostruktur wurden die in dieser Arbeit verwendeten Polyurethan-Schaumproben mit einem Cone-Beam-Setup tomographisch mit einer Auflösung von  $2,75\text{ }\mu\text{m}/\text{Voxel}$  abgebildet. Tabelle 4.3 zeigt die Abmessungen der verfügbaren Proben. In den

**Tabelle 4.3:** Bildgrößen und reale Abmessungen der tomographierten Polyurethanschaumproben bei einer Auflösung von  $2,75\text{ }\mu\text{m}/\text{PxI}$ .

	Bildgröße [Voxel]	Abmessung [mm]
<i>Probe A</i>	$962 \times 1034 \times 1500$	$2,65 \times 2,84 \times 4,13$
<i>Probe B</i>	$963 \times 693 \times 1000$	$2,65 \times 1,91 \times 2,75$
<i>Probe C</i>	$895 \times 726 \times 1100$	$2,46 \times 2,00 \times 3,03$

rekonstruierten Bildern ist ein typisches Bildrauschen zu erkennen, das sich durch ein lokales Schwanken der Grauwerte auszeichnet. Abbildung 4.10 zeigt eine exemplarische CT Aufnahme eines Schaums. Es ist ersichtlich, dass die Schaum-



**Abbildung 4.10:** CT-Aufnahme von *Probe B*. Die Kanten des Schaums und somit die Schaumkoordinaten (blau) fallen nicht mit den Bildkoordinaten (rot) zusammen.

koordinaten nicht mit den Bildkoordinaten übereinstimmen. Zur späteren Analyse richtungsabhängiger Eigenschaften muss die Ausrichtung der Bilder entsprechend angepasst werden.

## 4.4 Rekonstruktion

Für die Untersuchung des Zusammenhangs zwischen Mikrostruktur und makroskopischen mechanischen Eigenschaften werden aus den CT-Grauwertbildern binäre digitale Zwillinge rekonstruiert. Die Segmentierung erfolgt mit der in Unterabschnitt 4.2.2 beschriebenen U-Net-Architektur, die als überwachtes Lernverfahren segmentierte Referenzbilder benötigt. Um die aufwendige und fehleranfällige manuelle Annotation zu vermeiden, werden in dieser Arbeit künstlich erzeugte, physikalisch konsistente Ground-Truth-Daten verwendet.

Hierfür wurde eine zweistufige Methode entwickelt, die geometriebasierte Generierungsverfahren (vgl. Unterabschnitt 5.3.1) mit maschinellen Lernverfahren kombiniert, um die Ähnlichkeit zu realen Mikrostrukturen schrittweise zu erhöhen. Die Qualität der synthetischen Trainingsdaten wird gemäß Unterabschnitt 4.2.4 anhand von Bild-Embeddings eines vortrainierten Encoder-Netzwerks bewertet.

Auf Basis dieser synthetischen Trainingsdaten wird ein U-Net trainiert und zur Segmentierung der CT-Aufnahmen eingesetzt. Die Qualität der Segmentierungen wird abschließend durch den Vergleich der berechneten Porositäten mit experimentellen Referenzwerten überprüft.

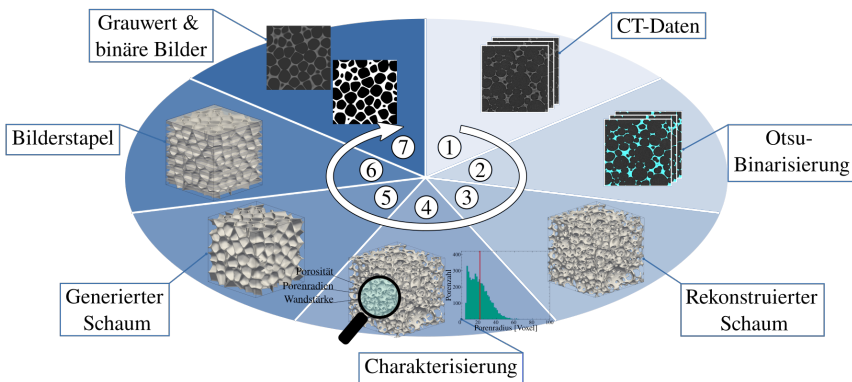
### 4.4.1 Datengenerierung

In diesem Abschnitt wird die entwickelte zweistufige Methode zur Generierung künstlicher Trainingsdaten vorgestellt, wie sie bereits in [149] veröffentlicht wurde. Ziel des Verfahrens ist es, die Qualität der synthetischen Daten in einem zweistufigen Prozess iterativ zu verbessern, um ein U-Net für die Segmentierung von Mikrostrukturen zu trainieren.

Beide Schritte umfassen dabei die Erzeugung synthetischer Graustufenbilder mit zugehörigen binären Ground-Truth-Daten sowie das anschließende Training eines U-Nets. Die beiden Prozessschritte der Datengenerierung werden im Folgenden detailliert beschrieben.

## Erster Schritt

Ziel des ersten Schritts ist es, ein U-Net zu trainieren, das eine verbesserte Bildsegmentierung bietet, auf der der zweite Schritt des Datengenerierungsprozesses aufbauen kann. Dazu synthetisiert der in Abbildung 4.11 gezeigte Prozess künstliche Trainingsdaten mittels geometriebasierter Generierungsalgorithmen.



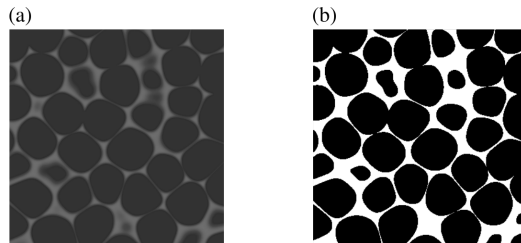
**Abbildung 4.11:** Der erste Schritt des Datengenerierungsprozesses. CT-Daten werden mit dem Otsu-Schwellenwertverfahren binarisiert, rekonstruiert und geometrisch charakterisiert (4). Aus den geometrischen Eigenschaften werden mit geometriebasierten Methoden künstliche Schaumstrukturen synthetisiert (5), in gestapelte Bilder geschnitten (6) und verarbeitet, um CT-Daten mit bekannter Ground-Truth zu ähneln (7). Adaptiert von [149].

Zunächst werden die verfügbaren CT-Aufnahmen der Schaumstrukturen mit einem globalen Otsu-Schwellenwert binarisiert. Aus den binarisierten Bildern wird ein dreidimensionales digitales Modell der Schaumstruktur rekonstruiert, das

anschließend hinsichtlich Porosität, Porengrößenverteilung und mittlerer Wandstärke analysiert wird. Basierend auf diesen Kenngrößen werden mithilfe des in PACE3D [150] implementierten geometriebasierten Algorithmus (vgl. Unterabschnitt 5.3.1) ähnliche Schaumstrukturen generiert. Alternativ könnten auch andere öffentlich verfügbare Generierungsverfahren eingesetzt werden.

Die synthetisierten Strukturen werden anschließend entlang ihrer Höhe in binäre Bildstapel zerlegt. Zur Erzeugung der entsprechenden Graustufenbilder werden die binären Bilder in 8-Bit-Bilder umgewandelt, mit einem Gaußschen Weichzeichner geglättet, im Kontrast angepasst und abschließend mit Gaußschem Rauschen überlagert. Diese Verarbeitungsschritte erfolgen vollständig in der quelloffenen Software *Fiji*. Das entsprechende Makro ist öffentlich verfügbar [151].

Damit steht der erste Trainingsdatensatz an dieser Stelle zur weiteren Verwendung zur Verfügung. Abbildung 4.12 zeigt ein exemplarisches synthetisches Grauwertbild und die zugehörige binäre Ground-Truth.



**Abbildung 4.12:** Künstliche Trainingsdaten, generiert im ersten Schritt des zweistufigen Generierungsprozess. Bild a) zeigt ein Grauwertbild, das vom binären Bild b) abgeleitet wurde.

## Modelltraining

Der erste Datensatz dient als Grundlage, um ein U-Net zu trainieren, das eine verbesserte Bildsegmentierung ermöglicht und damit den zweiten Schritt der Datengenerierung vorbereitet. Dazu werden 12000 synthetische Eingabe- und



Ausgabebilder erzeugt, deren strukturelle Eigenschaften gezielt um die in den rekonstruierten CT-Daten gemessenen Werte variiert werden. Dadurch entsteht eine vielfältige und informationsreiche Trainingsbasis.

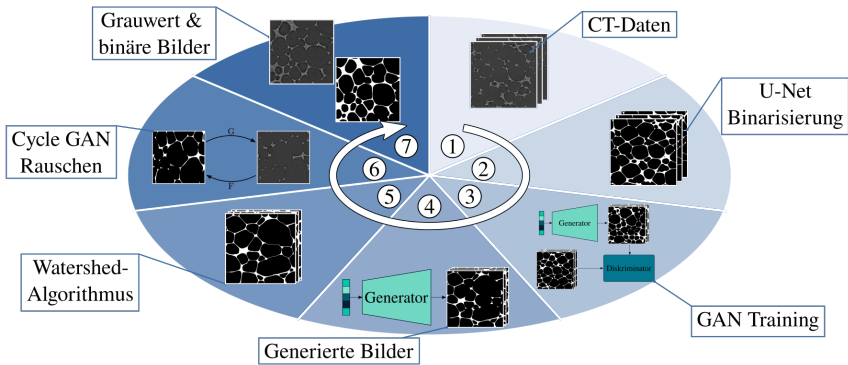
Für das Training kommt die Dice-Verlustfunktion zum Einsatz, um die Auswirkungen der unausgewogenen Verteilung von Vorder- und Hintergrundpixeln auf die Leistung des Modells zu reduzieren. Zusätzlich wird den Eingangsbildern während des Trainings Gaußsches Rauschen mit einer Standardabweichung von 0,05 vorverarbeitet. Diese Vorverarbeitung imitiert das in den realen CT-Bildern sichtbare Rauschen. Mit diesen Parametern erreicht das trainierte Modell auf dem ersten Datensatz eine Validierungsgenauigkeit von 99,8 %.

### **Zweiter Schritt**

Während das U-Net auf den geometriebasiert generierten Trainingsdaten bereits vielversprechende Ergebnisse liefert, bleibt die Bewertung seiner Leistung auf realen CT-Daten schwierig. Dies liegt daran, dass der erste Trainingsdatensatz möglicherweise nicht alle Details der realen CT-Daten erfasst und diese somit nicht genau nachahmen kann. Um diese Einschränkung zu überwinden, wird der nachfolgende Generierungsansatz für verbesserte synthetische Trainingsdaten eingeführt, der in der Abbildung 4.13 dargestellt ist.

Dieser Ansatz baut auf dem bereits trainierten U-Net auf. Ziel ist es, einen synthetischen Datensatz zu erzeugen, der die realen CT-Daten realistisch nachbildet. Dies soll es ermöglichen, die auf synthetischen Daten erzielte Genauigkeit des U-Net auf reale CT-Daten zu übertragen.

Der Prozess zur Generierung dieses Datensatzes folgt einem ähnlichen Verfahren wie der erste Schritt. Zunächst wird ein Generative Adversarial Network (GAN) auf den binarisierten CT-Bildern trainiert. Das erhaltene Netzwerk wird anschließend verwendet, um realistische binäre Bilder von Schaumstrukturen zu generieren. Nach weiterer Bearbeitung der GAN-synthetisierten binären Bilder,



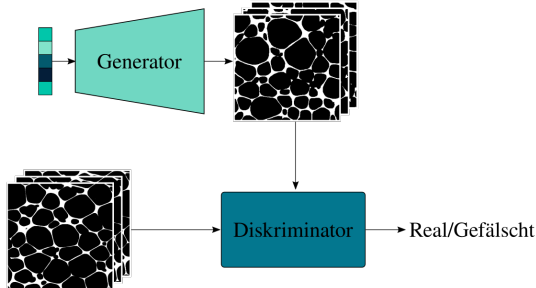
**Abbildung 4.13:** Der zweite Schritt des Datengenerierungsprozesses. CT-Scans werden unter Verwendung des im vorherigen Schritt unseres Verfahrens trainierten U-Net binarisiert. Anschließend wird ein GAN auf den binarisierten CT-Bildern trainiert, um realistische binäre Schaumstrukturen zu erzeugen. Die erhaltenen binären Bilder werden mit dem Watershed-Algorithmus verarbeitet, um fehlende Strukturmerkmale hinzuzufügen. Mit einem auf realen CT-Daten trainierten CycleGAN werden die binären Bilder in Graustufenbilder mit bekannter Ground-Truth umgewandelt.

werden aus ihnen CT-ähnliche Graustufenbilder abgeleitet. Hierzu kommt ein CycleGAN zum Einsatz, das auf realen CT-Daten trainiert wurde. Der beschriebene Ablauf erzeugt einen synthetischen Trainingsdatensatz, der aus realistischen Graustufenbildern und ihrer binären Ground-Truth besteht. Die genaue Arbeitsweise des zweiten Schritts wird nachfolgend beschrieben.

## Künstliche Generierung

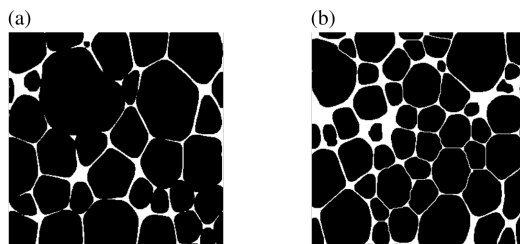
Die Generierung realistischer binärer Schaumstrukturen erfolgt mithilfe eines Generative Adversarial Networks (GAN), wie in Unterabschnitt 5.1.3 beschrieben. Als Trainingsdaten dienen die binarisierten CT-Bilder, die im ersten Schritt durch das U-Net segmentiert wurden. Es ergibt sich daraus die in Abbildung 4.14 schematisch dargestellte GAN-Architektur.

Nach dem Training des GAN mit diesen Daten kann der GAN-Generator realistische binäre Schaumstrukturen erzeugen. Es ist wichtig zu beachten, dass der



**Abbildung 4.14:** Schematische Darstellung der GAN-Architektur. Der Generator erzeugt synthetische Bilder binärer Schaumstrukturen, die darauf abzielen, den Diskriminator zu täuschen. Der Diskriminator versucht wiederum, zwischen echten und gefälschten Bildern zu unterscheiden.

GAN-Generator mit CT-Daten trainiert wurde, die vom ersten U-Net segmentiert wurden. Daher kann der GAN-Generator nur Bilder synthetisieren, die der Segmentierung dieses U-Nets ähneln. Strukturelle Merkmale in den Graustufen-CT-Daten, die vom U-Net nicht identifiziert wurden, werden in den generierten Bildern nicht vorhanden sein. Solche Merkmale wurden durch den manuellen Vergleich der segmentierten CT-Bilder mit ihren ursprünglichen Grauwertbildern identifiziert. Bei diesen handelt es sich um sehr dünne Porenwände. Um sicherzustellen, dass diese strukturellen Merkmale dennoch im neuen Datensatz vorhanden sind, wurden die vom GAN generierten binären Bilder mit dem Watershed-Algorithmus nachbearbeitet. Dieser identifiziert Poren im binären Bild und fügt dünne Wände ein, wodurch realistische, geschlossene Zellstrukturen entstehen. Um eine Verzerrung des Datensatzes durch das Hinzufügen dieser strukturellen Informationen zu vermeiden, erfolgt diese Nachbearbeitung nur auf der Hälfte aller vom GAN generierten Bilder. Abbildung 4.15 zeigt zwei beispielhafte vom GAN synthetisierte Strukturen, eine ohne und eine mit hinzugefügten strukturellen Informationen.



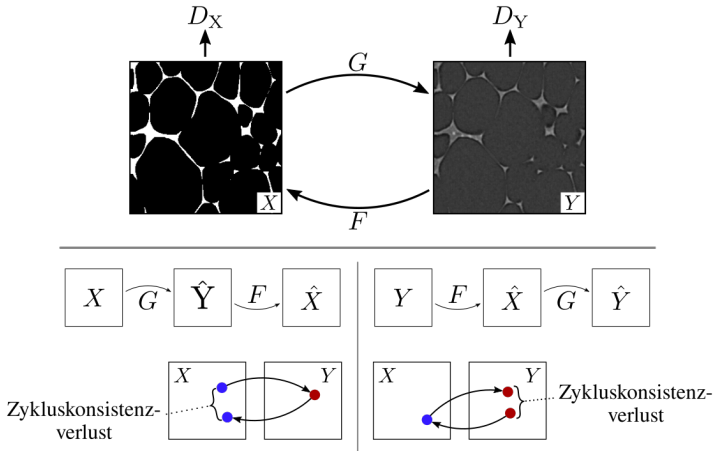
**Abbildung 4.15:** Bilder, die vom GAN-Generator erzeugt wurden. (a) Eine Ausgabe des GAN-Generators ohne weitere Nachbearbeitung. (b) Eine Mikrostruktur, die mit dem Watershed-Algorithmus nachbearbeitet wurde, um dünne strukturelle Elemente hinzuzufügen.

## Künstliches Rauschen

Um die verbesserten synthetischen Trainingsdaten zu vervollständigen, werden zu den vom GAN erzeugten binären Bildern passende Graustufenbilder generiert. Das Aufprägen realistisches Rauschens erfolgt mithilfe eines CycleGANs.

Abbildung 4.16 zeigt die Architektur des CycleGANs. Der Generator  $G$  lernt, binäre Bilder  $X$  in Graustufenbilder  $Y$  zu überführen, während der Diskriminator  $D_Y$  lernt, zwischen echten und gefälschten Graustufenbildern zu unterscheiden. Der Generator  $F$  hingegen übersetzt Graustufenbilder  $Y$  in binäre Bilder  $X'$  und der Diskriminator  $D_X$  unterscheidet zwischen echten und falschen binären Bildern. Die Verwendung eines Zykluskonsistenzverlusts stellt weiter sicher, dass ein binäres Bild  $X$ , das vom Generator  $G$  in ein Graustufenbild  $Y$  und anschließend vom Generator  $F$  wieder in ein binäres Bild  $X'$  übersetzt wurde, dem ursprünglichen  $X$  möglichst entspricht.

Da CycleGANs auf ungepaarten Daten trainiert werden können, wird das Netzwerk hier mit den vom GAN synthetisierten binären Bildern (siehe Abbildung 4.15) und den realen Graustufen-CT-Daten trainiert. Innerhalb der CycleGAN Komponenten, Generator und Diskriminator, kommen U-Nets zum Einsatz. Das Training der CycleGAN erzeugt einen Generator  $G$ , der binäre Bilder so manipulieren kann, dass sie Graustufen-CT-Bildern ähneln. Außerdem wird ein Generator  $F$  erstellt, der Graustufenbilder in binäre Bilder umwandeln kann.

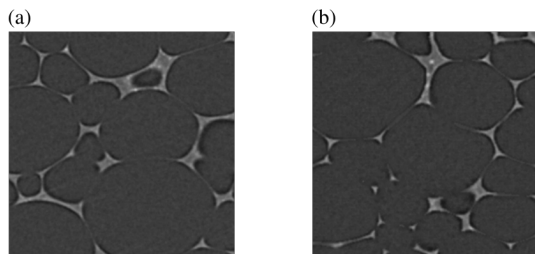


**Abbildung 4.16:** Schematische Darstellung der CycleGAN-Architektur. Zwei GANs werden gleichzeitig trainiert: eines transformiert Binär- in Grauwertbilder, das andere führt die inverse Abbildung durch. Darstellung adaptiert nach Zhu et al. [152].

Dieser wird später mittels Finetuning für die eigentliche Segmentierungsaufgabe herangezogen. Das Anwenden des Generators  $G$  auf die vom GAN synthetisierten binären Bilder finalisiert den zweiten synthetischen Datensatz und vervollständigt den zweistufigen Datengenerierungsprozess. Abbildung 4.17 stellt ein Graustufenbild des gerade erstellten Datensatzes einem realen CT-Bild gegenüber. Für das menschliche Auge ist das synthetisierte Graustufenbild fast nicht von den realen Daten zu unterscheiden, was auf eine enge Ähnlichkeit zwischen unseren Trainingsdaten und den realen Daten hinweist. Ob diese Ähnlichkeit auch objektiv messbar ist, wird im nächsten Abschnitt untersucht.

## Ähnlichkeitsmessung

Der beschriebene Generierungsprozess ermöglicht die Generierung hochrealistischer Trainingsdaten für Bildsegmentierungsaufgaben. Um die Ähnlichkeit zwischen realen und synthetischen Bildern quantitativ zu bewerten, wird die Fréchet



**Abbildung 4.17:** Vergleich zwischen CT-Bild und synthetischem Bild. (a) Ein reales CT-Bild. (b) Ein synthetisches Graustufenbild, das durch Hinzufügen von Rauschen zu einem vom GAN erzeugten Binärbild mit dem CycleGAN generiert wurde.

Inception Distance (FID) verwendet, die auf durch ein vortrainiertes Encoder-Netzwerk extrahierte Merkmalsrepräsentationen von Bilddatensätzen beruht. In der vorliegenden Arbeit wird ein ResNet50-Netzwerk verwendet, das auf dem *MicroNet*-Datensatz [117] trainiert wurde, einer öffentlich verfügbaren Sammlung von Graustufenmaterialbildern, die mittels Rasterelektronenmikroskopie oder Computertomographie aufgenommen wurden. Da dieser Datensatz hinsichtlich Struktur und Modalität stark unseren CT-Daten ähnelt, wird erwartet, dass die resultierenden Merkmalsrepräsentationen relevanter sind als die durch das häufig verwendete InceptionV3-Modell, das auf ImageNet vortrainiert wurde, extrahierten.

Um Merkmale aus jedem Bild zu extrahieren, wird die letzte Schicht des ResNet50-Modells entfernt. Das modifizierte Netzwerk gibt so einen 2048-dimensionalen Merkmalsvektoren aus, der zur Berechnung der FID-Werte verwendet wird. Da der verwendete Encoder vom standardmäßigen InceptionV3 abweicht, können bestehende Referenz-FID-Werte nicht direkt übernommen werden. Um sinnvolle Vergleichswerte zu schaffen, wird zunächst die FID zwischen den verschiedenen CT-Scans berechnet. Diese intra-CT-Vergleiche dienen anschließend als Referenz zur Bewertung der Ähnlichkeit zwischen realen und synthetischen Bildern. Für jeden dieser CT-Scans wurde ein synthetischer Datensatz generiert. Tabelle 4.4 zeigt die gemessenen FID-Werte zwischen den realen und synthetischen Datensätzen einschließlich der in der ersten Stufe des Generierungsprozesses

erzeugten Daten. In der Tabelle werden die in der zweiten Stufe erzeugten Daten als GAN und die aus der ersten Stufe als algorithmisch bezeichnet.

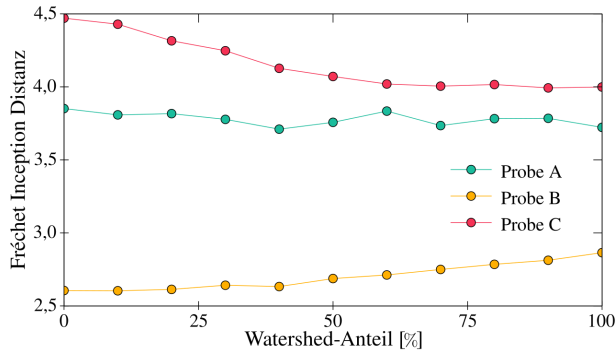
**Tabelle 4.4:** FID zwischen den realen und synthetischen Datensätzen. GAN bezeichnet die Trainingsdaten des zweiten Schritts und algorithmisch die des ersten.

	<i>Probe A</i>	<i>Probe B</i>	<i>Probe C</i>	GAN	Algorithmisch
<i>Probe A</i>	0	18,72	37,97	3,85	99,69
<i>Probe B</i>	18,72	0	13,62	2,61	78,10
<i>Probe C</i>	37,97	13,62	0	4,47	51,09

Die Referenzwerte der intra-CT-FID reichen von 13,62 bis 37,97. Im Gegensatz dazu sind die FID-Werte zwischen realen und durch GAN generierten Datensätzen deutlich geringer und liegen zwischen 2,61 und 4,47. Dies deutet darauf hin, dass die synthetischen Daten die statistischen Eigenschaften der realen Datensätze sehr gut nachbilden. Darüber hinaus ist eine deutliche Verbesserung der Realitätsnähe zu beobachten, wenn man GAN-generierte Bilder mit denen vergleicht, die durch einen herkömmlichen algorithmischen Ansatz erzeugt wurden.

Um den Einfluss der Nachbearbeitung zu bewerten, wird der Effekt eines variierenden Anteils an Bildern, die mit Watershed-Segmentierung nachbearbeitet wurden, auf die resultierenden FID-Werte untersucht. Abbildung 4.18 zeigt, dass dieser Nachbearbeitungsschritt die FID leicht beeinflusst, wobei der Effekt je nach Datensatz variiert. Hier kommt es teilweise zu einer Verbesserung, teils zu einer Verschlechterung der Ähnlichkeit.

Daher sollte die Entscheidung zur Anwendung der Nachbearbeitung fallabhängig getroffen werden. Um weiter zu untersuchen, wie gut die synthetischen Daten die Verteilung der realen CT-Daten erfassen, werden die generierten Daten mit dem t-Distributed Stochastic Neighbour Embedding (t-SNE) Algorithmus (vgl. Unterabschnitt 4.2.4) untersucht.

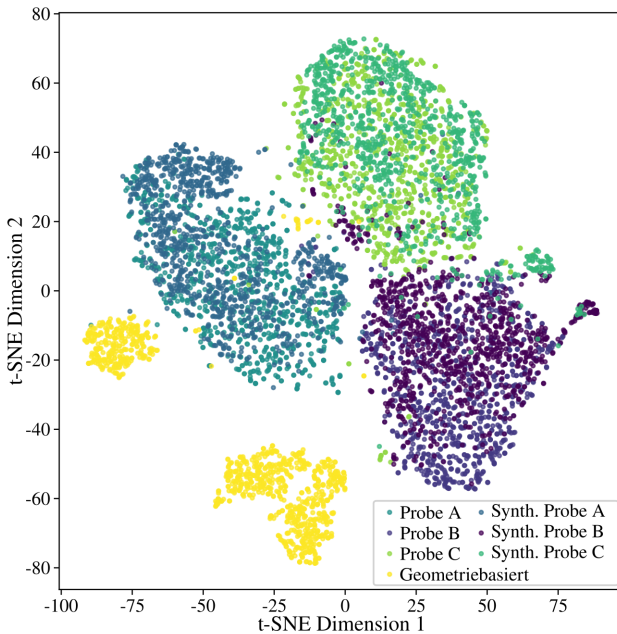


**Abbildung 4.18:** Einfluss der Watershed-Nachbearbeitung auf die Bildähnlichkeit (FID).

Für den vorliegenden Datensatz wurde die Kosinus-Distanzmetrik zur Messung der Ähnlichkeit der Datenpunkte verwendet. Vor der Anwendung von t-SNE wurde die Dimensionalität der Bilddaten durch Extraktion latenter Embeddings mittels des zuvor beschriebenen ResNet50-Modells, das auf dem MicroNet-Datensatz vortrainiert wurde, zunächst reduziert. Der Anteil der Watershed-verarbeiteten Bilder, die in die Analyse einbezogen wurden, wurde basierend auf der Konfiguration mit dem niedrigsten FID-Wert gewählt, wie in Abbildung 4.18 gezeigt. Die resultierenden Bild-Embeddings wurden mittels t-SNE visualisiert. Das Ergebnis ist in Abbildung 4.19 dargestellt. Jeder Punkt repräsentiert ein Graustufenbild. Die verschiedenen CT-Datensätze und ihre jeweiligen synthetischen Bilddaten werden in unterschiedlichen Farben dargestellt.

Wie in Abbildung 4.19 dargestellt, kann das Encoder-Netzwerk zwischen den verschiedenen CT-Datensätzen unterscheiden, die gut separierte Cluster bilden. Auffällig ist, dass die durch GAN generierten synthetischen Bilder eng mit den realen CT-Daten gruppiert sind, die sie nachbilden sollen, was einer starken Überlappung im Merkmalsraum entspricht. Dies deutet darauf hin, dass die synthetischen Daten die vollständige Verteilung der entsprechenden realen Datensätze effektiv abdecken. Im Gegensatz dazu bilden die algorithmischen Bilder (gelbe Punkte) klar getrennte Cluster, was die Realitätsnähe der GAN-basierten Generierung zusätzlich unterstreicht. Diese datengetriebenen Untersuchungen





**Abbildung 4.19:** t-SNE-Visualisierung der reduzierten Bild-Embeddings. Die GAN-generierten Bilder gruppieren sich eng mit ihren entsprechenden CT-Datensätzen, während die algorithmisch erzeugten Bilder des ersten Generierungsschritts klar getrennte Cluster bilden. Die Ähnlichkeit wurde mittels Kosinus-Distanz berechnet.

bestätigen, dass der entwickelte Generierungsprozess hochrealistische Bilddaten produziert, die eng mit den statistischen und strukturellen Eigenschaften realer CT-Scans übereinstimmen.

## 4.4.2 U-Net Finetuning

Auf Grundlage der generierten Trainingsdaten können überwachte Lernalgorithmen zur Segmentierung der CT-Daten trainiert werden. Aus Unterabschnitt 4.4.1 steht mit dem Generator  $F$  bereits ein Modell zur Verfügung, das Grauwert-CT-Daten in binäre Segmentierungen überführen kann. Dieser wurde jedoch im

Rahmen einer CycleGAN-Architektur trainiert und nicht explizit mit einer pixelweisen Verlustfunktion auf die Segmentierung von Grauwertbildern spezialisiert. Da  $F$  die Transformation *Grauwert*  $\rightarrow$  *Binär* bereits beherrscht, eignet er sich aber optimal für ein gezieltes Finetuning, um ihn zur Segmentierung einzusetzen.

## Trainingskonfiguration

Für das Finetuning wird der zuletzt gespeicherte Stand von  $F$  geladen. Als Trainingsdaten dienen 5 000 synthetisch erzeugte Bildpaare, die im vorherigen Schritt mithilfe des binären GAN-Generators und des Generators  $G$  erstellt wurden. Die binären Strukturen wurden zu 50 % mit dem Watershed-Algorithmus nachbearbeitet. Da  $F$  diese Daten zuvor nicht gesehen hat, ist ein Informationsleck (*data leakage*) ausgeschlossen. Um die Segmentierung des Netzwerks robuster zu gestalten, wurde zudem ein Gaußsches Rauschen mit einer Standardabweichung von 0,08 während des Trainings auf die Grauwertbilder aufgebracht.

Das Training selbst erfolgt mit einer reduzierten Lernrate von  $5e-5$  über 40 Epochen, um die vortrainierten Gewichte nur leicht anzupassen. Als Verlustfunktion wird eine gewichtete Kombination aus der Dice-Verlustfunktion  $\mathcal{L}_{\text{Dice}}$  und der L1-Verlustfunktion  $\mathcal{L}_{\text{L1}}$  verwendet:

$$\mathcal{L}_{\text{Seg}} = 0.8 \cdot \mathcal{L}_{\text{Dice}} + 0.2 \cdot \mathcal{L}_{\text{L1}}. \quad (4.8)$$

Hierbei optimiert  $\mathcal{L}_{\text{Dice}}$  insbesondere die Überschneidung der Segmentierung mit den Ground-Truth-Daten und somit die Form der Struktur, während  $\mathcal{L}_{\text{L1}}$  die absoluten Pixelunterschiede minimiert und so glatte, pixelgenaue Kanten begünstigt.

## Validierung

Die Modellgüte wird anhand von *Intersection over Union* (IoU) und *Dice-Koeffizient* bewertet. Obwohl beide Metriken stark korrelieren und in den Extremen (perfekte oder völlig falsche Segmentierung) identische Werte liefern, unterscheiden sie sich in der Gewichtung von Fehlern. IoU bestraft einzelne große Abweichungen stärker und ist damit empfindlicher gegenüber Ausreißern, während Dice die durchschnittliche Überschneidung betont und geringfügig robuster bei vereinzelt Fehlklassifikationen ist. Die Kombination beider Metriken ermöglicht daher eine differenziertere Bewertung der Segmentierungsgüte. IoU deutet dabei die Modellgüte für den schlechtesten Fall an während Dice die mittlere Modellgüte anzeigt.

Das Netz gibt für jeden Pixel  $x$  eine Wahrscheinlichkeit  $\hat{y}(x) \in [0, 1]$  aus. Für die Umwandlung in ein binäres Bild  $B$  muss also ein Schwellenwert  $s$  angewendet werden:

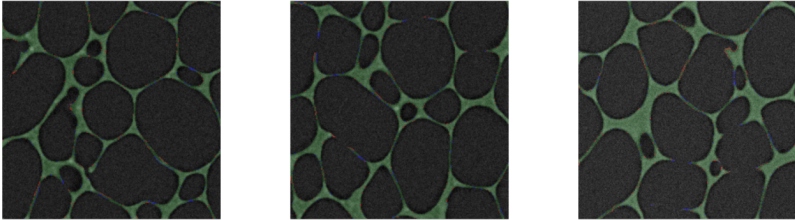
$$B(x) = \begin{cases} 1, & \hat{y}(x) \geq s \\ 0, & \hat{y}(x) < s. \end{cases} \quad (4.9)$$

Ein fester Wert von 0,5 ist dabei nicht immer optimal. Daher wird der Schwellenwert  $s$  zwischen 0,1 und 0,9 in Schritten von 0,01 variiert. Für jeden Wert wird die IoU und Dice auf den Validierungsdaten berechnet. Der beste Schwellenwert wird so identifiziert und für die Inferenz gespeichert. Die ermittelten Genauigkeiten des U-Nets auf den Validierungsdaten sind in Tabelle 4.5 gezeigt. Durch das Anpassen des Schwellenwerts konnten beide Metriken leicht auf IoU: 0,9574 und Dice: 0,9783 verbesserten werden. Allerdings erzielte das U-Net auch vorher bereits sehr hohe Genauigkeiten von IoU: 0,9568 und Dice: 0,9772.

**Tabelle 4.5:** Quantitative Ergebnisse des Finetunings.

Metrik	Standard-Schwellenwert	Optimaler Schwellenwert
IoU	0,9568	0,9574
Dice	0,9772	0,9783

Neben den Kennzahlen werden qualitative Beispiele betrachtet, um Stärken und Schwächen der Segmentierung zu veranschaulichen. Abbildung 4.20 zeigt für ausgewählte Validierungsbeispiele die Überlagerung der Grauwertbilder mit den Segmentierungen. In diesen sind korrekt segmentierte Gebiete grün, falsch-positive Bereiche rot und falsch-negative blau.



**Abbildung 4.20:** Überlagerung ausgewählter Validierungsbilder mit den Modellvorhersagen. Grün kennzeichnet korrekt segmentierte Bereiche, Rot falsch-positive und Blau falsch-negative.

Es zeigt sich, dass das Netz nur vereinzelt FP-Pixel aufweist. FN-Pixel treten häufiger in sehr dünnen oder schwach ausgeprägten Porenwänden auf.

Das Finetuning mit der kombinierten Dice–L1-Verlustfunktion erlaubt somit eine präzise Anpassung des vortrainierten Generators  $F$  an die Segmentierungsaufgabe. Die sehr hohen IoU- und Dice-Werte bestätigen die erfolgreiche Übertragung. Die qualitativen Beispiele zeigen jedoch, dass insbesondere an Übergangsbereichen zwischen Poren und Struktur gelegentlich Ungenauigkeiten auftreten. Insgesamt sind die Segmentierungen allerdings sehr genau und nur in Details fehlerhaft.

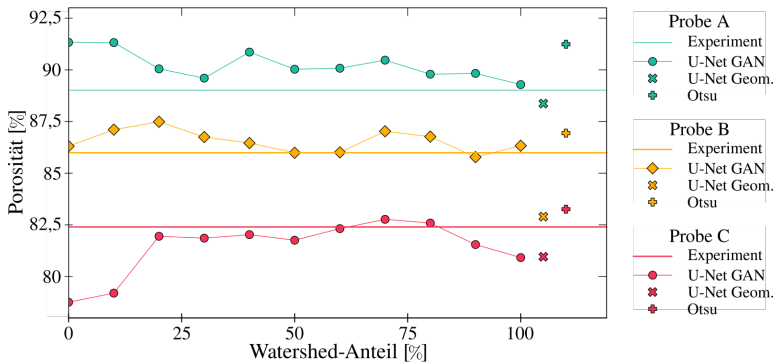
### 4.4.3 Experimentelle Untersuchung

Zur Quantifizierung der Segmentierungsgenauigkeit werden die aus den segmentierten CT-Daten abgeleiteten Porositäten mit den experimentell gemessenen Werten der entsprechenden Schaumproben verglichen (siehe Unterabschnitt 4.3.2).

Für die Segmentierung kommen mehrere über das in Unterabschnitt 4.4.2 beschriebene Finetuning trainierte U-Nets zum Einsatz.

Dabei wird der Einfluss der Watershed-basierten Nachbearbeitung in den Trainingsdaten untersucht, indem der Anteil der so verarbeiteten binären Strukturen in verschiedenen Trainingsdatensätzen variiert wird. Die so trainierten U-Nets segmentieren anschließend die CT-Daten, wodurch sich der Effekt des Watershed-Anteils direkt auf die Segmentierungsgenauigkeit bewerten lässt.

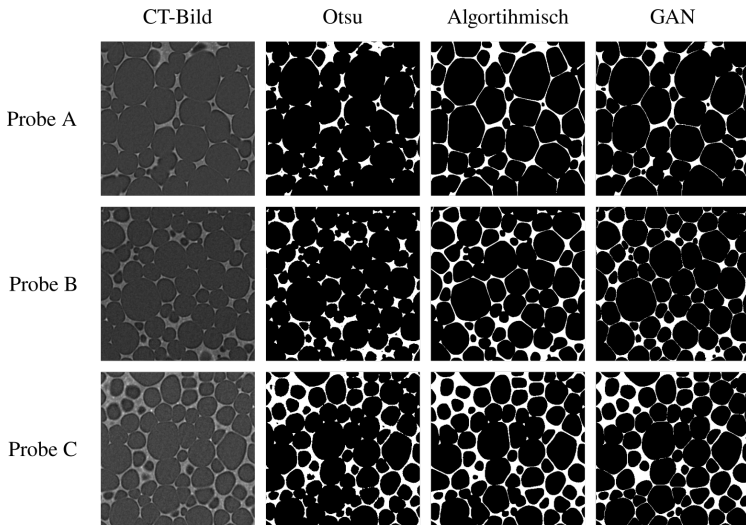
Abbildung 4.21 zeigt die Ergebnisse dieser Validierung. Unabhängig vom Datensatz übertreffen U-Nets, die auf GAN-generierten Daten trainiert wurden, jene, die auf geometriebasierten Daten basieren. Zudem führt ein höherer Watershed-Anteil in allen Fällen zu einer verbesserten Segmentierungsgenauigkeit.



**Abbildung 4.21:** Einfluss der Watershed-Nachbearbeitung auf die Segmentierungsgenauigkeit. Für jede Probe verbessern Watershed-verarbeitete Bilder die erreichte Segmentierungsgenauigkeit.

Zur qualitativen Veranschaulichung dieser Unterschiede zeigt Abbildung 4.22 beispielhafte Segmentierungsergebnisse.

Das auf GAN-Daten trainierte U-Net erkennt feine Strukturen und dünne Zellwände präziser, während das auf algorithmisch generierten Daten trainierte Modell



**Abbildung 4.22:** Einfluss der Watershed-Nachbearbeitung auf die Segmentierung der verfügbaren CT-Bilder. Für jeden Datensatz werden die Ergebnisse der Otsu-Schwellemnennungsmethode, eines U-Nets, das auf algorithmischen (Schritt 1) Daten trainiert wurde, sowie eines U-Nets, das auf GAN-basierten (Schritt 2) Daten trainiert wurde, dargestellt.

zu einer Übersegmentierung der Wandstärken neigt, was in einer Unterschätzung der Porosität resultiert. Diese Beobachtung stimmt mit den quantitativen Ergebnissen überein und unterstreicht den Nutzen GAN-basierter, synthetischer Trainingsdaten.

## 4.5 Geometrische Charakterisierung

Auf Basis des rekonstruierten digitalen Zwillings sollen in dieser Arbeit Struktureigenschaft-Beziehungen ermittelt werden. Da die rekonstruierte Schaumprobe nur einen Ausschnitt des untersuchten Schaums darstellt, muss zunächst sichergestellt werden, dass die darin ermittelten Struktureigenschaften repräsentativ sind. Zu diesem Zweck wird eine RVE-Analyse (siehe Abschnitt 2.4) durchgeführt. Ziel der RVE-Analyse ist zum einen die Prüfung, ob die gewählte CT-Probe

genügend Informationen über die Mikrostruktur des Schaummaterials aufweist, und zum anderen die Bestimmung der minimalen Subvolumengröße, ab der die gemessenen Eigenschaften als repräsentativ gelten. Auf diese Weise lässt sich die Größe der zu analysierenden Materialausschnitte und somit der Rechenaufwand auf ein Minimum reduzieren.

Die Bestimmung der RVEs erfolgt über den in Abschnitt 2.4 beschriebenen deterministischen Ansatz. Aus den rekonstruierten digitalen Zwillingen werden kubische Subvolumina mit sukzessiv steigender Kantenlänge ausgeschnitten und ihre Struktureigenschaften bestimmt. Für die vorliegenden Schaumproben wird die Kantenlänge  $L$  der Subvolumina für jede Messung um  $27,5\text{ }\mu\text{m}$  bzw. 10 Voxel erhöht, beginnend bei 50 Voxel. Dies ergibt insgesamt 55 mögliche Messpunkte.

Um zu ermitteln, ob ein RVE erreicht ist, wird die prozentuale Abweichung von den Eigenschaften der Gesamtstruktur herangezogen. Diese Abweichung wird über

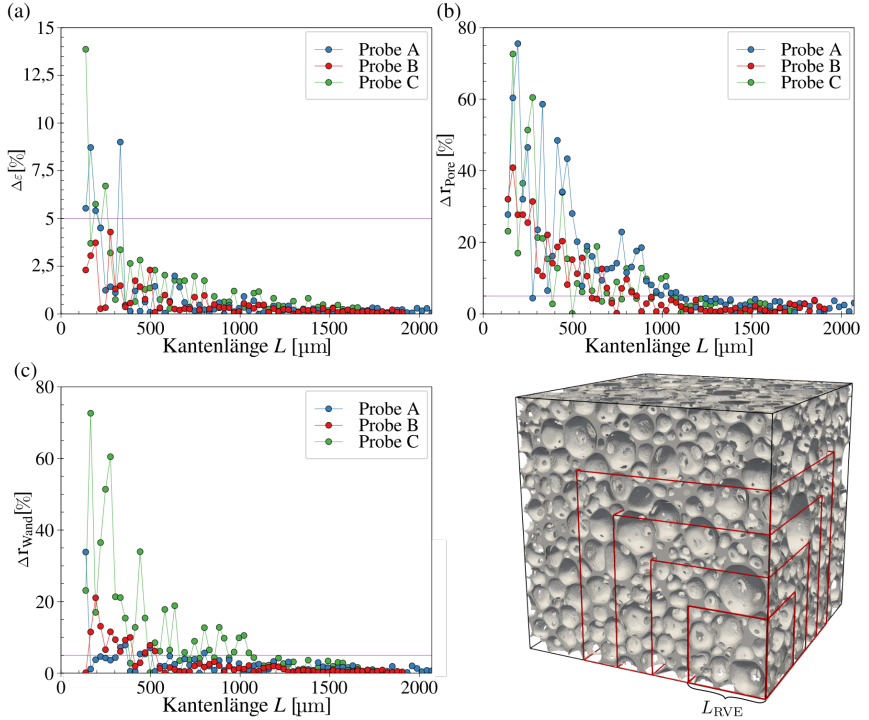
$$\Delta r_{VE} = \left| \frac{\kappa_{\text{Gesamt}} - \kappa_{VE}}{\kappa_{\text{Gesamt}}} \right| * 100\%, \quad (4.10)$$

berechnet.

Hierbei steht  $\kappa_{\text{Gesamt}}$  für die im Gesamtvolumen und  $\kappa_{VE}$  für die im Subvolumen gemessenen Struktureigenschaften. Unterschreitet  $\Delta r_{VE}$  den Schwellenwert von 5 %, gilt die entsprechende Kantenlänge  $L_{RVE}$  als RVE-Größe.

Da die Größe des RVEs von der untersuchten Eigenschaft abhängt, wurden für jede Schaumprobe RVE-Analysen für die Porosität  $\varepsilon$ , den mittleren Porenradius  $r_p$  und die mittlere Wanddicke  $r_w$  durchgeführt.

Für die Porosität (a) wird die 5 %-Grenze bereits bei einer Kantenlänge von  $300\text{ }\mu\text{m}$  (110 Voxel) unterschritten. Porenradius (b) und Wanddicke (c) erreichen erst ab  $1100\text{ }\mu\text{m}$  (400 Voxel) ein RVE. In allen Fällen ist eine asymptotische Annäherung an den Gesamtvolumenwert erkennbar. Es kann dementsprechend davon ausgegangen werden, dass die CT-Probe groß genug ist, um repräsentativ für den gesamten Schaum zu sein.



**Abbildung 4.23:** RVE Analyse der Struktureigenschaften. Für die Porosität a), den Porenradius b) und die Wandstärke c) werden unterschiedliche RVE-Größen identifiziert.

Für die weiteren Strukturanalysen wird eine Kantenlänge von 1 100  $\mu\text{m}$  (400 Voxel) gewählt, da ab dieser Größe für alle untersuchten Eigenschaften ein RVE erreicht wird. Die so ermittelten Werte sind als effektive Kenngrößen der jeweiligen Struktur interpretierbar.

Basierend auf den festgelegten RVE-Größen wurden die in Tabelle 4.6 dargestellten Struktureigenschaften der rekonstruierten digitalen Zwillinge bestimmt.



**Tabelle 4.6:** Ergebnisse der Strukturanalysen. Werte in  $\mu\text{m}$  und Voxel bei einer Voxelgröße von  $2,75 \mu\text{m}/\text{Voxel}$ .

	Porosität [%]	Porenradius [ $\mu\text{m}$ (Voxel)]	Wanddicke [ $\mu\text{m}$ (Voxel)]
<i>Probe A</i>	88,55	47,39 (17,2)	7,89 (2,9)
<i>Probe B</i>	83,50	43,77 (15,9)	8,45 (3,1)
<i>Probe C</i>	79,06	41,18 (15,0)	8,03 (2,9)

## 4.6 Einbindung in Kadi4Mat

Die in diesem Kapitel entwickelten Methoden zur Rekonstruktion und geometrischen Charakterisierung von CT-Daten wurden in Form von Workflows in KadiStudio modelliert, um eine reproduzierbare Anwendung der Methoden zu ermöglichen. Die Workflows wurden bewusst generisch gestaltet, sodass sie mit minimalen Anpassungen auch auf andere Anwendungsfälle übertragbar sind.

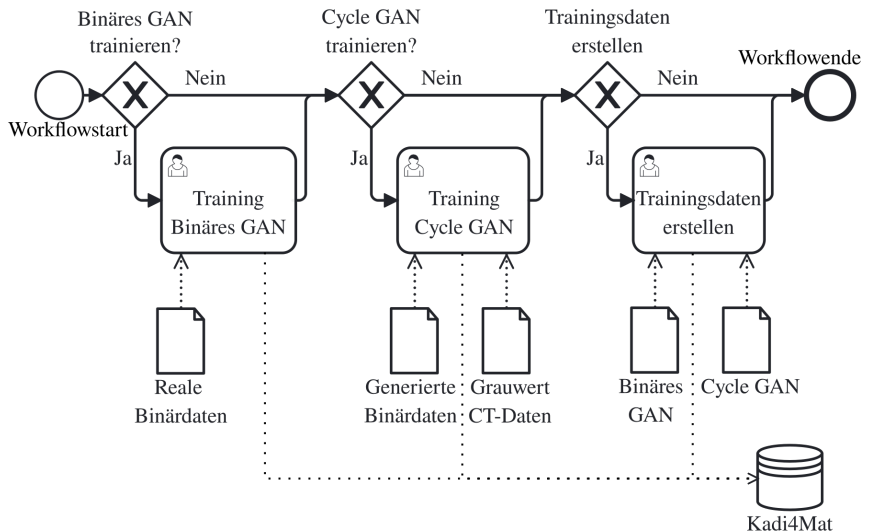
Neben der Prozessmodellierung wird auch die Datendokumentation in Kadi4Mat integriert. Jede untersuchte Struktur wird dabei als eigener Record in KadiWeb modelliert. Dieses Schema ermöglicht es, sämtliche relevanten Informationen einer Probe Schritt für Schritt zu ergänzen. Während nach der Rekonstruktion zunächst grundlegende Informationen wie Hersteller, Dichte oder Probenname erfasst werden, erweitert sich die Metadatenbeschreibung nach der Charakterisierung um Eigenschaften wie Porosität, mittlere Porengröße oder Wandstärke. Auf diese Weise wachsen die Metadaten inkrementell mit den durchgeführten Analysen an und bilden die Grundlage für eine konsistente, FAIR dokumentierte Strukturdatenbank. Im Folgenden werden beide Workflows im Detail beschrieben.

### 4.6.1 Rekonstruktion

Der in KadiStudio implementierte Workflow zur Rekonstruktion von CT-Daten bildet die in Abschnitt 4.4 beschriebenen Verarbeitungsschritte ab. Er ermöglicht sowohl die Generierung synthetischer Trainingsdaten als auch das Training bzw.

Finetuning eines U-Nets zur Segmentierung. Die Umsetzung erfolgt modular. Für jedes Arbeitspaket von der Datenerzeugung über das Netztraining bis zur Anwendung des trainierten Modells existiert ein eigener Unterprozess im Gesamtworkflow, der je nach Anwendungsfall individuell ausgeführt werden kann.

Abbildung 4.24 zeigt den Workflow zur Generierung künstlicher Trainingsdaten. Der vollständige KadiStudio-Workflow ist publiziert und frei verfügbar [153].

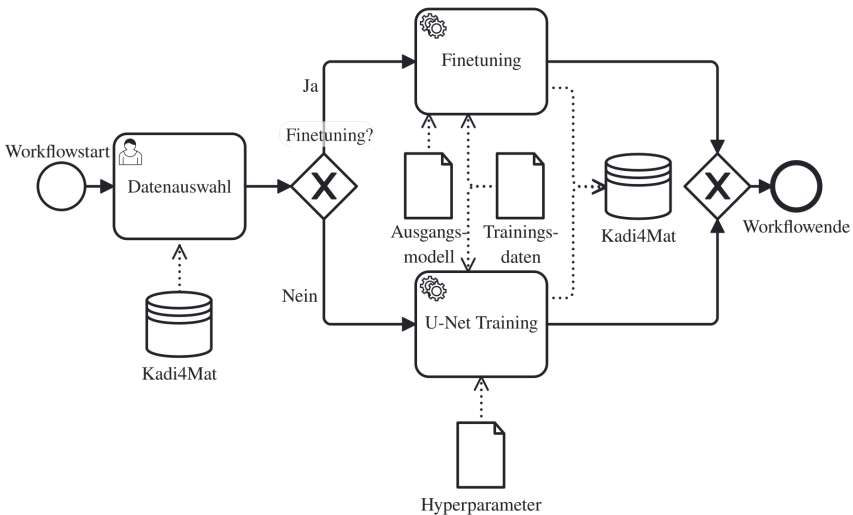


**Abbildung 4.24:** Workflow zur Generierung künstlicher Trainingsdaten. Die Erstellung der Trainingsdaten benötigt zwei neuronale Netzwerke – Binary GAN und CycleGAN. Das Training dieser Netzwerke ist ebenfalls im Workflow implementiert. Welche Subprozesse des Workflows ausgeführt werden, wählt der Nutzer.

Abhängig von der Nutzerauswahl implementiert der Workflow zur Datengenerierung unterschiedliche Funktionen. Im simpelsten Fall können direkt Trainingsdaten erzeugt werden. Dazu werden zwei neuronale Netze benötigt: ein GAN zur Generierung von Binärdaten und ein CycleGAN zur Überführung der Binärdaten zu Grauwertbildern.

Sollten diese Netze nicht verfügbar sein, bietet der Workflow die Möglichkeit diese Netzwerke zu trainieren. Für das binäre GAN werden dazu realistische Binärdaten benötigt und für das CycleGAN, neben den Binärdaten, Grauwertbilder, die imitiert werden sollen.

Die generierten Trainingsdaten können anschließend im Trainingsworkflow genutzt werden, um ein U-Net für die Segmentierung zu trainieren. Abbildung 4.25 zeigt den Workflow schematisch. Der Nutzer hat dabei die Möglichkeit, entweder

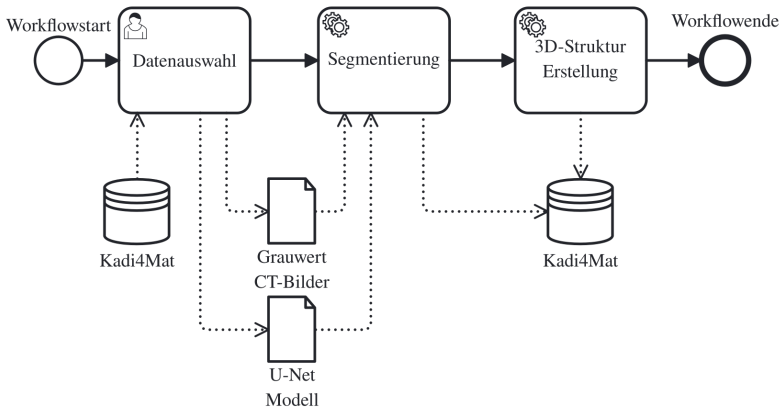


**Abbildung 4.25:** Workflow zum Training eines Segmentierungsnetzwerks in Form eines U-Nets. Der Workflow benötigt Trainingsdaten bestehend aus Grauwertbildern und binären Ziellabeln. Das trainierte U-Net wird abschließend auf Kadi4Mat hochgeladen.

ein bestehendes Netzwerk mittels Finetuning anzupassen oder ein neues U-Net mit frei wählbaren Hyperparametern zu trainieren. Nach Abschluss des Trainings wird das Modell exportiert und auf Kadi4Mat gespeichert, sodass es im Segmentierungsworflow verfügbar ist.

Der eigentliche Segmentierungsworflow verarbeitet CT-Bilder mithilfe des trainierten U-Nets. Er ist in Abbildung 4.26 dargestellt und erzeugt segmentierte

CT-Bilder, die zu einer dreidimensionalen Struktur zusammengesetzt werden. Auch diese Ergebnisse werden automatisiert auf Kadi4Mat abgelegt.



**Abbildung 4.26:** Workflow zur Segmentierung. Neben einem trainierten U-Net Modell benötigt der Workflow die zu segmentierenden Grauwert CT-Bilder. Die segmentierten Bilder zu einer dreidimensionalen Struktur zusammengesetzt. Die segmentierten Bilder und die dreidimensionale Struktur werden auf Kadi4Mat hochgeladen.

Jede rekonstruierte Struktur wird zudem automatisch mit grundlegenden Metadaten versehen. Dazu gehören in dieser frühen Phase der Prozesskette vor allem die Größe der Struktur. Weitere Felder wie etwa Hersteller, Material oder effektive Dichte werden vorbereitet, müssen aber manuell ergänzt werden, da sie im Rekonstruktionsschritt selbst nicht ableitbar sind.

Für jede rekonstruierte Struktur wird in KadiWeb ein eigener Record angelegt, der diese Informationen enthält. Auf diese Weise wird jede untersuchte Probe eindeutig dokumentiert und kann im Verlauf der Arbeit sukzessive um weitere Metadaten ergänzt werden.

Neben den Metadaten verknüpft der Workflow die erzeugten Struktur-Records direkt mit den verwendeten Workflows und Modellen. Auf diese Weise ist die Datenherkunft jederzeit nachvollziehbar. Abbildung 4.27 zeigt exemplarisch einen solchen Record. Die zugehörigen Verlinkungen sind in Abbildung 4.30 dargestellt.

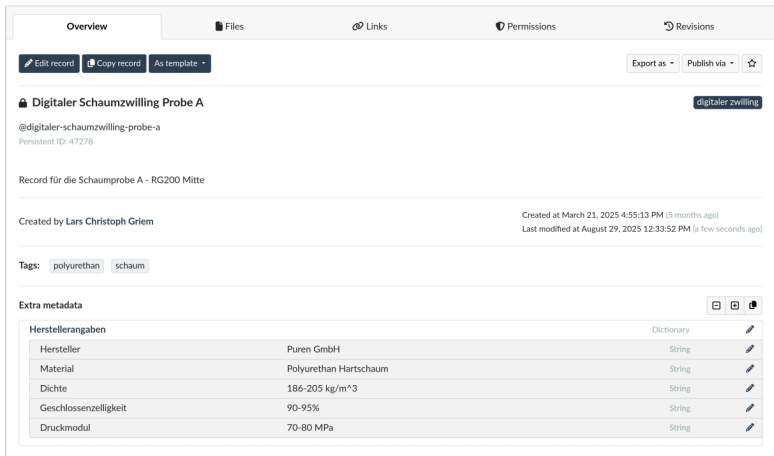


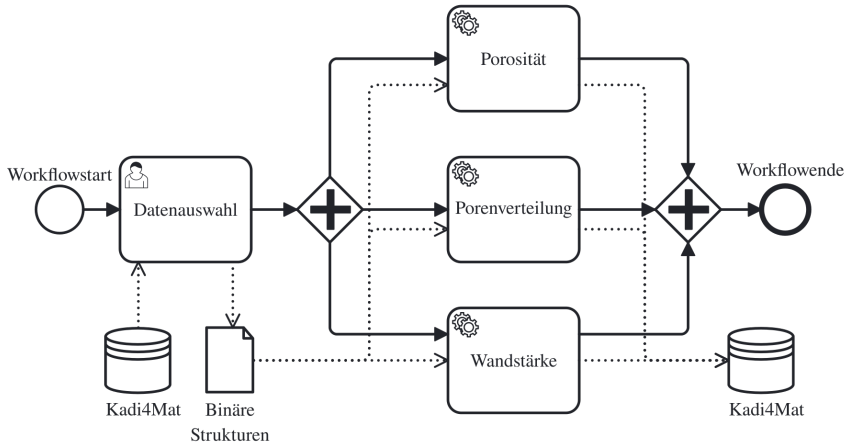
Abbildung 4.27: Record der rekonstruierten Schaumprobe. Die Herstellerangaben stellen die Basis-metadaten des Materials dar.

## 4.6.2 Charakterisierung

Die Bestimmung der Struktureigenschaften von digitalen Materialzwillingen wurde im Zuge dieser Arbeit in Form eines KadiStudio Workflows automatisiert.

Der Workflow ist modular aufgebaut und bildet die in Abbildung 4.28 skizzierten Verarbeitungsschritte ab. Jede zu bestimmende Strukturgröße wird in einem eigenen Unterprozess behandelt, sodass der Nutzer individuell auswählen kann, welche Eigenschaften ermitteln lassen möchte. Die Umsetzung erfolgte bewusst generisch, um eine Übertragbarkeit auf andere Materialsysteme zu ermöglichen. Der eigentliche KadiStudio-Workflow ist publiziert und frei zugänglich [153].

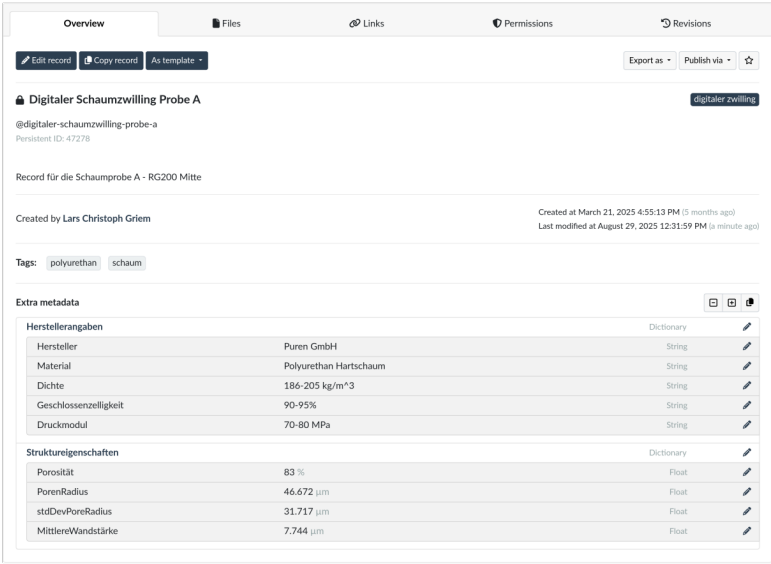
In der vorliegenden Arbeit sind Subprozesse für die Bestimmung von Wanddicke, Porengröße, Porosität und Strukturoberfläche implementiert. Eine Erweiterung des Workflows um zusätzliche Kennwerte ist jederzeit möglich. Als Eingabe können sowohl lokal vorliegende Strukturen als auch bereits in Kadi4Mat abgelegte Datensätze genutzt werden. Die berechneten Werte werden nach Abschluss der Analyse automatisch auf Kadi4Mat hochgeladen.



**Abbildung 4.28:** Workflow zur Strukturanalyse. Jede Strukturgröße wird in einem eigenen Unterprozess ermittelt. Der Nutzer entscheidet welche Werte ermittelt werden sollen. Die ermittelten Werte werden auf Kadi4Mat dokumentiert.

Für jede untersuchte Struktur wird dabei auf den bestehenden Record aus der Rekonstruktion zurückgegriffen, der durch die neuen Informationen ergänzt wird. Gemäß dem in Unterabschnitt 3.3.1 beschriebenen Konzept der inkrementellen Metadaten wachsen die Records so mit jedem Arbeitsschritt an. Darüber hinaus wird jeder aktualisierte Record automatisch mit den verwendeten Workflows und Modellen verknüpft, sodass die gesamte Datenherkunft und Prozesskette nachvollziehbar dokumentiert ist.

Abbildung 4.29 zeigt exemplarisch einen solchen Record, in dem die berechneten Strukturparameter maschinenlesbar in den Metadaten abgelegt sind. Zudem sind die Verlinkungen zu den verwendeten Methoden und Modellen dargelegt. Diese klare Strukturierung erleichtert in späteren Schritten, insbesondere in Kapitel 7, die Analyse und Verknüpfung der Strukturdatenbank.



**Abbildung 4.29:** Dokumentation der Charakterisierung der rekonstruierten CT-Daten. Die Strukturparameter sind in den Metadaten, maschinenlesbar abgelegt.

## 4.7 Diskussion

In diesem Kapitel wurde der erste Schritt der computergestützten Materialentwicklung umgesetzt: die Rekonstruktion und Charakterisierung digitaler Zwillinge auf Grundlage hochauflösender CT-Daten. Die entwickelten Workflows sind vollständig in Kadi4Mat eingebunden, sodass sämtliche Prozessschritte und Ergebnisse automatisch dokumentiert werden und eine nachvollziehbare sowie wiederverwendbare Datenbasis entsteht.

Die vorgestellten Methoden zeigen, dass durch synthetische Datengenerierung und Finetuning eines U-Nets hochgenaue Segmentierungen von CT-Aufnahmen geschlossenzelliger Schäume möglich sind. Mit IoU- und Dice-Werten von über 0,95 wurde eine Segmentierungsqualität erreicht, die sowohl für geometrische Analysen als auch für die Ableitung von Struktur-Eigenschafts-Beziehungen

geeignet ist. Die Validierung über experimentell bestimmte Porositäten bestätigt die Übertragbarkeit auf reale Daten. Damit stellt die Methodik eine robuste und effiziente Alternative zu Schwellwertmethoden dar.

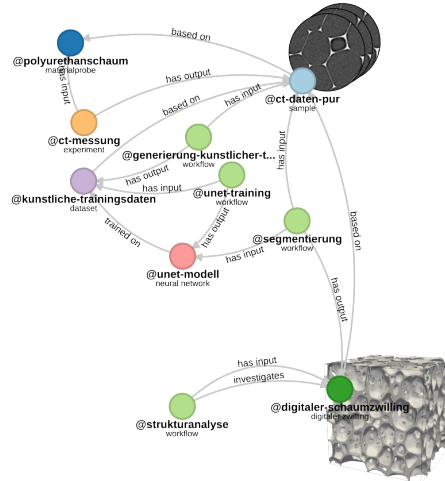
Trotz dieser Stärken bestehen geringe Unsicherheiten. An Übergangsbereichen zwischen Poren und Struktur können sehr dünne Zellwände lokal falsch segmentiert werden. Diese Abweichungen sind jedoch klein und haben im vorliegenden Anwendungsfall keinen erkennbaren Einfluss auf die bestimmten mikrostrukturellen Eigenschaften wie Porosität, Porenradius oder Wandstärke. Sollte in anderen Materialien oder Anwendungen eine höhere Sensitivität gegenüber solchen Abweichungen erforderlich sein, bietet die Anbindung leistungsstarker, domänenübergreifender Segmentierungsansätze wie Cellpose [154] oder Segment Anything [155] eine vielversprechende Erweiterung. Für die hier untersuchten Daten erwies sich hingegen das Finetuning des U-Nets als hinreichend präzise und zugleich methodisch besser in den Workflow integrierbar.

Darüber hinaus konzentrierte sich die Charakterisierung in dieser Arbeit auf grundlegende Kenngrößen wie Porosität, mittlere Porengröße und Wandstärke. Diese Größen erfassen wesentliche geometrische Aspekte der untersuchten Strukturen, bilden jedoch nicht die gesamte Vielfalt möglicher Merkmale ab. Für künftige Anwendungen könnten zusätzliche Strukturgrößen wie Tortuosität oder Konnektivität berücksichtigt werden, um ein umfassenderes Bild der Mikrostruktur zu erhalten.

Zusammenfassend belegt dieses Kapitel, dass durch die vorgestellten Methoden eine vollständige, reproduzierbare Prozesskette von der CT-Aufnahme über rekonstruierte digitale Zwillinge bis hin zur strukturellen Charakterisierung aufgebaut werden kann. Die erzielten Ergebnisse bestätigen die Eignung synthetischer Trainingsdaten für die Segmentierung hochauflösender CT-Daten und unterstreichen den Nutzen eines durchgängigen Forschungsdatenmanagements. Gleichzeitig eröffnen sich klare Perspektiven für weiterführende Arbeiten, die insbesondere die Übertragbarkeit auf weitere Materialsysteme und zusätzliche Strukturmerkmale adressieren sollten.



Der erreichte Stand der Methodik sowie der aktuelle Fortschritt im Materialentwicklungsprozess sind in Abbildung 4.30 dargestellt. Ausgehend von CT-Daten



**Abbildung 4.30:** Zusammenfassung der Strukturrekonstruktion und -analyse auf Kadi4Mat. Ausgehend von CT-Daten wird ein U-Net auf Grundlage künstlicher Trainingsdaten trainiert. Dieses U-Net dient zur Segmentierung der CT-Daten.

realer Schaumstrukturen werden digitale Zwillinge über ML-gestützte Segmentierung rekonstruiert, geometrisch charakterisiert und in strukturierter Form auf Kadi4Mat abgelegt. Die so erzeugte Datenbasis ist nicht nur reproduzierbar und erweiterbar, sondern auch unmittelbar für die in den folgenden Kapiteln behandelten Generierungs- und Simulationsansätze nutzbar.

## 5 ML-gestützte Generierung digitaler Zwillinge

Die Rekonstruktion realer Mikrostrukturen aus CT-Daten (Abschnitt 4.4) liefert detailgetreue digitale Zwillinge bestehender Materialien. Um belastbare Struktur-Eigenschaft-Beziehungen ableiten zu können, wird jedoch eine möglichst diverse Strukturdatenbank benötigt, die ein breites Spektrum unterschiedlicher Morphologien abdeckt. Da sich eine solche Vielfalt nicht ausschließlich durch Rekonstruktionen realer Strukturen erreichen lässt, müssen digitale Zwillinge gezielt in ihren geometrischen Eigenschaften variiert und künstlich generiert werden.

Im Rahmen der in dieser Arbeit entwickelten Methodik wird daher die Generierung künstlicher digitaler Zwillinge von Mikrostrukturen untersucht. Ziel ist es, mithilfe unterschiedlicher Ansätze eine Methode zu entwickeln, die Mikrostrukturen systematisch variiert und kontrolliert Parameter wie Porengröße, Porosität oder Wanddicke verändert, um so ein breites morphologisches Spektrum zu erzeugen. Dies bildet die Grundlage für die spätere Analyse und das Ableiten von Struktur-Eigenschaft-Beziehungen.

Zunächst werden etablierte Ansätze aus der Literatur vorgestellt, die sich in drei Hauptkategorien einteilen lassen: geometriebasierte, physikbasierte und ML-basierte Methoden. Anschließend werden die im Rahmen dieser Arbeit verfügbaren geometriebasierten Algorithmen auf ihre Eignung zur Modellierung realer Schaumstrukturen untersucht. Darauf aufbauend wird ein eigener ML-basierter Generierungsansatz entwickelt, der unabhängig vom Anwendungsfall eingesetzt und umfassend validiert wird. Abschließend werden die Verfahren in Form von Workflows automatisiert, um ihre Anwendung zu erleichtern.

## 5.1 Stand der Forschung

Neben der in Abschnitt 4.4 vorgestellten Rekonstruktion aus bildgebenden Verfahren stellt die künstliche Generierung digitaler Abbilder poröser Mikrostrukturen eine weit verbreitete Methode dar, die in zahlreichen Anwendungsgebieten eingesetzt wird. Grundsätzlich lassen sich drei Ansätze unterscheiden: Geometriebasierte Methoden, die die Mikrostruktur mithilfe mathematischer oder konstruktiver Modelle erzeugen, physikbasierte Methoden, die den physikalischen Entstehungsprozess des Materials nachbilden sowie ML-basierte Methoden, die auf generativen neuronalen Netzen basieren und die Verteilung realer Mikrostrukturen aus Beispieldaten erlernen. Allen Ansätzen ist gemeinsam, dass sie prinzipiell eine gezielte Einstellung der geometrischen Eigenschaften der resultierenden Mikrostruktur ermöglichen. Sie unterscheiden sich jedoch deutlich hinsichtlich ihres Realismus, ihres Rechenaufwands sowie der Flexibilität bei der Anpassung an neue Zielstrukturen. Da in dieser Arbeit sowohl geometriebasierte als auch ML-basierte Generierungsmethoden betrachtet und deren Ergebnisse anhand einheitlicher Kriterien bewertet werden, bildet der nachfolgende Überblick über den Stand der Forschung bei den genannten Ansätzen die Grundlage für die spätere vergleichende Analyse.

### 5.1.1 Geometriebasierte Generierung

Bei der geometriebasierten Generierung wird die Mikrostruktur auf Basis geometrischer Konstruktionsmethoden erzeugt. Ein etabliertes Verfahren ist die Verwendung von Voronoidiagrammen, die in [156] ausführlich beschrieben werden und sich bereits als geeignetes Modell für poröse Mikrostrukturen erwiesen haben [157–160]. Bei diesen Verfahren wird ein dreidimensionaler Raum in Zellen unterteilt, die jeweils den Raumanteil beschreiben, der einem zufällig platzierten Voronoi Punkt am nächsten liegt. Alsayednoor & Harrison [161] konnten zeigen, dass eine rein zufällige Platzierung der Voronoi Punkte häufig anisotrope Strukturen erzeugt. Durch Erhöhung der Irregularität der Punktverteilung lässt sich dieser Effekt zwar reduzieren, jedoch entstehen dabei häufig unnatürlich geformte

Zellen. Eine Möglichkeit zur Umgehung unrealistisch geformter Zellen ist der *Lloyd-Algorithmus* [162], der die Voronoiunkte iterativ in den Schwerpunkt ihrer Zellen verschiebt und so zentroidale, gleichmäßigere Strukturen erzeugt. Für eine gezielte Einstellung der Zellgrößenverteilung kann zudem eine *Laguerre-Voronoi-Zerlegung* verwendet werden, bei der jedem Punkt ein Gewicht zugeordnet wird. Dieses Gewicht vergrößert das Einzugsgebiet der entsprechenden Voronoi-zelle [163].

Voronoi-basierten Methoden bieten den Vorteil, dass die geometrischen Eigenschaften der resultierenden Strukturen über Parameter wie Punktzahl, Gewichtsverteilung oder Lloyd-Iterationen gezielt eingestellt werden können. Aufgrund ihrer Konstruktion sind die entstehenden Schaumstrukturen jedoch meist unrealistisch geformt und weisen scharfe Kanten auf, die abgerundet werden müssen, um realistischere Schaumstrukturen zu erhalten.

In der Simulationsumgebung PACE3D sind entsprechende Voronoi-basierte Generierungsalgorithmen bereits implementiert und wurden erfolgreich in unterschiedlichen Anwendungsgebieten eingesetzt. Altschuh [97] verwendet sie zur Generierung realistischer Membranstrukturen, während Reder & Holland-Cunz et al. [164] sie zur Modellierung offenzelliger Schaumstrukturen einsetzen. In einer Arbeit von August et al. [165] werden sie zudem zur Generierung offenporiger Metallschäume für den Einsatz in Wärmetauschern verwendet. Darüber hinaus kommen ähnliche Verfahren auch zur Modellierung von Kornstrukturen in metallischen Werkstoffen zum Einsatz [39, 166].

Neben voronoi-basierten Generierungsalgorithmen können zur Erstellung poröser Mikrostrukturen Schüttungsalgorithmen verwendet werden, die aus definierten Partikeln eine Gesamtstruktur zusammensetzen. Diese Algorithmen platzieren dazu zunächst Partikel in einem dreidimensionalen Raum. Diese Platzierung erfolgt gewöhnlich zufällig oder die Partikel werden in das Volumen gefüllt und anschließend über *Drop and Roll* Algorithmen in eine stabile Position gebracht. Diese rein geometrische Platzierung der Partikel ist sehr schnell, allerdings sind die entstehenden Mikrostrukturen nicht realistisch und können nur bedingt für

Simulationen verwendet werden. Daher erfolgt nach der geometrischen Platzierung in aktuellen Algorithmen die Simulation der Verdichtung der Packung durch Schütteln oder Pressen mit Hilfe eines Physiklösers [167]. Diese simulative Anpassung der Mikrostruktur erhöht nicht nur die Packungsdichte, sondern auch die Ähnlichkeit zu realen Mikrostrukturen. Solche Packungsalgorithmen sind in der PACE3D Simulationsumgebung implementiert und wurden bereits erfolgreich zur Simulation geologischer Strukturen [168, 169] angewendet.

### 5.1.2 Physikbasierte Generierung

Im Gegensatz zur geometriebasierten Generierung, bei der die Mikrostruktur explizit modelliert wird, ahmt die physikbasierte Generierung den physikalischen Entstehungsprozess des Materials nach. Dadurch können besonders realistische Strukturen mit organischen Formen und variablen Porengeometrien erzeugt werden. Gleichzeitig erlaubt dieser Ansatz, den Einfluss von Prozessparametern auf die resultierende Mikrostruktur gezielt zu untersuchen.

Ein etablierter Ansatz ist die Phasenfeldmethode, bei der die Mikrostruktur durch die Minimierung eines freien Energiefunktionalen evolviert. Diese Methode löst die relevanten Transport- und Grenzflächengleichungen numerisch auf und kann so Prozesse wie Diffusion, Konvektion oder kapillargetriebene Formänderungen abbilden [170]. Im Kontext poröser Materialien wird dies häufig eingesetzt, um die Bildung und Alterung von Schäumen zu simulieren. Dabei wird eine initiale Schaumstruktur über eine definierte Zeitspanne evolviert, wodurch sich ihre Geometrie infolge von Koaleszenz und Oswaldreifung verändert [171]. Vakili et al. [172] nutzen diesen Ansatz zur Generierung metallischer Schäume unter Berücksichtigung von Schaumalterung und gekoppelter Flüssigkeitsströmung. Dazu werden initiale Keimporen des betrachteten Materials simuliert und ihre Vergrößerung und Koaleszenz berücksichtigt. Da durch dieses Vorgehen die reale Schaumformierung nachempfunden wird, entstehen dabei realitätsnahe Schaumstrukturen mit unregelmäßigen Blasengrößen und Wanddicken sowie organischen Blasenformen. Mithilfe der simulativen Anpassung lässt sich bei dieser Methode

die Konnektivität der Poren sowie ihre Größenverteilung gezielt anpassen. Zudem können die Einflüsse der Generierungsparameter auf die Mikrostruktur direkt untersucht werden. Ein weiterer simulativer Ansatz ist die *spinodale Entmischung*, basierend auf dem Cahn-Hilliard-Ansatz. Hierbei wird die Entmischung einer binären Flüssigkeitsmischung simuliert, was zu einer charakteristischen vernetzten Porenstruktur führt. Dieses Verfahren wird unter anderem zur Generierung von Polymermembranen eingesetzt, bei denen der Einfluss der Prozessparameter auf die resultierende Mikrostruktur gezielt untersucht werden kann [170]. Auch Sinterprozesse lassen sich simulativ nachbilden, um die Mikrostruktur gesinterter Materialien zu erzeugen [173].

Die physikbasierte Generierung lässt sich in einer Vielzahl von Anwendungsgebieten verwenden und liefert in der Regel realistischere Mikrostrukturen als die rein geometrische Modellierung. Allerdings ist sie mit hohem Rechenaufwand und entsprechend langen Generierungszeiten verbunden, sodass der Aufbau einer umfangreichen Strukturdatenbank deutlich zeitintensiver ist.

### 5.1.3 ML-basierte Generierung

Eine weitere Möglichkeit zur Generierung digitaler Zwillinge von Mikrostrukturen ist der Einsatz generativer neuronaler Netze. Diese Modelle haben sich in den letzten Jahren als leistungsfähige Werkzeuge etabliert und können die Verteilung realer Mikrostrukturen erlernen, um daraus neue, synthetische Strukturen zu erzeugen. Gegenüber geometriebasierten und physikbasierten Methoden bieten sie den Vorteil, dass nach abgeschlossenem Training neue Mikrostrukturen schnell und mit geringem Rechenaufwand generiert werden können.

Die Generierung erfolgt dabei in der Regel, indem ein niedrigdimensionaler Eingabevektor, häufig reines Rauschen, über einen Generator in eine vollständige Struktur überführt wird. Die wichtigsten Architekturansätze sind im Folgenden beschrieben. Bei diesen handelt es sich um *Variational Autoencoder* (VAE), *Generative Adversarial Networks* (GAN) und *Denoising Diffusion Probabilistic Models*.

Die Generierung poröser Mikrostrukturen mit Hilfe von VAEs wurde bereits in einer vorangegangenen Arbeit untersucht [174]. VAEs generieren neue Strukturen, indem sie Punkte aus einem latenten Raum auswählen und daraus mit einem Generator oder Decoder eine vollständige Mikrostruktur erzeugen. Der Vorteil eines VAEs ist, dass aus dem latenten Raum neue Strukturen erzeugt werden können, deren Eigenschaften gezielt eingestellt werden können. Zhao et al. [174] verwenden dazu eine Bayes'sche Optimierung. Ein Nachteil der VAE Architektur ist hingegen die meist schlechte Qualität der generierten Strukturen. Diese sind häufig verschwommen und weisen keine definierten Umrisse auf. In letzter Zeit werden VAEs daher kaum noch für generative Zwecke eingesetzt und dienen häufiger der Gruppierung und Klassifizierung von Strukturen.

Im Gegensatz dazu erzeugen GAN-Generatoren hochaufgelöste, realistische Strukturen mit klaren Umrisse. Daher wurden sie bereits kurz nach ihrer Einführung [19] erfolgreich zur Generierung zweidimensionaler Abbilder poröser Mikrostrukturen eingesetzt [175]. Weitere Entwicklungen von GANs konzentrieren sich darauf, die Morphologie der generierten Strukturen gezielt einzustellen. Der Generator erhält dabei zusätzliche Informationen, die die Morphologie quantifizieren oder er wird durch *reinforcement learning* zur Umsetzung der Eigenschaften geführt [176]. Mit *SliceGAN* [177] existiert weiterhin eine GAN-basierte Methode, die aus zweidimensionalen Bildern dreidimensionale Strukturen erzeugt. Diese Methode wurde bereits zur Erzeugung unterschiedlicher Mikrostrukturen erprobt [178]. Da GANs zwei gegeneinander antretende Netzwerke gleichzeitig trainiert, ist der Zielzustand ein Nash-Gleichgewicht, das schwer einzustellen ist. Das Training eines GANs ist entsprechend instabil und kann zudem zu Modecollapse führen. Das GAN generiert dabei wiederholt die gleichen Bilder, ohne den gesamten Datenbereich der Trainingsdaten abzudecken.

Als weitere Architektur zur Generierung künstlicher Mikrostrukturen sind Denoising Diffusion Probabilistic Models, nachfolgend als Diffusionsmodelle bezeichnet, zu nennen. Der Vorteil von Diffusionsmodellen gegenüber GANs ist das stabilere Training, da eine definierte Verlustfunktion optimiert wird und kein Nash-Gleichgewicht angestrebt wird. Im Gegensatz zu VAEs und GANs erfolgt

die Erzeugung dieser Mikrostrukturen nicht in einem einzigen, sondern in mehreren nacheinander ablaufenden Schritten. Der iterative Charakter der Generierung mittels Diffusionsmodellen, ermöglicht eine hohe Detailtreue und das Abdecken der gesamten Datenverteilung.

Diese Eigenschaften konnten bereits an verschiedenen Materialien nachgewiesen werden [179]. Die Generierung komplexer Mikrostrukturen konnte dabei von Diffusionsmodellen genauer und zuverlässiger durchgeführt werden als von VAEs und GANs. Diffusionsmodelle bieten sich weiterhin zur konditionierten Generierung an, wobei die Eigenschaften der finalen Struktur in den Generierungsprozess einfließen. Sie benötigen zur Erstellung neuer Datenpunkte allerdings eine Vielzahl an Diffusionsschritten. Das macht sie bei der Erzeugung dreidimensionaler Strukturen langsamer und rechenintensiver als GANs oder VAEs.

Um die hohe Qualität von Diffusionsmodellen mit der Geschwindigkeit von GANs zu kombinieren, wurden *Denoising Diffusion GANs* (DD-GAN) entwickelt [34]. Hier bewertet ein Diskriminator die einzelnen Entrauschungsschritte, dadurch kann die Anzahl der notwendigen Schritte zum kompletten Rauschen um mehrere Größenordnungen reduziert werden [34]. DD-GANs ermöglichen so die schnelle und diverse Generierung von dreidimensionalen Strukturen.

Zusammenfassend stellen maschinelle Lernverfahren somit eine Möglichkeit zur Verfügung, komplexe intrinsische Geometrien von Mikrostrukturen aus Beispieldaten zu identifizieren und zu imitieren. Der Vorteil gegenüber geometriebasierten und physikbasierten Methoden ist dabei, dass keine neuen Konstruktionsalgorithmen oder Modellierungsansätze entwickelt werden müssen. Nachteilig ist hingegen der Bedarf an Daten für das Training eines Netzes. Die Vorteile der maschinellen Lernverfahren überwiegen allerdings. Insbesondere die hohe Generierungsgeschwindigkeit eines trainierten Netzwerks.



## 5.2 Validierung digitaler Zwillinge

Die im vorangegangenen Abschnitt beschriebenen Generierungsansätze unterscheiden sich nicht nur in ihrer methodischen Grundlage, sondern auch in der Qualität und Variabilität der erzeugten Mikrostrukturen. Um diese Unterschiede systematisch zu erfassen und bewerten zu können, ist eine objektive Validierung unerlässlich.

In dieser Arbeit erfolgt diese Bewertung anhand von zwei zentralen Kriterien: Die Einstellbarkeit der Struktureigenschaften und der Realismus der generierten Strukturen.

Beide Kriterien sind entscheidend für den späteren Einsatz der generierten Strukturen in Simulationsstudien. Nur wenn sowohl die Struktureigenschaften gezielt eingestellt als auch die mikrostrukturellen Merkmale realistisch abgebildet werden können, lassen sich fundierte Aussagen über Struktur-Eigenschaft-Beziehungen treffen und die Simulationsergebnisse zuverlässig auf reale Materialien übertragen.

Der folgende Abschnitt beschreibt die in dieser Arbeit verwendete Methodik zur Validierung digitaler Zwillinge. Diese adressiert beide Bewertungsriterien gleichermaßen und bildet die Grundlage für alle späteren Analysen der Generierungsergebnisse.

### 5.2.1 Einstellbarkeit der Struktureigenschaften

Die Einstellbarkeit der Eigenschaften ist entscheidend, um gezielt Mikrostrukturen mit gewünschten Merkmalen erzeugen zu können. Für die Überprüfung der Eigenschaften kommen die in Abschnitt 4.5 beschriebenen Methoden zum Einsatz. Ziel ist es, mit der jeweiligen Generierungsmethode eine möglichst große Vielfalt an Struktureigenschaften abbilden zu können. Darüber hinaus wird betrachtet, inwieweit die Eigenschaften direkt vorgegeben werden können oder ob sie nur indirekt über die Parametrisierung des Generierungsalgorithmus angenähert werden.

## 5.2.2 Realismus der Strukturen

Der Realismus der generierten Strukturen ist von großer Bedeutung, da die späteren simulativen Untersuchungen der mechanischen Eigenschaften auf Grundlage der vollständig aufgelösten Mikrostruktur durchgeführt werden. Nur wenn die generierten Strukturen in ihrer mikrostrukturellen Morphologie den realen Schäumen entsprechen, sind die Ergebnisse mechanischer Simulationen auf reale Strukturen übertragbar. Daher müssen die Strukturen nicht nur in abgeleiteten Struktureigenschaften wie Porosität, Porenradius oder Wandstärke mit den realen Daten übereinstimmen, sondern auch in ihrer mikrostrukturellen Morphologie.

Zur Bewertung wird eine Zweipunktkorrelation (ZPK) in Kombination mit einer Hauptkomponentenanalyse (PCA) verwendet. Die genaue Vorgehensweise ist in Unterabschnitt 4.2.4 dargelegt. Für eine konsistente Vergleichbarkeit wird die PCA in dieser Arbeit ausschließlich auf den realen CT-Daten berechnet. Die generierten Strukturen werden anschließend mit der so bestimmten Transformation in denselben Hauptkomponentenraum projiziert. Liegen die Projektionen zweier Strukturen räumlich nahe beieinander, deutet dies auf ähnliche Morphologien der Mikrostrukturen hin. Dies ermöglicht eine systematische und objektive Bewertung des Realismus generierter Mikrostrukturen und bildet zugleich die Grundlage für sämtliche Bewertungen der Generierungsergebnisse in den folgenden Kapiteln.

## 5.3 Geometriebasierte Generierung

Die PACE3D-Simulationsumgebung stellt eine Reihe von Algorithmen zur Verfügung, mit denen digitale Abbilder poröser Mikrostrukturen gezielt konstruiert werden können. Diese decken ein breites Anwendungsspektrum ab. Von Grünkörpern für Sinterprozesse [180], über Schüttungen für geologische Untersuchungen [169], Packungen für Batteriematerialien [181, 182], bis hin zu Kornstrukturen für metallische Legierungen [39, 166] oder Membranstrukturen für Flüssigkeitstransport [136]. Da diese Algorithmen in der vorliegenden Arbeit nicht zum Einsatz kommen, wird auf ihre Funktionsweise nicht weiter eingegangen. Stattdessen wird

an dieser Stelle auf die zugehörigen Publikationen verwiesen. Es sei lediglich angemerkt, dass die Möglichkeit besteht, für verschiedene Anwendungsfälle eine geeignete Strukturdatenbank mit PACE3D zu erstellen.

Im Folgenden wird auf die in dieser Arbeit relevante geometriebasierte Methode zur Generierung synthetischer Schaumstrukturen eingegangen. Dabei wird zunächst der verwendete Voronoi-basierte Generierungsalgorithmus beschrieben, bevor auf die notwendige Nachbearbeitung eingegangen wird, mit der die kantige Geometrie der initialen Strukturen realistisch abgerundet wird.

### 5.3.1 Schaumgenerierung

Für die Generierung synthetischer Schaumstrukturen kommt in dieser Arbeit ein weiterentwickelter geometriebasierter Algorithmus aus PACE3D zum Einsatz. Dieser basiert auf einer angepassten Version des von Altschuh et al. [136] vorgestellten Voronoi-Algorithmus.

Der Generierungsprozess des verwendeten Algorithmus lässt sich in zwei Schritte unterteilen:

1. **Erzeugung eines dreidimensionalen Voronoi-Diagramms:** Eine definierte Anzahl von Voronoi-Punkten wird mithilfe eines deterministischen Zufallszahlengenerators im Raum platziert. Durch das Festlegen eines *seed*-Werts bei der Erzeugung, kann jede Struktur gezielt erneut erzeugt werden. Aus den Nachbarschaftsbeziehungen dieser Punkte entstehen polyedrische Voronoi-Zellen [156].
2. **Aufbau der Strukturphase:** Die Schnittflächen der Voronoi-Zellen werden mit einer Strukturphase fester Dicke gefüllt, wodurch ein geschlossenes Zellgerüst entsteht.

Beide Schritte können über verschiedene Parameter gezielt beeinflusst werden. Die wichtigsten werden im Folgenden erläutert.

## Voronoizerlegung

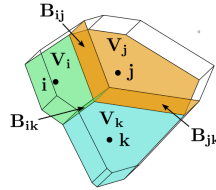
Für die Voronoizerlegung eines dreidimensionalen Raums  $\Omega \in \mathbb{R}^3$  wird eine definierte Menge an Punkten  $\mathbf{P} = \{x_i\}_{i=1}^n \subset \Omega$  platziert. Das zugehörige Voronoigebiet  $V_i$  eines Punkts  $x_i$  ist definiert als:

$$V_i = \{\mathbf{x} \in \Omega \mid \|\mathbf{x} - \mathbf{x}_i\| < \|\mathbf{x} - \mathbf{x}_j\|, j \neq i\}. \quad (5.1)$$

Die Grenzfläche zwischen zwei Voronoigebieten  $V_i$  und  $V_j$  ist gegeben durch:

$$\mathbf{B}_{ij} = \{\mathbf{x} \in \Omega \mid \|\mathbf{x} - \mathbf{x}_i\| = \|\mathbf{x} - \mathbf{x}_j\|\}. \quad (5.2)$$

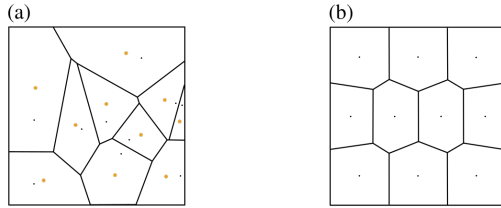
Eine Visualisierung dieser Aufteilung ist in Abbildung 5.1 dargestellt.



**Abbildung 5.1:** Voronoizerlegung eines dreidimensionalen Raums. Jeder Voronoipunkt definiert ein Volumen, das ihm am nächsten liegt. Die Flächen  $\mathbf{B}_{ij}$  bilden die Grenzflächen zwischen benachbarten Zellen. Grafik adaptiert von [97].

Im ersten Generierungsschritt liegen die wesentlichen Stellgrößen in der Anzahl der Voronoipunkte sowie in ihrer räumlichen Verteilung. Diese Parameter beeinflussen direkt die Porengröße und Porosität der späteren Struktur.

Um gleichmäßigere, schaumähnlichere Zellgeometrien zu erzeugen, wurde der ursprüngliche Generierungsalgorithmus um die Möglichkeit erweitert, den Lloyd-Algorithmus [162] anzuwenden. Dieser verschiebt jeden Voronoipunkt iterativ in den Schwerpunkt seiner jeweiligen Zelle, wodurch sogenannte *zentroidale*



**Abbildung 5.2:** Zentroidales Voronoigebiet. (a) Zufällig gewählte Voronoipunkte (schwarz) mit den Schwerpunkten der jeweiligen Voronoigebiete (orange). Der Lloyd-Algorithmus verschiebt die Punkte schrittweise in Richtung dieser Schwerpunkte. (b) Zentroidales Voronoidiagramm, bei dem die Voronoipunkte exakt in den Schwerpunkten ihrer Gebiete liegen.

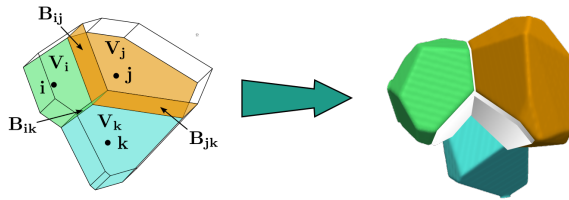
*Voronoi-Diagramme* entstehen [183]. Die Anzahl der Iterationen des Lloyd-Algorithmus erlaubt dabei eine gezielte Steuerung der Regelmäßigkeit der Zellstruktur. Abbildung 5.2 zeigt den Effekt dieses Algorithmus auf die Anordnung der Voronoipunkte.

Neben den beschriebenen Parametern stehen weitere Optionen zur Verfügung, die die Voronoi-Konstruktion beeinflussen. Da diese für den weiteren Verlauf dieser Arbeit nicht von Relevanz sind, werden sie nicht detailliert aufgeführt. Für weitere Informationen wird auf die Veröffentlichungen von Altschuh et al. [136] und August et al. [165] verwiesen.

## Strukturkonstruktion

Zur Erstellung der eigentlichen Mikrostruktur werden die Halbierungsebenen  $\mathbf{B}_{ij}$  des Voronoidiagramms verwendet. Auf diesen wird wie in Abbildung 5.3 gezeigt eine Strukturphase mit einer definierbaren Stärke platziert. In der Abbildung ist diese über die weißen Bereiche gekennzeichnet. Dadurch ergibt sich eine geschlossene Strukturphase auf den Halbierungsebenen.

Mit dem beschriebenen Generierungsalgorithmus lassen sich vielfältige Strukturen erzeugen, deren Eigenschaften gezielt angepasst werden können. Eine direkte Vorgabe spezifischer Zielwerte, etwa für Porengröße oder Porosität, ist bei der



**Abbildung 5.3:** Konstruktion eines Schaums aus der Voronoizerlegung. Entlang der Grenzflächen wird eine Strukturphase mit definierter Dicke aufgebaut. Bild adaptiert von [97].

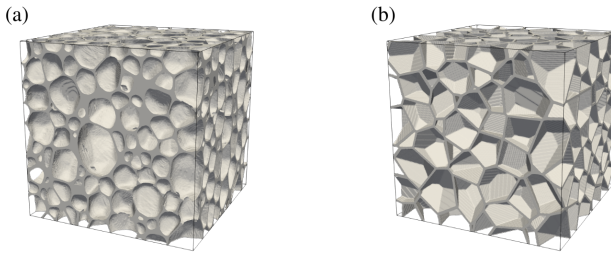
Voronoizerlegung jedoch nicht möglich. Stattdessen müssen diese über die gezielte Variation von Parametern wie der Anzahl der Voronoipunkte, der Wandstärke und der Anzahl der Lloyd-Iterationen iterativ angenähert werden.

### 5.3.2 Nachbearbeitung

Mit dem beschriebenen Algorithmus der Mikrostrukturgenerierung können zuverlässig Schaumstrukturen mit einstellbaren Geometrien erzeugt werden. Diese Strukturen weisen allerdings, wie zu erwarten [161], für Schäume untypische Winkel und Kanten auf. Abbildung 5.4 visualisiert den Unterschied zwischen realen und generierten Schäumen. Es ist erkennbar, dass die Größenverteilungen und der generelle Aufbau der Schäume übereinstimmt. Allerdings sind die Poren im generierten Schaum scharfkantige Polyeder, wohingegen der reale Schaum über abgerundete Poren verfügt. Eine Nachbearbeitung ist daher erforderlich, um die Porenform realistischen Strukturen anzunähern.

Im Rahmen dieser Arbeit wurden dazu zwei unterschiedliche Methoden zur Nachbearbeitung untersucht. Zum einen durch eine Simulation und zum anderen durch Bildbearbeitungsmethoden. Nachfolgend werden beide Methoden vorgestellt und hinsichtlich ihrer Eignung zur Generierung realistischer Schäume analysiert. Dazu wird, wie in Abschnitt 5.2 beschrieben, die Einstellbarkeit der Struktureigenschaften und der Realismus der Strukturen bewertet.

Als Ausgangsbasis dienen 230 unterschiedlich parametrisierte Strukturen, die mit dem beschriebenen geometriebasierten Algorithmus erzeugt wurden. Bei deren

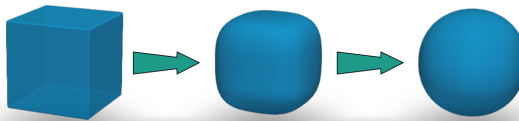


**Abbildung 5.4:** Vergleich eines realen Schaums (a) mit einem geometriebasiert generierten Schaum (b) vor der Nachbearbeitung.

Erstellung wurde darauf geachtet, dass ihre Struktureigenschaften den Bereich der realen Schäume vollständig abdecken und darüber hinaus erweitern. Die zu Vergleichszwecken herangezogenen CT-Daten stammen aus Subvolumen mit einer Kantenlänge von 400 Voxel, die aus den Bereichen *Probe A*, *Probe B* und *Probe C* extrahiert wurden.

### Simulation

Die Idee hinter der simulativen Nachbearbeitung der Strukturen ist es, die Poren der generierten Strukturen als Flüssigkeitstropfen zu modellieren, die bestrebt sind, ihre Oberfläche zu minimieren. Abbildung 5.5 visualisiert diese Idee. Die Poren werden dadurch schrittweise runder und nähern sich somit den Poren der realen Schaumstrukturen an. Zur Simulation kommt ein Allen-Cahn Phasenfeldmodell



**Abbildung 5.5:** Visualisierung der simulativen Nachbearbeitung. Die Poren werden als Flüssigkeitstropfen interpretiert und ihre innere Energie durch Formänderung optimiert. Ausgehen von polyedrischen Strukturen entsteht über die Simulation eine perfekte Kugel.

zum Einsatz, welches die Simulation von Flüssigkeitstropfen ermöglicht [35]. Dieses Modell verwendet das Energiedichtefunktional der Form

$$\mathcal{F}(\phi, \nabla\phi) = \int_V \varepsilon a(\phi, \nabla\phi) + \frac{1}{\varepsilon} w(\phi) + f_{VP}(\phi) dV \quad (5.3)$$

zur Modellierung der inneren Energie des Systems. Der erste Teil der Gleichung beschreibt dabei die Energie der Grenzfläche zwischen den vorliegenden Phasen und wird über ein Ginzburg-Landau typisches Funktional der Form

$$\begin{aligned} \mathcal{F}(\phi, \nabla\phi) &= \int_V W_I(\phi, \nabla\phi) dV \\ &= \int_V \varepsilon a(\phi, \nabla\phi) + \frac{1}{\varepsilon} w(\phi) dV \end{aligned} \quad (5.4)$$

formuliert. Der Parameter  $\varepsilon$  skaliert dabei die Breite des diffusen Phasenübergangs. Die reine Allen-Cahn Gleichung ist, wie in [35] detailliert beschrieben, über eine nicht erhaltende Größe beschrieben. Das bedeutet, dass die zeitliche Volumenänderung eines Tropfens zur Minimierung der freien Energie negativ ist. Der Tropfen wird demnach so lange verkleinert, bis er komplett verschwunden ist. Um dem entgegenzuwirken, führt das verwendete Modell den Term  $f_{VP}(\phi)$  ein. Dieser garantiert, dass die Volumina der simulierten Flüssigkeitstropfen erhalten bleiben. Dazu wird zunächst die Volumenänderung während der Simulation berechnet und anschließend über die treibende Kraft  $f_{VP}(\phi)$  korrigiert. Das verwendete Modell ignoriert den Einfluss der Gravitation, so dass die Tropfen in einer Art Schwebezustand evolviert werden. Eine detaillierte Herleitung und Beschreibung des gewählten Ansatzes ist in [35] zu finden.

Das beschriebenen Modell evolviert die Tropfen so lange, bis sie eine perfekte Kugelform eingenommen haben. Ein Zeitschritt der Simulation dauert etwa 96 s. Dabei kann es zur Verschmelzung aller Tropfen zu einem einzigen großen Tropfen kommen. Zudem können alle Poren zu perfekten Kugeln evolviere, was wiederum nicht den realen Strukturen entspricht. Aus diesen Gründen, muss die Simulation früher abgebrochen werden.

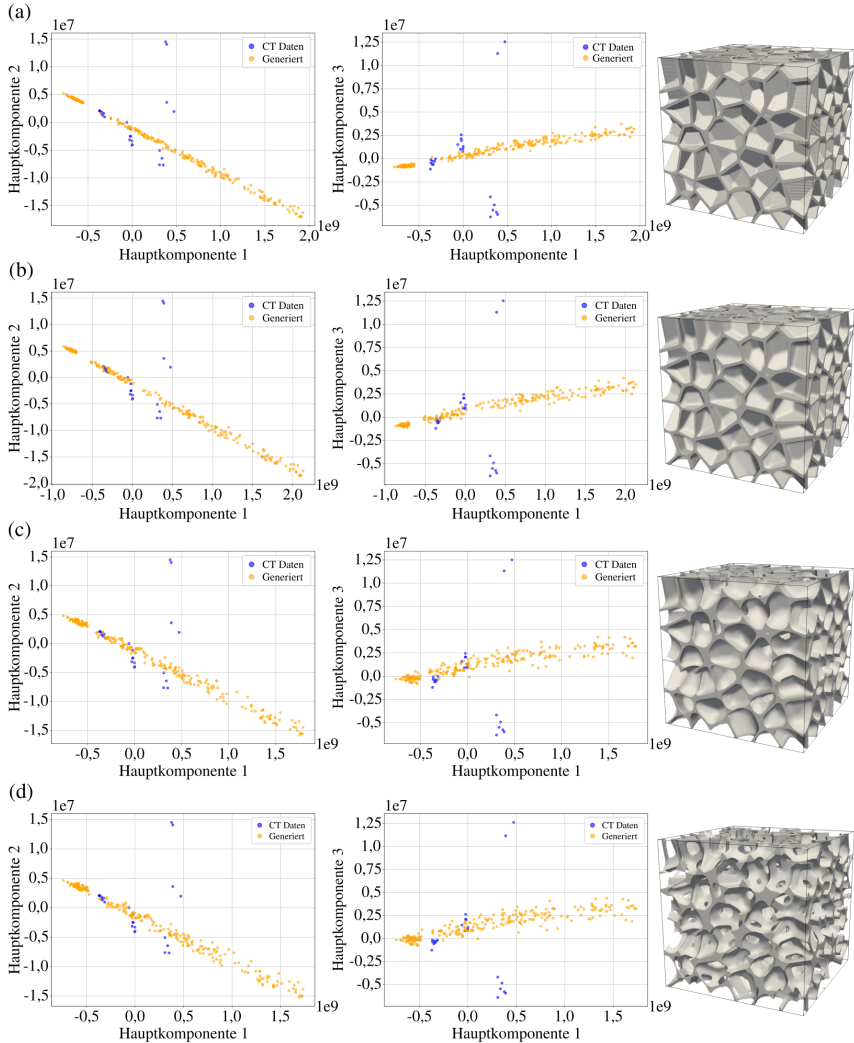


Um den optimalen Abbruchzeitpunkt zu bestimmen, wird der Realismus der Strukturen in regelmäßigen Intervallen mit den realen CT-Daten verglichen. Dabei kommt die in Unterabschnitt 4.2.4 beschriebene Kombination aus ZPK und PCA zum Einsatz. Die PCA selbst wird ausschließlich auf den CT-Daten durchgeführt. Die generierten Strukturen werden anschließend mittels der ermittelten Transformation in den PCA-Raum projiziert. Dies gewährleistet eine konsistente Vergleichbarkeit über alle Simulationszeitschritte hinweg. Abbildung 5.6 zeigt die zeitliche Entwicklung des Realismus zu unterschiedlichen Zeitschritten. Neben den ermittelten Korrelationen wird auch eine exemplarische Mikrostruktur gezeigt.

Den Diagrammen ist zu entnehmen, dass die CT-Daten über die gesamte Nachbearbeitung von den generierten Strukturen klar getrennte Cluster bilden. Die Nachbearbeitung verschiebt dabei die Punkteverteilung der generierten Strukturen kaum. Es ist lediglich eine leichte Verbreiterung der linearen Punkteverteilung zu erkennen. Dabei nähern sich die generierten Strukturen insbesondere im PC1-PC3 Diagramm immer weiter den CT-Daten an und decken deren Datenverteilung etwas besser ab.

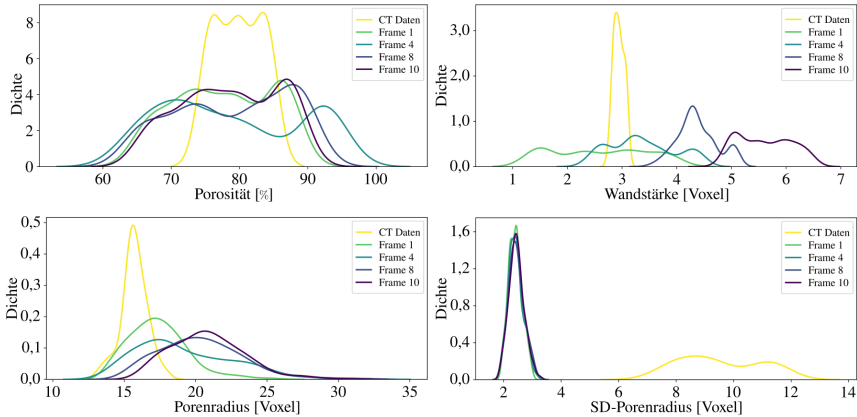
In den gezeigten Strukturen ist diese Entwicklung auch erkennbar. Im achten Zeitschritt (c) scheint die Schaumstruktur den realen Strukturen am ähnlichsten. Dies entspricht einer Simulationsdauer von etwa 13 min für eine Struktur mit einer Kantenlänge von 400 Voxel. Eine weitere Simulation der Strukturen ist nicht sinnvoll, da die Poren ab spätestens dem zehnten Zeitschritt (d) beginnen zu verschmelzen und die Porenwände sich auflösen.

Neben dem Realismus wird auch die Einhaltung der Struktureigenschaften über die Nachbearbeitungszeit betrachtet. Die Ergebnisse stellt Abbildung 5.7 in Form einer *Kerndichteschätzung* (KDE) dar. Diese erstellt auf Basis der gemessenen Struktureigenschaften in allen Proben eine kontinuierliche Schätzung der Verteilungen der Eigenschaften. Diese statistische Glättung kann zu scheinbar unphysikalischen Werten führen, so zeigt die KDE für die Porosität auch Werte oberhalb von 100 %. Dennoch gibt die KDE einen verständlichen Überblick über die Eigenschaftsverteilungen in den Strukturen.



**Abbildung 5.6:** Entwicklung des Realismus der Schäume über die Simulationsdauer als Hauptkomponentenanalyse. Die Ausgangsstruktur ist in (a) gezeigt. Die nachfolgenden Grafiken (b), (c) und (d) zeigen den vierten, achten und zehnten Simulationsschritt. Die realen Daten sind in blau, die generierten Daten in orange dargestellt.

Im Diagramm sind die Eigenschaften der CT-Daten gelb markiert. Grundlage bilden Subvolumina mit Kantenlängen von 400 Voxel, die zufällig aus *Probe A*, *Probe B* und *Probe C* ausgeschnitten und vermessen wurden.



**Abbildung 5.7:** Entwicklung der Struktureigenschaften über die Nachbearbeitung mittels einer Simulation.

Es ist ersichtlich, dass die Porosität der realen Daten sehr gut von den generierten Strukturen abgedeckt wird. Die Verteilung der Porositätswerte bleibt zudem über alle Simulationsschritte hinweg weitgehend konstant. Bei den übrigen Eigenschaften treten jedoch deutliche Abweichungen auf.

So liegt die Wandstärke zu Beginn in einem relativ gleichmäßigen Bereich zwischen 1,5 Voxel und 4,5 Voxel. Mit fortschreitender Simulation verschiebt sich die Verteilung jedoch zu größeren Wandstärken. Ursache ist das gewählte Nachbearbeitungsverfahren. Beim simulativen Evolvieren der Strukturen in Richtung eines Energieminimums lagert sich Material bevorzugt an bereits dickeren Stellen an, während dünne Stege verschwinden und deren Volumen in größere Strukturbereiche übergeht.

Für den Porenradius ist etwas ähnliches erkennbar. Während die initiale Verteilung der Porenradien gut mit den CT-Daten übereinstimmt, verschiebt sich diese immer

weiter in Richtung großer Porenradien. Dies ist damit zu begründen, dass kleine Poren gegebenenfalls komplett verschwinden und ihre Volumen stattdessen in größere Poren verschoben wird. Andererseits verschmelzen benachbarte Poren miteinander und bilden dadurch größere zusammenhängende Poren. Eine mögliche Gegenmaßnahme besteht darin, jeder Pore eine eigene Phase zuzuweisen. So wird verhindert, dass Poren verschmelzen oder vollständig verschwinden, da der Volumenanteil jeder Phase konstant bleiben muss. Diese Methode wurde für zweidimensionale Strukturen mit einer Kantenlänge von 400 Voxel erfolgreich getestet, erforderte jedoch bereits 24 s Rechenzeit pro Zeitschritt. Für dreidimensionale Strukturen ist sie daher praktisch nicht umsetzbar.

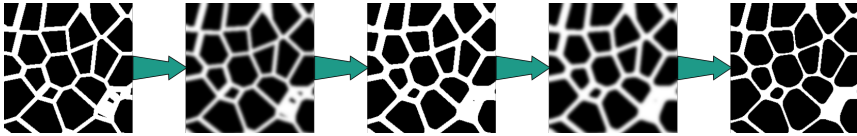
Die Standardabweichung des Porenradius (SD-Porenradius) zeigt weiterhin, dass die generierten Strukturen kaum in der Lage sind, die Größenverteilung der Poren realistisch abzubilden. Über alle Zeitschritte hinweg bleibt die Standardabweichung nahezu konstant, anstatt die in realen Strukturen beobachtete Variabilität zu zeigen.

Insgesamt zeigt sich, dass die ursprünglich generierten Schäume die Struktureigenschaften realer Schäume zunächst gut wiedergeben. Durch die Nachbearbeitung verschieben sich diese Eigenschaften jedoch, was eine gezielte Einstellung deutlich erschwert. Zudem erreicht die Nachbearbeitung keine hohe strukturelle Ähnlichkeit zu den realen CT-Daten.

## **Bildbearbeitung**

Der zweite untersuchte Ansatz zur Nachbearbeitung der Schaumgeometrie basiert auf der iterativen Anwendung einfacher Bildverarbeitungsmethoden. Bei diesen handelt es sich konkret um eine Bildglättung durch einen  $3 \times 3$  Gaußfilter und einer anschließenden Binarisierung durch einen globalen Schwellwert. Der Ablauf ist dabei so gestaltet, dass zunächst ein Schwellwert kleiner als 0,5 verwendet wird. Dadurch wird die Strukturphase aufgedickt. Anschließend folgt ein zweiter Durchlauf mit einem Schwellwert größer als 0,5, wodurch die Struktur wieder

ausgedünnt wird. Auf diese Weise bleibt die Porosität annäherungsweise erhalten, während die Schaumgeometrie schrittweise geglättet und abgerundet wird. Abbildung 5.8 illustriert diesen Vorgang anhand eines Beispielbildes.

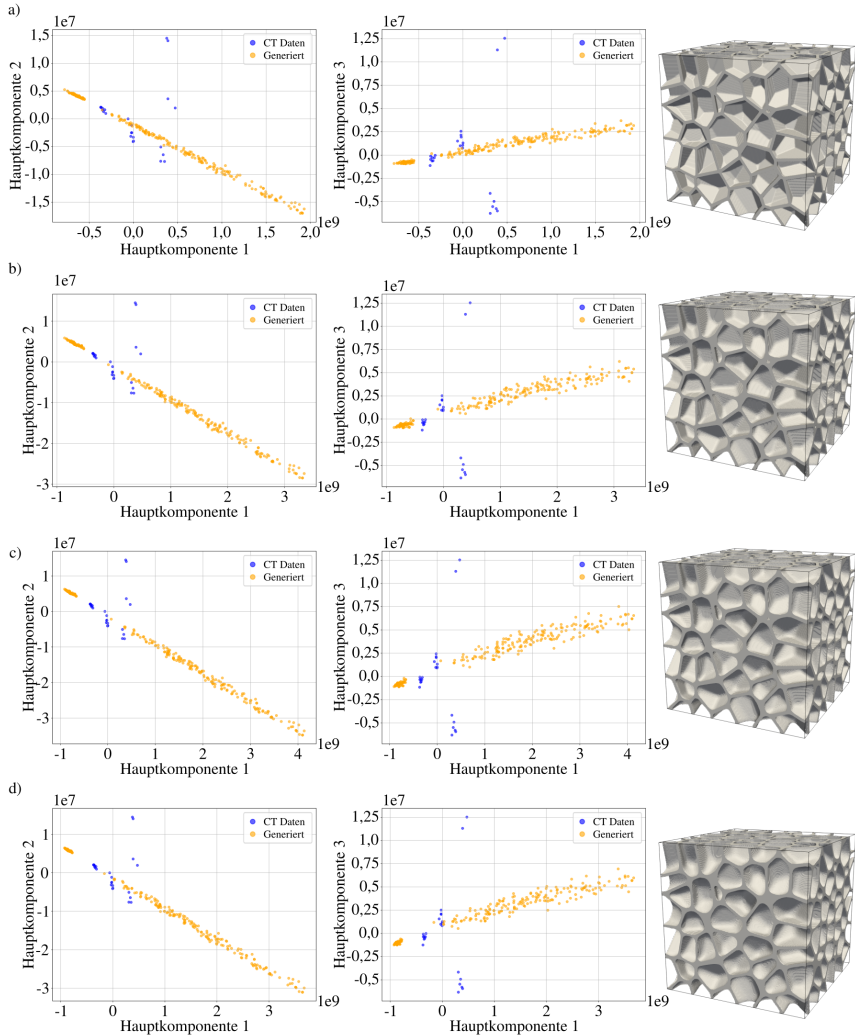


**Abbildung 5.8:** Eine Iteration der Nachbearbeitung der Struktur mittels Bildbearbeitungsmethoden. Die ursprüngliche Struktur wird mit einem Gaußfilter verschmiert und anschließend mit einem globalen Schwellwert  $< 0,5$  binarisiert, wodurch diese breiter und runder werden. Ein erneuter Gaußfilter-Schritt mit anschließendem Schwellwert  $> 0,5$  verdünnt die Struktur wieder. Das Ergebnis ist eine abgerundete Geometrie mit ähnlicher Porosität wie die Ausgangsstruktur.

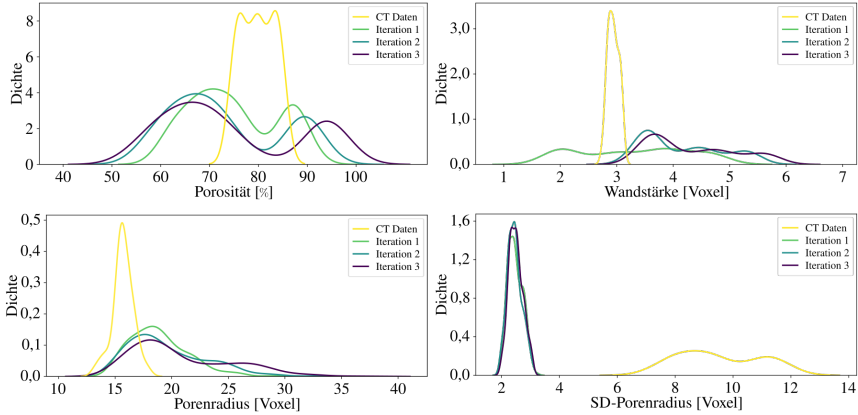
Im Vergleich zur simulativen Nachbearbeitung ist dieses Verfahren deutlich schneller. Für eine dreidimensionale Struktur mit einer Kantenlänge von 400 Voxel beträgt die Rechenzeit pro Iteration etwa 67 s.

Wie bei der simulativen Methode ist jedoch schwer abzuschätzen, wie viele Iterationen nötig sind, um einen möglichst realistischen Schaum zu erhalten. Deshalb wird auch hier nach jedem Bearbeitungsschritt der Realismus bewertet (Abbildung 5.9) und die Struktureigenschaften überprüft (Abbildung 5.10). Als Vergleich dienten die gleichen Strukturen wie bei der simulativen Nachbearbeitung.

Die PCA-Auswertung zeigt, dass die Bildbearbeitung die Verteilung der generierten Strukturen im Merkmalsraum verbreitert und damit einen größeren Bereich abdeckt. Eine direkte Überlappung mit den realen CT-Daten wird jedoch nicht erreicht. Hinsichtlich der Struktureigenschaften sind die Veränderungen geringer als bei der simulativen Nachbearbeitung, insbesondere bei Wandstärke und Porenradius. Die Ursache liegt daran, dass die Ausgangsstruktur durch das iterative Aufdicken und Verdünnen selbst nicht geändert wird, sondern lediglich abgerundet wird. Dies führt allerdings auch dazu, dass ab einer gewissen Iteration kaum noch Veränderungen erkennbar sind. In Abbildung 5.9 ist beispielsweise zwischen der vierten (c) und der zehnten Iteration (d) nahezu kein Unterschied in Mikrostruktur



**Abbildung 5.9:** Entwicklung des Realismus der Schäume über die Bildbearbeitungsschritte als Hauptkomponentenanalyse. Die Ausgangsstruktur ist in (a) gezeigt. Die nachfolgenden Grafiken (b), (c) und (d) zeigen den ersten, zweiten und zehnten Bearbeitungsschritt. Die realen Daten sind in blau, die generierten Daten in orange dargestellt.



**Abbildung 5.10:** Entwicklung der Struktureigenschaften über die Nachbearbeitung mittels Bildbearbeitungsmethoden.

und PCA-Position feststellbar. Die Nachbearbeitung mittels Bildbearbeitungsmethoden führt somit auch nicht zu einer realistischen Nachahmung der realen CT-Daten.

Insgesamt konnten weder die simulative noch die bildbasierte Nachbearbeitung zufriedenstellend realistische Mikrostrukturen erzeugen. Zudem verlängert die Nachbearbeitung die Generierungsdauer erheblich. Aus diesen Gründen wird im folgenden Kapitel ein datengetriebener Ansatz zur Generierung von Schaumstrukturen vorgestellt und validiert.

## 5.4 Datengetriebene Generierung mittels Diffusionsmodellen

Die zuvor untersuchten geometriebasierten Algorithmen konnten weder ausreichend realistische noch in ihren Eigenschaften gezielt steuerbare Strukturen erzeugen. Aus diesem Grund wird in diesem Kapitel ein datengetriebener Ansatz auf

Basis generativer neuronaler Netze entwickelt, der realistische Schaumstrukturen mit kontrollierbaren Struktureigenschaften erzeugen kann.

Generative Modelle lernen die Verteilung realer Daten und sind dadurch in der Lage, neue, synthetische Strukturen zu erzeugen. Durch geeignete Konditionierungen lassen sich gezielt morphologische Merkmale wie Porosität, Wandstärke oder Porengröße steuern. Unter den aktuellen Ansätzen haben sich Diffusionsmodelle als besonders leistungsfähig erwiesen, da sie eine hohe Detailtreue und Flexibilität bieten [184]. Ihr Nachteil ist jedoch die vergleichsweise lange Rechenzeit, da viele aufeinanderfolgende Diffusionsschritte erforderlich sind. Dieses Problem, verstärkt sich noch bei dreidimensionalen Strukturen.

Um diesem Problem durch die Reduzierung der benötigten Diffusionsschritte entgegenzuwirken, wird in dieser Arbeit ein Diffusion-GAN (DD-GAN) eingesetzt. Diese Architektur kombiniert die hohe Datenvielseitigkeit von Diffusionsmodellen mit der schnellen Generierungsfähigkeit von GANs [34]. Im Folgenden wird zunächst kurz auf beide Modelltypen eingegangen, bevor die konkrete DD-GAN-Architektur vorgestellt wird.

### **5.4.1 DD-GAN**

Diffusion-GANs vereinen die Stärken von Denoising Diffusion Modellen und Generative Adversarial Networks (GAN). Um die Vorteile dieser Kombination zu verstehen, werden zunächst beide Architekturen einzeln vorgestellt.

Diffusionsmodelle erzeugen neue Datenpunkte, indem sie aus einem initial stark verrauschten Datensatz schrittweise Rauschen entfernen. In jedem Iterationsschritt wird normalverteiltes Rauschen abgezogen, bis schließlich ein realistisch wirkendes Bild oder Volumen entsteht. Dieser fein abgestufte Generierungsprozess führt zu einer hohen Detailtreue und realistischen Ergebnissen. Allerdings erfordert er viele Iterationen, wodurch insbesondere bei dreidimensionalen Strukturen erheblicher Zeit- und Rechenaufwand entsteht.



Generative Adversarial Networks arbeiten hingegen mit zwei gleichzeitig trainierten neuronalen Netzen, dem Generator und dem Diskriminator. Der Generator erzeugt aus zufälligen Eingabewerten in einem einzigen Generierungsschritt einen synthetischen Datenpunkt. Der Diskriminator bewertet die Ähnlichkeit der synthetischen Daten zu realen Daten. Während der Generator versucht, den Diskriminator zu täuschen, lernt dieser, synthetische von echten Daten zu unterscheiden. Die beiden Netzwerke werden gleichzeitig trainiert und beeinflussen sich gegenseitig. Daher wird im Gegensatz zu anderen Generierungsmethoden keine feste Verlustfunktion optimiert, sondern vielmehr das dynamische Zusammenspiel beider Netzwerke. Dies kann zu instabilem Training führen und begünstigt den sogenannten Mode Collapse, bei dem der Generator nur wenige, sehr ähnliche Ausgaben produziert und nicht den gesamten Datenbereich abdeckt.

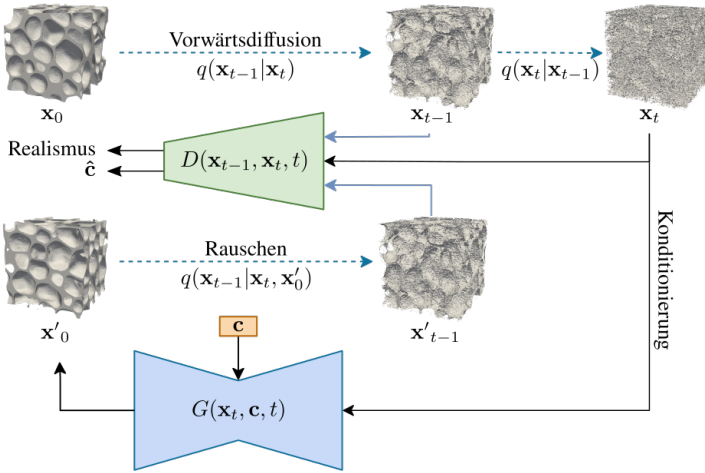
Ein Diffusion-GAN kombiniert nun beide Architekturen, um deren Vorteile auszunutzen und ihre Nachteile zu minimieren. Der Generierungsprozess folgt dem iterativen Entrauschen eines Diffusionsmodells, jedoch wird in jedem Schritt multimodales Rauschen entfernt. Die Bewertung der Zwischenergebnisse übernimmt dabei ein GAN-Diskriminator, der die Qualität jedes Entrauschungsschrittes einschätzt. Dadurch kann die Zahl der notwendigen Iterationen deutlich reduziert werden, was die Methode auch für die Generierung komplexer dreidimensionaler Strukturen praktikabel macht.

Ein weiterer Vorteil des DD-GAN ist seine Eignung für Konditionierung. Das ist die gezielte Steuerung des Generierungsprozesses durch zusätzliche Eingaben, beispielsweise Strukturparameter oder Referenzbilder einer Zielstruktur. Zwar lassen sich auch GANs konditionieren, doch erleichtert der mehrstufige Generierungsprozess von Diffusionsmodellen die Integration solcher Zusatzinformationen deutlich.

Im Folgenden wird der konkrete Aufbau des hier verwendeten DD-GAN beschrieben und auf die einzelnen Komponenten im Detail eingegangen.

### 5.4.1.1 Modell

Der grundlegende Ablauf eines Entrauschungsschritts im hier eingesetzten DD-GAN ist in Abbildung 5.11 dargestellt. Ausgangspunkt ist eine verrauschte Struktur  $\mathbf{x}_t$  im aktuellen Zeitschritt  $t$  sowie eine Konditionierung  $\mathbf{c}$ , die hier die gewünschten strukturellen Eigenschaften der Zielstruktur beschreibt.



**Abbildung 5.11:** Schematischer Ablauf eines Entrauschungsschritts im eingesetzten DD-GAN. Der Generator erhält die verrauschte Struktur  $\mathbf{x}_t$ , den Zeitschritt  $t$  sowie die Konditionierung  $\mathbf{c}$  und erzeugt daraus eine entrauschte Struktur  $\mathbf{x}'_0$ . Diese wird mit Rauschen entsprechend dem vorherigen Zeitschritt  $t - 1$  versehen, um die nächste Struktur  $\mathbf{x}'_{t-1}$  zu erzeugen. Der Diskriminator bewertet sowohl den Realismus als auch die korrekte Umsetzung der Konditionierung.

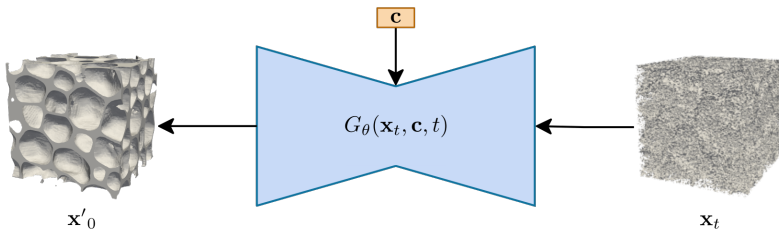
Der Generator erstellt aus diesen Eingaben eine vollständig entrauschte Struktur  $\mathbf{x}'_0$ , die anschließend mit Rauschen des vorherigen Zeitschritts  $t - 1$  versehen wird. So entsteht  $\mathbf{x}'_{t-1}$ , die als Ausgangspunkt für den nächsten Entrauschungsschritt dient. Dieser Ablauf wiederholt sich iterativ, bis am Ende des Diffusionsprozesses eine vollständig entrauschte, synthetische Mikrostruktur vorliegt.

Parallel dazu verarbeitet der Diskriminator den aktuellen Zeitschritt  $t$ , die Ausgangsstruktur  $\mathbf{x}_t$  sowie entweder die reale Struktur  $\mathbf{x}_{t-1}$  oder die vom Generator erzeugte Struktur  $\mathbf{x}'_{t-1}$ . Der Diskriminator bewertet basierend darauf zum einen den Realismus und zum anderen die Umsetzung der Konditionierung. Er versucht dazu, reale von synthetischen Strukturen zu unterscheiden und die Struktureigenschaften vorherzusagen, um sie mit der Konditionierung zu vergleichen.

Das Training des DD-GANs optimiert somit gleichzeitig zwei getrennte Netzwerke: Einen Generator, der realistische und korrekt konditionierte Strukturen erzeugt, und einen Diskriminator, der zur Klassifikation und zur Charakterisierung der Strukturen verwendet werden kann. Im Folgenden wird zunächst der Aufbau des Generators und anschließend die Architektur des Diskriminators im Detail beschrieben.

## Generator

Der Generator des eingesetzten DD-GAN hat die Aufgabe, aus einer verrauschten Mikrostruktur  $\mathbf{x}_t$  eine realistische und vollständig entrauschte Struktur  $\mathbf{x}'_0$  zu rekonstruieren, die den vorgegebenen Struktureigenschaften entspricht. Neben der verrauschten Eingabe erhält er den aktuellen Zeitschritt  $t$  sowie einen Konditionierungsvektor  $\mathbf{c} \in \mathbb{R}^4$ , der Porosität, Wandstärke, mittleren Porenradius und Standardabweichung des Porenradius der Zielstruktur beschreibt.



**Abbildung 5.12:** Schematische Funktionsweise des Generators. Die Eingabe besteht aus einer verrauschten Struktur  $\mathbf{x}_t$ , dem Zeitschritt  $t$  sowie der Konditionierung  $\mathbf{c}$ . Der Generator liefert daraus eine entrauschte Struktur  $\mathbf{x}'_0$ .

Formal lässt sich der Generator demnach als Funktion

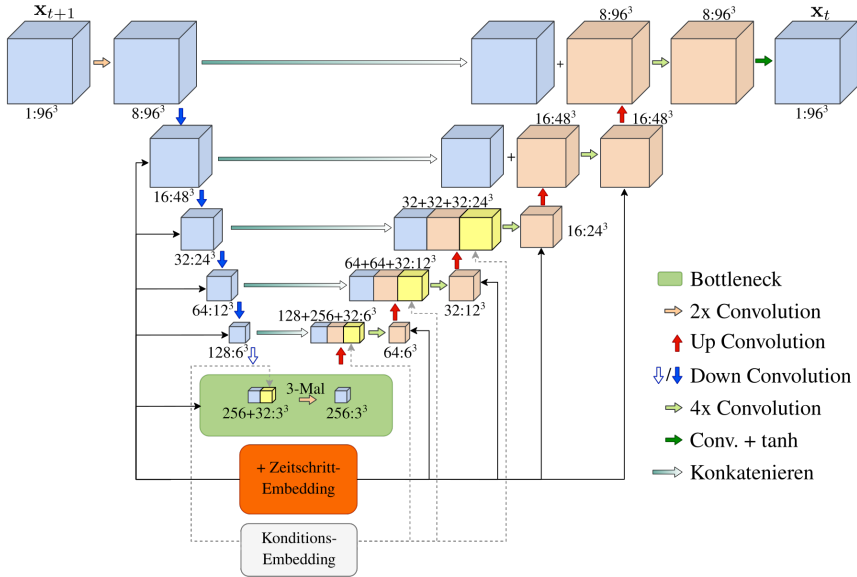
$$G_{\theta}(\mathbf{x}_t, \mathbf{c}, t) : \mathbb{R}^N \times \mathbb{R}^4 \times \mathbb{R} \rightarrow \mathbb{R}^N$$

beschreiben.

Die Architektur des Generators ist in Abbildung 5.13 schematisch dargestellt und basiert auf der Architektur zur Bildgenerierung von Xiao et al. [34], die weiterentwickelt wurde, um dreidimensionale Strukturen zu erzeugen. Darüber hinaus unterscheidet sich das entwickelte Modell in einem wesentlichen Punkt vom Original. Während das ursprüngliche DD-GAN eine latente Zufallsvariable  $\mathbf{z}$  zur bedingten Generierung einsetzt, verzichtet die hier implementierte Architektur vollständig auf diese Komponente. Stattdessen erfolgt die Steuerung ausschließlich über die explizite Konditionierung  $\mathbf{c}$ .

Der Vektor  $\mathbf{c} \in \mathbb{R}^4$  wird zunächst durch ein kleines neuronales Netz mit zwei vollständig verbundenen Schichten und dazwischenliegender ReLU-Aktivierung in einen latenten Merkmalsraum projiziert. Der resultierende Konditionsvektor besitzt eine feste Dimension von 32. Dieses Embedding wird an zwei Stellen in das Netzwerk eingespeist: Im Bottleneck, also im zentralen Teil des U-Nets, wird es auf die räumliche Dimension der Feature Maps expandiert und entlang der Kanalachse an diese angehängt; in den drei untersten Stufen des Decoders erfolgt ein erneutes Konkatenieren mit den jeweils aktuellen Feature Maps. Die tiefe Einbindung ist bewusst gewählt, da hier semantisch abstraktere Repräsentationen verarbeitet werden. Die Konkatenation anstelle einer Addition ermöglicht es dem Netz, Bildinformationen und Konditionierung getrennt zu halten und flexibel zu lernen, in welcher Ausprägung die vorgegebenen Eigenschaften berücksichtigt werden sollen.

Der Zeitschritt  $t \in \mathbb{N}$  wird über ein sinusoidales Positional Encoding [185] in einen Vektor der Länge 256 kodiert. Die diskreten Zeitschrittwerte werden so in einen kontinuierlichen Vektor umgewandelt. Nach Projektion auf die benötigte Kanalzahl durch eine lineare Schicht mit vorgeschalteter SiLU-Aktivierung wird er auf die räumliche Dimension erweitert und auf die Feature Maps addiert. Das



**Abbildung 5.13:** Generator des DD-GANs. Die Architektur ist ein 3D-U-Net, das verrauschte Strukturen verarbeitet. Die hellgrauen Blöcke markieren die Einbindung der Konditionierung. Die angegebenen Werte beziehen sich auf *Kanäle:räumliche Größe*. In jeder Faltungsstufe wird zusätzlich ein Zeitembedding addiert.

Zeitembedding wird an dieser Stelle auf die Features addiert, um diese je nach Zeitschritt zu verschieben. So bleibt die Anzahl der Features konstant, das Netz kann aber über die Verschiebung den Zeitschritt kodieren.

Das Ziel des Trainings für den Generator besteht darin, Mikrostrukturen zu generieren, die sowohl hinsichtlich ihres Aufbaus realistisch sind als auch die vorgegebenen strukturellen Eigenschaften gemäß der Konditionierung umsetzen. Die Zielfunktion des Generators lautet demnach:

$$\mathcal{L}_G = \mathcal{L}_{\text{GAN}} + \lambda_c \cdot \mathcal{L}_{\text{cond}} + \mathcal{L}_{\text{FM}}. \quad (5.5)$$

Dabei bezeichnet  $\mathcal{L}_{\text{GAN}}$  den klassischen adversarialen Verlust des Generators (vgl. Gleichung 2.18), der den Realismus der generierten Mikrostruktur maximiert.

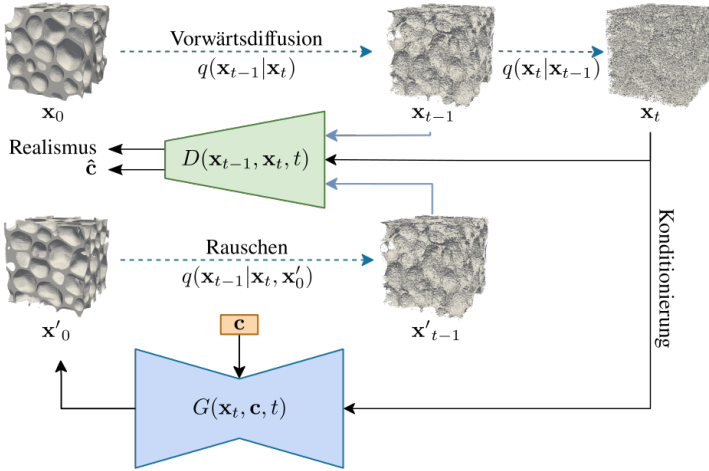
Der Term  $\mathcal{L}_{\text{cond}}$  misst die Übereinstimmung zwischen den vom Diskriminator vorhergesagten Mikrostruktureigenschaften und den vorgegebenen Konditionierungswerten und ist als mittlere quadratische Abweichung (MSE) implementiert. Der Hyperparameter  $\lambda$  steuert den Einfluss der Konditionierungsabweichung. Der Parameter  $\lambda$  wird im Verlauf der ersten 1000 Trainings-Epochen linear von 0 auf 1 erhöht ( $\lambda = \frac{E}{1000}$ ) und bleibt anschließend konstant bei 1. Der dritte Term  $\mathcal{L}_{\text{FM}}$  bezeichnet den Feature-Matching-Verlust, der die L1-Distanz zwischen den internen Feature-Repräsentationen realer und generierter Strukturen in ausgewählten Schichten des Diskriminators minimiert. Der Feature-Matching-Term wird über alle Schichten des hinweg gleich gewichtet, wobei jeweils die absolute Differenz (L1-Norm) der Feature Maps verwendet wird. Der Feature-Matching-Verlust kommt hier zum Einsatz, um starke Schwankungen im Training zu vermeiden und es so zu stabilisieren.

Zur Vermeidung eines Mode Collapse wird im Diskriminator zusätzlich *Mini-batch Discrimination* eingesetzt. Dieser Mechanismus fördert die Diversität der generierten Strukturen, indem er Korrelationen innerhalb eines Minibatches berücksichtigt.

## Diskriminator

Der Diskriminator erfüllt in der verwendeten DD-GAN Architektur zwei Aufgaben: Die Bewertung des Realismus und die Vorhersage der Konditionierung. Dazu erhält der Diskriminator eine verrauschte Struktur  $\mathbf{x}_t$  sowie eine weitere Struktur, die entweder real  $\mathbf{x}_{t-1}$  oder generiert  $\mathbf{x}'_{t-1}$  ist. Zudem erhält er den aktuellen Zeitschritt  $t$ . Abbildung 5.14 stellt die Vorgehensweise des Diskriminators schematisch dar.

Auf Basis dieser Eingangsdaten versucht der Diskriminator nun zu bewerten, ob es sich um eine reale Struktur handelt oder nicht, und leitet gleichzeitig eine



**Abbildung 5.14:** Schematische Funktionsweise des Diskriminators. Eine verrauschte Struktur  $\mathbf{x}_t$  sowie eine etwas entrauschte Struktur  $\mathbf{x}_{t-1}$  oder  $\mathbf{x}'_{t-1}$  werden dem Diskriminator gegeben. Der Diskriminator bewertet nun den Realismus der entrauschten Struktur und sagt die Konditionierung  $\hat{\mathbf{c}}$  voraus. Ziel des Diskriminators ist es die Konditionierung möglichst genau vorherzusagen und reale und generierte Strukturen zuverlässig voneinander über den Realismus zu unterscheiden.

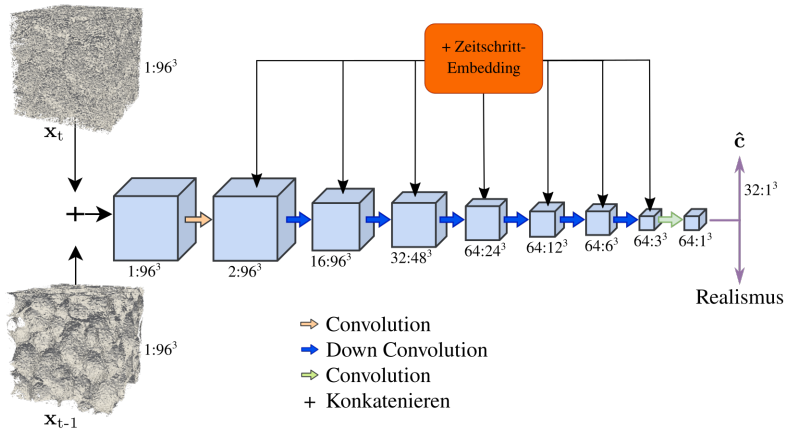
Vorhersage für die zugrunde liegende Konditionierung ab. Formal lässt sich der Diskriminator daher als:

$$D_\phi(\mathbf{x}_{t-1}, \mathbf{x}_t, t) : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R} \rightarrow [0, 1] \times \hat{\mathbf{c}}$$

beschreiben.

Wie in Abbildung 5.15 dargestellt handelt es sich bei dem Diskriminator um einen Encoder auf Basis eines Convolutional Neural Networks.

Die Architektur basiert auf dem Modell von Xiao et al. [34], wurde jedoch weiterentwickelt. Einerseits wurde der Diskriminator zur Verarbeitung dreidimensionaler Strukturen erweitert. Andererseits wurde die Architektur um einen zweiten Ausgabepfad erweitert, über den die Vorhersage der Mikrostruktureigenschaften erfolgt.



**Abbildung 5.15:** Diskriminator des DD-GANs. Die Architektur ist weitgehend von [34] übernommen. Die Größe der verarbeiteten Eingänge ist mit „Kanäle:Größe“ neben den Blöcken notiert. Wie der Generator erhält auch der Diskriminator ein Zeitschrittembedding, um ihn im Diffusionsprozess verwenden zu können.

Zu Beginn des Diskriminators werden die beiden Eingänge  $x_t$  und  $x_{t-1}$  entlang der Kanaldimension miteinander concatenated. Der daraus entstehende Tensor durchläuft zunächst eine 3D-Faltungsschicht, gefolgt von sechs sogenannten *DownConv*-Blöcken. Jeder dieser Blöcke besteht aus zwei Faltungsschichten mit vorgeschalteter LeakyReLU-Aktivierung sowie einer optionalen Downsampling-Operation über Average-Pooling. Zusätzlich wird in jedem Block wie im Generator ein Zeitembedding additiv zur Feature Map hinzugefügt.

Das Zeitembedding wird analog zum Generator aus dem Zeitschritt  $t$  über ein sinusoidales Positional Encoding erzeugt und durch ein kleines neuronales Netz mit zwei linearen Schichten transformiert. Das erstellte Embedding wird in jedem DownConv-Block auf die benötigte Kanalgröße projiziert und auf die Feature Map aufaddiert. Das Zeitembedding wird hier aus dem gleichen Grund wie im Generator aufaddiert: Die Verschiebung der Features um das Zeitembedding ermöglicht es dem Netz die Zeit zu kodieren, ohne die Featureanzahl zu erhöhen. Die Dimensionen der Tensoren innerhalb des Netzwerks sind in Abbildung 5.15 über „Kanäle:Größe“ zusammengefasst.



Nach dem letzten Downsampling-Schritt ist die räumliche Dimension des Tensors auf eine Größe von  $64 : 1^3$  reduziert. Auf diesem Tensor wird die Minibatch Discrimination implementiert. Bei dieser Methode zur Vermeidung von Mode Collapse wird die Standardabweichung innerhalb von Minibatches berechnet, den der Diskriminator verarbeitet und an die Feature Maps angehängt. In der vorliegenden Arbeit bestehen die Minibatches aus jeweils vier Strukturen. Diese Technik fördert die Diversität der generierten Strukturen, indem sie die Varianz der untersuchten Strukturen in die Bewertung einfließen lässt. Im Anschluss folgt eine weitere Faltungsschicht, gefolgt von einer LeakyReLU-Aktivierung. Das Ergebnis dieser Faltung wird in zwei getrennte MLPs geleitet, um die Vorhersage für Realismus und die Konditionierung  $\hat{c}$  zu erzeugen. Der erste Pfad dient der Realismusbewertung und besteht aus einer linearen Schicht, die einen Wert im Bereich  $[0; 1]$  ausgibt. Je näher der Wert an 1 liegt, desto höher wird der Realismus der Struktur bewertet. Der Pfad zur Vorhersage der Konditionierung besteht aus zwei vollvernetzten Schichten mit dazwischenliegender LeakyReLU-Aktivierung.

Das Trainingsziel des Diskriminators besteht darin, einerseits reale und generierte Strukturen hinsichtlich ihres Realismus zu klassifizieren und andererseits die strukturellen Eigenschaften möglichst genau vorherzusagen. Entsprechend setzt sich die Verlustfunktion aus mehreren Termen zusammen:

$$\mathcal{L}_D = \mathcal{L}_{\text{adv}} + \lambda \cdot \mathcal{L}_{\text{cond}}. \quad (5.6)$$

$\mathcal{L}_{\text{adv}}$  beschreibt hierbei den Kreuzentropieverlust der Unterscheidung realer von generierter Strukturen. Der  $\mathcal{L}_{\text{cond}}$  hingegen beschreibt den mittleren quadratischen Fehler (MSE) zwischen der vorhergesagten und der tatsächlichen Konditionierung. Genau wie im Generator wird der Konditionierungsverlust über  $\lambda$  linear über die ersten 1 000 Trainings-Epochen von 0 auf 1 skaliert und danach auf 1 festgelegt. Die Erweiterung der klassischen GAN-Zielfunktion um den Konditionierungsloss erlaubt es dem Diskriminator, die Umsetzung der Konditionierung durch den Generator gezielt zu bewerten. Dies ermöglicht es, die Konditionierung in den generierten Strukturen überhaupt erst zu überprüfen.

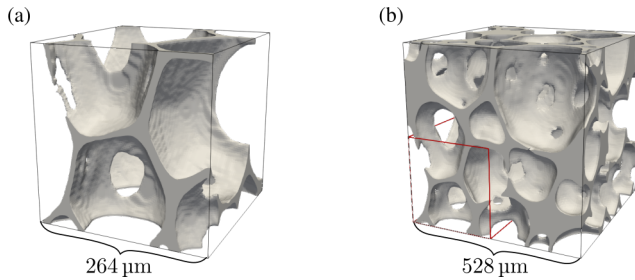
### 5.4.2 Trainingsdaten

Zum Training des beschriebenen DD-GANs werden die rekonstruierten CT-Daten aus Abschnitt 4.4 genutzt. Diese liegen als sehr große Volumendatensätze vor:

- *Probe A*:  $962 \times 1034 \times 1500$  Voxel
- *Probe B*:  $963 \times 693 \times 1000$  Voxel
- *Probe C*:  $895 \times 726 \times 1100$  Voxel

Um die Daten verarbeitbar zu machen, werden kleinere Subvolumen aus den CT-Daten extrahiert. Da die Architektur des Generators die räumliche Auflösung in fünf Stufen halbiert, muss die Kantenlänge der Subvolumen ein Vielfaches von  $2^5 = 32$  betragen. Es werden  $96 \times 96 \times 96$  Voxel große Ausschnitte verwendet, die ein gutes Gleichgewicht zwischen Strukturdarstellung und Speicherbedarf bieten.

Aufgrund der Inhomogenität der Mikrostrukturen könnte ein solches Subvolumen jedoch zu wenig strukturelle Information enthalten. Daher werden die CT-Daten vor dem Zuschchnitt auf  $5,5 \mu\text{m}/\text{Voxel}$  heruntergesampelt. Ein  $96^3$ -Ausschnitt entspricht damit einem Bereich von  $192 \times 192 \times 192$  Voxel in den unskalierten Daten, also  $528 \times 528 \times 528 \mu\text{m}$ . Abbildung 5.16 stellt ein skaliertes Subvolumen einem nicht skalierten Subvolumen gegenüber und verdeutlicht den erzielten Informationsgewinn.



**Abbildung 5.16:** Vergleich des Informationsgehalts in den Subvolumen: (a) Original, (b) skaliertes Subvolumen. Die rote Box in (b) markiert die Position des ursprünglichen Volumens.

Da das DD-GAN konditioniert trainiert wird, müssen die Struktureigenschaften jedes Subvolumens vorliegen. Um das Training durch die Messung dieser Werte nicht zu verlangsamen, werden diese vorab berechnet. Dafür werden 500000 unterschiedliche Subvolumen aus den CT-Daten extrahiert und mit den GPU-beschleunigten Bibliotheken *Cucim* und *Cupy* verarbeitet. *Cucim* baut auf *Cupy* auf. Letzteres stellt ein Python-Framework mit einer zu *NumPy* analogen API bereit, das auf *NVIDIA*-GPUs beschleunigte Operationen nutzt. Über diese Tools werden die Porosität, die Wandstärke, die mittlere Porengröße und die Standardabweichung der Porengröße ermittelt. Die Ergebnisse der Analyse werden zusammen mit den Koordinaten des verwendeten Subvolumens in einer Datei gespeichert. Während des Trainings werden die Subvolumina entsprechend den Koordinaten ausgeschnitten und die gespeicherten Struktureigenschaften als Konditionierung verwendet. Für eine größere Varianz der Strukturen werden diese während des Trainings zufällig gespiegelt oder gedreht. Dies verändert die Mikrostruktur, die Struktureigenschaften bleiben allerdings erhalten.

Um sicherzustellen, dass die verwendeten GPU beschleunigten Programme die Struktureigenschaften zuverlässig ermitteln, werden die Ergebnisse mit den in PACE3D verfügbaren Methoden verglichen. Wichtig bei diesem Vergleich ist jedoch nicht die absolute Differenz der Messmethoden, sondern vielmehr, dass die Abweichungen relativ konstant bleiben. In diesem Fall können die Konditionierungen einfach um diese Abweichung angepasst werden.

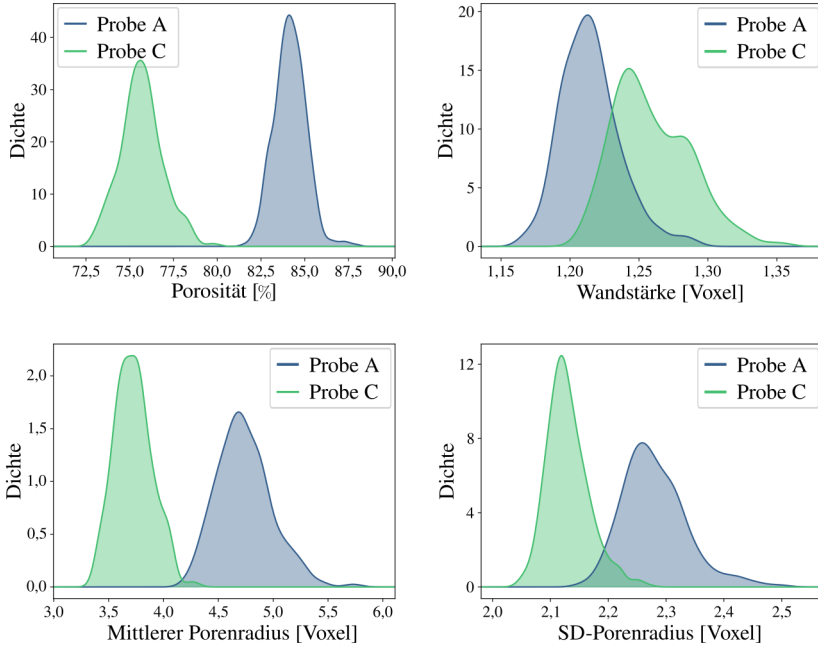
Eigenschaft	Methode	Kantenlänge der Ausschnitt [Voxel]				
		50	100	200	250	300
Porosität [%]	GPU	76	79	83	83	83
	PACE3D	75	79	83	83	82
	Diff.	+1	+0	+0	+0	+1
Wandstärke [Voxel]	GPU	2,40	2,42	2,38	2,38	2,41
	PACE3D	2,49	2,95	2,86	2,84	2,89
	Diff.	-0,09	-0,53	-0,49	-0,46	-0,47
Mittlerer Porenradius [Voxel]	GPU	9,37	10,88	11,64	11,44	11,58
	PACE3D	4,24	16,09	13,37	14,39	14,98
	Diff.	+5,13	-5,22	-1,73	-2,94	-3,40
SD-Porenradius [Voxel]	GPU	5,32	7,10	7,89	8,14	8,53
	PACE3D	0,87	7,70	9,39	9,67	10,11
	Diff.	+4,45	-0,60	-1,50	-1,52	-1,57

**Tabelle 5.1:** Vergleich der Eigenschaftsberechnungen von GPU und PACE3D bei verschiedenen Volumengrößen. Die Differenz ist definiert als  $\text{Wert}_{\text{GPU}} - \text{Wert}_{\text{PACE}}$ .

Tabelle 5.1 zeigt diesen Vergleich. In dieser Tabelle ist insbesondere die Domänengröße 200 Voxel interessant, da diese den verwendeten Ausschnitten von  $96^3$  Voxel am nächsten kommt. Die Messwerte wurden in nicht vergrößerten Strukturen ermittelt. Es ist erkennbar, dass ab einer Kantenlänge von 200 Voxel die Abweichungen relativ konstant bleiben. Lediglich die Porengröße schwankt etwas stärker. Dies ist darauf zurückzuführen, dass das verwendete PACE3D Tool die Randporen nicht betrachtet. Diese sind als solche definiert, die nicht vollständig im Gebiet liegen und dementsprechend einen kleineren Radius haben. In den verwendeten GPU beschleunigten Methoden ist dies nicht implementiert. Der Fehler hängt entsprechend mit der Domänengröße und den darin enthaltenen Randporen zusammen. Nachfolgend werden die ermittelten Konditionierungen als genau genug betrachtet und im Training verwendet.

Für das erste Training kommen nur die Datensätze von *Probe A* und *Probe C* zum Einsatz, *Probe B* wird zurückgehalten und erst später für ein Finetuning des

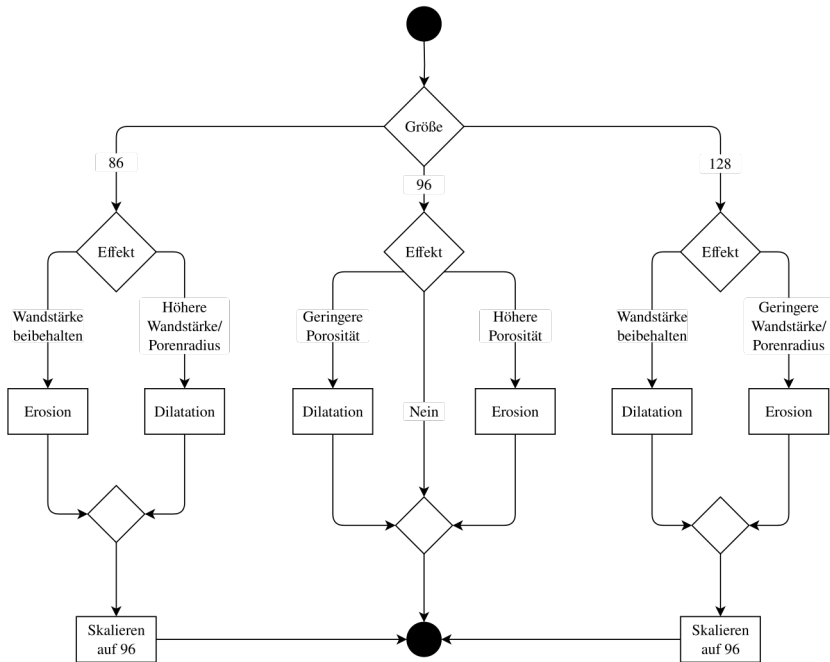
DD-GANs verwendet. Die in Abbildung 5.17 gezeigte Analyse der Struktureigenschaften zeigt, dass sich die Werte innerhalb eines CT-Scans eng um einen Mittelwert gruppieren und bestimmte Eigenschaftsbereiche kaum abgedeckt sind.



**Abbildung 5.17:** Verteilung der Struktureigenschaften in den Subvolumen aus *Probe A* und *Probe C*. Für jeden CT-Scan sind die Werte um einen Mittelwert normalverteilt.

Diese ungleichmäßige Verteilung birgt mehrere Risiken. Durch die ungleichmäßige Datenverteilung könnte der Diskriminator lernen, stets einen der Mittelwerte der Eigenschaften vorherzusagen. Dies führt zwar zu einer vermeintlichen hohen Genauigkeit der Eigenschaftsvorhersage, der Diskriminator lernt jedoch nicht wirklich die Eigenschaften zu extrahieren. Die Verteilung der Strukturdaten bewirkt zudem, dass der Diskriminator erkennen kann, dass reale Daten immer bestimmte Eigenschaften besitzen. Eine synthetische Struktur mit anderen Eigenschaften kann direkt als solche von den Eigenschaften abgeleitet werden. Ein

weiteres Problem ist, dass es dem Generator schlicht an Beispieldaten für bestimmte Strukturen fehlt. Er hat entsprechend keine geeignete Ausgangsstruktur um mache Eigenschaften sinnvoll zu lernen. Um diese Risiken zu minimieren, werden die Daten gezielt so vorverarbeitet, dass die Struktureigenschaften gleichmäßig verteilt sind. Die Logik dieser Verarbeitung ist in Abbildung 5.18 dargestellt.



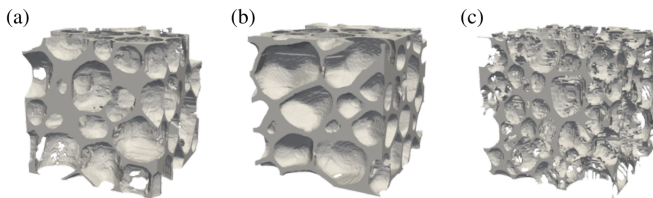
**Abbildung 5.18:** Logik der Bildverarbeitung zum Schließen der Eigenschaftslücken. Es werden Subvolumen einer zufälligen Kantenlänge  $l \in \{86; 96; 128\}$  Voxel ausgeschnitten. Durch Dilatation oder Erosion werden die Eigenschaften verändert. Abschließend werden die Subvolumen auf eine Kantenlänge von  $l = 96$  skaliert.

Die in Abbildung 5.18 dargestellte Vorgehensweise beginnt mit der zufälligen Auswahl eines Subvolumens aus den CT-Daten. Dabei wird die Kantenlänge gleichverteilt aus den Werten  $l \in \{86, 96, 128\}$  Voxel gewählt. Anschließend wird

das Subvolumen durch gezielte Bildbearbeitung so verändert, dass entweder die ursprüngliche Wandstärke weitgehend erhalten bleibt oder gezielt stark abweichende Eigenschaften erzeugt werden. Hierzu kommen Dilatations- und Erosionsoperationen zum Einsatz, deren Auswahl von der Subvolumengröße abhängt. Bei 96 Voxel wird in der Hälfte der Fälle keine Änderung vorgenommen, während in jeweils 25 % der Fälle Dilatation oder Erosion angewandt werden.

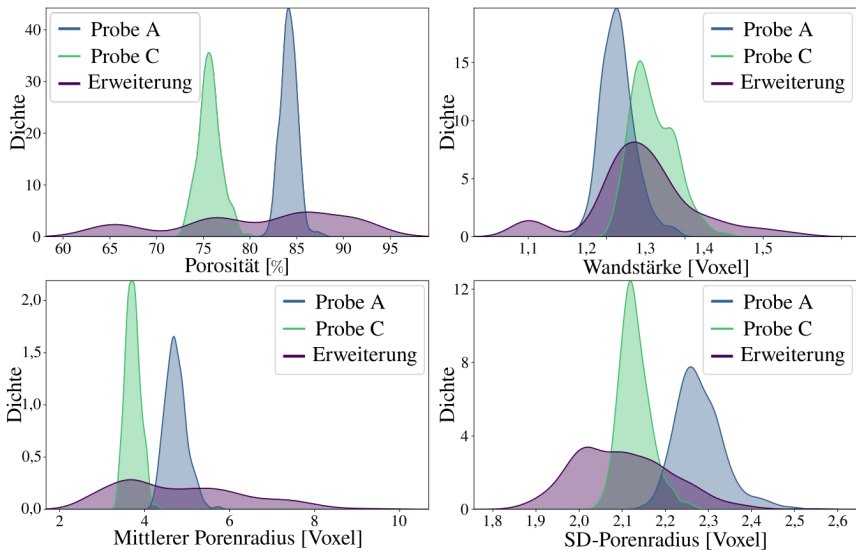
Bei 86 Voxel erfolgt in 75 % der Fälle eine Erosion, um die durch das Hineinzoomen erhöhte Wandstärke zu normalisieren, während seltener eine Dilatation zur Erzeugung besonders dicker Wände eingesetzt wird. Eine Dilatation kommt hier nur in 25 % der Strukturen zum Einsatz, da hier zwar die Strukturvarianz erhöht wird, allerdings auch unrealistische Strukturen erzeugt werden können.

Für 128 Voxel gilt das umgekehrte Vorgehen. In 75 % wird dilatiert, um die durch Herauszoomen verringerten Wandstärken zu erhöhen, und nur selten erodiert, um besonders dünne Wände zu erzeugen. Nachdem der gewählte Effekt angewandt wurde, wird das Subvolumen auf eine einheitliche Kantenlänge von 96 Voxel skaliert. Abschließend reduziert ein Closing-Algorithmus potenzielle Artefakte in den bearbeiteten Strukturen. Dieses Verfahren ermöglicht sowohl eine gezielte Erweiterung des Wertebereichs der Struktureigenschaften als auch eine gleichmäßigere Abdeckung des gesamten Eigenschaftsraums. Abbildung 5.19 zeigt für jede der drei Größen eine exemplarische verarbeitete und skalierte Mikrostruktur. In (a) ist eine erodierte 86 Voxel, in (b) eine dilatierte 96 Voxel und in (c) eine erodierte 128 Voxel Struktur dargestellt.



**Abbildung 5.19:** In der Datenerweiterung erstellte Mikrostrukturen. (a) zeigt einen 86 Voxel Ausschnitt der erodiert wurde. (b) zeigt einen 96 Voxel Ausschnitt auf den Dilatation angewendet wurde. (c) zeigt einen 128 Voxel Ausschnitt nach einer Erosion.

Diese Vorverarbeitung der Daten wird direkt in die beschriebene Strukturvorbereitung integriert und steht im Training somit direkt zur Verfügung. Um die Datenverteilung weiter zu homogenisieren, werden häufig auftretende Parameterkombinationen gezielt gefiltert. Abbildung 5.20 zeigt das Ergebnis der Datenerweiterung. In grün und blau sind die ursprünglichen Datenverteilungen dargestellt, während die neue Datenverteilung in lila dargestellt ist. Die neuen Daten decken den Wertebereich der unterschiedlichen Struktureigenschaften gleichmäßiger ab. Dies verbessert die Robustheit des Trainings und reduziert das Risiko von Mode Collapse sowie einer Überanpassung an enge Eigenschaftsbereiche.



**Abbildung 5.20:** Vergleich der initialen Datenverteilung mit den Ergebnissen der Datenvorverarbeitung. Während die initialen Daten um zwei Mittelwerte normalverteilt sind, deckt die erweiterte Datenverteilung einen größeren Wertebereich gleichmäßiger ab.



### 5.4.2.1 Training

Das Training des beschriebenen DD-GANs basiert auf den zuvor aufbereiteten Daten und erfolgt mit einer Batchsize von 8 über insgesamt 4000 Epochen. Jede Epoche besteht aus 800 Iterationen, wobei eine Iteration dem Durchlauf eines Batches durch das Netzwerk entspricht. Da kein fester, vorab definierter Datensatz existiert, sondern die Trainingsstrukturen in jedem Schritt neu aus der Datenbasis gesampelt werden, muss die Anzahl der Iterationen pro Epoche explizit festgelegt werden.

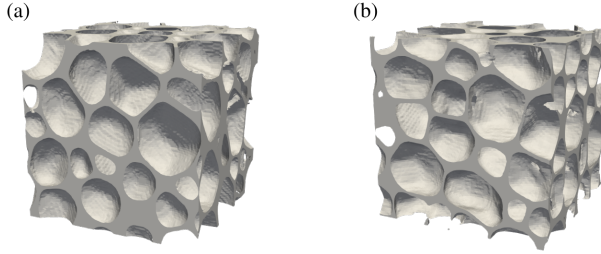
Für die Optimierung werden unterschiedliche Lernraten eingesetzt:  $1,25e-4$  für den Diskriminator und  $2,0e-4$  für den Generator. Diese bewusst unterschiedliche Wahl hängt mit der Beobachtung zusammen, dass der Generator in der Regel langsamer lernt als der Diskriminator. Ohne diese Unterscheidung würde der Diskriminator sehr schnell in der Lage sein, reale und synthetische Strukturen zu unterscheiden, wodurch der Generator kaum Gelegenheit hätte, seine Generierung zu verbessern. Um diesem Ungleichgewicht zusätzlich entgegenzuwirken, werden in den ersten 750 Epochen 8 % der Labels des Diskriminators invertiert. Reale Strukturen werden ihm in diesem Fall als synthetische präsentiert und umgekehrt. Diese Technik zwingt den Diskriminator dazu, seine Entscheidungen sorgfältiger zu treffen, und gibt dem Generator zu Beginn des Trainings zusätzlichen Spielraum zur Verbesserung seiner Ergebnisse.

Das Training erfolgt auf zwei *NVIDIA A100-SXM4* GPUs mit jeweils 80 GB RAM. Auf diesen benötigt das Training pro Epoche 9,3 min, was bei 4000 Epochen einer Gesamtdauer von 26 d entspricht.

### 5.4.3 Validierung

Durch das Training werden ein Generator sowie ein Diskriminator erzeugt, die zur Synthese und Analyse von Schaumstrukturen eingesetzt werden. Der Generator kann auf den ersten Blick hochrealistische Strukturen erzeugen, die visuell kaum von realen Schäumen zu unterscheiden sind. Abbildung 5.21 zeigt exemplarisch

den Vergleich eines aus CT-Daten rekonstruierten Schaums mit einer vom Generator erzeugten Struktur. Um die Leistungsfähigkeit beider Modelle zu bewerten,



**Abbildung 5.21:** Vergleich eines realen Subvolumens (a) mit einer generierten Struktur (b).

werden sie im Folgenden getrennt untersucht. Die Analyse konzentriert sich auf zwei Aspekte: Erstens wird überprüft, ob der Generator die vorgegebenen konditionierten Zielwerte einhält. Zweitens wird bewertet, inwiefern die generierten Strukturen realistisch erscheinen. Der Diskriminator wird ergänzend hinsichtlich seiner Fähigkeit evaluiert, Mikrostruktureigenschaften zuverlässig vorherzusagen.

#### 5.4.3.1 Struktureigenschaften

Zur Bewertung der Eigenschaftseinhaltung werden gezielt Mikrostrukturen mit definierten Konditionierungsvektoren  $\mathbf{c}$  erzeugt. Die Evaluation verfolgt zwei Ziele: Zum einen wird untersucht, wie gut das Modell Strukturen mit Eigenschaften innerhalb des Wertebereichs der Trainingsdaten generieren kann. Zum anderen soll analysiert werden, inwieweit auch außerhalb dieses Bereichs liegende Zielwerte zuverlässig umgesetzt werden, um die Generalisierungsfähigkeit des Generators zu bewerten.

Für die systematische Evaluation wird der kontinuierliche Konditionsraum  $\mathcal{C}$  in diskrete Intervalle unterteilt. Dazu wird der Wertebereich jeder Eigenschaft  $\mathbf{c}_i$  in Intervalle unterteilt. Die  $j$ -te Intervallgrenze der  $i$ -ten Eigenschaft wird als  $a_{i,j}$

bezeichnet, sodass ein Intervall als  $[a_{i,j}; a_{i,j+1}]$  notiert wird. Die Intervallgrenzen werden auf Basis der Verteilung der jeweiligen Eigenschaft berechnet. Der Startwert  $a_{i,1}$  ergibt sich aus

$$a_{i,1} = \mu_i - s_i \cdot \sigma_i,$$

wobei  $\mu_i$  und  $\sigma_i$  den Mittelwert und die Standardabweichung der Eigenschaft  $c_i$  bezeichnen. Der Skalierungsfaktor  $s_i$  legt den Bereich der betrachteten Werte fest. Die Schrittweite  $\delta_i$  für die Intervallbildung ergibt sich durch

$$\delta_i = \frac{1}{k_i} \cdot \sigma_i,$$

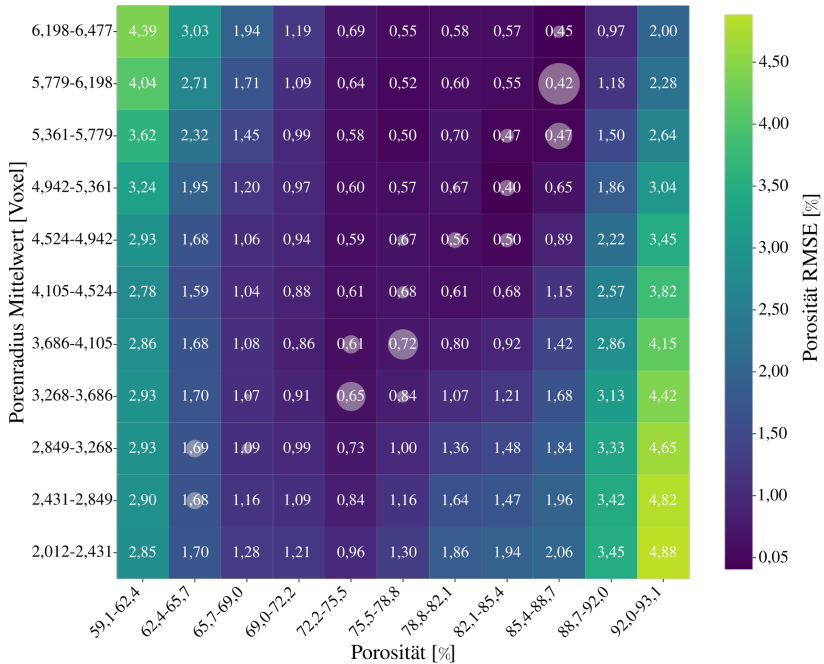
wobei der Parameter  $k_i$  die Auflösung der Diskretisierung bestimmt. Die restlichen Intervallgrenzen werden durch wiederholtes Addieren von  $\delta_i$  bis zum symmetrischen Endwert  $a_{i,M} = \mu_i + s_i \cdot \sigma_i$  berechnet.

Für die hier vorgestellte Untersuchung wurde ein einheitlicher Skalierungsfaktor von  $s_i = 2$  für alle Eigenschaften gewählt. Die Schrittweiten hingegen wurden individuell festgelegt: Für Porosität und mittleren Porenradius wurde  $k_i = 8$  verwendet, für Wandstärke und Porenradius-Standardabweichung  $k_i = 4$ . Durch Kombination der Eigenschaftsintervalle entstehen *Sammlungen* möglicher Konditionen. Eine Sammlung entspricht geometrisch einem Teilvolumen des diskretisierten Eigenschaftsraums. Im Fall von drei betrachteten Eigenschaften kann dies als Würfel interpretiert werden, bei vier Eigenschaften entspricht sie einem Hyperwürfel.

Aus jeder Sammlung werden  $K = 2$  Konditionierungsvektoren  $\mathbf{c}$  gezogen, wobei jede Eigenschaft innerhalb ihres Intervalls gleichverteilt gesampelt wird. Diese Vektoren dienen als Eingabe für den Generator, der damit Mikrostrukturen erzeugt. Zur effizienten Verarbeitung werden mehrere Sammlungen zu sogenannten *Sammlungsgruppen* zusammengefasst. Jede Gruppe enthält maximal  $B/K$  Sammlungen, wobei  $B$  die Batchgröße darstellt. Bei einer Batchgröße von  $B = 16$  und

$K = 2$  können somit acht Sammlungen pro Sammlungsgruppe parallel verarbeitet werden. Dies ermöglicht eine speichereffiziente und parallelisierbare Evaluierung.

Die generierten Mikrostrukturen werden hinsichtlich ihrer tatsächlichen Eigenschaften ausgewertet, indem der MSE für jede Eigenschaft zum Zielwert berechnet wird. Zur Visualisierung der Fehler wird die positive Wurzel des MSE, der RMSE, verwendet, um die gleichen Einheiten wie die gemessene Struktureigenschaft aufzuweisen. Die Ergebnisse der Validierung sind als Heatmap in Abbildung 5.22 dargestellt. Diese zeigt die RMSE der Porosität in Abhängigkeit von Porosität und mittlerem Porenradius. Die beiden restlichen Eigenschaften (Wandstärke und



**Abbildung 5.22:** Visualisierung des RMSE der Porosität. Die Werte sind gemittelt über  $s$  und  $\sigma_{PR}$ .

SD-Porenradius) wurden zufällig gewählt und der Einfluss entsprechend gemittelt.

In der Heatmap selbst sind die Trainingsdaten durch graue Kreise angedeutet. Der Radius der Kreise skaliert dabei mit der Anzahl an Trainingsdaten in der betrachteten Sammlung.

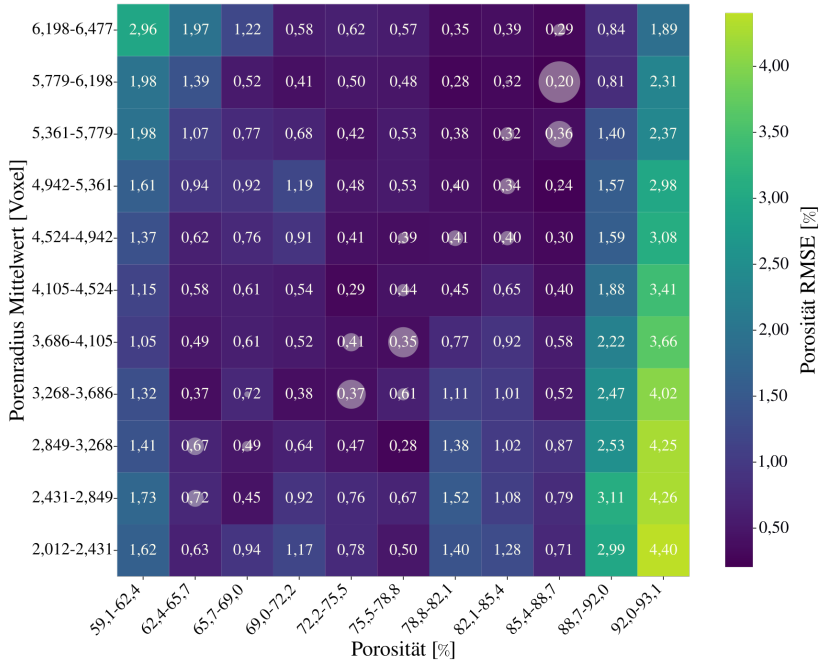
In den zentralen Bereichen des Eigenschaftsraums treten geringe Fehlerwerte auf. Die RMSE der Porosität liegt hier typischerweise bei etwa 0,006, was auf eine präzise Einhaltung der Zielwerte hinweist. In Randbereichen, insbesondere bei Kombinationen aus hoher Porosität und geringem Porenradius, steigen die Fehler erwartungsgemäß an, da hier kaum Trainingsdaten vorhanden sind.

Derzeit werden die Konditionierungen  $s$  und  $\sigma_{PR}$  zufällig gewählt und die Abweichungen der Porosität aus den so generierten Strukturen gemittelt. Um den Einfluss der Eigenschaften Wandstärke und Porenradius-Standardabweichung systematisch zu untersuchen, wird eine zweite Heatmap berechnet. Dabei wird für jede Kombination aus Porosität und mittlerem Porenradius diejenige Kombination von  $s$  und  $\sigma_{PR}$  gewählt, bei der die folgende Metrik minimiert wird:

$$\sum_{i=1}^N \frac{MSE_i}{\sigma_i^2}.$$

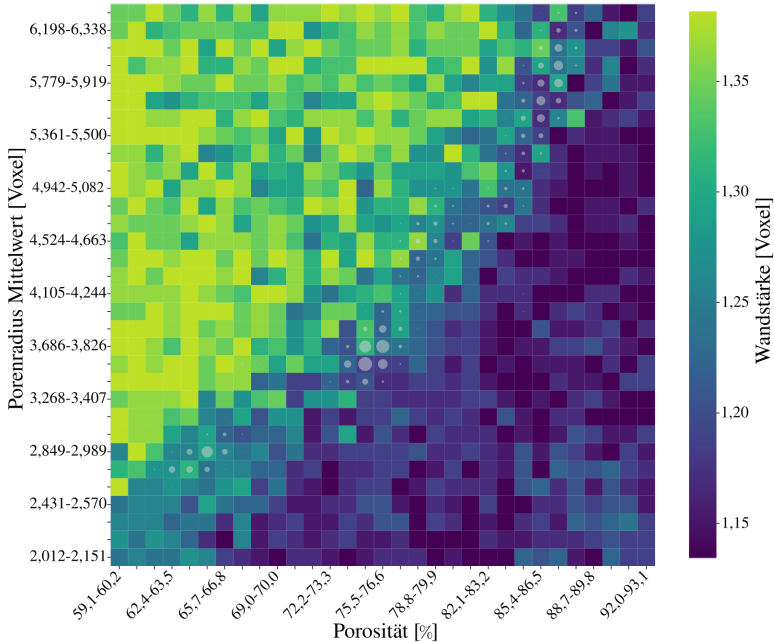
Dabei ist  $MSE_i$  der quadratische Fehler der Eigenschaft  $c_i$  im evaluierten Eigenschaftsbereich, normiert durch die Varianz. Durch diese Normierung nach der Varianz jeder Eigenschaft entsteht ein ausgewogenes Fehlermaß über alle Konditionen hinweg. Durch die Minimierung dieser Fehlersumme durch die Variation der Eigenschaften  $s$  und  $\sigma_{PR}$  kann für jede Sammlung die optimale Einstellung dieser Werte gefunden werden. Die resultierende Heatmap zeigt den minimalen RMSE der Porosität über den optimierten Eigenschaftskombinationen (Abbildung 5.23).

Im Bereich der Trainingsdaten Porosität  $\in [62,4\%; 88,7\%]$  zeigt sich eine flächendeckend gute Einhaltung der Zielporosität. Nur in wenigen Randbereichen, beispielsweise bei hoher Porosität und kleinem mittleren Porenradius, treten größere Fehler. Diese lassen sich durch die in den Trainingsdaten beobachtete Korrelation zwischen Porosität und Porenradius erklären. Höhere Porosität geht hier tendenziell mit größeren Poren einher. Hohe Porositäten bei gleichzeitig



**Abbildung 5.23:** Visualisierung des RMSE der Porosität unter Verwendung optimaler Werte für  $s$  und  $\sigma_{PR}$ .

niedrigen Porenradien kommen nicht vor. Für die anderen Konditionierungen – mittlerer Porenradius, Wandstärke, Standardabweichung des Porenradius – wurden ebenfalls Heatmaps erstellt, die den RMSE zeigen. Diese sind im Anhang zu finden (vgl. Abbildung A.1–A.3). An dieser Stelle ist es interessant zu untersuchen, welche Eigenschaftskombinationen als optimal identifiziert wurden. Dazu zeigen Abbildung 5.24 und Abbildung 5.25 die optimalen Werte für die Wandstärke und die Standardabweichung der Porenradien in den unterschiedlichen Sammlungen. Speziell Abbildung 5.24 zeigt, dass das Modell Korrelationen innerhalb des Eigenschaftsraums erlernt hat: Bei höherer Porosität und größeren Poren werden tendenziell kleinere Wandstärken gewählt. Ein Verhalten, das mit den bei der Augmentierung eingesetzten Operationen (Dilatation und Erosion) übereinstimmt.

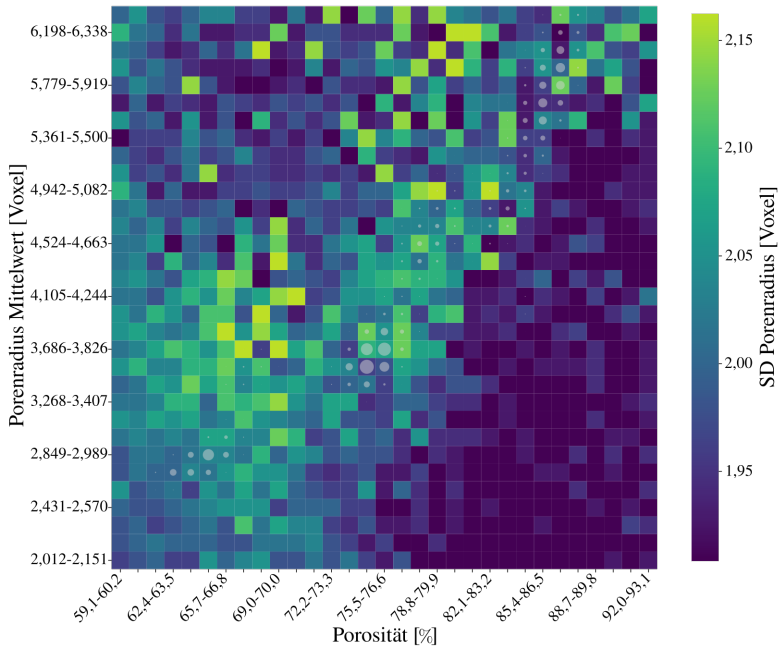


**Abbildung 5.24:** Optimale Parameterwahl für die Wandstärke. Ein klarer Trend ist zu erkennen.

Für die Standardabweichung des Porenradius ist keine solche klare Trennung sichtbar. Im rechten unteren Bereich treten allerdings tendenziell niedrigere Werte der Standardabweichung häufiger auf. Da ein niedriger mittlerer Porenradius eine breitere Porenradienverteilung erschwert ist dies zu erwarten.

Im restlichen Diagramm können allerdings keine klaren Trends erkannt werden, vielmehr sind hohe und niedrige Werte für die Standardabweichung gleichermaßen vertreten. Die Erkenntnis zur optimalen Wahl der Wandstärke und der SD-Porenradius für bestimmte Kombinationen von Porenradius und Porosität sind sehr wertvoll für die gezielte Erstellung von Strukturen.

Die durchgeführte Validierung zeigt, dass der Generator in der Lage ist, Zielwerte für alle vier betrachteten Mikrostruktureigenschaften zuverlässig zu erfüllen. Insbesondere im Kernbereich des Trainingsdatenraums sind die Fehler gering.



**Abbildung 5.25:** Optimale Parameterwahl für die SD des Porenradius. Kein klarer Trend ist erkennbar.

Auch außerhalb des Trainingsbereichs lassen sich plausible Mikrostrukturen erzeugen, wenngleich die Genauigkeit dort leicht abnimmt. Die gezielte Betrachtung und Optimierung aller vier Eigenschaften zeigt, dass das Modell wesentliche Zusammenhänge zwischen den Struktureigenschaften gelernt hat.

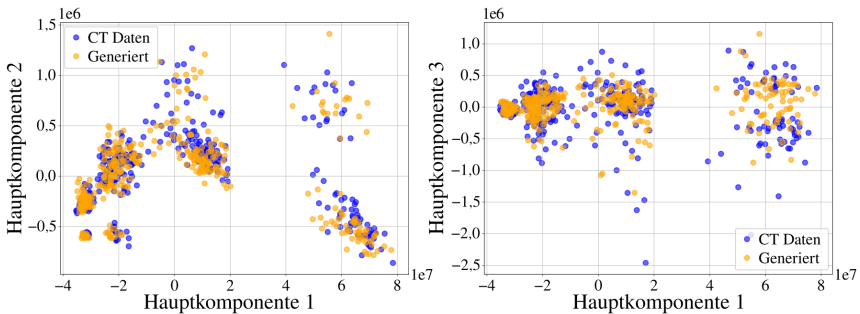
### 5.4.3.2 Realismus

Neben der Einhaltung der Zielgrößen muss geprüft werden, ob die vom Generator erzeugten Mikrostrukturen der Morphologie realer Schaumstrukturen ähneln. Diese Bewertung ist insbesondere mit Blick auf anschließende mechanische



Simulationen wichtig, da diese auf der geometrisch vollständig aufgelösten Mikrostruktur durchgeführt werden. Nur bei strukturell vergleichbaren Materialien ist eine Übertragbarkeit mechanischer Simulationsergebnisse möglich.

Zur Bewertung der strukturellen Qualität wird das in Unterabschnitt 4.2.4 eingeführte Verfahren aus Zweipunktkorrelation (ZPK) und Hauptkomponentenanalyse (PCA) auf den Generator angewendet. Dazu wird ein Testdatensatz aus 400 Subvolumina der rekonstruierten CT-Daten erzeugt, deren ZPK berechnet und gemeinsam einer PCA unterzogen werden. Das so erzeugte PCA-Modell dient als Referenz. Die generierten Strukturen werden in dieses Modell projiziert, sodass beide Datensätze im selben Hauptkomponentenraum vergleichbar sind.



**Abbildung 5.26:** Vergleich der Hauptkomponentenverteilung des Generators (orange) mit den realen Daten (blau).

Abbildung 5.26 zeigt die Verteilung der realen (blau) und generierten (orange) Strukturen im Raum der ersten drei Hauptkomponenten, aufgeteilt in zwei Diagramme: PC1-PC2 und PC1-PC3.

In der Projektion auf die ersten beiden Hauptkomponenten (PC1-PC2) ist eine gute Übereinstimmung der generierten mit den realen Strukturen erkennbar. Die Punktwolken überlagern sich weitgehend, was auf eine strukturelle Ähnlichkeit der beiden Datensätze im Sinne ihrer räumlichen Korrelationen hinweist. Die

Hauptkomponente PC1 erklärt dabei den mit Abstand größten Anteil der Gesamtvarianz mit 99,76 %, während PC2 lediglich 0,03 % und PC3 0,015 % zur erklärten Varianz beitragen.

Kleinere Abweichungen sind in Randbereichen zu beobachten, etwa bei  $PC1 \approx 5,0$  und  $PC2 \approx 1,0$ , wo vereinzelte reale Strukturen keine entsprechenden generierten Gegenstücke aufweisen. Diese Unterschiede betreffen allerdings nur Bereiche mit geringer realer Datendichte und beeinflussen den gesamten Trend nur geringfügig.

Etwas deutlichere Unterschiede treten in der Projektion auf PC1 und PC3 auf. Zwar ist auch hier eine gewisse Überlappung zwischen realen und generierten Strukturen erkennbar, jedoch werden bestimmte Bereiche nicht vollständig abgedeckt. Insbesondere sind das die negativen Werte in PC3 sowie der Bereich zwischen 1 und 7 in PC1. Auch Werte um 0,7 in PC3 und um 1 in PC1 sind unterrepräsentiert.

Da PC3 jedoch nur einen sehr geringen Anteil der Gesamtvarianz beschreibt, spiegeln diese Unterschiede vor allem feine Details der Strukturmorphologie wider. Die Abweichungen deuten darauf hin, dass der Generator die Hauptdatenverteilung sehr gut erfasst, in höherdimensionalen Details der räumlichen Korrelation jedoch nicht die volle Varianz reproduziert. Im Vergleich zu den geometriebasierten Verfahren zeigt sich dennoch eine deutlich bessere Abdeckung der Datenbereiche, was auf einen hohen Realismus der generierten Strukturen hinweist.

Zusammenfassend lässt sich festhalten, dass der Generator in der Lage ist, den wesentlichen Strukturraum der realen Daten zu erfassen. Die starke Überschneidung in der Hauptkomponentenanalyse spricht für einen hohen Grad an struktureller Ähnlichkeit. Die in weniger varianzstarken Hauptkomponenten verbleibenden Unterschiede sind erwartbar und deuten auf Verbesserungspotenziale in der Modellierung feiner Struktureigenschaften hin, ohne die Gesamtergebnisse wesentlich zu beeinträchtigen. Damit gilt die Fähigkeit des Generators, realistische Strukturen zu erzeugen, an dieser Stelle als validiert.

### 5.4.4 Effizienz des Generators

Abschließend wird die Effizienz des trainierten Generators hinsichtlich Laufzeit und Speicherbedarf untersucht. Diese Bewertung ist insbesondere für praktische Anwendungen relevant, bei denen eine große Anzahl an Strukturen generiert werden soll, beispielsweise für simulationsbasierte Materialanalysen.

Analog zur vorangegangenen Validierung des Realismus werden 400 Mikrostrukturen mit einer Kantenlänge von 216 Voxel erzeugt. Diese Größe wurde gewählt, da sie sowohl ein Vielfaches von 32 ist als auch nach Hochskalierung eine Strukturgröße mit einer Kantenlänge von über 400 Voxel ermöglicht. Diese Größe liegt im Bereich des RVE der geometrischen und mechanischen Eigenschaften und macht die so erzeugten Strukturen direkt in Simulationen verwendbar (vgl. Abschnitt 4.5 und Unterabschnitt 6.2.3).

Die Strukturgenerierung erfolgt auf einer *NVIDIA A100-SXM4* GPU mit 80 GB Arbeitsspeicher. Um den Einfluss der Batch-Größe auf Laufzeit und Speicherverbrauch zu untersuchen, wird die Generierung mit verschiedenen Batch-Größen durchgeführt. Die Ergebnisse sind in Tabelle 5.2 dargestellt.

Batch-Größe	Anzahl Batches	Zeit pro Batch [s]	GPU-Speicher [GB]	Zeit pro Struktur [s]
1	400	12,11	4,05	12,11
8	50	17,98	26,91	2,25
16	25	24,30	53,04	1,50
20	20	27,30	66,10	1,36
24	17	OOM (Out of Memory)		

**Tabelle 5.2:** Effizienz des Generators bei verschiedenen Batch-Größen auf einer *NVIDIA A100-SXM4* GPU mit 80 GB. OOM bedeutet, dass nicht genügend Arbeitsspeicher zur Verfügung stand.

Die Generierungszeit pro Struktur sinkt mit steigender Batch-Größe deutlich. Während bei Einzelgenerierung (Batch-Größe 1) rund 12,11 s pro Struktur benötigt werden, reduziert sich dieser Wert bei einer Batch-Größe von 16 bereits auf 1,50 s, bei Batch-Größe 20 sogar auf nur 1,36 s. Gleichzeitig steigt jedoch der Speicherbedarf stark an. Bei Batch-Größe 20 werden über 66 GB benötigt. Eine Batch-Größe von 24 überschreitet die verfügbaren Ressourcen und führt zu einem Speicherüberlauf (OOM). Für eine Nutzung auf weniger leistungsfähiger Hardware ist daher die Einzelgenerierung oder eine kleinere Batch-Größe erforderlich. Dennoch zeigt sich, dass der Generator prinzipiell in der Lage ist, große Mengen an Strukturen innerhalb kurzer Zeit zu erzeugen, ein entscheidender Vorteil gegenüber konventionellen Methoden.

Zum Vergleich wird die Generierungsdauer der geometriebasierten Methode aus Unterabschnitt 5.3.1 betrachtet. Die dabei erzeugten Strukturen haben eine Kantenlänge von 200 Voxel. Zur besseren Vergleichbarkeit wird auch die Dauer für Strukturen mit Kantenlänge 400 Voxel ausgewiesen, da diese für Simulationen erforderlich wären. Die Ergebnisse sowie die Werte des Generators sind in Tabelle 5.3 zusammengefasst.

Generator	Strukturgröße [Voxel]	Zeit pro Mikrostruktur [s]
Generator-Modell (Batch-Größe 1)	$216^3$	12,11
Generator-Modell (Batch-Größe 20)	$216^3$	<b>1,36</b>
Geometriebasierter Generator	$200^3$	205
Geometriebasierter Generator	$400^3$	560

**Tabelle 5.3:** Vergleich der Effizienz des Generator-Modells mit dem geometriebasierten Generator aus Unterabschnitt 5.3.1 bei unterschiedlichen Strukturgrößen.

Im Vergleich zeigt sich, dass der trainierte Generator die geometriebasierte Methode deutlich übertrifft. Bereits ohne Batching erzeugt das Modell eine  $216^3$  Struktur etwa 17 mal schneller als der geometriebasierte Ansatz eine  $200^3$  Struktur.

Bei Verwendung einer Batch-Größe von 20 beträgt der Unterschied zur geometriebasierten Methode für  $400^3$  Strukturen sogar den Faktor 411. Anschaulich ausgedrückt: Während der geometriebasierte Generator noch an der ersten Struktur arbeitet, hat der trainierte Generator bereits alle 400 Teststrukturen fertiggestellt. Hinzu kommt, dass in diesen Zeiten die notwendige Nachbearbeitung der geometriebasierten Strukturen noch nicht berücksichtigt ist (vgl. Unterabschnitt 5.3.2). Je nach Verfahren verlängert diese die Generierungsdauer pro Struktur um weitere 768 s (Simulation) bzw. 268 s (Bildbearbeitung).

Diese massive Beschleunigung macht das Modell besonders attraktiv für den Einsatz in datengetriebenen Materialentwicklungsprozessen. Selbst die notwendige Trainingszeit von 26 Tagen ist angesichts der enormen Zeitersparnis bei der Strukturgenerierung gut investiert.

### 5.4.5 Validierung des Diskriminators

Nachdem der Generator umfassend validiert wurde, wird nun auch der Diskriminator hinsichtlich seiner Eignung zur Vorhersage struktureller Eigenschaften untersucht. Die Bewertung konzentriert sich auf die Genauigkeit der vorhergesagten Konditionierungswerte sowie auf die Übereinstimmung ihrer Verteilung mit den realen Daten.

Der Diskriminator erhält als Eingabe zwei Strukturen  $\mathbf{x}_t$  und  $\mathbf{x}_{t-1}$  sowie den jeweiligen Zeitschritt  $t$ . Da er als Teil des Diffusionsmodells arbeitet, erfolgt die Eigenschaftsvorhersage typischerweise auf verrauschten Strukturen. Entsprechend wird die Vorhersagegüte in Abhängigkeit vom Zeitschritt untersucht. Grundlage für die Analyse bildet der zur Generatorvalidierung erstellte Testdatensatz mit 400 Strukturen.

Tabelle 5.4 zeigt den mittleren absoluten prozentualen Fehler (MAPE) für die vier vorhergesagten Mikrostruktureigenschaften über alle Zeitschritte  $t \in [1; 8]$ . Auffällig ist, dass die höchste Genauigkeit nicht bei vollständig entrauschten Strukturen ( $t = 1$ ), sondern bei mittleren Zeitschritten erreicht wird. Die höchste

Genauigkeit für Porosität und Wandstärke wird bei  $t = 4$  erzielt, während die Standardabweichung des Porenradius bei  $t = 1$  am genauesten geschätzt wird. Der mittlere Porenradius hingegen weist über alle Zeitschritte hinweg eine vergleichsweise hohe Abweichung von über 5 % auf. Insgesamt zeigt sich, dass die Vorhersagegenauigkeit nicht ausreicht, um die analytische Berechnung der Eigenschaften zu ersetzen.

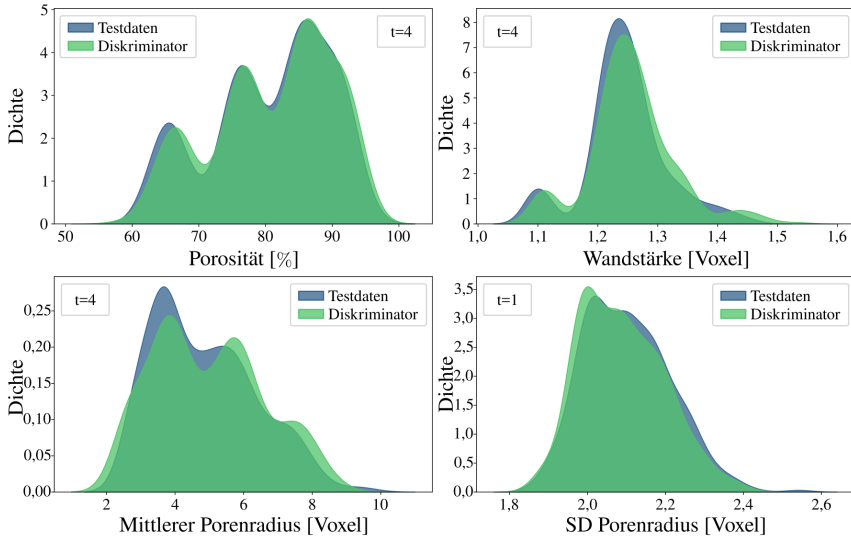
Zeitschritt	Porosität [%]	Wandstärke [%]	Porenradius-Mittelwert [%]	Porenradius-Standardabw. [%]
1	1,751	1,689	6,454	1,197
2	2,021	1,582	6,747	1,264
3	1,733	1,466	6,303	1,326
4	1,215	1,287	5,527	1,277
5	1,309	1,310	5,840	1,346
6	1,513	1,625	9,229	1,437
7	1,962	2,093	11,305	2,023
8	3,062	4,236	12,217	3,695

**Tabelle 5.4:** Mittlerer absoluter prozentualer Fehler (MAPE) der Eigenschaftsvorhersage des Diskriminators je Zeitschritt  $t \in [1; 8]$ . Der Diskriminator erhält  $(\mathbf{x}_{t-1}, \mathbf{x}_t)$  und sagt die Eigenschaften  $\hat{\mathbf{c}}$  voraus.

Ergänzend zur Vorhersagegenauigkeit wird untersucht, ob sich die Verteilung der vorhergesagten Eigenschaften signifikant von der Verteilung der realen Eigenschaften unterscheidet. Abbildung 5.27 zeigt hierzu die geschätzten Dichtefunktionen (KDE) für reale (blau) und vorhergesagte (grüne) Werte, jeweils für den Zeitschritt mit dem geringsten MAPE pro Eigenschaft.

Mit Ausnahme des mittleren Porenradius stimmen die Verteilungen gut überein. Hier zeigt das Modell eine Tendenz zur Überschätzung bei mittleren bis hohen Werten. Auffällig ist zudem die Bildung lokaler Extrempunkte. Während die reale Verteilung im Bereich  $\mu_{\text{pr}} \approx 5\text{Voxel}$  keine ausgeprägte Änderung zeigt, weist die

Vorhersage dort ein lokales Minimum auf. Ein ähnliches Verhalten ist bei der Wandstärke erkennbar. Hier tritt in der Vorhersage ein zusätzliches Maximum bei  $s \approx 1.46$  auf, das in den realen Daten nicht in gleicher Ausprägung existiert. Das Modell differenziert also stärker zwischen mittleren und sehr hohen Wandstärken, als es im realen Datensatz der Fall ist.



**Abbildung 5.27:** Vergleich der Verteilung der vorhergesagten Eigenschaften (grün) mit den berechneten Eigenschaften (blau) des Testdatensatzes des Diskriminators. Bei jeder Eigenschaft werden die Vorhersagen des Zeitschritts mit dem niedrigsten durchschnittlichen Fehler verwendet. Zur Visualisierung wird eine Kerndichteschätzung (KDE) verwendet.

Trotz der erkennbaren Abweichungen ist die generelle Übereinstimmung der Verteilungen ausreichend, um dem Diskriminator eine effektive Rolle im Trainingsprozess des Generators zu ermöglichen. Die Eigenschaftsvorhersage erfolgt zudem äußerst effizient. Der gesamte Testdatensatz kann in unter 12 s verarbeitet werden. Daraus ergibt sich eine durchschnittliche Laufzeit von nur 0,0313 s pro Struktur und 0,0039 s pro Zeitschritt. Diese hohe Geschwindigkeit macht den

Diskriminator zu einem praktikablen Werkzeug für das Training, auch wenn er die analytische Eigenschaftsberechnung nicht ersetzen kann.

Die Validierungen des Generators und des Diskriminators lassen den Schluss zu, dass mit dem DD-GAN ein Modell trainiert wurde, das zuverlässig hochrealistische Schaumstrukturen erzeugen kann. Der Generator kann dabei sowohl innerhalb als auch außerhalb der Trainingsdaten erfolgreich eingesetzt werden. Obwohl die Konditionierungen nicht immer genau umgesetzt werden, hält der Generator diese bei geeigneter Kombination der Konditionierung gut ein. Eine Einschränkung des gewählten Modells ist jedoch, dass es auf Schäume beschränkt ist. Dies ist vor allem durch den Trainingsprozess und hier durch die Konditionierung gegeben, die auf der Analyse des Schaums basieren. Für andere Mikrostrukturen wie geologische Schüttungen oder Batteriematerialien, sind diese Methoden möglicherweise nicht anwendbar und müssten angepasst werden. Aus diesem Grund soll eine andere Form der Konditionierung betrachtet werden. Dabei werden dem Generator statt der Struktureigenschaften direkt Bilder der gewünschten Mikrostruktur übergeben.

### **5.4.6 Bildkonditioniertes DD-GAN**

Der bisher verwendete Generator ist auf die Konditionierung durch explizit berechnete Struktureigenschaften ausgelegt. Dieses Vorgehen ist für Schaumstrukturen gut geeignet, erfordert jedoch bei einer Übertragung auf andere Materialsysteme stets eine Anpassung und Validierung der Eigenschaftsberechnungen. Damit ist die Architektur nicht ohne Weiteres auf neue Anwendungsfälle übertragbar.

Um diese Einschränkung zu umgehen, wird im Folgenden untersucht, ob anstelle expliziter Eigenschaften direkt Bilddaten als Konditionierung verwendet werden können. Ein solcher Ansatz hätte den Vorteil, dass für das Training eines neuen Generators lediglich Bildausschnitte der Zielstruktur benötigt würden, sodass domänenspezifische Analysealgorithmen vollständig entfallen. Dies würde den Einsatz auf andere Materialsysteme deutlich vereinfachen.



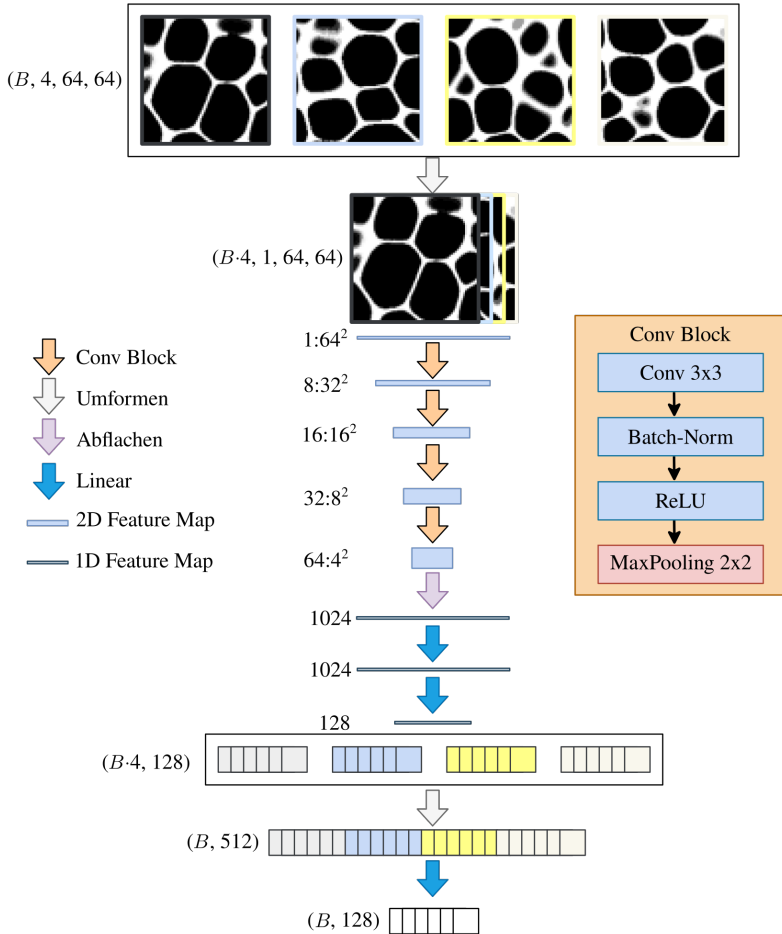
## Modellanpassungen

Die Umstellung auf eine bildbasierte Konditionierung erfordert Änderungen sowohl im Generator als auch im Diskriminator. Im Generator betreffen die Anpassungen ausschließlich die Erzeugung des Konditionierungsvektors, während die restliche Architektur unverändert bleibt (vgl. Abschnitt 5.4.1.1). Anstelle expliziter Struktureigenschaften basiert der neue Konditionierungsraum auf automatisch aus Bildern extrahierten Merkmalen. Hierzu werden pro Trainingsbeispiel vier zweidimensionale Bildausschnitte der Größe  $64 \times 64$  Pixel aus der  $96^3$  Voxel großen 3D-Struktur entnommen. Die Ausschnitte werden entlang einer zufällig gewählten Achse ( $x$ ,  $y$  oder  $z$ ) an einer ebenfalls zufällig gewählten Schnittposition geschnitten. Die Bildgröße von  $64 \times 64$  ist bewusst kleiner als die Gesamtstruktur gewählt, um eine größere Varianz möglicher Konditionierungen zu ermöglichen und gleichzeitig die Rechenlast bei der Verarbeitung durch Convolutional Neural Networks (CNNs) zu reduzieren. Als Potenz von zwei ist sie zudem optimal für wiederholte Downsampling-Operationen wie Max-Pooling geeignet.

Zur Umwandlung der vier Bildausschnitte in ein konditionierendes Merkmalembedding wird ein kleines CNN eingesetzt. Das CNN orientiert sich in seiner Struktur an der *SimpleNet*-Architektur [186], wurde jedoch bewusst kompakter gewählt, um die Parametrisierung schlank zu halten. Während das Originalmodell aus 13 Schichten besteht und etwa 5,4 Millionen Parameter besitzt, ist der hier eingesetzte Bildkodierer auf etwa 1,3 Millionen Parameter reduziert.

Das Kodierernetzwerk besteht aus vier sogenannten *Conv-Blöcken*. Jeder Block umfasst eine  $3 \times 3$ -Faltung, gefolgt von einer Batch-Normalisierung, einer ReLU-Aktivierung sowie einem  $2 \times 2$  Max-Pooling zur Reduktion der räumlichen Auflösung. Nach dem letzten Block wird der resultierende Tensor in einen eindimensionalen Tensor umgewandelt und durch zwei lineare Schichten mit anschließender ReLU verarbeitet, um ein Zwischen-Embedding der Größe  $d_{\text{cond}} = 128$  zu erzeugen.

Die vier Bildausschnitte werden unabhängig voneinander durch das CNN verarbeitet. Zu diesem Zweck wird die Eingabe mit ursprünglicher Form  $(B, 4, 64, 64)$



**Abbildung 5.28:** Netzwerk zur Erstellung der Bildkonditionierung. Es werden vier  $64 \times 64$  große Ausschnitte als Eingabe verwendet. Durch mehrere Faltungsschichten wird ein Embedding mit 128 Einträgen erstellt, das dem Generator übergeben wird.

in eine Batchstruktur ( $B \cdot 4, 1, 64, 64$ ) überführt. Dies erlaubt eine getrennte, aber effiziente Verarbeitung mit 2D-Faltung, ohne Verknüpfungen zwischen den Bildern herzustellen. Eine 3D-Faltung ist nicht sinnvoll, da die Ausschnitte zufällig und unabhängig gewählt werden.

Die resultierenden vier Zwischen-Embeddings der Bilder  $d_{\text{cond}}$  werden anschließend zu einem Tensor der Form  $(B, 4 \cdot d_{\text{cond}})$  zusammengefügt und über eine weitere linearen Schicht auf die finale Größe  $d_{\text{cond}} = 128$  reduziert. Das so erzeugte Embedding der Bilder dient als Konditionierungsvektor für den Generator. Die Schnittstelle zum Generator ist hier die gleiche wie in Abschnitt 5.4.1.1 beschrieben.

Neben dem Generator muss auch der Diskriminator für die Verwendung der Bildkonditionierung angepasst werden. Da in der neuen Architektur keine expliziten Eigenschaften mehr als Zielgrößen verwendet werden, entfällt der Eigenschaftspfad im Diskriminator. Dieser Pfad wurde zuvor eingesetzt, um den Generator zusätzlich durch die Vorhersage der Struktureigenschaften zu lenken. In der bildbasierten Variante entfällt diese Funktion, da keine expliziten Merkmale mehr bekannt sind oder berechnet werden.

Die Entfernung dieses Pfades aus dem Diskriminator hat einige positive Nebenefekte. Die Modellarchitektur wird vereinfacht und das Training kann auf beliebigen Mikrostrukturen angewendet werden, ohne dass Eigenschaftsberechnungen oder analytische Modelle notwendig sind. Dadurch wird der Trainingsprozess auch effizienter, da weniger Parameter aktualisiert und keine Zielgrößen vorberechnet werden müssen.

## Anpassung der Verlustfunktion

Die Umstellung der Generatorarchitektur auf eine bildbasierte Konditionierung hat auch Auswirkungen auf die Verlustfunktion. Mit dem Wegfall der konditionierten Eigenschaftsausgabe im Diskriminator entfällt der zugehörige Verlustterm. Die

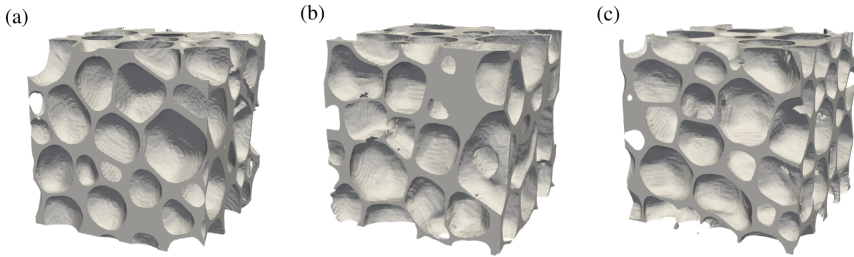
Verlustfunktion des Diskriminators reduziert sich somit wieder auf die ursprüngliche Verlustfunktion eines GANs.

Auch der Generatorverlust wird entsprechend angepasst. Nach Entfernung des konditionellen Anteils basiert dieser wieder ausschließlich auf der GAN Verlustfunktion nach, wie in Gleichung 2.18 beschrieben. Zusätzlich bleibt das bereits in Abschnitt 5.4.1.1 eingeführte Feature Matching erhalten. Dieses trägt insbesondere zur Stabilisierung des Trainings bei und hilft darüber hinaus, Unterschiede zwischen realen und generierten Strukturen in den Zwischenrepräsentationen des Diskriminators zu minimieren. Da keine expliziten Zielgrößen mehr vorliegen, wird erwartet, dass das Feature Matching auch die Qualität der gelernten Konditionierung positiv beeinflusst.

Ein Nachteil der angepassten Verlustfunktion liegt in der geringeren Transparenz des Trainingsprozesses. Durch den Wegfall der Eigenschaftsvorhersage im Diskriminator entfallen zentrale Indikatoren, um die Einhaltung der Zielgrößen quantitativ zu überprüfen. Auch der Verlauf des Feature-Matching-Verlusts erlaubt nur eingeschränkt Rückschlüsse auf die strukturelle Qualität der erzeugten Ergebnisse. Um dennoch eine fortlaufende Kontrolle zu gewährleisten, werden in jeder Epoche generierte Beispielstrukturen sowie die zugehörigen Konditionsbilder abgespeichert. Dadurch kann das Training manuell überwacht werden und möglichen Trainingsinstabilitäten wie Overfitting gezielt entgegengewirkt werden.

## Ergebnis

Auf Grundlage der beschriebenen Anpassungen wurde ein bildkonditioniertes DD-GAN erfolgreich trainiert. Abbildung 5.29 vergleicht eine reale rekonstruierte Schaumstruktur (a) mit einer eigenschaftskonditioniert (b) und einer bildkonditioniert (c) generierten Struktur.



**Abbildung 5.29:** Vergleich einer realen Struktur (a) mit dem eigenschaftskonditionierten (b) und dem neuen bildkonditionierten Generator (c).

Die folgende Validierung des bildkonditionierten Generators verfolgt zwei Ziele. Erstens die Bestimmung optimaler Bildausschnitt-Kombinationen zur Konditionierung und zweitens der direkte Vergleich des bildkonditionierten Generators  $G_{\text{Bild}}$  mit dem bisherigen, eigenschaftsbasierten Generator  $G_{\text{Struktur}}$ .

### 5.4.7 Evaluation des bildkonditionierten Generators

Für die Validierung des trainierten bildkonditionierten Generators  $G_{\text{Bild}}$  wird zunächst die optimale Kombination an Eingabebildern bestimmt. Da die Konditionierung auf vier zweidimensionalen Schnitten der 3D-Struktur basiert, beeinflusst die Auswahl dieser Ausschnitte die resultierenden Mikrostrukturen erheblich. Ungünstige Kombinationen können markante Eigenschaften wie große Poren unzureichend abbilden und so zu ungenauen Generierungen führen.

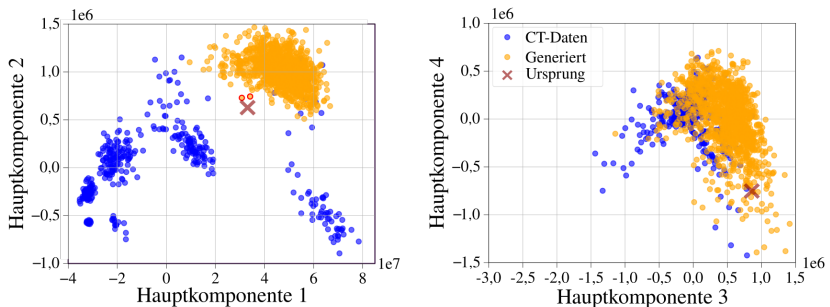
#### Wahl der Eingabebilder

Theoretisch existieren bei vier Ausschnitten und freier Positionierung im  $96^3$  Voxel großen Subvolumen insgesamt  $(96 \times 3)^4 = 6879707136$  Möglichkeiten. Eine vollständige Analyse ist daher in vertretbarer Zeit nicht realisierbar. Um dennoch eine fundierte Auswahl zu ermöglichen, wird nachfolgend das Ziel verfolgt, eine möglichst repräsentative und optimale Kombination zu identifizieren.

Zur Reduktion des Suchraums wird die Position der Schnitte zunächst auf zwei Werte je Achse beschränkt. Diese sind entlang der Achsen die Position 32 sowie 48, die der Mitte des Ausschnitts und der Position zwischen Mitte und Rand entsprechen. Diese Wahl liefert ausreichend Variation in allen Raumrichtungen, hält den Rechenaufwand jedoch überschaubar. Die Anzahl der möglichen Kombinationen reduziert sich dadurch auf  $(2 \times 3)^4 = 1296$ .

Dazu wird  $G_{\text{Bild}}$  mit einer initial gewählten Kombination konditioniert und auf den Testdatensatz angewendet. Für jede generierte Struktur wird eine Zweipunkt-Autokorrelation berechnet und zusammen mit den realen Testdaten einer Hauptkomponentenanalyse (PCA) unterzogen. In der Projektion in den PCA-Raum wird die Struktur mit dem größten euklidischen Abstand zur realen Referenz identifiziert und als „schlechteste“ Generierung ausgewählt.

Für diese „schlechteste“ Struktur wird anschließend eine vollständige Analyse aller 1296 Kombinationen durchgeführt. Jede Kombination dient als Kondition, die resultierenden Strukturen werden per Zweipunktkorrelation analysiert und über PCA projiziert. Abbildung 5.30 zeigt, dass der reale Datenpunkt (rotes Kreuz) außerhalb der generierten Punktwolke (orange) liegt.



**Abbildung 5.30:** PCA der „schlechtesten“ bildgenerierten Strukturen. Das rote Kreuz markiert die zur Konditionierung verwendete Struktur. Die generierten Strukturen (orangene Punkte) verschieben sich in Richtung besser abgedeckter Datenbereiche. Die genauesten Strukturen sind mit einem roten Rand versehen.

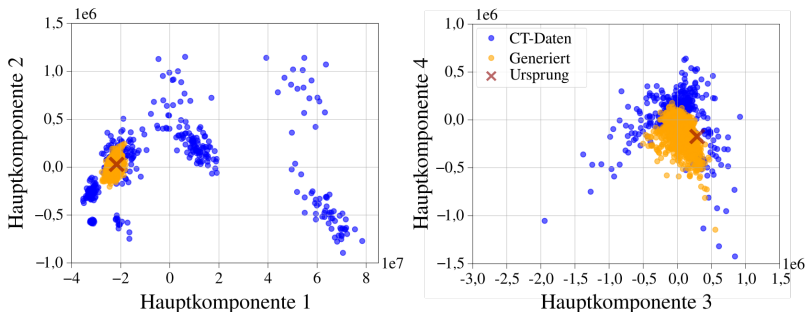
Dies deutet darauf hin, dass der Generator bei ungünstigen Eingaben dazu neigt, in Regionen hoher Trainingsdatendichte auszuweichen. Zwei Kombinationen erzeugen jedoch Strukturen, die der realen Referenz im PCA-Raum besonders nahekommen. Diese sind im PC1-PC2-Diagramm rot umrandet.

Basierend auf dieser Analyse wird aus allen getesteten Kombinationen die optimale Eingabekombination gesucht. Dazu wird für jede generierte Struktur die euklidische Distanz zur realen Struktur im PCA-Raum berechnet. Um semantisch sinnvolle Eingaben zu garantieren, werden Kombinationen mit mehrfach verwendeten Schnitten (Duplikate) ausgeschlossen. Als beste Kombination ohne Duplikate stellt sich

$$[(x, 48), (y, 48), (y, 32), (z, 32)]$$

heraus. Dabei steht der erste Eintrag für die Achse und der zweite für die Position des Schnitts entlang dieser Achse.

Zur Einordnung der Ergebnisse wird zusätzlich eine besonders gut abgedeckte Struktur untersucht. Diese liegt in einem Bereich hoher Trainingsdatendichte. Die in Abbildung 5.31 visualisierte PCA-Analyse zeigt hier, dass sich die generierten Punkte gleichmäßig um die reale Struktur gruppieren. Dies stützt die Hypothese,



**Abbildung 5.31:** PCA der besten bildgenerierten Strukturen. Das rote Kreuz markiert die Struktur die zur Konditionierung verwendet wurde. Die orangenen Punkte zeigen die davon generierten Strukturen. Es ist eine gute Übereinstimmung der Strukturen erkennbar.

dass die Genauigkeit des Generators neben der Auswahl der Konditionierungsausschnitte auch stark von der Trainingsdatendichte abhängt. Je niedriger diese ist, desto größer werden die Abweichungen.

Die identifizierte optimale Ausschnittkombination wird im Folgenden als Standard-Eingabe verwendet, um  $G_{\text{Bild}}$  direkt mit dem bisherigen, eigenschaftsbasierten Generator  $G_{\text{Struktur}}$  zu vergleichen.

### Vergleich von $G_{\text{Bild}}$ und $G_{\text{Struktur}}$

Für die nachfolgende Evaluation wird der trainierte bildkonditionierte Generator  $G_{\text{Bild}}$  mit der zuvor bestimmten optimalen Eingabekombination

$$[(x, 48), (y, 48), (y, 32), (z, 32)]$$

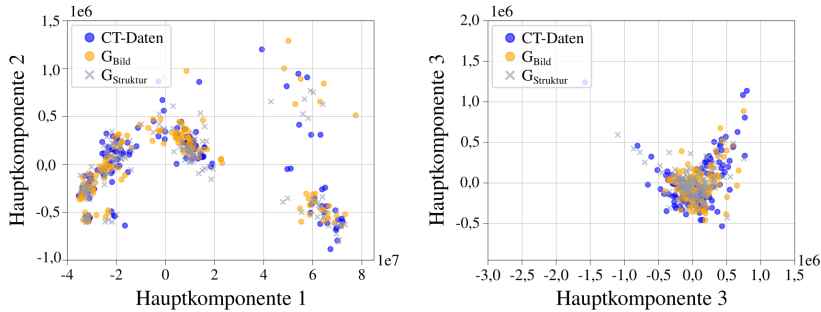
parametrisiert. Ziel ist es, seine Leistungsfähigkeit direkt mit der des eigenschaftsbasierten Generators  $G_{\text{Struktur}}$  zu vergleichen.

Dazu wird der gleiche Testdatensatz wie in der bisherigen Analyse verwendet. Der Ablauf unterscheidet sich lediglich in der Erstellung der Eingabe. Während  $G_{\text{Struktur}}$  explizit berechnete Struktureigenschaften nutzt, erhält  $G_{\text{Bild}}$  seine Eingaben in Form der vier ausgewählten Bildausschnitte.

Nach der Generierung der Strukturen wird für alle Ergebnisse die Zweipunkt-Autokorrelation berechnet und per Hauptkomponentenanalyse (PCA) visualisiert. Die PCA wird dabei ausschließlich auf Basis der realen Testdaten trainiert, um eine konsistente Vergleichsbasis zu gewährleisten. In der Abbildung 5.32 sind reale Daten in Blau,  $G_{\text{Bild}}$  in Orange und  $G_{\text{Struktur}}$  in Grau dargestellt. Zur Übersichtlichkeit ist die Darstellung auf die ersten 150 Strukturen beschränkt.

Die Ergebnisse zeigen, dass beide Modelle den Kernbereich der Datenverteilung gut treffen. Insbesondere bei niedrigen PC1-Werten in Kombination mit PC2-Werten nahe Null. In diesem zentralen Bereich weist  $G_{\text{Struktur}}$  jedoch eine etwas größere Streuung auf, was auf eine bessere Abbildung der Variabilität in den





**Abbildung 5.32:** PCA-Projektion der generierten Strukturen aus  $G_{\text{Bild}}$  (orange) im Vergleich zu den realen Daten (blau) und den Ergebnissen von  $G_{\text{Struktur}}$  (grau). Grundlage ist ein PCA-Modell auf den CT-Daten.

Trainingsdaten hindeutet.  $G_{\text{Bild}}$  hingegen generiert Strukturen, die dichter um ein Zentrum gruppieren, was zu einer geringeren Streuung im Vergleich zu den realen Daten führt. Bereiche mit hohen PC2-Werten werden vom bildkonditionierten Modell zudem seltener erreicht.

Am Rand der Verteilung, insbesondere bei hohen PC1- und gleichzeitig niedrigen PC2-Werten, zeigt sich ein anderes Bild.  $G_{\text{Bild}}$  deckt hier die reale Streuung besser ab als  $G_{\text{Struktur}}$ , dessen Ergebnisse dort deutlich kompakter sind. In der PC3–PC4-Projektion zeigt sich, dass  $G_{\text{Struktur}}$  die komplexe Datenverteilung insgesamt vollständiger abbildet.  $G_{\text{Bild}}$  hingegen erzeugt hingegen eine einseitige Verteilung. Der Bereich mit hohen PC3-Werten wird gut getroffen, bei negativen PC3-Werten finden sich jedoch kaum Punkte.

Zusammenfassend gelingt es  $G_{\text{Bild}}$ , im Kernbereich des Testdatensatzes eine solide Übereinstimmung mit den realen Strukturen zu erreichen. In Bereichen mit größerer struktureller Vielfalt oder seltener auftretenden Merkmalen ist das Modell jedoch limitiert. Ein möglicher Grund hierfür liegt in der Art der Konditionierung. Die gewählten Ausschnitte beschreiben zwar repräsentative Querschnitte, abstrahieren jedoch die Struktur auf einen eher globalen Charakter. Insbesondere abseits häufiger Muster, also in weniger dicht besetzten Regionen des PCA-Raums,

scheint das Modell dazu zu tendieren, bekannte Morphologien zu rekonstruieren, anstatt spezifische Details aus den Eingabebildern exakt wiederzugeben.

Diese Beobachtung deckt sich mit den Ergebnissen aus dem vorhergehenden Abschnitt zur Evaluationsvorbereitung. Dort war sichtbar, dass der Generator bei isolierten Datenpunkten Schwierigkeiten hat, die zugrundeliegenden Merkmale präzise zu rekonstruieren. Die starke Leistung im Zentrum der Verteilung und die limitierten Fähigkeiten in Randbereichen zeigen das Potenzial und gleichzeitig die Grenzen der Bildkonditionierung im gewählten Architekturansatz auf.

Da der verwendete Diskriminator lediglich zur Bewertung des Realismus der generierten Strukturen eingesetzt wird, soll dieser an dieser Stelle nicht gesondert validiert werden. Die Realismus Bewertung ist durch die Bewertung des Generators bereits implizit erfolgt.

## 5.4.8 Finetuning

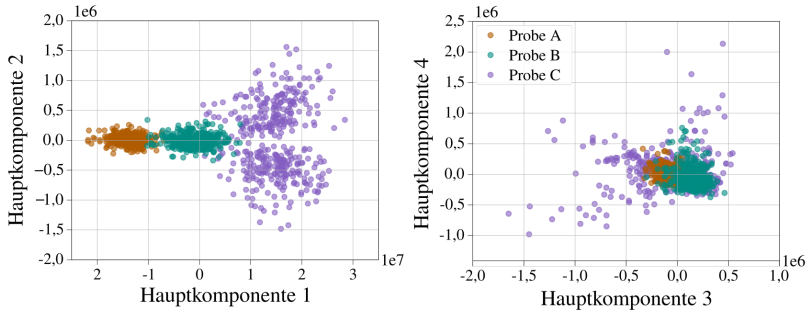
Bisher wurde das *DD-GAN* ausschließlich auf einem erweiterten Datensatz trainiert, der auf den CT-Scans *Probe A* und *Probe C* basiert. Der Datensatz *Probe B* blieb ungenutzt, um im Anschluss die Fähigkeit des Modells zum Finetuning zu untersuchen. Dieser Ansatz ist praxisnah, da Bildkonditionierungen im Gegensatz zu expliziten Struktureigenschaften meist direkt aus Messdaten gewonnen werden können.

### 5.4.8.1 Analyse der Ausgangsdaten

Vor Beginn des Finetunings wird der neue Datensatz *Probe B* im Kontext der bereits verwendeten Datensätze *Probe C* und *Probe A* eingeordnet. Zu diesem Zweck wird eine ZPK mit PCA auf allen nicht augmentierten Subvolumen durchgeführt. In Abbildung 5.33 ist die Verteilung der drei Datensätze dargestellt.

Die drei Datensätze lassen sich anhand dieser Projektion klar voneinander unterscheiden. Die Punkte aus *Probe B* (türkis) liegen im PCA-Raum zwischen den

bisherigen Trainingsdaten *Probe A* (orange) und *Probe C* (lila). Dies deutet auf eine Mikrostrukturverteilung hin, die im bisherigen Training nicht abgedeckt wurde. Im PC3–PC4-Diagramm ist *Probe B* zudem kompakter verteilt, wenngleich sich keine vollständig klare Trennung ergibt.



**Abbildung 5.33:** PCA-Vergleich der nicht augmentierten CT-Daten *Probe A*, *Probe C* und *Probe B*. Die Darstellung erfolgt im Raum der ersten beiden Hauptkomponenten.

Die klare Trennung der Verteilungen rechtfertigt die Verwendung von *Probe B* als separate Zieldomäne für das Finetuning. Das Ziel besteht darin, die Lernfähigkeit des Generators im Umgang mit bisher ungesehenen Strukturen zu analysieren.

#### 5.4.8.2 Trainingsvorgang

Das Finetuning erfolgt auf Basis des vollständig trainierten DD-GANs. Generator und Diskriminator werden mit den bisherigen Gewichten initialisiert und anschließend auf den neuen Datensatz *Probe B* weitertrainiert. Die zuvor gelernten Gewichte der neuronalen Netze bleiben erhalten, sodass das Modell bereits vorhandene Generierungsfähigkeiten beibehält und gezielt anpasst. Die Trainingsparameter entsprechen dabei exakt denen des ursprünglichen Trainings, sodass der Lerneffekt klar auf die neuen Daten zurückzuführen ist. Wie zuvor erfolgt die Konditionierung über die festgelegte Kombination  $[(x, 48), (y, 48), (y, 32), (z, 32)]$ ,

um die Vergleichbarkeit zwischen den Modellen zu gewährleisten. Während des Trainings werden in regelmäßigen Abständen Zwischenstände gespeichert, um die Entwicklung nachvollziehen zu können.

#### 5.4.8.3 Evaluation des Finetunings

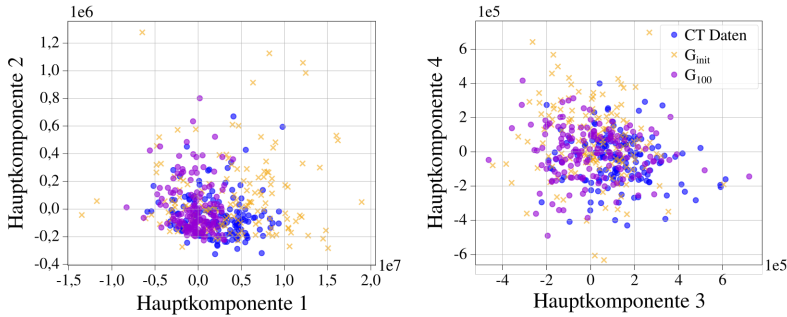
Zur Beurteilung der Finetuning-Leistung wird der Generator erneut mithilfe einer Hauptkomponentenanalyse (PCA) evaluiert. Der zugrundeliegende Testdatensatz besteht aus 150 zufällig ausgewählten Subvolumen aus *Probe B*. Für jede Teststruktur wird die zugehörige Kondition erzeugt und zur Generierung einer Struktur verwendet. Diese generierten Mikrostrukturen werden anschließend im PCA-Raum visualisiert.

Abbildung 5.34 zeigt die resultierenden Punktwolken der realen Daten (blau), der durch das ursprüngliche Modell generierten Strukturen (orange) und der durch das Finetuning angepassten Varianten (lila). Im PC1-PC2-Diagramm ist deutlich zu erkennen, dass das Finetuning die Verteilung der realen Strukturen besser abbildet. Das ursprüngliche Modell weist eine deutlich größere Streuung auf und tendiert dazu, sich auf einen Teilbereich der realen Verteilung zu fokussieren. Insbesondere Datenpunkte mit niedrigen PC1-Werten werden durch das Modell nach dem Finetuning nicht vollständig erreicht. Insgesamt nähert sich das Modell jedoch sichtbar an die Zielverteilung an.

Ein ähnliches Bild zeigt sich im PC1-PC3-Diagramm. Auch hier ist die Streuung der generierten Punkte nach dem Finetuning kompakter und näher an der realen Verteilung. Die Ergebnisse belegen die Fähigkeit des DD-GAN, sich durch gezieltes Finetuning an neue Mikrostrukturverteilungen anzupassen.

#### 5.4.8.4 Fazit zum Finetuning

Das Finetuning auf *Probe B* zeigt, dass der bildkonditionierte Generator in der Lage ist, an neue Strukturverteilungen angepasst zu werden. Der Vergleich der



**Abbildung 5.34:** Vergleich der PCA-Projektionen für *Probe B*: reale Daten (blau), Generator vor dem Finetuning (orange), Generator nach dem Finetuning (lila).

generierten Strukturen vor und nach dem Finetuning im PCA-Raum belegt eine verbesserte Abdeckung der Zieldomäne. Die Streuung des neuen Modells ist reduzierter, und die Ähnlichkeit zur tatsächlichen Verteilung nimmt zu. Diese Ergebnisse unterstreichen das Potenzial des Finetunings, um vortrainierte Modelle schrittweise an domänenspezifische Anforderungen anzupassen.

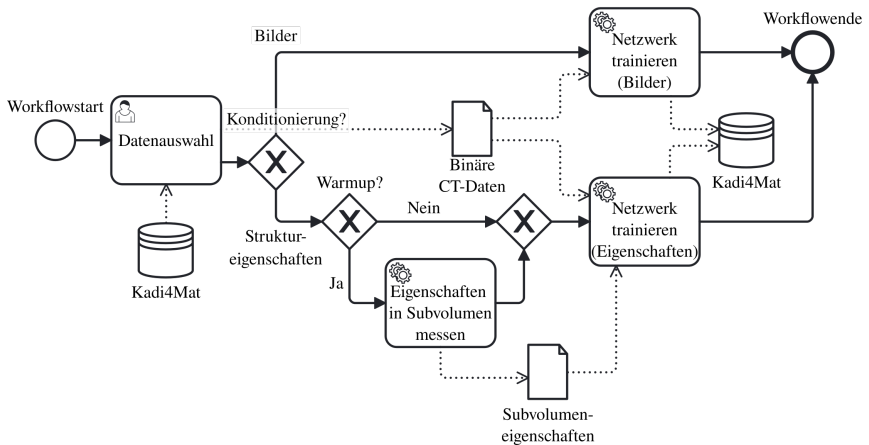
## 5.5 Einbindung in Kadi4Mat

In den vorangegangenen Abschnitten wurden unterschiedliche Möglichkeiten zur Generierung digitaler Schaumzwillinge untersucht. Da sich gezeigt hat, dass die datengetriebene Generierung mittels Diffusionsmodellen den klassischen geometriebasierten Ansätzen in Realismus und Effizienz überlegen ist, wird in diesem Kapitel ausschließlich diese Methode dokumentiert und automatisiert. Damit wird die Generierung nicht nur technisch verfügbar, sondern zugleich FAIR in die übergreifende Methodik eingebettet. Um die entwickelte Generierungsmethode für andere Anwendungsfälle nutzbar zu machen und die in dieser Arbeit erzeugten Strukturen reproduzierbar bereitzustellen, wurden in KadiStudio zwei generische Workflows implementiert: Das Training eines DD-GANs und die Generierung neuer Strukturen.

Beide Workflows sind modular aufgebaut und decken unterschiedliche Varianten ab. Das Training kann bild- oder eigenschaftskonditioniert erfolgen, während die Generierung sowohl zufällig als auch gezielt durchgeführt werden kann. Beide Workflows sind publiziert und frei zugänglich [187]. Im Folgenden werden die Workflows detailliert vorgestellt.

### 5.5.1 DD-GAN Training

Der in Abbildung 5.35 dargestellte Workflow realisiert das Training eines bild- oder eigenschaftskonditionierten DD-GANs. Als Eingabe dienen binäre CT-Daten, die direkt aus Kadi4Mat geladen werden können. Bei der eigenschaftskonditionierten Variante wird zusätzlich eine Eigenschaftsdatei benötigt, die die Werte der verwendeten Subvolumina enthält.



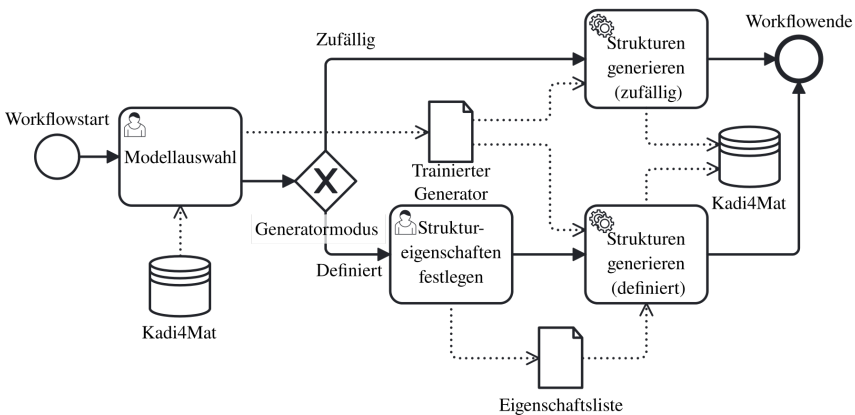
**Abbildung 5.35:** Workflow zum Training des DD-GANs. Das Netzwerk kann sowohl über Bilder als auch über Struktureigenschaften konditioniert werden. Das trainierte Netzwerk wird auf Kadi4Mat hochgeladen und mit den Trainingsdaten verlinkt.

Liegt diese noch nicht vor, kann sie im Workflow über ein vorgeschaltetes *Warmup* automatisch erstellt werden. Das Training eines bildkonditionierten DD-GANs benötigt keine weiteren Eingaben vom Nutzer.

Nach Abschluss des Trainings werden Generator und Diskriminator auf Kadi4Mat hochgeladen und mit den verwendeten Trainingsdaten verlinkt. Jedes Modell wird dabei als eigener Record dokumentiert und erhält Metadaten zu Trainingsparametern (z. B. Lernrate, Epochenzahl, Batchgröße) sowie zur Art der Konditionierung. Die Verlinkungen zwischen Modell, Trainingsdaten und Workflow erlauben eine transparente Nachvollziehbarkeit der Datenherkunft.

## 5.5.2 Generierung von Strukturen

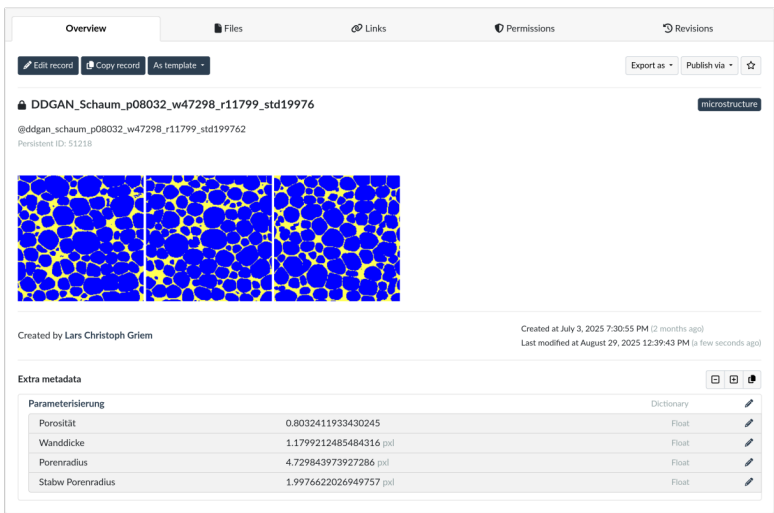
Die eigentliche Generierung von Schaumstrukturen erfolgt im in Abbildung 5.36 dargestellten Workflow. Zunächst wählt der Nutzer den zu verwendenden Generator aus Kadi4Mat aus. Abhängig vom gewählten Modell können anschließend



**Abbildung 5.36:** Workflow zur Generierung von Schaumstrukturen mit einem DD-GAN Generator. Es können sowohl zufällige Strukturen als auch solche mit definierten Struktureigenschaften generiert werden.

entweder zufällige Strukturen erzeugt oder gezielt Strukturen mit vordefinierten Struktureigenschaften generiert werden. Im zweiten Fall wird eine Eigenschaftsliste mit den Zielwerten erstellt und an den Generator übergeben.

Die generierten Strukturen werden jeweils als eigenständige Records auf Kadi4Mat abgelegt. Die zugehörigen Metadaten umfassen die verwendeten Generierungsparameter und wachsen damit analog zum in Unterabschnitt 3.3.1 beschriebenen Prinzip inkrementell an. Zusätzlich wird jede generierte Struktur automatisch mit dem verwendeten Workflow und dem Generator verlinkt. Damit wird nicht nur die Datenherkunft dokumentiert, sondern auch eine spätere Wiederverwendung in anderen Projekten ermöglicht. Abbildung 5.37 zeigt exemplarisch einen solchen Struktur-Record nach der Generierung.



**Abbildung 5.37:** Exemplarischer Record einer generierten Struktur. Die Metadaten umfassen nach der Generierung nur die Parametrisierungswerte.



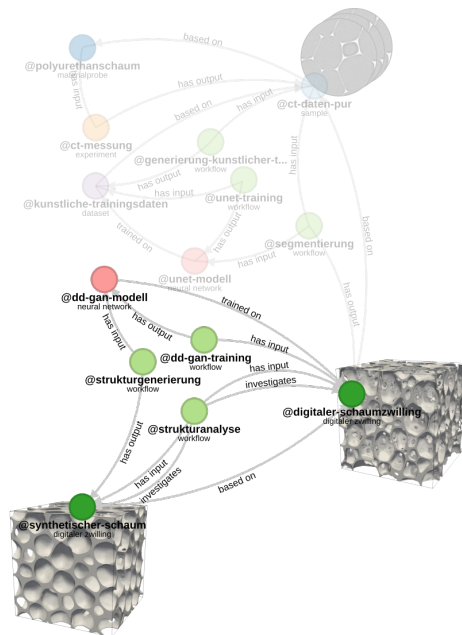
## 5.6 Diskussion

In diesem Kapitel wurde der zweite Schritt der digitalen Materialentwicklung umgesetzt: die datengetriebene Generierung digitaler Zwillinge. Die entwickelten Workflows sind vollständig in Kadi4Mat eingebunden, sodass sowohl Trainings- als auch Generierungsprozesse reproduzierbar dokumentiert und die erzeugten Strukturen direkt mit den verwendeten Daten verknüpft werden. Damit entsteht eine erweiterbare Datenbasis, die unmittelbar für die nachfolgenden Simulations- und Analyseansätze nutzbar ist.

Die Ergebnisse zeigen, dass der entwickelte DD-GAN-Generator hochrealistische Schaumstrukturen erzeugt, die visuell kaum von CT-Daten unterscheidbar sind. Im Gegensatz zu geometriebasierten Methoden lassen sich die Struktureigenschaften zudem präzise einstellen. Besonders hervorzuheben ist die Übertragbarkeit der Methode: Da der Generator ausschließlich auf CT-Daten basiert, kann er prinzipiell direkt auf andere Materialsysteme angewendet werden. Bei der eigenschaftskonditionierten Variante ist hierfür lediglich die Berechnung geeigneter Merkmale erforderlich, während der bildkonditionierte Generator keinerlei analytische Vorverarbeitung benötigt und damit auch für Anwendungen ohne einfach extrahierbare Eigenschaften geeignet ist. Die erfolgreiche Demonstration eines ersten Finetunings bestätigt zudem die Eignung des Ansatzes für ein Transfer Learning.

Einschränkungen bestehen vor allem in der Robustheit und im Ressourcenbedarf. Der bildkonditionierte Generator weicht in schlecht abgedeckten Datenbereichen gelegentlich auf häufigere Strukturen aus. Zudem ist das Training sehr rechenintensiv, sodass für viele Anwendungsfälle eher Finetuning-Strategien in Frage kommen. Diese Limitierungen sind jedoch praktischer Natur und mindern nicht die grundsätzliche Stärke des Ansatzes. Im direkten Vergleich übertrifft der DD-GAN klassische geometriebasierte Methoden sowohl in Realismus als auch in Effizienz und eröffnet die Möglichkeit, realistische Strukturen in großem Umfang und mit gezielter Eigenschaftssteuerung zu erzeugen.

Zusammenfassend wurde gezeigt, dass die vorgestellten Methoden eine vollständige, reproduzierbare Prozesskette zur Generierung digitaler Zwillinge bereitstellen. Mit dem DD-GAN steht ein generischer Ansatz zur Verfügung, der flexibel an unterschiedliche Anwendungsfälle anpassen lässt. Perspektivisch eröffnen sich durch Transfer Learning und multimodale Konditionierungen weitere Möglichkeiten, die Methode auf andere Materialsysteme zu übertragen und ihre Robustheit zu steigern. Der erreichte Stand der Methodik sowie der aktuelle Fortschritt im Materialentwicklungsprozess sind in Abbildung 5.38 dargestellt. Ausgehend von



**Abbildung 5.38:** Datengetriebene Generierung auf Kadi4Mat: Auf rekonstruierten CT-Daten trainiertes DD-GAN erzeugt synthetische digitale Zwillinge.

CT-Daten können nun digitale Zwillinge synthetisch erzeugt und FAIR dokumentiert werden. Die so geschaffene Datenbasis bildet die Grundlage für die im folgenden Kapitel behandelten Simulation makroskopischer Eigenschaften.



## 6 Bestimmung der physikalischen Schaumeigenschaften

Nachdem in den vorangegangenen Kapiteln die Erstellung und mikrostrukturelle Charakterisierung digitaler Materialzwillinge sowie deren gezielte Generierung beschrieben wurde, liegt der Fokus nun auf der makroskopischen Simulation effektiver Eigenschaften. Damit wird die letzte methodische Komponente der entwickelten Methodik eingeführt, die notwendig ist, um eine strukturierte und reproduzierbare Strukturdatenbank aufzubauen, welche systematisch hinsichtlich Struktur-Eigenschaft-Beziehungen untersucht werden kann.

In dieser Arbeit werden ausschließlich die mechanischen Eigenschaften der Schaumstrukturen simulativ bestimmt. Grundlage bildet die in der Simulationssoftware PACE3D implementierte Phasenfeldmethode, die sich in einer Vielzahl materialwissenschaftlicher Anwendungen bewährt hat. Der hier verwendete Mechaniksolver erlaubt die Modellierung elastischer und plastischer Verformungen und stellt damit eine geeignete Methode für die Simulation poröser Strukturen dar. Vor den eigentlichen Simulationen wird zunächst für jede Schaumprobe eine RVE-Analyse durchgeführt, um die Aussagekraft der Ergebnisse sicherzustellen. Anschließend wird das Modell über experimentelle Druckversuche parametrisiert, wobei die ermittelten Materialkennwerte des Polyurethans als Eingabe dienen.

Um die Methode reproduzierbar und effizient einzusetzen, wird der gesamte Simulations- und Auswertungsprozess in einem Workflow automatisiert. Dieser Ansatz ermöglicht es, die Simulationen konsistent auf unterschiedliche Strukturen anzuwenden und die Ergebnisse standardisiert auszuwerten.

## 6.1 Stand der Forschung

Die numerische Simulation ist ein zentrales Werkzeug zur Bestimmung physikalischer Eigenschaften von Materialien. Je nach Problemstellung kommen dabei unterschiedliche Modellierungsansätze zum Einsatz, darunter klassische Methoden wie die Finite-Elemente-Methode (FEM) im Kontext singularer Grenzflächen sowie die Phasenfeldmethode. Während FEM auf einem starren oder lokal angepassten Gitter arbeitet und Phasengrenzen explizit verfolgt werden müssen, beschreibt die Phasenfeldmethode Phasenübergänge kontinuierlich über einen Ordnungsparameter. Dadurch entfällt die aufwändige Nachführung beweglicher Gitter, was die Methode insbesondere für Probleme mit komplexer Grenzflächenentwicklung attraktiv macht.

Ein wesentlicher Vorteil der Phasenfeldmethode liegt in ihrer Flexibilität. Durch geeignete Formulierung der Energieterme und Kopplung mit weiteren physikalischen Gleichungen kann sie an eine Vielzahl von Anwendungsfällen angepasst werden. Entsprechend wird sie in unterschiedlichsten Bereichen der Materialwissenschaften eingesetzt. Ben Said [35], Altschuh [97], Kunz [133] und Reder [40] befassten sich in ihren Arbeiten mit der Anwendung der Phasenfeldmethode zur Fluidsimulation. Ben Said, Altschuh und Kunz setzten dabei die kapillargetriebene Benetzung von porösen Strukturen um, die unter anderem die Simulation von medizinischen Diagnostikmembranen und den Flüssigkeitstransport in geologischen Strukturen ermöglicht. Reder erweiterte das Fluidmodell um die Wechselwirkungen mit Festkörpern. Dies ermöglicht unter anderem die Simulation des Sedimenttransports in Flüssen und des Herstellungsprozesses von Batterieelektroden.

Die Simulation von Batteriematerialien insbesondere der Eigenschaften und der zeitlichen Entwicklung des Batteriematerials wurde von Hoffrogge [188] und Daubner [189] umgesetzt. Die Entwicklungen von Hoffrogge ermöglichen es die Nickelvergrößerung in Festoxidbrennstoffanoden zu simulieren. Dadurch können mögliche Einflussparameter auf die Mikrostruktur und die Haltbarkeit der

Brennstoffzelle gezielt untersucht werden. Daubner setzt hingegen eine Modellformulierung um, die die Simulation der Alterung von Natriumbatterien aufgrund von Laderaten induzierten Spannungen im Batteriematerial ermöglicht. Weitere Anwendungsgebiete und Modellformulierungen entstammen der Untersuchung geologischer Prozesse. Späth [168] und Prajapati [190] beschäftigen sich dabei mit der Ausbreitung von Rissen in Gesteinen und dem Kristallwachstum neuer Gesteinskörner in den entstandenen Rissen.

In Arbeiten von Schneider [191], Schoof [38] und Herrmann [192] wurde ein Mechaniksolver mit der Phasenfeldmethode umgesetzt, der sowohl elastische als auch elastoplastische Verformungen umsetzt. Dieser wurde erfolgreich zur Untersuchung unterschiedlicher Phänomene wie der Wärmebehandlung von Stählen und der Rissbildung in Gusseisen angewandt. Neben den genannten Veröffentlichungen existieren weitere Arbeiten hinsichtlich der Benetzung von Oberflächen [193–195], dem Wachstum von Kornstrukturen [196, 197] und Sinterprozessen [173, 198]. Alle diese Methoden sind in der Software PACE3D implementiert und können verwendet werden. Die Phasenfeldmethode in Form von PACE3D bietet somit eine geeignete Grundlage für die Untersuchung der physikalischen Eigenschaften und wird in dieser Arbeit daher verwendet. Das für diese Arbeit relevante Modell basiert auf dem erwähnten Mechaniksolver und wurde bereits von Reder & Holland-Cunz et al. [164] erfolgreich zur Simulation poröser Mikrostrukturen angewandt. Nachfolgend erfolgt die Beschreibung des Modells.

## **6.2 Phasenfeldmethode für Mechanikmodellierung**

Der in PACE3D implementierte Mechaniksolver ermöglicht die simulative Untersuchung der mechanischen Eigenschaften von Mikrostrukturen. In dieser Arbeit wird er genutzt, um die makroskopischen mechanischen Eigenschaften der digitalen Schaumzwillinge zu bestimmen. Die folgenden Abschnitte beschreiben das zugrunde liegende Modell sowie das verwendete Simulationssetup.

### 6.2.1 Modell

Die in Abschnitt 2.3 skizzierte Phasenfeldmethode lässt sich durch geeignetes Anpassen des Energiefunktional (Gleichung 2.24) auf ein breites Anwendungsspektrum anwenden. Die in dieser Arbeit verwendete Simulationssoftware PACE3D bietet bereits ein großes Repertoire unterschiedlicher Problemmodellierungen.

Zur Simulation der Spannungen und Dehnungen im Phasenfeldkontext steht ein von Schneider et al. [199] entwickeltes Modell zur Verfügung. Diese Modell wurde von Hermann [192] und Schoof [38] weiterentwickelt, um auch plastische Deformationen in Materialien zu simulieren. Diese Modellierung basiert auf kleinen Deformationen und einer geometrischen Linearisierung.

Das Modell erweitert das Energiefunktional (vgl. Gleichung 2.24) um die elastischen und plastischen Energiedichten des Materials zu

$$\mathcal{F}(\phi, \nabla\phi, \mathbf{u}) = \int_V W_{\text{intf}}(\phi, \nabla\phi) + \bar{W}_{\text{elast}}(\phi, \mathbf{u}) + \bar{W}_{\text{plast}}(\phi, \mathbf{u}) \, dV. \quad (6.1)$$

Dabei beschreibt  $W_{\text{intf}}(\phi, \nabla\phi)$  die Grenzflächenenergiedichte, welche die Energie über die diffuse Grenzfläche modelliert.  $\bar{W}_{\text{elast}}(\phi, \mathbf{u})$  und  $\bar{W}_{\text{plast}}(\phi, \mathbf{u})$  modellieren die elastische bzw. plastische Energie in den Bulkphasen. Durch  $\mathbf{u}$  wird hier das Verschiebungsfeld beschrieben. Eine detaillierte Definition dieser Energieterm erfolgt nicht, da das Modell in dieser Arbeit lediglich zur Bestimmung physikalischer Eigenschaften verwendet und nicht weiterentwickelt wurde. Ausführliche Beschreibungen finden sich in [38].

Zur Simulation des beschriebenen Systems erfolgt die Variation des Energiefunktional aus Gleichung 6.1:

$$\forall \delta\phi, \delta\mathbf{u} : \quad \delta\mathcal{F} = \int_V (\cdot) \delta\phi + (\cdot) \delta\mathbf{u} \, dV = 0. \quad (6.2)$$

Dabei entspricht  $\phi$  dem Ordnungsparameter und  $\mathbf{u}$  dem Verschiebungsfeld. Da das Phasenfeld  $\phi$  hier konstant bleibt, entfällt der Term  $(\cdot)\delta\phi$ . Der verbleibende Term kann wie folgt geschrieben werden:

$$(\cdot)\delta\mathbf{u} = \text{div}(\mathcal{C}[\tilde{\varepsilon} - \tilde{\varepsilon}]) \cdot \delta\mathbf{u}, \quad \varepsilon = \frac{1}{2}(\nabla\mathbf{u} + \nabla^\top\mathbf{u}) \quad (6.3)$$

mit  $\tilde{\varepsilon}$  als inelastischem Dehnungsanteil und  $\mathcal{C}$  als Steifigkeitstensor. Diese Minimierung der freien Energie mittels Variationsrechnung ergibt somit die lokale Impulsbilanz als Bestimmungsgleichung. Der Spannungstensor wird über eine lineare Interpolation der Steifigkeiten,

$$\mathcal{C} = \sum_{\alpha=1}^N \phi^\alpha \mathcal{C}^\alpha \quad (6.4)$$

berechnet [200]. Hier sind  $\mathcal{C}^\alpha$  die konstanten phasenspezifischen Steifigkeitstensoren.

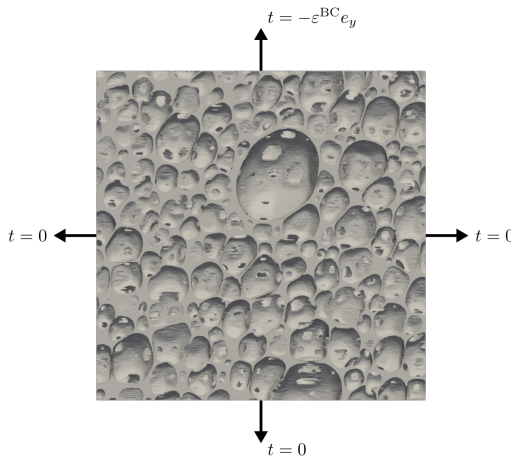
Für die untersuchten Schaumstrukturen hat die Porenphase eine verschwindend kleine Steifigkeit, weshalb die Steifigkeit der Strukturphase über die Grenzflächenbreite linear abnimmt. Über die Impulsbilanz  $\text{div}(\boldsymbol{\sigma}) = 0$  und  $\boldsymbol{\sigma} = \mathcal{C}[\varepsilon]$  mit den elastischen Dehnungen  $\varepsilon := \tilde{\varepsilon} - \tilde{\varepsilon}$  erfolgt die Lösung nach dem Verschiebungsfeld  $\mathbf{u}$ . Für jeden Zeitschritt  $t$  ergibt sich so in jeder Zelle ein Spannungs- und Dehnungstensor, woraus Spannungs-Dehnungsdiagramme bestimmt werden können.

Aus dem Spannungs-Dehnungsdiagramm wird der Druckmodul aus der Steigung des linearen Verformungsbereichs bestimmt. Die Stauchgrenze ergibt sich an dem Punkt, an dem die um 0,2 % nach rechts verschobene Gerade mit der Steigung des Druckmodul die Spannungs-Dehnungskurve schneidet.



## 6.2.2 Simulationssetup

Zur Simulation der mechanischen Eigenschaften wird eine uniaxiale Verschiebung auf die Schäume aufgebracht. Da diese Verschiebung in experimentellen Untersuchungen lediglich an der Oberseite aufgebracht wird, erfolgt dieser Belastungsfall auch in der Simulation. Abbildung 6.1 zeigt den schematischen Aufbau der Simulation. Die Verschiebung wird in der Simulation linear aufgebracht und imitiert den Ablauf eines Druckversuchs.



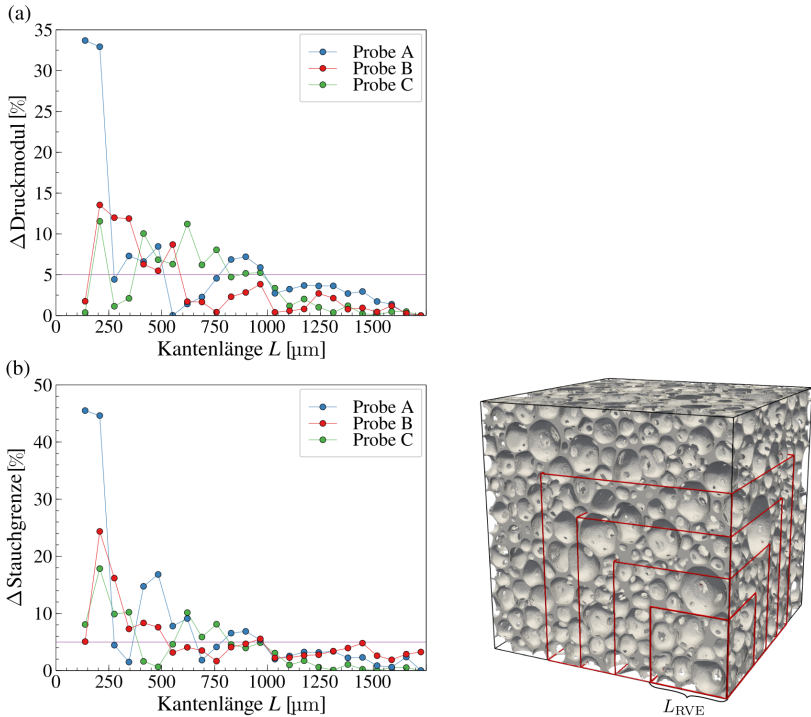
**Abbildung 6.1:** Simulationssetup: Auf den untersuchten Schaum wird eine Verschiebung entlang der y-Achse, analog zum modellierten Druckversuch, aufgebracht. Die Verschiebung wird über eine lineare Funktion langsam an das Maximum herangeführt, um den Druckversuch zu imitieren.

## 6.2.3 RVE Bestimmung

Wie bereits bei der Bestimmung der Struktureigenschaften, ist auch für eine verlässliche Simulation der mechanischen Eigenschaften eine RVE-Analyse erforderlich. Hierfür wird der deterministische Ansatz aus Abschnitt 2.4 verwendet.

Aus der rekonstruierten Schaumprobe werden zunächst Subvolumina mit einer Kantenlänge von 150 Voxel extrahiert. Diese Kantenlänge wird in Schritten von 75 Voxel sukzessive vergrößert, und für jedes Subvolumen werden die mechanischen Eigenschaften simuliert.

Die ermittelten Werte werden anschließend mit den Eigenschaften der Gesamtprobe verglichen, um ein relatives Fehlermaß zu berechnen. Fällt die Abweichung unter einen Schwellenwert von 5 %, gilt die zugehörige Kantenlänge  $L$  als RVE-Größe. Abbildung 6.2 zeigt die RVE-Analysen für alle drei Schaumproben. Es



**Abbildung 6.2:** Bestimmung der RVE-Größe: Sobald die Abweichung unter 5 % liegt ist ein RVE erreicht.

ist zu erkennen, dass die relative Abweichung der Simulationen vom Zielwert, ab

einer Kantenlänge von  $1\,100\,\mu\text{m}$  (400 Voxeln) konstant unter 5 % liegt. Nachfolgend werden die Untersuchungen der Schaumstrukturen demnach in Subvolumen mit einer Kantenlänge von 400 Voxeln durchgeführt.

#### **6.2.4 Parametrisierung der Simulation**

Für eine realistische Simulation müssen die mechanischen Eigenschaften der beiden Phasen Polyurethan und Luft bekannt sein. Die Materialparameter für Luft werden der Literatur entnommen. Der Druckmodul liegt hier bei  $1,01\text{e}-4\text{ GPa}$ . Eine Streckgrenze existiert nicht, da Luft als Gas keine plastische Verformung aufweist. Für Polyurethan liegen keine Bulk-Werte aus der Literatur vor, da dieses Material meist geschäumt vorliegt. Die Werte werden daher experimentell approximiert. Dazu werden Druckversuche an realen Polyurethanschaumproben durchgeführt. Die so ermittelten Spannungs-Dehnungsdiagramme dienen als Referenz. Die Simulationsparameter des Polyurethans werden anschließend iterativ angepasst, bis die simulierten Kurven den experimentellen möglichst genau entsprechen.

### **6.3 Experimentelle Untersuchungen**

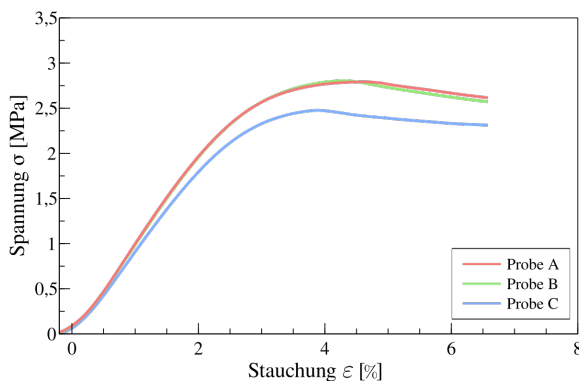
Um die Simulationen korrekt zu parametrisieren und zu validieren, werden experimentelle Druckversuche an repräsentativen Proben des Polyurethanschaums durchgeführt. Diese Experimente liefern Referenzwerte für die Parametrisierung der Simulation und ermöglichen den Abgleich zwischen numerischen und physikalischen Ergebnissen. Im Folgenden werden das Vorgehen und die erhaltenen Ergebnisse präsentiert.

### 6.3.1 Ziel und Versuchsaufbau

Zur Parametrisierung der Simulation hinsichtlich der mechanischen Eigenschaften des Polyurethanschaums werden normgerechte Druckversuche nach DIN EN ISO 844:2021 durchgeführt. Proben werden an verschiedenen Positionen des Schaummaterials entnommen, entsprechend den in den CT-Scans *Probe A*, *Probe B* und *Probe C* untersuchten Bereichen. Die Schaumproben haben eine Größe von  $10 \times 10 \times 5$  cm und sind somit deutlich größer als die in den CT-Daten abgebildeten Ausschnitte, wodurch die Messwerte repräsentativ für die jeweiligen Entnahmebereiche sind. Die Belastung erfolgt quasistatisch mit einer Prüfgeschwindigkeit von 5 mm/min entlang der Aufschäumrichtung, während kontinuierlich Traversenweg und Kraft gemessen werden, woraus Kraft-Weg-Kurven und anschließend Spannungs-Stauchungs-Diagramme abgeleitet werden.

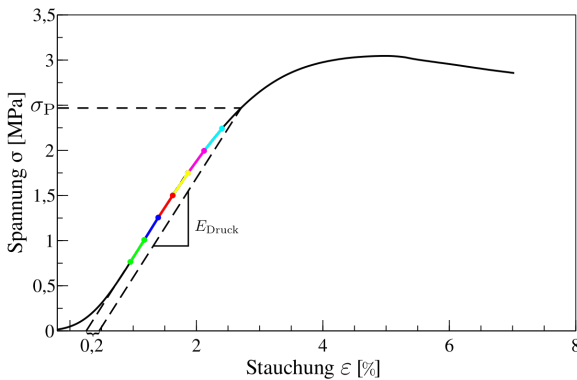
### 6.3.2 Auswertung der Druckversuche

Aus den Spannungs-Stauchungs-Kurve wird der Druckmodul und die Stauchgrenze ermittelt. Abbildung 6.3 zeigt die aufgezeichneten Kurven für alle drei Schaumproben entlang der Aufschäumrichtung ( $z$ -Achse). Alle drei dargestellten



**Abbildung 6.3:** Druckversuche des Schaumproben in Aufschäumrichtung.

Kurven weisen einen Kurvenfuß um den Koordinatenursprung auf, der in der Berechnung des Druckmoduls berücksichtigt werden muss. Dazu wird der Druckmodul aus den dargestellten Kurven, wie in der Norm DIN EN:ISO 844:2021 beschrieben, aus dem steilsten Abschnitt der Stauchung zwischen 25 % und 75 % der maximalen Kraft bestimmt. Dieser Abschnitt wird hierfür in sechs gleich große Segmente unterteilt und in diesen jeweils die Steigung errechnet, siehe Abbildung 6.4. Die beiden benachbarten Segmente deren kombinierte Steigung



**Abbildung 6.4:** Methode zur Bestimmung des Druckmoduls nach DIN EN:ISO 844. Der Kurvenbereich zwischen 25 % und 75 % der Maximalkraft wird in 6 gleich große Segmente geteilt. Die benachbarten Segmente, deren Summe ihrer Steigung am größten ist, definieren den Druckmodul des untersuchten Materials. Der Schnittpunkt einer um 0,2 % nach rechts verschobener Geraden mit der Steigung des Druckmoduls, definiert die Stauchgrenze.

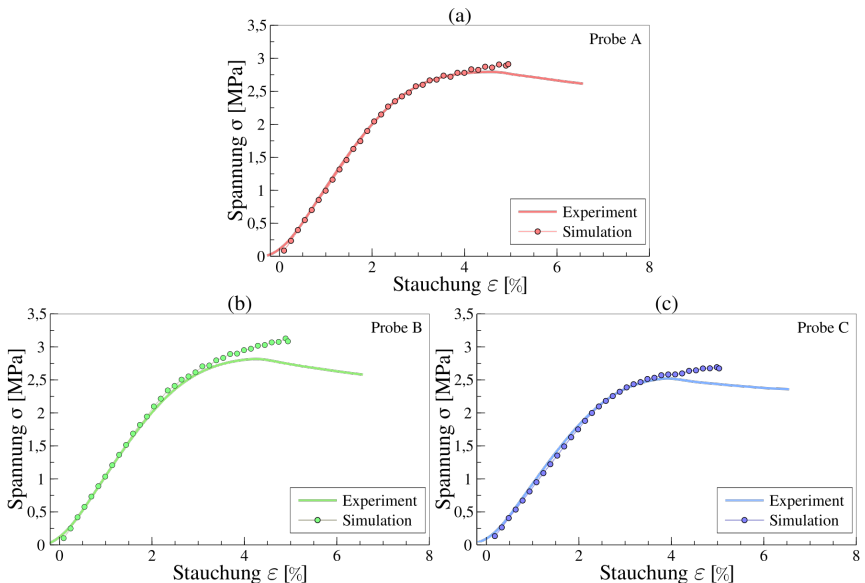
am höchsten ist, werden zur Berechnung des Druckmoduls verwendet. Der Kurvenfuß der experimentell ermittelten Diagramme wird durch die Verlängerung des linearen Bereichs ausgehend von der Stauchung um 25 % der maximalen Kraft entfernt.

Die Stauchgrenze wird an dem Punkt definiert, an dem eine plastische Verformung von 0,2 % vorliegt. Die exakte Bestimmung des Übergangs von elastischer zu plastischer Verformung ist aus dem Diagramm sonst nur schwer ableitbar.

### 6.3.3 Abgleich mit Simulationen

Auf Grundlage dieser Experimente konnte die Simulation parametrisiert werden. Dazu wurde der Druckmodul des Polyurethans iterativ angepasst, bis die simulierten Diagramme mit den experimentell ermittelten zusammenfielen. Abbildung 6.5 stellt die simulierten und experimentellen Untersuchungen gegenüber.

Für alle drei Proben wird derselbe Druckmodul der Bulkphase verwendet. Die Übereinstimmung der experimentellen und simulierten Kurven validiert an dieser Stelle die Simulation. Nachfolgend werden alle Simulationen mit den ermittelten Parametern durchgeführt, wobei der Druckmodul der Bulkphase 1,92 GPa und die Stauchgrenze 52 MPa beträgt.

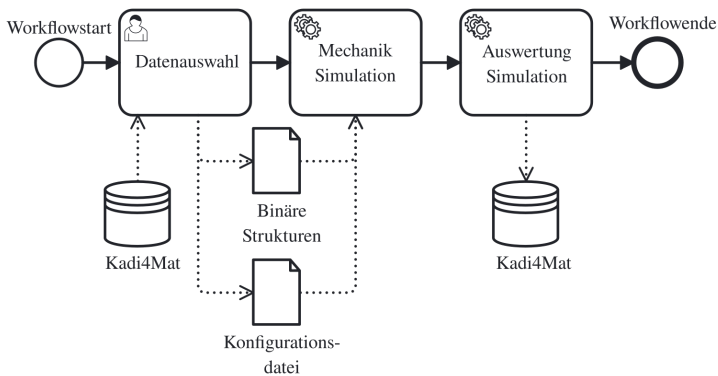


**Abbildung 6.5:** Gegenüberstellung der experimentellen und simulativen Untersuchungen der Schäume. Die Simulation wird auf den Druckversuchen von Probe A parametrisiert. In (a) ist das Ergebnis der Parametrisierung gezeigt. Zur Validierung der Parametrisierung stellen (b) und (c) die Experimente den Simulationen der anderen Proben gegenüber.

## 6.4 Einbindung in Kadi4Mat

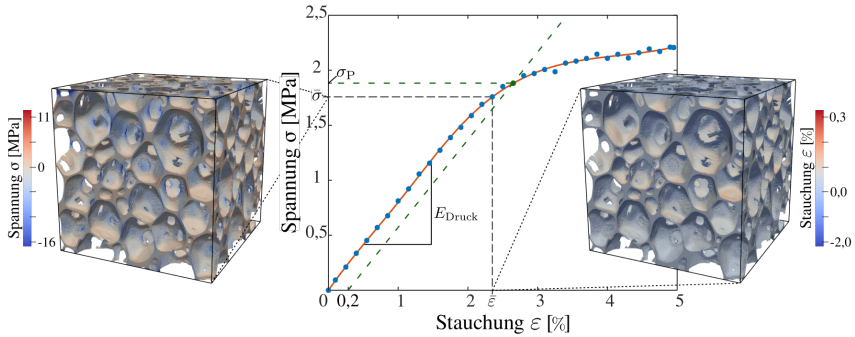
Zur Automatisierung und FAIRen Dokumentation der Simulationen wurde ein generischer Workflow in KadiStudio entwickelt, der flexibel auch für andere Anwendungsfälle genutzt werden kann. Nutzer können während der Ausführung sowohl die gewünschte Simulationskonfiguration als auch die zu untersuchenden Mikrostrukturen auswählen. Dadurch lassen sich unterschiedliche Konfigurationsdateien einbinden, die jeweils verschiedene Eigenschaften simulieren.

Die Automatisierung der Bestimmung effektiver Materialparameter erfolgt in Form eines KadiStudio-Workflows. Eine schematische Übersicht des Ablaufs ist in Abbildung 6.6 als BPMN-Diagramm dargestellt. Der vollständige Workflow ist auf Zenodo publiziert und frei zugänglich [201].



**Abbildung 6.6:** Schematische Darstellung des Simulationsworkflows. Der Workflow implementiert die mechanische Belastung einer Struktur und die Auswertung der Ergebnisse, kann aber um andere Anwendungsbeispiele erweitert werden.

Der Workflow nutzt die Simulationssoftware PACE3D, die ein breites Spektrum physikalischer Vorgänge modellieren kann. Ergänzend Analysewerkzeuge ermöglichen es, direkt aus den Simulationen effektive Materialparameter abzuleiten.



**Abbildung 6.7:** Analyse der Simulationsergebnisse. Blaue Punkte: Spannungs- und Stauchungswerte pro Zeitschritt; orange Kurve: Interpolation. Der lineare Verlauf der Kurve entspricht dem effektiven Druckmodul. Der Schnittpunkt einer um 0,2 % nach rechts verschobenen Geraden mit der Steigung des Druckmoduls mit der interpolierten Kurve bestimmt die Stauchgrenze  $\sigma_P$ .

Die hier vorgestellte Implementierung fokussiert auf mechanische Belastungen und die automatische Extraktion der Kennwerte.

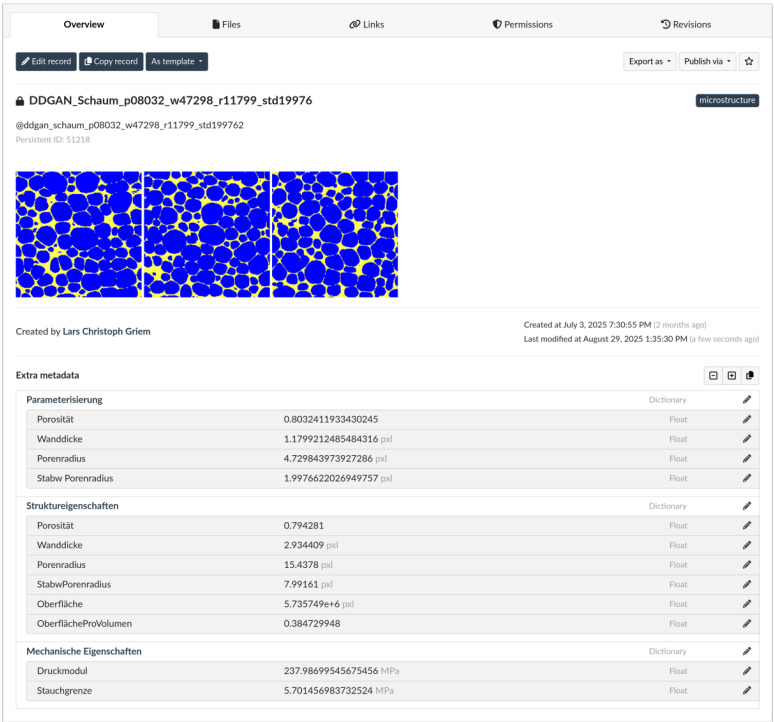
PACE3D wird über Konfigurationsdateien gesteuert, die die physikalischen Prozesse definieren. Während der Ausführung wählt der Nutzer die gewünschte Eigenschaft und die passende Konfiguration wird geladen. Abmessungen und Materialkennwerte der Vollmaterialien (Druckmodul und Stauchgrenze) werden dabei automatisch an die jeweilige Mikrostruktur angepasst.

Nach der Simulation erfolgt die Auswertung: Für jeden Zeitschritt werden mittlere Spannungs- und Dehnungswerte berechnet, interpoliert und als Spannungs-Dehnungskurve dargestellt. Daraus wird der Druckmodul als Steigung im linearen Bereich bestimmt. Der Schnittpunkt einer um 0,2 % verschobenen Geraden mit der Steigung des Druckmoduls definiert die Stauchgrenze. Abbildung 6.7 visualisiert diesen Analyseprozess.

Die auf diese Weise bestimmten mechanischen Eigenschaften werden anschließend automatisch als strukturierte Metadaten in den jeweiligen Struktur-Record auf Kadi4Mat geschrieben. Gemäß dem in Unterabschnitt 3.3.1 vorgestellten Konzept der inkrementellen Metadaten wächst der Record damit um die simulierten



Eigenschaften an. Abbildung 6.8 zeigt einen exemplarischen Record, in dem die mechanischen Eigenschaften zusätzlich zu den bereits dokumentierten strukturellen Eigenschaften abgelegt sind. Neben der reinen Dokumentation der Ergebnisse werden die Records zudem automatisch mit den verwendeten Workflows und Simulationsmodellen verknüpft. Auf diese Weise entsteht eine lückenlose Abbildung der Datenherkunft, die die Reproduzierbarkeit und Nachvollziehbarkeit sämtlicher Schritte gewährleistet. Die entsprechenden Relationen sind in Abbildung 6.9 gezeigt.



**Abbildung 6.8:** Record der die mechanischen und strukturellen Eigenschaften einer Mikrostruktur beschreibt. Die Metadaten wurden durch die Workflowanalysen automatisiert hinzugefügt.

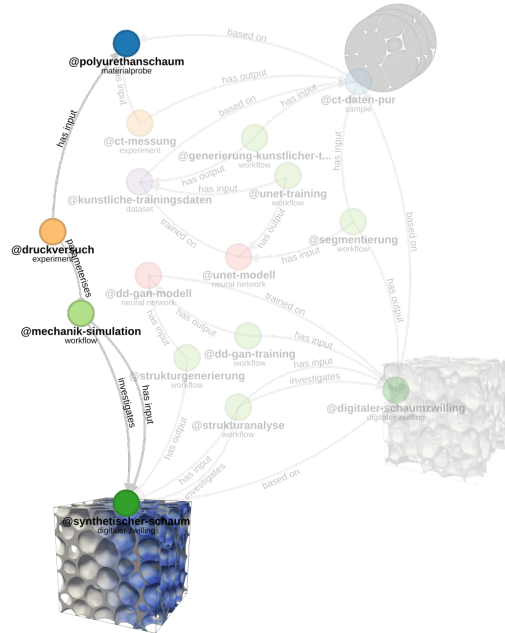
## 6.5 Diskussion

In diesem Kapitel wurde die letzte methodische Komponente der entwickelten Materialdesign-Methodik umgesetzt: die Simulation makroskopischer mechanischer Eigenschaften digitaler Schaumzwillinge. Die entwickelten Workflows sind vollständig in Kadi4Mat eingebunden, sodass sämtliche Simulationen und Auswertungen automatisch dokumentiert werden und eine nachvollziehbare sowie wiederverwendbare Datenbasis entsteht.

Die Ergebnisse zeigen, dass durch die Parametrisierung des Mechaniksolvers auf Basis experimenteller Druckversuche eine enge Übereinstimmung zwischen Simulation und Experiment erzielt werden kann. Damit lassen sich aus der Mikrostruktur zuverlässig die makroskopischen Kennwerte Druckmodul und Stauchgrenze ableiten. Der automatisierte Workflow erlaubt es, diese Berechnungen konsistent auf beliebige Strukturen anzuwenden und die resultierenden Eigenschaften direkt in strukturierter Form in der Datenbank zu dokumentieren. Damit ist der Übergang von der rein geometrischen Charakterisierung zur physikalischen Bewertung der Materialeigenschaften vollzogen.

Gleichzeitig bestehen Einschränkungen. Die Simulationsumgebung wurde in dieser Arbeit ausschließlich zur Bestimmung mechanischer Eigenschaften eingesetzt, obwohl prinzipiell auch andere physikalische Prozesse abgebildet werden können. Zudem ist die Parametrisierung auf Polyurethanschaum zugeschnitten, sodass eine Übertragung auf andere Materialien zunächst die Bestimmung geeigneter Bulk-Parameter erfordert.

Zusammenfassend bietet der entwickelte Workflow eine robuste und reproduzierbare Grundlage, um Mikrostrukturen mit ihren mechanischen Eigenschaften zu verknüpfen. Seine modulare Struktur ermöglicht es, ihn in zukünftigen Arbeiten flexibel um weitere Modelle und Materialsysteme zu erweitern. Damit fügt er sich nahtlos in die Gesamtmethodik ein und vervollständigt die Prozesskette vom CT-Datensatz bis zur physikalischen Charakterisierung. Der aktuelle Stand des Materialentwicklungsprozesses ist in Abbildung 6.9 dargestellt.



**Abbildung 6.9:** Zusammenfassung der Simulationen. Ein auf Druckversuchen validierter Simulationsworkflow bestimmt die mechanischen Eigenschaften der Schäume.

Damit stehen alle Bausteine für eine konsistente, datengetriebene Materialentwicklung zur Verfügung. Das folgende Kapitel widmet sich der systematischen Anwendung dieser Methoden, um Struktur-Eigenschaft-Beziehungen zu identifizieren und die Struktur-Eigenschafts-Beziehungen der aufgebauten Datenbank zu analysieren.

## 7 Identifikation von Struktur-Eigenschaft-Beziehungen

In diesem Kapitel wird die letzte methodische Komponente der in dieser Arbeit entwickelten Methodik vorgestellt: die Identifikation von Struktur-Eigenschaft-Beziehungen. Während die vorangegangenen Kapitel Methoden zur Rekonstruktion, Generierung, Charakterisierung und Simulation von Mikrostrukturen etabliert haben, wird nun gezeigt, wie diese Bausteine systematisch zusammengeführt und ausgewertet werden können. Der Schwerpunkt liegt hierbei auf der Entwicklung eines generisch einsetzbaren Workflows, der auch für andere Materialsysteme nutzbar ist. Zur Identifikation der Struktur-Eigenschaft-Beziehungen wird auf neuronale Netze zurückgegriffen, da diese im Gegensatz zu vielen analytischen Modellen nicht auf spezifische Anwendungsfälle beschränkt sind und komplexe, nichtlineare Zusammenhänge zwischen Struktur und Eigenschaft erfassen können.

Nachfolgend wird zunächst der Stand der Forschung skizziert, der aktuelle Entwicklungen und Methoden zur Identifikation von Struktur-Eigenschaft-Beziehungen anhand ausgewählter Beispiele aufzeigt. Anschließend wird die Erstellung einer strukturell vielfältigen und ausgewogenen Datenbank beschrieben, die im darauffolgenden Abschnitt mithilfe maschineller Lernverfahren analysiert wird. Ziel ist die Entwicklung eines interpretierbaren ML-Modells, aus dem verlässliche Struktur-Eigenschaft-Beziehungen abgeleitet und in verständlicher Form dargestellt werden können.

## 7.1 Stand der Forschung

Die Identifikation von Struktur-Eigenschaft-Beziehungen ist ein zentrales Thema in den Materialwissenschaften, da sie sowohl ein tieferes Verständnis von Materialsystemen als auch die Vorhersage des Materialverhaltens ermöglichen. Struktur-Eigenschaft-Beziehung verknüpfen strukturelle Merkmale eines Materials mit dessen makroskopischen Eigenschaften [202]. Typischerweise wird beschrieben, wie mikroskopische oder mesoskopische Strukturmerkmale – Porosität, Korngröße oder Kristallorientierung – die makroskopischen Materialeigenschaften – Festigkeit, Leitfähigkeit oder Permeabilität – beeinflussen.

Grundsätzlich lassen sich Struktur-Eigenschaft-Beziehungen entweder analytisch durch physikalische Modelle oder datengetrieben mithilfe statistischer bzw. maschineller Lernverfahren erfassen [202]. Je nach Fragestellung können Struktur-Eigenschaft-Beziehungen dabei sowohl zur Lösung des vorwärtsgerichteten Problems (von Struktur zur Eigenschaft) als auch des inversen Problems (von gewünschter Eigenschaft zur Struktur) eingesetzt werden.

Für datengetriebene Verfahren ist die Verfügbarkeit umfangreicher und qualitativ hochwertiger Daten entscheidend. Einige Arbeiten, wie die von Liu et al. [203] und Lin et al. [204], nutzen bestehende Materialdatenbanken, um durch Struktur-Eigenschaft-Beziehung-Analysen neue Materialien zu identifizieren.

Andere, wie Danesh et al. [205], erzeugen gezielt neue Datenpunkte, beispielsweise mithilfe aktiven Lernens, um Strukturen mit hohem Informationsgehalt bezüglich der Struktur-Eigenschaft-Beziehung zu identifizieren. Dabei ist es entscheidend, den relevanten Datenraum möglichst vollständig und gleichmäßig abzudecken, um Verzerrungen und Überanpassung zu vermeiden [204]. Bei der Auswahl oder Erstellung solcher Datenbanken ist darauf zu achten, dass der relevante Datenraum möglichst vollständig und gleichmäßig abgedeckt wird, um Verzerrungen zu vermeiden und belastbare Vorhersagen zu ermöglichen [204].

Zur Modellierung von Struktur-Eigenschaft-Beziehungen können sowohl direkt messbare physikalische Größen als auch komplexere, aus Daten abgeleitete Merkmale verwendet werden. Erstere kommen beispielsweise in analytischen Modellen wie der Kozeny–Carman-Gleichung zum Einsatz, wie in der Arbeit von Kunz et al. [206] demonstriert wird, in der die Permeabilität poröser Medien in Abhängigkeit struktureller Eigenschaften modelliert wird. Ein verwandter datenbasierter Ansatz findet sich bei Saunders et al. [207], die physikalisch motivierte Strukturparameter als Eingangsdaten für ein neuronales Netz nutzen, um mechanische Eigenschaften metallischer Materialien vorherzusagen. Dieses Vorgehen ermöglicht es, datengetriebene Modellierung mit einer gewissen physikalischen Interpretierbarkeit zu kombinieren.

Im Gegensatz zu physikalischen Größen ermöglichen korrelative Methoden wie  $n$ -Punktkorrelationen und Hauptkomponentenanalyse (PCA) oder neuronale Netze eine ganzheitlichere Beschreibung der Materialstruktur [205, 208, 209]. Diese Verfahren sind zwar meist weniger interpretierbar, können jedoch komplexe, nichtlineare Zusammenhänge erfassen.

Während die PCA eine lineare Dimensionsreduktion mit interpretierbaren Hauptkomponenten ermöglicht [205], können neuronale Netze auch nichtlineare, hochdimensionale Struktureigenschaften extrahieren, erfordern jedoch größere Datenmengen und liefern weniger intuitive nachvollziehbare Merkmale.

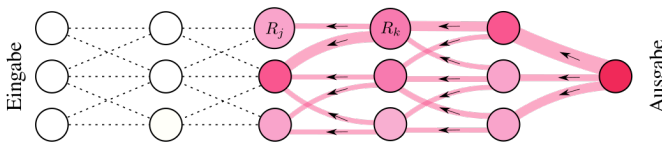
Ein etablierter Ansatz zur Visualisierung und Analyse von Struktur-Eigenschaft-Beziehungen sind sogenannte Materialkarten. Diese setzen strukturbezogene Parameter in Beziehung zu makroskopischen Eigenschaften und ermöglichen eine intuitive Darstellung funktionaler Abhängigkeiten. In der Arbeit von Kunz et al. [206] wird dieser Ansatz beispielsweise genutzt, um die Flüssigkeitsausbreitung in porösen Medien in Abhängigkeit von strukturellen Parametern zu analysieren. Die Aussagekraft solcher Karten hängt dabei entscheidend von der Auswahl geeigneter struktureller Merkmale ab. Unabhängig davon, ob diese physikalisch gemessen oder datenbasiert extrahiert wurden.

## 7.2 Methoden zur Analyse neuronaler Netze

Neuronale Netze gelten häufig als Blackboxes. Sie verarbeiten Eingaben und erzeugen Ausgaben, ohne dass die internen Entscheidungsprozesse direkt nachvollziehbar sind. Die Explainable AI bietet Methoden, die diese Prozesse transparenter machen [210]. Eine zentrale Methode ist die *Layer-wise Relevance Propagation* (LRP) [211], die den Beitrag einzelner Eingaben zu einer Vorhersage sichtbar macht. Dabei wird die Ausgabe des Netzes schrittweise von der Ausgangsbeschicht bis zu den Eingaben zurückverteilt.

Die LRP verläuft in zwei Schritten: Zunächst erfolgt eine normale Vorhersage, während der die Aktivierungen in jeder Schicht gespeichert werden. Anschließend wird die Relevanz von der Ausgangsbeschicht bis zur Eingabeschicht nach festgelegten Regeln rückverteilt. Als Relevanz wird dabei die Ausgabe der letzten Schicht definiert. Für Regressionsaufgaben ist das der vorhergesagte Wert, für Klassifikationen die Klassenzugehörigkeiten.

Eine zentrale Bedingung bei der Rückverteilung ist die Relevanzerhaltung. Für die Relevanz eines Neurons  $R_k$  und die Relevanzanteile  $R_{j \leftarrow k}$  die von von Neuron  $R_k$  an die vorherigen Schicht zurückgegeben werden, gilt  $\sum_j R_{j \leftarrow k} = R_k$ . So bleibt die Gesamtrelevanz über alle Schichten hinweg konstant, wodurch die Vorhersage vollständig durch Eingaben erklärbar bleibt. Abbildung 7.1 stellt diesen Prozess schematisch dar.



**Abbildung 7.1:** Schematische Darstellung der LRP: Die Ausgabe wird schrittweise bis zu den Eingaben zurückgeführt. Adaptiert aus [212].

Die LRP lässt sich nicht nur auf MLPs, sondern auch auf CNNs anwenden und eignet sich daher als generischer Ansatz zur Analyse neuronaler Netze.

## 7.3 Aufbau der Datenbank

Eine gut strukturierte und ausgewogene Datenbasis ist entscheidend für die zuverlässige Identifikation von Struktur-Eigenschaft-Beziehungen. Insbesondere bei der Anwendung maschineller Lernverfahren ist es erforderlich, dass der betrachtete Datenraum möglichst vollständig und gleichmäßig abgedeckt wird. Eine ungleichmäßige oder lückenhafte Verteilung der Daten kann die Lernfähigkeit von Modellen stark beeinträchtigen, zu instabilen Vorhersagen führen und letztlich die Aussagekraft der resultierenden Struktur-Eigenschaft-Beziehungen einschränken [213].

Dies gilt umso mehr, wenn interpretierbare Modelle entwickelt werden sollen. Nur wenn der zugrunde liegende Merkmalsraum hinreichend gut repräsentiert ist, lassen sich zuverlässige Aussagen über den Einfluss struktureller Parameter auf makroskopische Materialeigenschaften treffen. Entsprechend ist eine informationsreiche und ausbalancierte Datenbasis ein zentraler Bestandteil für die erfolgreiche Identifikation von Struktur-Eigenschaft-Beziehungen.

Zum Aufbau der Datenbank kommt zunächst das in Abschnitt 5.4 vorgestellte eigenschaftskonditionierte DD-GAN zum Einsatz, um unterschiedlichste Schaumstrukturen zu erstellen. Als Orientierung für die Generierung werden zunächst die in Abschnitt 4.5 ermittelten strukturellen Eigenschaften der realen Schaumstrukturen als Eingang verwendet. Auf Grundlage dieser Eigenschaften erzeugt das Diffusionsmodell synthetische Schaumstrukturen mit ähnlichen Struktureigenschaften.

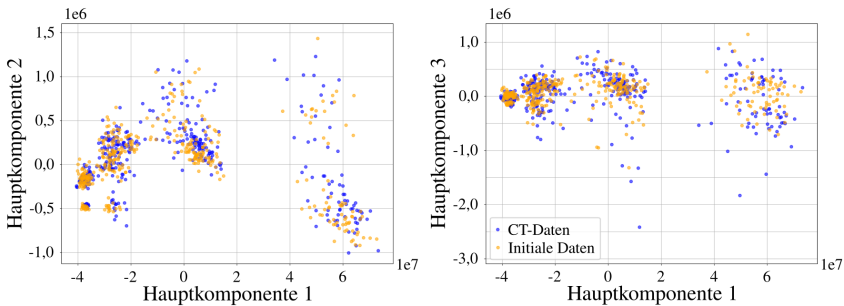
Um die strukturelle Vielfalt der erstellten initialen Datenbank zu bewerten, wird zunächst die Verteilung der Daten analysiert. Da im Vorfeld nicht bekannt ist, welche strukturellen Merkmale für die Struktur-Eigenschaft-Beziehungen entscheidend sind, wird bewusst auf eine manuelle *a-priori*-Auswahl von Strukturparametern verzichtet.

Stattdessen wird die in Unterabschnitt 4.2.4 beschriebene, datengetriebene Analyse eingesetzt. Diese basiert auf einer ZPK-Darstellung mit anschließender PCA



und ermöglicht eine ganzheitliche Erfassung der mikrostrukturellen Eigenschaften. Die Projektion aller Datenpunkte in diesen Merkmalsraum erlaubt eine visuelle Bewertung der Verteilung und macht strukturelle Lücken unabhängig von konkreten Strukturparametern erkennbar.

Abbildung 7.2 zeigt die Projektion sowohl realer als auch generierter Strukturen in diesen Merkmalsraum. Wie in den vorangegangenen Kapiteln erfolgt die Berechnung der PCA ausschließlich auf den CT-Daten. Die synthetischen Daten werden lediglich über die identifizierte Transformation in den Merkmalsraum projiziert. In dem dargestellten Ergebnis beschreibt die erste Hauptkomponente 98,7 % der Varianz, während PC2 0,08 % und PC3 sowie PC4 jeweils etwa 0,03 % erklären.



**Abbildung 7.2:** Projektion realer und generierter Strukturen im PCA-Raum, basierend auf der 2-Punkt-Korrelation. Die generierten Strukturen überlagern den Wertebereich der realen Strukturen weitgehend und erweitern diesen zudem. Gleichzeitig treten im Merkmalsraum Lücken zwischen einzelnen Clustern auf, in denen bislang keine Strukturen vorhanden sind. Diese Bereiche markieren gezielt zu ergänzende Regionen zur Vollständigkeit der Datenbank.

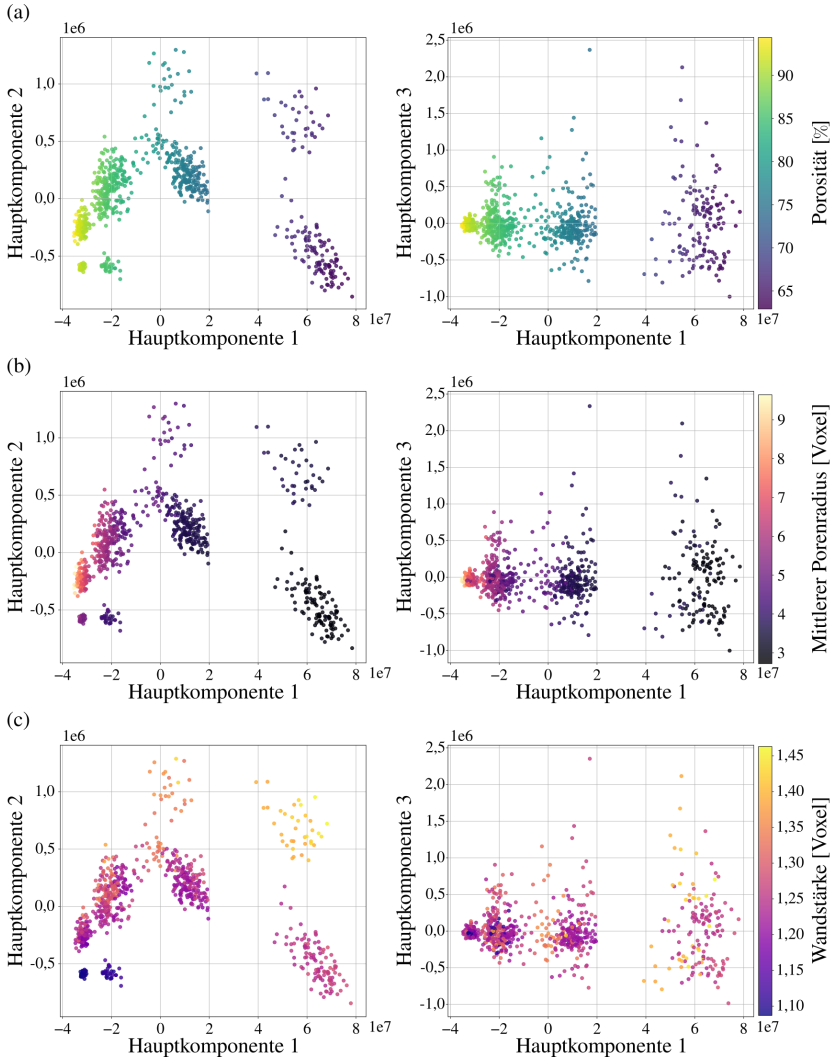
Es zeigt sich, wie bereits in Unterabschnitt 4.4.1 dargelegt, dass die synthetisierten Daten die realen Daten gut abdecken und auch erweitern. Allerdings ist zu erkennen, dass sich entlang der ersten Hauptkomponente Cluster bilden, die nicht miteinander verbunden sind. Zwar schließen die generierten Strukturen diese Lücken bereits teilweise, dennoch bleibt insbesondere im Bereich um eine Wert

von 1 der PC1 eine deutliche Lücke bestehen. Diese deutet auf eine unzureichende Abdeckung dieses Datenbereichs durch die bisherigen Strukturen hin.

Um diese Lücke nun gezielt durch Strukturen zu füllen, ist es notwendig zu verstehen welche Strukturmerkmale entlang der einzelnen Hauptkomponenten variiert werden. Erst dadurch kann das Generierungsmodell richtig parametrisiert werden. Zu diesem Zweck wurden in Abbildung 7.3 separate PCA-Projektionen erstellt, in denen die Datenpunkte jeweils nach bekannten Strukturparametern (Porosität, mittlerem Porenradius und Wanddicke) eingefärbt wurden. Da die Lücken in Abbildung 7.2 am ausgeprägtesten in den PC1-PC2- und PC1-PC3-Diagrammen zu erkennen sind, wurden lediglich diese gemäß ihrer Struktureigenschaften eingefärbt.

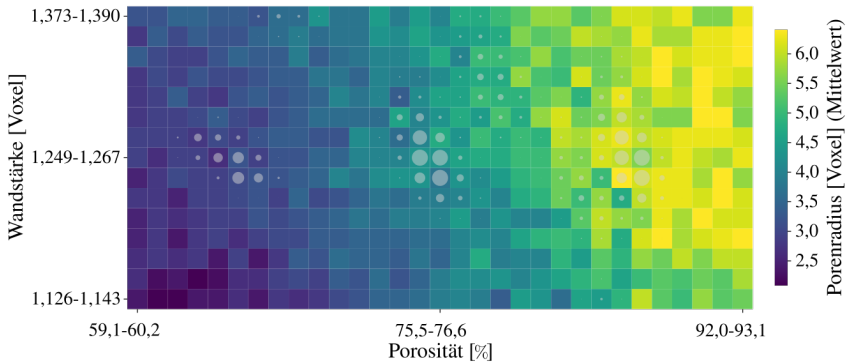
Der Farbverlauf im PC1-PC2-Diagramm zeigt deutlich, dass PC1 vor allem die Porosität abbildet. Damit lässt sich die in Abbildung 7.2 zuvor identifizierte Lücke als Bereich mittlerer Porosität interpretieren, der bislang kaum durch reale oder generierte Strukturen abgedeckt ist. Dies deckt sich mit der in Unterabschnitt 4.4.1 identifizierten Datenverteilung der CT-Daten, die für Porositäten im Bereich um 70 % und 80 % eine Lücke aufweisen (vgl. Abbildung 5.17). In den weiteren Projektionen, insbesondere im PC1-PC3-Diagramm, treten auch für Wanddicke und Porenradius erkennbare Muster auf, die sich in Form isolierter Bereiche zeigen. Diese deuten darauf hin, dass eine vollständige Abdeckung des Merkmalsraums nicht allein durch Variation der Porosität, sondern auch durch die gezielte Kombinationen dieser Strukturparameter erreicht werden muss.

Vor diesem Hintergrund konzentriert sich die Ergänzung der Datenbank im ersten Schritt auf die gezielte Variation der Porosität. Mit dem Generierungsnetzwerk wurden neue Strukturen erzeugt, deren Porositäten im Bereich von 65 % bis 75 % vorgegeben wurden. Der Generator benötigt neben der Porosität auch die Parameter Porenradius und Wanddicke. Um sicherzustellen, dass realistische Strukturen generiert werden ist eine sinnvolle Auswahl der anderen Parameter essentiell. Die in Abbildung 7.4 dargestellte Heatmap unterstützt bei der Auswahl geeigneter Parameter. Diese Heatmap wurde in Unterabschnitt 5.4.3 zur Validierung des DD-GANs erstellt und zeigt die Verteilung der Wanddicken und Porenradien für



**Abbildung 7.3:** Hauptkomponentenanalyse der Zweipunkt-Korrelation. Die Datenpunkte sind gemäß der Struktureigenschaften eingefärbt.

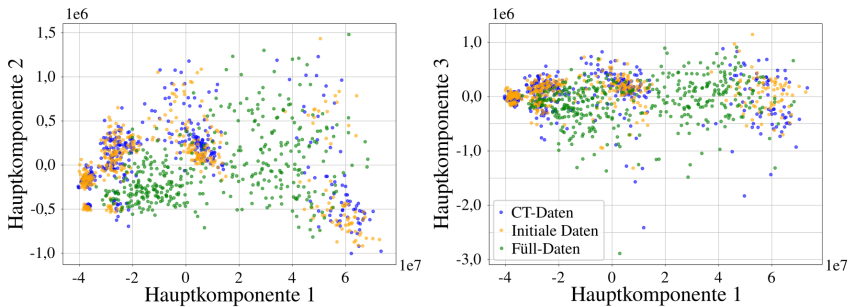
bestimmte Porositäten in den realen CT-Daten. Daraus können manuell passende Werte für Wandstärke und Porenradius ausgewählt werden. Da die manuelle Auswahl bei großen Datenmengen sehr zeitaufwändig ist, wurde eine automatische Parametrisierung implementiert (vgl. Abschnitt 7.5).



**Abbildung 7.4:** Heatmap der Generator-Parameterkombinationen. Jede Box zeigt Eigenschaftsbereiche, die ein gezieltes Einstellen des Generators ermöglichen. Graue Kreise markieren die Verteilung der Trainingsdaten, ihr Radius skaliert mit der Datenmenge.

Diese ermittelt für einen vorgegebenen Porositätsbereich und den darin liegenden Strukturen den Mittelwert und die Standardabweichung der Wandstärke und des Porenradius. Basierend auf diesen Werten erstellt sie neue Parametrisierungen für den Generator. Ein entsprechendes Vorgehen wurde verwendet, um Parameterkombinationen zu finden, die die im PC1-PC3 identifizierten Cluster miteinander verbinden. Für Strukturen mit geringer Porosität (hohe PC1-Werte) wurde die Wanddicke um den Wert 1,35 und der Porenradius zwischen 3,0 und 4,0 variiert, um eine Verbindung der isolierten Regionen zu ermöglichen. Dasselbe Vorgehen wurde für hohe Porositäten (niedrige PC1-Werte) angewendet, um die auf der linken unteren Seite des PC1-PC3-Diagramms isolierte Region mit benachbarten Bereichen zu verbinden.

Ziel war es, eine kontinuierliche Verteilung im Merkmalsraum zu erzeugen und die bisher getrennten Regionen zu verbinden, um eine solide Grundlage für die spätere Analyse von Struktur-Eigenschaft-Beziehungen zu erzeugen. Um zu prüfen, ob die so erzeugten Strukturen die identifizierten Lücken schließen, wurden alle Strukturen in Abbildung 7.5 erneut im Merkmalsraum visualisiert. Es zeigt



**Abbildung 7.5:** PCA aller Schaumstrukturen. Die realen Schaumstrukturen sind blau, die zunächst synthetisierten in orange dargestellt. Die gezielte Generierung weiterer Strukturen erzeugt die grünen Datenpunkte. Der Datenbereich konnte auf diese Weise vollständig abgedeckt werden.

sich, dass die ergänzten Datenpunkte die zuvor identifizierten Lücken schließen und die Regionen miteinander verbinden. Besonders im Bereich mittlerer Porositäten konnte so eine gleichmäßigere Verteilung erreicht werden, wodurch der strukturelle Merkmalsraum nun zusammenhängend abgedeckt ist.

An dieser Stelle ist zu erwähnen, dass die Hauptkomponentenanalyse hier lediglich zur Identifikation fehlender Datenpunkte diente. Eine weitere Verwendung der ermittelten Werte zur Identifikation von Struktur-Eigenschaft-Beziehungen ist nicht angedacht. Zur Ergänzung mechanischer Eigenschaften wurde der in Abschnitt 6.4 beschriebene, experimentell validierte Simulationsworkflow eingesetzt. Da dieser lediglich bestehende Strukturen um mechanische Eigenschaften erweitert, ohne den Merkmalsraum zu vergrößern, wird hier nicht weiter auf ihn eingegangen.

Damit ist die Datenbank vollständig erzeugt. Für die Generierung, geometrische und mechanische Charakterisierung der Schaumstrukturen kamen die in Abschnitt 5.5, 6.4 und 4.6.2 beschriebenen Workflows zum Einsatz. Alle erzeugten Daten werden mitsamt Metadaten auf Kadi4Mat abgelegt und mit den verwendeten Ressourcen verknüpft. Die gesamte Datenbank ist somit automatisch auf Kadi4Mat dokumentiert. Abbildung 7.6 zeigt die Visualisierung in KadiWeb sowie einen exemplarischen Record. Die Verknüpfungen mit den Workflows und die jeweilige Parametrisierung machen die Datenherkunft nachvollziehbar und sichern die Reproduzierbarkeit. Auf dieser strukturierten und FAIR dokumentierten Basis werden im folgenden Kapitel mögliche Struktur-Eigenschaft-Beziehungen untersucht.



**Abbildung 7.6:** Datenbank in KadiWeb mit exemplarischem Record. Die Punktwolke zeigt die abgelegten Strukturen, die mit den zugehörigen Workflows verknüpft sind. Einheitlich strukturierte Metadaten ermöglichen Suche und Reproduktion.

## 7.4 Analyse der Datenbank

Die im vorangegangenen Kapitel beschriebene Datenbank bietet durch ihre gleichmäßige Datenverteilung eine ideale Grundlage für die Anwendung maschineller Lernverfahren zur Identifikation von Struktur-Eigenschaft-Beziehungen. Im Gegensatz zur Datenbankerstellung, bei der abgeleitete Merkmale wie die Zweipunktkorrelation verwendet wurden, basiert die Analyse hier direkt auf den grundlegenden, leicht extrahierbaren Strukturparametern: Porosität, mittlerer Porenradius, Standardabweichung des Porenradius, innere Oberfläche und mittlere Wandstärke. Der Vorteil dieser Parameter liegt in ihrer direkten Interpretierbarkeit und in der geringen Rechenzeit für ihre Ermittlung.

Ziel der Analyse ist es Struktur-Eigenschaft-Beziehungen zu identifizieren, die die mechanischen Eigenschaften – Druckmodul, Stauchgrenze – durch maximal drei Strukturparameter beschreiben. Die Reduktion auf wenige, relevante Merkmale erlaubt es, die Ergebnisse in Form sogenannter Materialkarten darzustellen. Diese Karten ermöglichen nicht nur eine intuitive Visualisierung der Struktur-Eigenschaft-Beziehungen, sondern lassen sich auch sowohl für vorwärtsgerichtete Aufgaben (Eigenschaftsvorhersage) als auch für inverse Fragestellungen (Strukturauslegung auf gewünschte Eigenschaften) nutzen.

### 7.4.1 Modellarchitektur und Training

Für die Analyse kommt ein neuronales Netzwerk zum Einsatz, das ausschließlich aus vollständig verbundenen Schichten (englisch *dense layers*) besteht. Die Wahl dieser Architektur begründet sich einerseits durch die kompakte Form der Eingangsdaten (Struktureigenschaften), andererseits durch die gute Kompatibilität mit Methoden der erklärbaren künstlichen Intelligenz (*Explainable AI*). Letztere werden im Anschluss eingesetzt, um die Relevanz einzelner Eingangsgrößen für die Vorhersage zu analysieren.

Das verwendete Netzwerk besteht aus einer Eingabeschicht, fünf versteckten Schichten mit 16, 32, 64, 32 und 16 Neuronen sowie einer Ausgabeschicht. Als

Aktivierungsfunktion kommt in allen versteckten Schichten die *Leaky ReLU*-Funktion mit einem Gradienten von 0,1 zum Einsatz. Aufgrund der niedrigen Dimension der Eingangsdaten in Form weniger Strukturparameter sowie der Aufgabenstellung einer Regression ist die Netzgröße bewusst kompakt gehalten (etwa 5 346 trainierbare Parameter). Diese bewusst kompakte Architektur bietet gegenüber komplexeren Architekturen einige Vorteile. So reduziert die geringe Anzahl trainierbarer Parameter die Gefahr von Overfitting, was insbesondere bei der begrenzten Größe der verfügbaren Datenbank wichtig ist. Zum anderen lassen sich solch kleinen neuronalen Netzwerke auch auf gewöhnlichen Computern problemlos trainieren. Für die in dieser Arbeit entwickelten Workflows und Methoden ist dies von besonderer Bedeutung, um sie Forschenden besonders zugänglich zu machen.

Vor dem Training werden sowohl die Eingangs- als auch die Ausgangswerte normalisiert, um stabile Gradientenverläufe zu gewährleisten. Das Training selbst erfolgt über 100 Epochen mit einer Lernrate von  $1e-3$ . Optimiert wird das Modell mit dem Adam-Optimizer, als Verlustfunktion kommt der mittlere absolute Fehler (MAE) zum Einsatz. Sowohl die Definition des Modells als auch das gesamte Trainingsverfahren wurden mithilfe des ML-Frameworks *cids* [214] umgesetzt.

Die verwendeten Hyperparameter des Modells (Anzahl der Schichten, Neuronen, Lernrate und Epochen) wurden iterativ durch manuelle Variation und Bewertung der Modellgenauigkeit bestimmt. Eine weitergehende systematische Hyperparameter-Optimierung wurde aufgrund des bereits erreichten Qualitätsniveaus und des geringen zusätzlichen Nutzens für die Kernfragestellung dieser Arbeit nicht durchgeführt. Zur Überprüfung der Modellstabilität wird auf dem finalen Modell jedoch ergänzend eine Cross-Validation durchgeführt.

## 7.4.2 Identifikation relevanter Strukturparameter

Zur Identifikation von Struktur-Eigenschaft-Beziehungen wurde im ersten Schritt ein neuronales Netz mit der beschriebenen Architektur trainiert, das alle verfügbaren



Strukturparameter als Eingang erhält und die mechanischen Eigenschaften vorhersagt. Dieses Vorgehen wurde gewählt, da im Vorhinein keine Annahme getroffen werden soll, welche strukturellen Eigenschaften für die mechanischen Eigenschaften relevant sind. Über das beschriebene Training konnte so ein Netzwerk trainiert werden, das die mechanischen Eigenschaften mit einer durchschnittlichen prozentualen Abweichung von 2,63 % für den Druckmodul und 2,47 % für die Stauchgrenze vorhersagt.

Um diese Vorhersagen in eine Materialkarte umwandeln zu können, muss die Zahl der Eingangsparameter reduziert werden. Zur Identifikation der relevanten Parameter wird das trainierte Netzwerk mittels Layer-wise Relevance Propagation (LRP) analysiert. Diese Explainable AI Methode durchläuft ein neuronales Netzwerk von der Ausgabe- zur Eingabeschicht und analysiert dabei die Beiträge jedes Eingangs zur Vorhersage. Eine detaillierte Beschreibung der Methode ist in Abschnitt 7.2 zu finden.

Das Anwenden dieser Methode zeigt den Einfluss der einzelnen Parameter auf die Vorhersage. Tabelle 7.1 fasst die Relevanzanalyse zusammen. In der Tabelle steht  $\varepsilon$  für die Porosität,  $s$  für die Oberfläche,  $r_{\text{Pore}}$  für den Porenradius,  $\sigma_{r_{\text{Pore}}}$  für die Standardabweichung des Porenradius,  $r_{\text{Wand}}$  für die mittlere Wanddicke und  $\frac{s}{V_{\text{Struktur}}}$ , das die Oberfläche ins Verhältnis zum Strukturvolumen setzt.

**Tabelle 7.1:** Relevanz der einzelnen Eingabewerte. Der Informationsbeitrag wurde über Layer-wise Relevance Propagation ermittelt.

	$\varepsilon$	$s$	$r_{\text{Pore}}$	$\sigma_{r_{\text{Pore}}}$	$r_{\text{Wand}}$	$\frac{s}{V_{\text{Struktur}}}$
Informationsbeitrag	39 %	18 %	6 %	0 %	20 %	14 %

Es ist erkennbar, dass die Porosität und die mittlere Wanddicke am meisten zur Vorhersage beitragen. Die Relevanzen sind dabei mit 39 % und 20 % zudem bedeutend höher als die der übrigen Parameter. Lediglich die Oberfläche trägt mit 18 % einige Informationen zur Vorhersage bei. Da es sich bei ihr allerdings um einen Absolutwert handelt, ist er sehr stark von der gewählten Größe des

untersuchten Ausschnitts abhängig. Als allgemein beschreibender Parameter ist er entsprechend nicht verwendbar.

### 7.4.3 Modellreduktion und Validierung

Basierend auf dieser Analyse wird ein neues Netzwerk mit nur zwei Eingangsgrößen – Porosität und Wandstärke – trainiert, bei ansonsten unveränderter Architektur und identischen Hyperparametern. Das so trainierte Netz kann ebenfalls die mechanischen Eigenschaften zuverlässig vorhersagen. Tabelle 7.2 zeigt die Validierungsergebnisse die mit diesem Netzwerk auf den zurückgehaltenen Daten erzielt wurden. Die Vorhersagen beider mechanischen Eigenschaften sind mit  $R^2$  Werten von über 0,99 sehr genau und die Fehlermaße bewegen sich im niedrigen einstelligen Prozentbereich.

**Tabelle 7.2:** Genauigkeit des trainierten neuronalen Netzwerks. Als Eingangsparameter dienen die Porosität und die Wandstärke.

Datenaufteilung	MAE [MPa]	RMSE [MPa]	MAPE [%]	$R^2$
Druckmodul	2,6422	3,3270	1,6322	0,9977
Stauchgrenze	0,0715	0,1008	1,7363	0,9966

Um sicherzustellen, dass dieses Ergebnis nicht der gewählten Datenaufteilung geschuldet ist, sondern dass das Netzwerk in der Lage ist gut zu generalisieren, wird eine 5-fache Kreuzvalidierung durchgeführt. Die verfügbaren Daten werden dazu in 5 gleichgroße Teilmengen unterteilt. Anschließend werden 5 neuronale Netze der gleichen Architektur trainiert. Dabei wird immer eine der Teilmengen zurückgehalten und das Netzwerk auf den verbleibenden Teilmengen trainiert. Die Bewertung des Netzwerks findet anschließend auf den zurückgehaltenem Datensatz statt. Diese Kreuzvalidierung stellt sicher, dass die Generalisierungsfähigkeit des Netzes nicht von einer einzelnen Datenaufteilung abhängt, sondern dass das

Modell tatsächlich allgemein gültige Zusammenhänge in Form von Struktur-Eigenschaft-Beziehungen lernt. Tabelle 7.3 und Tabelle 7.4 fassen die Ergebnisse der Kreuzvalidierung für den Druckmodul und die Streckgrenze zusammen. Über alle Datenaufteilungen hinweg konnten mit der gewählten Architektur sehr genaue neuronale Netzwerke trainiert werden.

**Tabelle 7.3:** Resultate der Kreuzvalidierung bezogen auf den Druckmodul.

Datenaufteilung	MAE [MPa]	RMSE [MPa]	MAPE [%]	$R^2$
1	2,67	3,70	1,75	0,9969
2	2,64	4,14	1,81	0,9972
3	6,26	7,76	5,69	0,9836
4	4,45	5,34	2,46	0,9948
5	3,40	4,12	2,91	0,9952
Mittelwert	0,0388	0,0501	2,92	0,9935
Standardabweichung	0,0136	0,0148	1,45	0,0051

Für den Druckmodul liegen die  $R^2$ -Werte zwischen 0,9836 und 0,9972, die MAE-Werte bewegen sich im Bereich von 0,0264 bis 0,0626 MPa, und die RMSE-Werte zwischen 0,0370 und 0,0776 MPa. Der MAPE bleibt durchgängig unter 6 %.

Ähnlich stabil verläuft die Prognose der Stauchgrenze mit  $R^2$  von 0,9855 bis 0,9977, MAE zwischen 0,0825 und 0,1528 MPa, RMSE zwischen 0,0887 und 0,1545 MPa und MAPE unter 5 %. Die Datenaufteilung 3 weist erwartungsgemäß die geringste Genauigkeit auf, während Datenaufteilung 1 (bzw. 4) die besten Kennzahlen liefert. Selbst im schlechtesten Fall liegen alle Fehlermaße jedoch innerhalb akzeptabler Toleranzen. Die sehr geringen Standardabweichungen aller Kennzahlen belegen zudem, dass die Modellleistung kaum von der Wahl der Datenaufteilung abhängt und damit tatsächliche Struktur-Eigenschaft-Beziehungen durch das Netzwerk identifiziert werden. Das die Netzwerke in der Kreuzvalidierung insgesamt etwas schlechter als das initial trainierte Netzwerk abschneiden, lässt sich damit erklären, dass bei der 5-fach Kreuzvalidierung 80 % der Daten für

**Tabelle 7.4:** Resultate der Kreuzvalidierung bezogen auf die Stauchgrenze.

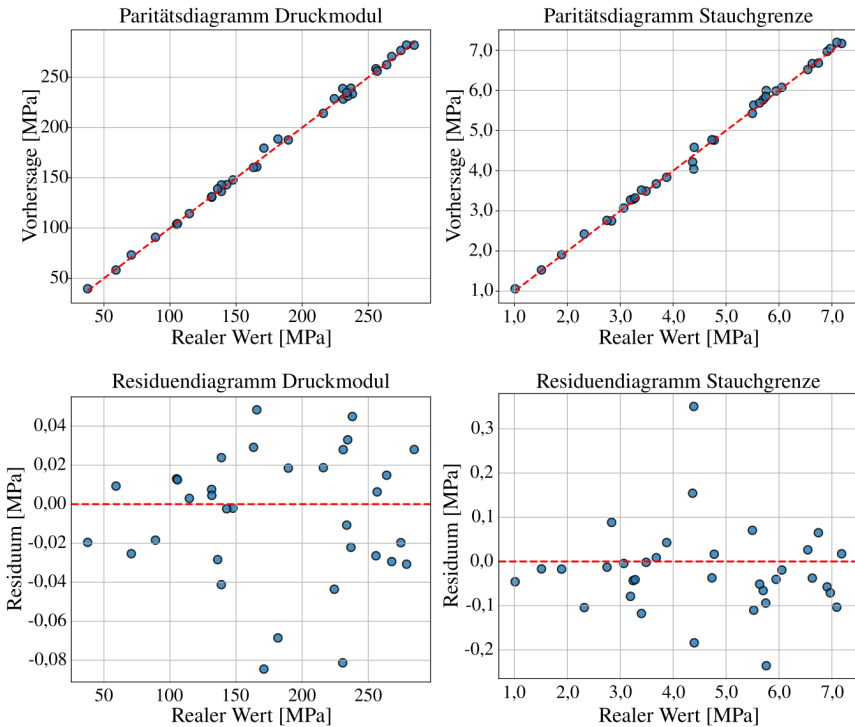
Datenaufteilung	MAE (MPa)	RMSE (MPa)	MAPE (%)	$R^2$
1	0,1140	0,1545	2,60	0,9910
2	0,0825	0,1194	2,13	0,9961
3	0,1528	0,1815	4,74	0,9855
4	0,0702	0,0887	1,81	0,9977
5	0,0989	0,1274	3,21	0,9926
Mittelwert	0,1037	0,1343	2,90	0,9926
Standardabweichung	0,0287	0,0316	1,03	0,0043

das Training verwendet werden, während das ursprüngliche Netzwerk 90 % der Daten verwendet.

### 7.4.4 Fehleranalyse

Um zu analysieren, wie der wenn auch kleine Fehler im initial trainierten Netzwerk (vgl. Tabelle 7.2) verteilt ist, werden seine Vorhersagen in Paritäts- und Residuendiagrammen visualisiert. Dadurch lässt sich die Verteilung der Fehler über die Vorhersagen hinweg besser nachvollziehen, was wiederum eine fundiertere Einschätzung der Vorhersagegenauigkeit der späteren Materialkarte in spezifischen Bereichen ermöglicht. Das Paritätsdiagramm stellt dabei die Vorhersage den realen Werten gegenüber. Bei einer perfekten Vorhersage liegen alle Punkte in diesem Diagramm auf der Diagonalen. Das Residuendiagramm hingegen stellt den Fehler jeder Vorhersage dar. Daraus lässt sich die Fehlerverteilung ableiten und beurteilen, ob die Netzwerke eine systematische Abweichung aufweisen oder ob der Fehler von der vorhergesagten Eigenschaft abhängt.

Abbildung 7.7 zeigt die Paritäts- und Residuendiagramme für das initial trainierte Netzwerk. Wie aufgrund der  $R^2$  Werten zu erwarten, weichen die Vorhersagen nur gering von der idealen Diagonalen im Paritätsdiagramm ab. Ein Trend in der



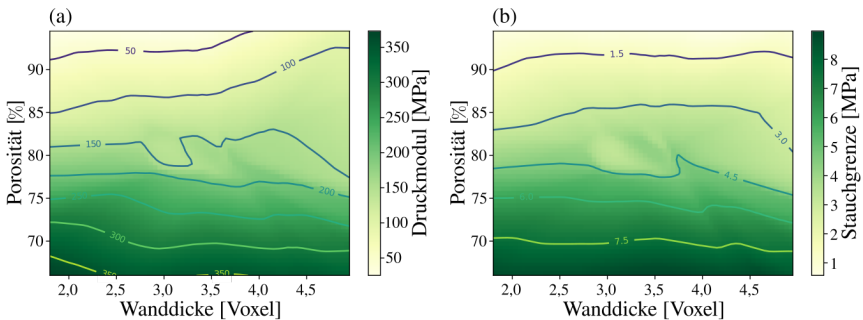
**Abbildung 7.7:** Paritäts- und Residuendiagramm des auf der Porosität und der Wanddicke trainierten Netzwerks.

Abweichung lässt sich nicht erkennen. Die Residuendiagramme zeigen zudem, dass die Fehler gleichmäßig um das Optimum streuen. Die Vorhersagegenauigkeit zeigt keine signifikante Abhängigkeit vom Wertebereich. Sowohl niedrige als auch hohe Werte werden mit vergleichbarer Präzision vorhergesagt. Lediglich in den mittleren Bereichen gibt es einige wenige Ausreißer, die schlechter vorhergesagt werden. In den nachfolgend erstellten Materialkarten ist demzufolge kein systematischer Fehler zu erwarten, sondern eine über den gesamten Wertebereich gleichmäßige Güte.

### 7.4.5 Erstellung der Materialkarten

Um aus dem trainierten Netzwerk Materialkarten abzuleiten, wird es systematisch durchlaufen. Dazu werden zunächst Intervalle für die Eingangsparameter (Porosität und Wanddicke) definiert. Hier ist darauf zu achten, dass die Intervalle auch in den Trainingsdaten abgedeckt sind. Ansonsten kann eine genaue Vorhersage des Netzwerkes nicht garantiert werden. Für die Porosität ergibt sich ein Intervall von 66 % bis 95 % und für die Wandstärke 1,8 Voxel bis 5,0 Voxel.

Diese Intervalle werden nun systematisch iteriert, um Eingangstupel für das neuronale Netz zu erstellen. Die Vorhersage für jeden Eingang wird gespeichert. Aus diesen Werten ergeben sich die in Abbildung 7.8 gezeigten Materialkarten für den Druckmodul und die Stauchgrenze.



**Abbildung 7.8:** Vom neuronalen Netzwerk abgeleiteten Materialkarten. (a) zeigt den Druckmodul und (b) die Stauchgrenze.

In den Materialkarten ist eine klare Verteilung der mechanischen Eigenschaften entlang der Strukturparameter zu erkennen. Wie zu erwarten steigen mit abnehmender Porosität und einem somit zunehmenden Strukturanteil die mechanischen Eigenschaften. Die Wanddicke hingegen hat nur einen geringen Einfluss auf die mechanischen Eigenschaften. In beiden Karten ist zudem erkennbar, dass die in den Residuendiagrammen identifizierten ungenauen Bereiche des Druckmoduls [1,5 MPa; 2,0 MPa] und der Stauchgrenze [4,0 MPa; 5,0 MPa] auch in den Materialkarten auftreten. In diesen sind sie durch zackige Höhenlinien erkennbar. Da aus

den Residuendiagrammen allerdings bekannt ist, dass die Fehler an diesen Stellen mit etwa 6 % immer noch relativ gering sind, haben die Ungenauigkeiten keine große Auswirkung auf die Aussagekraft der Materialkarte.

Vielmehr lassen sich aus diesen Diagrammen schnell sowohl die mechanischen Eigenschaften einer existierenden Struktur ablesen als auch die notwendigen Strukturparameter für bestimmte gesuchte mechanische Eigenschaften. Damit konnte aus der bestehenden Materialdatenbank erfolgreich eine Struktur-Eigenschaft-Beziehung identifiziert und verständlich dargestellt werden.

## 7.5 Einbindung in Kadi4Mat

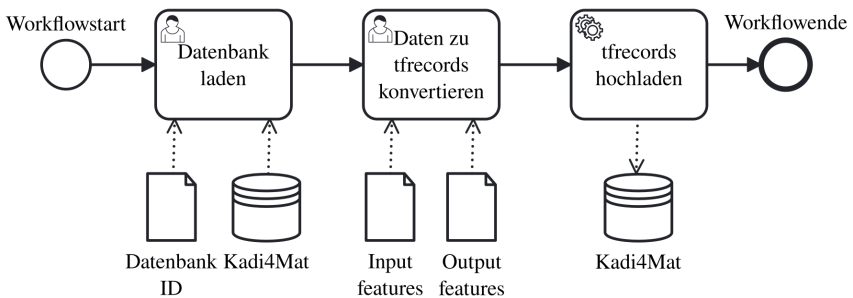
Wie in den vorangegangenen Kapiteln sind auch die in diesem Kapitel entwickelten Methoden als wiederverwendbare Workflows in KadiStudio implementiert. Dadurch sind die Analysen und Ergebnisse nicht nur vollständig reproduzierbar, sondern lassen sich auch auf andere Anwendungsfälle übertragen. Die Workflows sind alle publiziert und frei zugänglich [215].

Der Aufbau der Workflows orientiert sich am in Unterabschnitt 3.2.1 vorgestellten Konzept zur Strukturierung von Forschungsprozessen in Vor-, Haupt-, und Nachbearbeitung. Für die Automatisierung der Identifikation der Struktur-Eigenschaft-Beziehungen wurden entsprechend drei separate Workflows entwickelt, die jeweils unterschiedliche Phasen des Gesamtprozesses abbilden: (i) Datenvorbereitung, (ii) Training eines neuronalen Netzwerks und (iii) Erstellung von Materialkarten.

Die Aufteilung in drei eigenständige Workflows macht diese selbst übersichtlicher und leichter anpassbar. Zudem können so je nach Anwendungsfall gezielt nur bestimmte Teile des Gesamtprozesses ausgeführt werden. Die in diesem Kapitel beschriebene Kuratierung der Datenbank durch das Auffüllen potenzieller Lücken in der Datenverteilung ist nicht Teil der hier dargestellten Workflows. Diese Funktionalität ist bereits in den in Abschnitt 5.5 und Abschnitt 6.4 vorgestellten Workflows umgesetzt und kann entsprechend genutzt werden. Nachfolgend wird der Aufbau sowie die Funktionsweise der erstellten Workflows im Detail erläutert.

## 7.5.1 Datenvorbereitung

Ziel des ersten Workflows ist es, die auf Kadi4Mat verfügbaren Daten in ein Format zu überführen, das eine effiziente Nutzung in neuronalen Netzen ermöglicht. Hierfür sollen die Daten als sogenannten *tfrecords* bereitgestellt werden. Der Workflow zur Erstellung dieser *tfrecords* ist in Abbildung 7.9 als BPMN-Diagramm dargestellt.



**Abbildung 7.9:** Workflow zur Vorbereitung der Datenbank. Die zu untersuchende Collection wird ausgewählt. Aus den Metadaten der darin enthaltenen Records werden die Eingangs- und Ausgangswerte gewählt und in *tfrecords* geladen.

Der Workflow gliedert sich in drei aufeinanderfolgende Schritte. Zunächst legt der Nutzer fest, welche Datenbank als Grundlage der Analyse dienen soll, indem er die entsprechende Collection ID angibt. Der Aufbau einer solchen Collection ist in einem vorherigen Abschnitt dieses Kapitels in Abbildung 7.6 gezeigt. Die Metadaten aller Records, die sich in dieser Collection befinden, werden anschließend von Kadi4Mat heruntergeladen und stehen für den nächsten Arbeitsschritt zur Verfügung. In diesem wählt der Nutzer aus den ermittelten Metadaten, die Einträge, die als Eingangs- und Ausgabedaten für das zu trainierende neuronale Netzwerk verwendet werden sollen. Die Auswahl findet dabei über eine graphische Benutzeroberfläche statt in der die entsprechenden Einträge ausgewählt werden können. Abschließend werden die ausgewählten Metadaten automatisch



in tfrecords konvertiert und wieder auf Kadi4Mat hochgeladen. Um die Nachverfolgbarkeit der Datenherkunft sicherzustellen, werden die erzeugten tfrecords der ursprünglich ausgewählten Collection hinzugefügt. Dadurch ist jederzeit ersichtlich, aus welcher Datenbank die Trainingsdaten abgeleitet wurden.

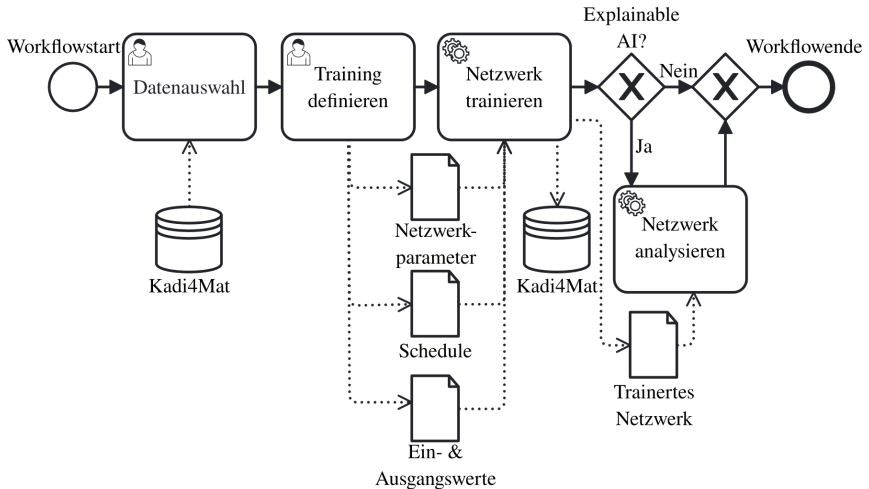
Nach Abschluss dieses Workflows liegen die Daten der ausgewählten Collection in einem standardisierten Format vor, das direkt für den Einsatz in maschinellen Lernverfahren genutzt werden kann.

### 7.5.2 Training

Ausgehend von der vorbereiteten Datenbank implementiert der zweite Workflow das Training eines neuronalen Netzwerks zur Identifikation der Struktur-Eigenschaft-Beziehungen in den vorbereiteten Daten. Wie in Abbildung 7.10 dargestellt, umfasst der Workflow insgesamt vier Aufgaben, wobei die Analyse des Netzwerks optional durchgeführt wird.

Im ersten Schritt wählt der Nutzer die zu verwendenden Daten aus, indem die entsprechenden tfrecords von Kadi4Mat heruntergeladen werden. Anschließend wird das eigentliche Training definiert. Hierbei legt der Nutzer zunächst die Architektur des neuronalen Netzwerks fest, indem er die Anzahl der versteckten Schichten bestimmt. Basierend darauf können anschließend die Anzahl der Neuronen pro Schicht sowie die jeweilige Aktivierungsfunktion ausgewählt werden. Darüber hinaus definiert der Nutzer den Ablauf des Trainings, indem er Parameter wie die Anzahl der Epochen und die Lernrate festlegt. Neben diesen architekturspezifischen Einstellungen wird auch bestimmt, welche Merkmale aus den tfrecords als Eingabe- und welche als Zielgrößen verwendet werden sollen. Alle gewählten Einstellungen werden in Form von json-Dateien gespeichert.

Im nächsten Schritt wird das neuronale Netzwerk auf Basis dieser json-Dateien trainiert. Nach Abschluss des Trainings werden das Netzwerk und die json-Dateien, die zur Parametrisierung verwendet wurden auf Kadi4Mat hochgeladen und mit den verwendeten tfrecords verlinkt. Dies gewährleistet eine vollständige



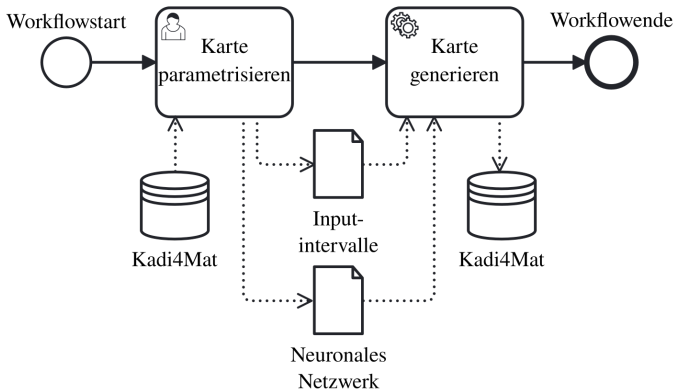
**Abbildung 7.10:** Workflow zum Training des Netzwerks zur Datenbankanalyse. Neben den Trainingsdaten, werden die Netzwerkparameter, der Trainingschedule sowie die Eingangs- und Ausgangsdaten des Netzwerks definiert. Nach Abschluss des Trainings können durch Explainable AI die wichtigsten Eingabewerte identifiziert werden.

Reproduzierbarkeit des gesamten Trainingsprozesses. Optional kann das trainierte Netzwerk anschließend mithilfe von Layer-wise Relevance Propagation analysiert werden. Auf diese Weise lassen sich die Merkmale identifizieren, die den größten Einfluss auf die Vorhersage der Struktur-Eigenschaft-Beziehungen haben.

An dieser Stelle ist der Workflow zum Training des neuronalen Netzwerks abgeschlossen und es steht ein trainiertes Netzwerk für die weitere Nutzung zur Verfügung.

### 7.5.3 Erstellung der Materialkarten

Der dritte Workflow automatisiert die Erstellung der Materialkarten auf Grundlage des zuvor trainierten neuronalen Netzwerks. Abbildung 7.11 zeigt den Ablauf dieses Workflows in Form eines BPMN-Diagramms.



**Abbildung 7.11:** Workflow zur Erstellung der Materialkarten. Ein trainiertes Netzwerk wird geladen und Intervalle der Eingabedaten definiert. Die Vorhersagen des Netzwerks werden als Materialkarte visualisiert.

Die Erstellung der Materialkarte gliedert sich in zwei Arbeitsschritte. Im ersten Schritt wählt der Nutzer aus, welches trainierte Netzwerk verwendet werden soll und legt fest, welche Intervalle für die Eingabeparameter berücksichtigt werden. Das benötigte Netzwerk wird dabei direkt aus Kadi4Mat geladen.

Im zweiten Arbeitsschritt wird basierend auf den gewählten Parametern eine Materialkarte erstellt. Diese wird im Anschluss auf Kadi4Mat hochgeladen. Über die Verknüpfungen des verwendeten neuronalen Netzwerks in Kadi4Mat ist bekannt, mit den Daten welcher Collection es trainiert wurde. Die Karten werden der so identifizierten Collection hinzugefügt. Auf diese Weise ist innerhalb der Collection direkt ersichtlich, welche Struktur-Eigenschaft-Beziehungen in den enthaltenen Records vorliegen.

Mit Abschluss dieses Workflows liegt eine Materialkarte vor, die die identifizierten Struktur-Eigenschaft-Beziehungen verständlich darstellt und somit ein beschleunigtes Materialdesign ermöglicht.

## 7.6 Diskussion

In diesem Kapitel wurde die letzte methodische Komponente des Workflows implementiert: die Identifikation von Struktur-Eigenschaft-Beziehungen auf Basis rekonstruierten und generierten Schaumstrukturen. Damit vollzieht sich der Übergang von der Datenbereitstellung hin zur systematischen Analyse und die Methodik für digitales Materialdesign wird abgeschlossen.

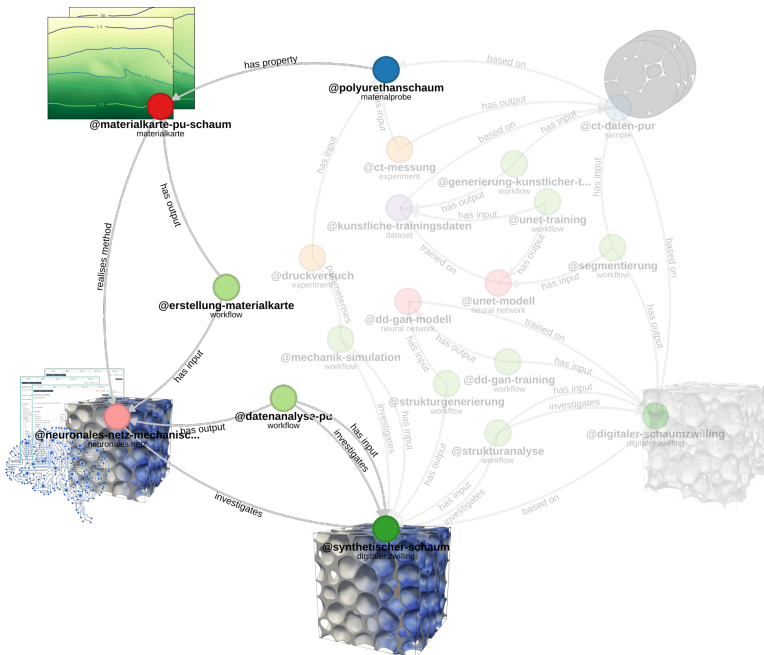
Die Ergebnisse zeigen, dass sich durch die Kombination generativer Modelle, datengetriebener Charakterisierung und Simulation eine vielfältige Datenbasis aufbauen lässt, die eine belastbare Grundlage für die Analyse von Struktur-Eigenschaft-Beziehungen bildet. Besonders hervorzuheben ist die gezielte Ergänzung bislang unzureichend abgedeckter Bereiche, wodurch Verzerrungen reduziert und eine gleichmäßige Datenverteilung erreicht wurden.

Darauf aufbauend konnte ein neuronales Netz trainiert werden, das die mechanischen Eigenschaften Druckmodul und Stauchgrenze aus wenigen Strukturparametern mit hoher Genauigkeit vorhersagt. Die Analyse mittels Explainable AI belegte den dominanten Einfluss von Porosität und Wanddicke. Durch die Reduktion auf diese beiden Parameter entstand ein kompaktes, interpretierbares Modell, dessen Materialkarten Struktur-Eigenschaft-Beziehungen anschaulich darstellen und sowohl Vorhersagen als auch inverse Strukturauslegungen ermöglichen. Statt klassischer statistischer Verfahren wurde bewusst die LRP gewählt, da sie für nichtlineare Zusammenhänge geeignet ist und sich sowohl auf MLPs als auch auf CNNs anwenden lässt. Sie bietet damit größere Flexibilität und unterstützt das Ziel einer einheitlichen Methodik über verschiedene Datenarten hinweg.

Auch wenn kleinere lokale Ungenauigkeiten in den Vorhersagen zu geringfügigen Unregelmäßigkeiten in den Materialkarten führen, beeinträchtigen diese die Aussagekraft nicht. Vielmehr belegen die insgesamt niedrigen Fehlermaße die Zuverlässigkeit der identifizierten Zusammenhänge und bestätigen die Eignung des gewählten Ansatzes. Die Umsetzung der entwickelten Methodne als Workflows dokumentiert sämtliche Analysen und Ergebnisse reproduzierbar. Gleichzeitig

ermöglicht die modulare Struktur der Workflows eine einfache Übertragbarkeit auf andere Materialsysteme.

Zusammenfassend zeigt dieses Kapitel, dass mit der vorgestellten Methodik nicht nur eine vollständige Prozesskette von der Rekonstruktion bis zur Simulation aufgebaut wurde, sondern auch konkrete Struktur-Eigenschaft-Beziehungen identifiziert werden konnten. Damit bildet die Arbeit eine Brücke zwischen methodischer Innovation und materialwissenschaftlicher Erkenntnis und eröffnet zugleich neue Möglichkeiten für datengetriebenes, beschleunigtes Materialdesign. Der abschließende Stand des Materialentwicklungsprozesses ist in Abbildung 7.12 dargestellt. Alle eingangs identifizierten Methoden wurden erfolgreich umgesetzt.



**Abbildung 7.12:** Zusammenfassung der Identifikation von Struktur-Eigenschaft-Beziehungen. Die Workflows zur Generierung und Simulation ermöglichen den Aufbau einer analysierbaren Datenbank.

## 8 Zusammenfassung und Ausblick

### 8.1 Zusammenfassung

In dieser Arbeit wurde der vollständige digitale Materialentwicklungsprozess exemplarisch am Beispiel offenzelliger Polyurethanschaumstrukturen umgesetzt. Für jeden Prozessschritt wurden generische, modulare Workflows konzipiert, die reproduzierbar, skalierbar und auf andere Materialsysteme übertragbar sind. Als methodische Grundlage wurde ein eigenes Konzept für FAIRes Forschungsdatenmanagement erarbeitet, das sämtliche Arbeitsschritte und Forschungsdaten in einer virtuellen Forschungsumgebung dokumentiert, verknüpft und damit eine durchgängige Nachvollziehbarkeit sicherstellt.

Das methodische Fundament der Arbeit bildet ein eigens entwickeltes Konzept zum FAIRen Materialdesign, das Prozesse und Daten systematisch strukturiert. Durch die iterative Anwendung des Input–Process–Output-Prinzips konnten alle relevanten Schritte – von der Rekonstruktion und Charakterisierung digitaler Zwillinge über die Generierung synthetischer Strukturen und die Simulation makroskopischer Eigenschaften bis hin zur datengetriebenen Analyse – in atomare, wiederverwendbare Arbeitspakete zerlegt und formalisiert werden. Dieses Modellierungskonzept wurde im Workflowsystem KadiStudio praktisch umgesetzt und steuert den Aufbau aller folgenden Workflows. Zudem wurde in den Methoden das entwickelte Konzept der wachsenden Metadaten erprobt. Die untersuchten Schäume wurden dazu inkrementell entlang des gesamten Forschungsprozesses mit Metadaten versehen.

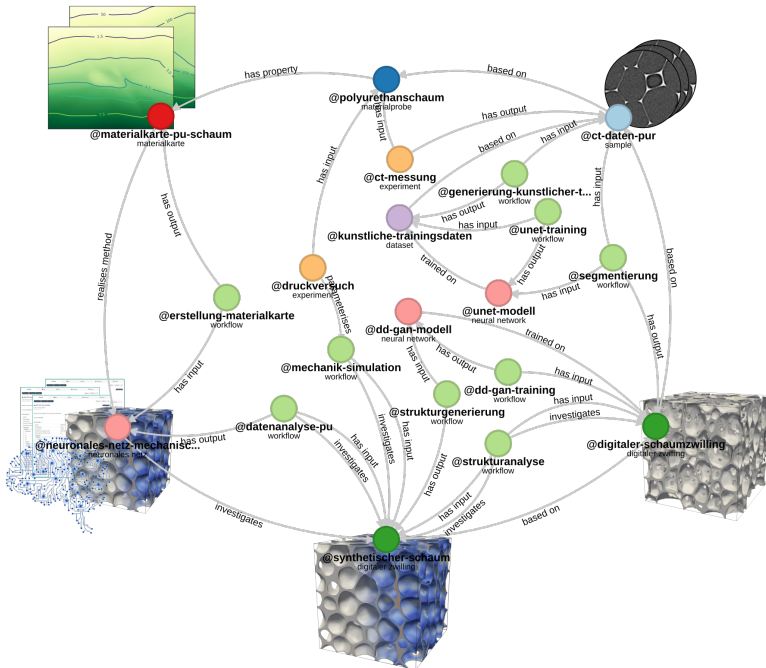
Für den ersten Schritt des Materialdesigns wurde eine Methode zur automatisierten Erzeugung synthetischer Trainingsdaten für ML-basierte Segmentierungen entwickelt, die eine schnelle und präzise Rekonstruktion digitaler Zwillinge aus CT-Daten ermöglicht. Die Segmentierung und anschließende geometrische Analyse, unter anderem hinsichtlich der Struktureigenschaften Porosität, Porengrößenverteilung und Wandstärke, wurden vollständig in Workflows überführt und FAIR dokumentiert.

Zur gezielten Erweiterung des Strukturraums wurde ein generatives maschinelles Lernverfahren in Form eines 3D-Diffusion-GANs entwickelt. Dies ermöglicht die effiziente Erzeugung realistischer Mikrostrukturen mit kontrollierbaren Eigenschaften. Neben einer eigenschaftskonditionierten Variante wurde auch eine bildkonditionierte Version implementiert, die keine explizite Merkmalerfassung voraussetzt. Beide Ansätze wurden in reproduzierbare Trainings- und Generierungsworkflows integriert und in der Forschungsumgebung Kadi4Mat verankert, wodurch eine systematische Erstellung vielfältiger Strukturen möglich wurde.

Für die anschließende Analyse makroskopischer Eigenschaften wurde ein generischer Workflow zur Durchführung mechanischer Simulationen auf Basis des PACE3D-Mechaniksolvers entwickelt. Das Modell wurde anhand experimenteller Daten parametrisiert und validiert. Der hier entwickelte Workflow erlaubt eine zuverlässige und reproduzierbare Ermittlung mechanischer Kennwerte wie Druckmodul und Stauchgrenze und kann flexibel auf weitere physikalische Fragestellungen ausgedehnt werden.

Alle generierten und simulierten Daten flossen in eine gezielt aufgebaute, strukturell vielfältige Materialdatenbank ein. Diese Datenbasis wurde anschließend mithilfe eines generischen, datengetriebenen Analyseworkflows ausgewertet. Dieser kombiniert das Training eines neuronalen Netzwerks mit Methoden der Explainable AI, um Zusammenhänge zwischen dem mikrostrukturellen Aufbau und den makroskopischen mechanischen Eigenschaften der Schaumstrukturen transparent und nachvollziehbar zu identifizieren. Das trainierte Netzwerk ist in der Lage, mechanische Eigenschaften zuverlässig aus wenigen, direkt messbaren

Strukturparametern vorherzusagen. Mithilfe der erklärbaren Analyse konnten insbesondere Porosität und Wandstärke als zentrale Einflussgrößen herausgearbeitet werden. Auf Grundlage des neuronalen Netzwerks wurden interpretierbare Materialkarten erstellt, die eine gezielte Strukturauslegung ermöglichen. Abbildung 8.1 fasst diesen durchgängigen Materialentwicklungsprozess mitsamt Methoden und Forschungsdatenmanagement schematisch zusammen.



**Abbildung 8.1:** Zusammenfassung des gesamten entwickelten Materialentwicklungsprozesses. Ausgehend von CT-Daten konnte eine Prozesskette bis zur Erstellung von Materialkarten entwickelt werden.

Insgesamt wurde gezeigt, dass sich der Materialentwicklungsprozess – von der Datenerfassung über die Strukturmodellierung und Simulation bis hin zur datengetriebenen Analyse – in einer einheitlichen, reproduzierbaren und skalierbaren



Struktur abbilden lässt. Die entwickelten Workflows sind generisch, modular aufgebaut und FAIR dokumentiert, sodass sie über den hier behandelten Anwendungsfall hinaus eingesetzt werden können. Der gezielte Einsatz maschineller Lernverfahren ermöglichte die Automatisierung zentraler Schritte und trug wesentlich zur Beschleunigung des gesamten Entwicklungsprozesses bei.

Am Beispiel offenzelliger Polyurethanschäumstrukturen im Fahrzeugleichtbau konnte zudem gezeigt werden, dass sich mechanische Kennwerte wie Druckmodul und Stauchgrenze mit hoher Genauigkeit auf Basis weniger aus der Mikrostruktur abgeleiteter Parameter vorhersagen lassen. Dies reduziert den Bedarf an zeit- und kostenintensiven experimentellen Testreihen erheblich und ermöglicht die gezielte Auswahl vielversprechender Strukturvarianten. Die Arbeit leistet damit nicht nur einen methodischen Beitrag zum generischen, datengetriebenen Materialdesign, sondern auch einen konkreten Mehrwert für die effiziente Entwicklung energieabsorbierender Leichtbauelemente.

## 8.2 Ausblick

Der folgende Ausblick skizziert zentrale Entwicklungspfade, die sich aus den in dieser Arbeit entwickelten Methoden und Konzepten ableiten lassen. Dabei werden sowohl kurzfristig umsetzbare Schritte als auch längerfristige Perspektiven betrachtet, die den digitalen Materialentwicklungsprozess erweitern und vertiefen.

### Skalierbare Übertragbarkeit

Ein naheliegender nächster Schritt liegt in der Überprüfung der Übertragbarkeit der entwickelten Workflows und Konzepte auf andere Materialsysteme. Während die Methoden in dieser Arbeit validiert und am Beispiel von Polyurethanschäumstrukturen erfolgreich eingesetzt wurden, stellt sich die entscheidende Frage, wie robust und anpassungsfähig sie in anderen Domänen sind. Besonders aussagekräftig ist hier die sogenannte *Cross-Domain Evaluation*, bei der Modelle

und Workflows auf Daten angewandt werden, die aus einem gänzlich anderen, aber verwandten Materialsystem stammen. Solche Untersuchungen erlauben eine Bewertung der Generalisierbarkeit und liefern Hinweise, wie universell die entwickelten Methoden einsetzbar sind.

## Analysegetriebenes Design

Ein vielversprechender Entwicklungspfad liegt in der engen Verzahnung von *Active Learning* mit Strukturgenerierung und Simulation. Anstatt die datengetriebene Analyse erst am Ende des Prozesses durchzuführen, wird sie von Beginn an als steuerndes Element etabliert. Auf einer bestehenden, zunächst kleinen Materialdatenbank kann ein Modell trainiert werden, das Zusammenhänge zwischen mikrostrukturellem Aufbau und makroskopischen Eigenschaften der Materialien erlernt. Dieses Modell identifiziert gezielt Bereiche der Vorhersage mit hoher Unsicherheit und wählt daraus Strukturkandidaten aus, deren Untersuchung den größten Informationsgewinn verspricht. Die ausgewählten Strukturen werden generiert, simuliert oder experimentell charakterisiert, und die neu gewonnenen Daten fließen unmittelbar zurück in das Modell. Durch diese iterativen Optimierungen wird das Modell robuster, und die Anzahl notwendiger Simulationen oder Generierungen reduziert sich deutlich. Damit entsteht die methodische Grundlage für geschlossene Optimierungszyklen.

## Anbindung von Laboren

Während in dieser Arbeit ausschließlich digital erzeugte Daten genutzt wurden, könnten künftig auch experimentelle Ergebnisse automatisiert erfasst, strukturiert dokumentiert und in die bestehende virtuelle Forschungsumgebung integriert werden. Technisch erfordert dies geeignete Schnittstellen zu Laborgeräten sowie standardisierte Metadatenschemata, beispielsweise basierend auf Metadata4Ing [81], ergänzt durch geeignete Konvertierungsmechanismen zwischen unterschiedlichen

Schemata. Erste grundlegende Schritte zur automatisierten Dokumentation experimenteller Forschungsprozesse wurden bereits in [79, 216] aufgezeigt und können hier als methodische Basis dienen.

Besonders für experimentelle Untersuchungen, die im Vergleich zu Simulationen deutlich zeit- und kostenintensiver sind, bietet dabei das zuvor beschriebene Active Learning ein großes Potenzial. In Kombination mit einer Laboranbindung ermöglicht es die gezielte Auswahl der aussagekräftigsten Materialkandidaten, was sowohl eine effiziente Ressourcennutzung als auch eine deutlich beschleunigte Erkenntnisgewinnung zur Folge hat.

Die Kombination dieser Ansätze ebnet letztlich den Weg zu einer *Material Acceleration Platform* (MAP), in der digitale und experimentelle Daten in einer einheitlich strukturierten, kontinuierlich wachsenden Materialdatenbank zusammenfließen und geschlossene, selbstoptimierende Entwicklungszyklen ermöglichen [217, 218].

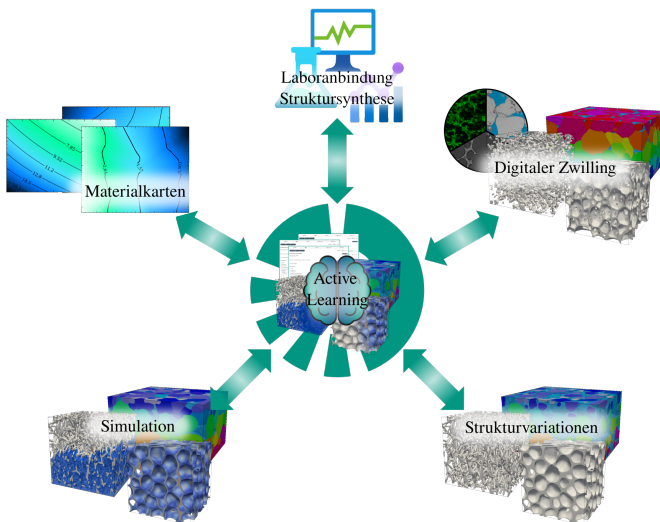
### Prozess–Struktur–Eigenschaft-Beziehungen

Ein weiterer langfristiger Entwicklungspfad liegt in der Integration der Herstellungsprozesse in den digitalen Materialentwicklungszyklus. Während diese Arbeit auf die digitale Generierung von Mikrostrukturen fokussiert ist, adressiert die Struktursynthese die reale Fertigung und Verarbeitung der Materialien. Erst die Verknüpfung beider Ansätze eröffnet die Möglichkeit, nicht nur Zielstrukturen zu definieren, sondern auch geeignete Herstellungsrouten abzuleiten.

Die Abbildung von Prozess–Struktur–Eigenschaft-Beziehungen gehört jedoch zu den anspruchsvollsten Fragestellungen der Materialwissenschaft. Sie setzt detaillierte Prozessdaten, robuste multiskalige Modelle und eine enge Validierung voraus. Dieser Themenkomplex ist daher Gegenstand umfangreicher Forschungsinitiativen. Gleichwohl stellt die perspektivische Einbindung realer Prozesssimulationen einen wesentlichen Schritt dar, um den Weg zu einer vollständig inversen Materialentwicklung zu ebnen.

## Langfristige Vision

Langfristig können diese Erweiterungen dazu beitragen, den digitalen Materialentwicklungsprozess zu einem vollständig autonomen, adaptiven System auszubauen. Durch die Integration von Active Learning, der direkten Laboranbindung und der Erweiterung um Prozess-Struktur-Eigenschaft-Beziehungen entsteht ein neuer, geschlossener Entwicklungszyklus, in dem die datengetriebene Analyse als zentrales, orchestrierendes Element fungiert. Dieser erweiterte Kreislauf ist in



**Abbildung 8.2:** Langfristige Vision für die Materialentwicklung. Erweiterter, adaptiver Entwicklungszyklus mit Active Learning, Laboranbindung und Prozess–Struktur–Eigenschafts-Beziehungen, in dem die datengetriebene Analyse als zentrales Steuerungselement ein selbstoptimierendes Materialdesignsystem ermöglicht.

Abbildung 8.2 dargestellt und verdeutlicht den Übergang von einem linearen, schrittweisen Vorgehen zu einem kontinuierlich lernenden, selbstoptimierenden System. Ein solches System würde fortlaufend aus digitalen und experimentellen Daten lernen, den Ressourceneinsatz minimieren und den Weg zu neuen Materialien erheblich beschleunigen.



# Literaturverzeichnis

- [1] European Organization For Nuclear Research und OpenAIRE, *Zenodo*, en, 2013. DOI: 10.25495/7GXX-RD71.
- [2] G. King, *An introduction to the dataverse network as an infrastructure for data sharing*, 2007.
- [3] B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke und I. Foster, „The materials data facility: data services to advance materials science research“, *Jom*, Jg. 68, Nr. 8, S. 2045–2052, 2016.
- [4] B. Blaiszik, L. Ward, M. Schwarting, J. Gaff, R. Chard, D. Pike, K. Chard und I. Foster, „A data ecosystem to support machine learning in materials science“, *MRS Communications*, Jg. 9, Nr. 4, S. 1125–1133, 2019.
- [5] H. Pampel, P. Vierkant, F. Scholze, R. Bertelmann, M. Kindling, J. Klump, H.-J. Goebelbecker, J. Gundlach, P. Schirmbacher und U. Dierolf, „Making research data repositories visible: the re3data.org registry“, *PloS one*, Jg. 8, Nr. 11, e78080, 2013.
- [6] Deutsche Forschungsgemeinschaft. „DFG RISources“. (2025), Adresse: <https://risources.dfg.de> (eingesehen am 18.06.2025).
- [7] N. CARPi, A. Minges und M. Piel, „eLabFTW: An open source laboratory notebook for research labs“, *Journal of Open Source Software*, Jg. 2, Nr. 12, S. 146, 2017. DOI: 10.21105/joss.00146.
- [8] P. Tremouilhac, A. Nguyen, Y.-C. Huang, S. Kotov, D. S. Lütjohann, F. Hübsch, N. Jung und S. Bräse, „Chemotion ELN: an Open Source electronic lab notebook for chemists in academia“, *Journal of Cheminformatics*, Jg. 9, S. 1–13, 2017.

- [9] J. Potthoff, P. Tremouilhac, P. Hodapp, B. Neumair, S. Bräse und N. Jung, „Procedures for systematic capture and management of analytical data in academia“, *Analytica chimica acta*: X, Jg. 1, S. 100 007, 2019.
- [10] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing und J. development team, „Jupyter Notebooks a publishing format for reproducible computational workflows“, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides und B. Schmidt, Hrsg., IOS Press, 2016, S. 87–90. DOI: 10.3233/978-1-61499-649-1-87.
- [11] E. Afgan, D. Baker, B. Batut, M. Van Den Beek, D. Bouvier, M. Čech, J. Chilton, D. Clements, N. Coraor, B. A. Grüning u. a., „The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update“, *Nucleic acids research*, Jg. 46, Nr. W1, W537–W544, 2018. DOI: <https://doi.org/10.1093/nar/gky379>.
- [12] S. P. Huber, S. Zoupanos, M. Uhrin, L. Talirz, L. Kahle, R. Häuselmann, D. Gresch, T. Müller, A. V. Yakutovich, C. W. Andersen u. a., „AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance“, *Scientific data*, Jg. 7, Nr. 1, S. 300, 2020.
- [13] M. Uhrin, S. P. Huber, J. Yu, N. Marzari und G. Pizzi, „Workflows in AiiDA: Engineering a high-throughput, event-based engine for robust and modular computational workflows“, *Computational Materials Science*, Jg. 187, S. 110 086, 2021.
- [14] nteract contributors, *nteract/papermill: Parameterize, execute, and analyze Jupyter Notebooks*, <https://github.com/nteract/papermill>, 2017.
- [15] T. G. Community, „The Galaxy platform for accessible, reproducible, and collaborative data analyses: 2024 update“, *Nucleic Acids Research*, Jg. 52, Nr. W1, W83–W94, Mai 2024. DOI: 10.1093/nar/gkae410.

- [16] A. Meyer, H. Kutabi, J. Bessai und D. Scholtyssek, „CLS-Luigi: Analytics Pipeline Synthesis“, in *International Conference on Learning and Intelligent Optimization*, Springer, 2024, S. 269–284.
- [17] J. Rehof und P. Urzyczyn, „Finite combinatory logic with intersection types“, in *International Conference on Typed Lambda Calculi and Applications*, Springer, 2011, S. 169–183.
- [18] Spotify, *Luigi: a Python module for building complex pipelines*, <https://github.com/spotify/luigi>, 2011.
- [19] I. Goodfellow, Y. Bengio, A. Courville und Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, Bd. 1.
- [20] F. Rosenblatt, „The perceptron: a probabilistic model for information storage and organization in the brain.“, *Psychological review*, Jg. 65, Nr. 6, S. 386, 1958.
- [21] X. Glorot und Y. Bengio, „Understanding the difficulty of training deep feedforward neural networks“, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop und Conference Proceedings, 2010, S. 249–256.
- [22] S. R. Dubey, S. K. Singh und B. B. Chaudhuri, „Activation functions in deep learning: A comprehensive survey and benchmark“, *Neurocomputing*, Jg. 503, S. 92–108, 2022.
- [23] V. Nair und G. E. Hinton, „Rectified linear units improve restricted boltzmann machines“, in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, S. 807–814.
- [24] M. A. Mercioni und S. Holban, „The most used activation functions: Classic versus current“, in *2020 International Conference on Development and Application Systems (DAS)*, IEEE, 2020, S. 141–145.
- [25] D.-A. Clevert, T. Unterthiner und S. Hochreiter, „Fast and accurate deep network learning by exponential linear units (elus)“, *arXiv preprint arXiv:1511.07289*, Jg. 4, Nr. 5, S. 11, 2015.



- [26] S. Elfving, E. Uchibe und K. Doya, „Sigmoid-weighted linear units for neural network function approximation in reinforcement learning“, *Neural networks*, Jg. 107, S. 3–11, 2018.
- [27] R. Kruse, S. Mostaghim, C. Borgelt, C. Braune und M. Steinbrecher, *Computational Intelligence - A Methodological Introduction, 3rd edition*. English. Springer Verlag, März 2022. DOI: 10.1007/978-3-030-42227-1.
- [28] D. P. Kingma und J. Ba, „Adam: A method for stochastic optimization“, *arXiv preprint arXiv:1412.6980*, 2014.
- [29] S. Ioffe und C. Szegedy, „Batch normalization: Accelerating deep network training by reducing internal covariate shift“, in *International conference on machine learning*, pmlr, 2015, S. 448–456.
- [30] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville und Y. Bengio, *Generative Adversarial Networks*, 2014.
- [31] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan und S. Ganguli, „Deep unsupervised learning using nonequilibrium thermodynamics“, in *International conference on machine learning*, pmlr, 2015, S. 2256–2265.
- [32] J. Ho, A. Jain und P. Abbeel, „Denoising diffusion probabilistic models“, *Advances in neural information processing systems*, Jg. 33, S. 6840–6851, 2020.
- [33] J. R. Norris, *Markov chains*. Cambridge university press, 1998.
- [34] Z. Xiao, K. Kreis und A. Vahdat, „Tackling the generative learning trilemma with denoising diffusion GANs“, *arXiv preprint arXiv:2112.07804*, 2021.
- [35] M. B. Said, „Phasenfeldmodellierung und Simulation des Benetzungsverhaltens von Flüssigkeiten auf festen Oberflächen“, Dissertation, Karlsruher Institut für Technologie, Karlsruhe, 2018.

- [36] J. v. d. Waals, „Thermodynamische theorie der kapillarität unter voraussetzung stetiger dichteänderung“, *Zeitschrift für Physikalische Chemie*, Jg. 13, Nr. 1, S. 657–725, 1894.
- [37] X. Li, J. Lowengrub, A. Rätz und A. Voigt, „Solving PDEs in complex geometries: a diffuse domain approach“, *Communications in mathematical sciences*, Jg. 7, Nr. 1, S. 81, 2009.
- [38] E. Schoof, „Chemomechanische Modellierung der Wärmebehandlung von Stählen mit der Phasenfeldmethode“, Diss., Karlsruher Institut für Technologie (KIT), 2020, 229 S. DOI: 10.5445/IR/1000117917.
- [39] N. Prajapati, C. Herrmann, M. Späth, D. Schneider, M. Selzer und B. Nestler, „Brittle anisotropic fracture propagation in quartz sandstone: insights from phase-field simulations“, *Computational Geosciences*, Jg. 24, S. 1361–1376, 2020.
- [40] M. D. Reder, „Modellierung und numerische Simulation der Interaktion von Mehrphasenströmungen mit Festkörpern basierend auf einem Phasenfeldansatz“, Diss., Karlsruher Institut für Technologie (KIT), 2024, 195 S. DOI: 10.5445/IR/1000167837.
- [41] B. Nestler, H. Garcke und B. Stinner, „Multicomponent alloy solidification: phase-field modeling and simulations“, *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, Jg. 71, Nr. 4, S. 041 609, 2005.
- [42] I. Gitman, H. Askes und L. Sluys, „Representative volume: Existence and size determination“, *Engineering fracture mechanics*, Jg. 74, Nr. 16, S. 2518–2534, 2007.
- [43] S. R. du Roscoat, M. Decain, X. Thibault, C. Geindreau und J.-F. Bloch, „Estimation of microstructural properties from synchrotron X-ray microtomography and determination of the REV in paper materials“, *Acta Materialia*, Jg. 55, Nr. 8, S. 2841–2850, 2007.

- [44] W. M. Harris und W. K. Chiu, „Determining the representative volume element size for three-dimensional microstructural material characterization. Part 1: Predictive models“, *Journal of Power Sources*, Jg. 282, S. 552–561, 2015.
- [45] C. Draxl und M. Scheffler, „Big data-driven materials science and its FAIR data infrastructure“, *Handbook of Materials Modeling: Methods: Theory and Modeling*, S. 49–73, 2020. DOI: [https://doi.org/10.1007/978-3-319-44677-6\\_104](https://doi.org/10.1007/978-3-319-44677-6_104).
- [46] M. Putnings, H. Neuroth und J. Neumann, Hrsg., *Praxishandbuch Forschungsdatenmanagement*. De Gruyter, 18. Jan. 2021. DOI: 10.1515/9783110657807.
- [47] Deutsche Forschungsgemeinschaft, „Leitlinien zum Umgang mit Forschungsdaten“, Deutsche Forschungsgemeinschaft, 30. Sep. 2015.
- [48] M. Kindling und P. Schirmbacher, „Die digitale Forschungswelt“ als Gegenstand der Forschung / Research on Digital Research / Recherche dans la domaine de la recherche numérique“, *Information - Wissenschaft & Praxis*, Jg. 64, Nr. 2-3, 1. Apr. 2013. DOI: 10.1515/iwp-2013-0017.
- [49] Publications Office of the European Union, *Einführung in das Metadatenmanagement*, 2013.
- [50] J. Riley, „Understanding metadata“, *Washington DC, United States: National Information Standards Organization (<http://www.niso.org/publications/press/UnderstandingMetadata.pdf>)*, Jg. 23, S. 7–10, 2017.
- [51] S. Büttner, H.-C. Hobohm und L. Müller, Hrsg., *Handbuch Forschungsdatenmanagement*. Bad Honnef: Bock + Herchen Verlag, 2011, 223 S., ISBN: 978-3-88347-283-6.
- [52] T. Berners-Lee, J. Hendler und O. Lassila, „The semantic web“, *Scientific american*, Jg. 284, Nr. 5, S. 34–43, 2001.

- [53] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne u. a., „The FAIR Guiding Principles for scientific data management and stewardship“, *Scientific data*, Jg. 3, Nr. 1, S. 1–9, 2016. DOI: <https://doi.org/10.1038/sdata.2016.18>.
- [54] P. B. Heidorn, „Shedding light on the dark data in the long tail of science“, *Library trends*, Jg. 57, Nr. 2, S. 280–299, 2008.
- [55] T. Goetz, „Freeing the dark data of failed scientific experiments“, *Wired Magazine*, Jg. 15, Nr. 10, S. 15–10, 2007.
- [56] J. Porter, „Chapter 3: Scientific Databases“, in *Ecological Data: Design, Management and Processing*, W. K. Michener und J. W. Brunt, Hrsg., Oxford, UK: Blackwell Science, 2000, S. 48–69.
- [57] H. K. Machina und D. J. Wild, „Electronic laboratory notebooks progress and challenges in implementation“, *Journal of laboratory automation*, Jg. 18, Nr. 4, S. 264–268, 2013.
- [58] E. D. Foster, E. C. Whipple und G. R. Rios, „Implementing an institution-wide electronic lab notebook initiative“, *Journal of the Medical Library Association: JMLA*, Jg. 110, Nr. 2, S. 222, 2022.
- [59] S. G. Higgins, A. A. Nogiwa-Valdez und M. M. Stevens, „Considerations for implementing electronic laboratory notebooks in an academic research environment“, *Nature Protocols*, Jg. 17, Nr. 2, S. 179–189, 2022.
- [60] A. ter Hofstede, W. M. van der Aalst, A. H. ter Hofstede und M. Weske, „Business process management: A survey“, in *Business Process Management: International Conference, BPM 2003 Eindhoven, The Netherlands, June 26–27, 2003 Proceedings I*, Springer, 2003, S. 1–12.
- [61] E. Deelman, D. Gannon, M. Shields und I. Taylor, „Workflows and e-Science: An overview of workflow system features and capabilities“, *Future generation computer systems*, Jg. 25, Nr. 5, S. 528–540, 2009.
- [62] V. Curcin und M. Ghanem, „Scientific workflow systems-can one size fit all?“, in *2008 Cairo International Biomedical Engineering Conference*, IEEE, 2008, S. 1–9.

- [63] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo und C. Notredame, „Nextflow enables reproducible computational workflows“, *Nature biotechnology*, Jg. 35, Nr. 4, S. 316–319, 2017.
- [64] F. Mölder, K. P. Jablonski, B. Letcher, M. B. Hall, C. H. Tomkins-Tinch, V. Sochat, J. Forster, S. Lee, S. O. Twardziok, A. Kanitz u. a., „Sustainable data analysis with Snakemake“, *F1000Research*, Jg. 10, 2021.
- [65] S. Bowers, „Scientific workflow, provenance, and data modeling challenges and approaches“, *Journal on Data Semantics*, Jg. 1, S. 19–30, 2012.
- [66] R. H. Schmitt, S. Başkaya, F. Claus, J. Schwibs, R. Gerike, G. Jagusch, M. S. Müller, W. Stille, C. Bischof, P. Pelz u. a., „NFDI4Ing-the national research data infrastructure for engineering sciences“, *Werkstoffsynthese und Herstellungsverfahren*, Techn. Ber., 2020.
- [67] N. Brandt, L. Griem, C. Herrmann, E. Schoof, G. Tosato, Y. Zhao, P. Zschumme und M. Selzer, „Kadi4Mat: A Research Data Infrastructure for Materials Science“, *Data Science Journal*, Jg. 20, Nr. 1, 2021. DOI: <http://doi.org/10.5334/dsj-2021-008>.
- [68] L. Griem, P. Zschumme, M. Laqua, N. Brandt, E. Schoof, P. Altschuh und M. Selzer, „KadiStudio: FAIR Modelling of Scientific Research Processes“, Englisch, *Data Science Journal*, Jg. 21, Nr. 1, Art.–Nr: 16, 2022, 43.31.01; LK 01. DOI: 10.5334/dsj-2022-016.
- [69] A. Goel, *Computer fundamentals*. Pearson Education India, 2010, ISBN: 978-81-317-3309-7.
- [70] J. M. Zelle, *Python programming: an introduction to computer science*. Franklin, Beedle & Associates, Inc., 2004.
- [71] M. R. Crusoe u. a., „Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language“, *Commun. ACM*, Jg. 65, Nr. 6, S. 54–63, Mai 2022. DOI: 10.1145/3486897.

- [72] M. R. Berthold, N. Cebon, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel und B. Wiswedel, „KNIME-the Konstanz information miner: version 2.0 and beyond“, *AcM SIGKDD explorations Newsletter*, Jg. 11, Nr. 1, S. 26–31, 2009.
- [73] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevár, M. Milutinović, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik und B. Zupan, „Orange: Data Mining Toolbox in Python“, *Journal of Machine Learning Research*, Jg. 14, S. 2349–2353, 2013.
- [74] Kadi4Mat Team and Contributors, *kadistudio: 0.1.0.alpha1*, Version 0.1.0.alpha1, Juli 2022. DOI: 10.5281/zenodo.6810891.
- [75] P. e. a. Dmitry, *Qt5 Node Editor*, <https://github.com/paceholder/nodeeditor>, 2017.
- [76] G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari und B. Kozinsky, „AiiDA: automated interactive infrastructure and database for computational science“, *Computational Materials Science*, Jg. 111, S. 218–230, 2016. DOI: <https://doi.org/10.1016/j.commatsci.2015.09.013>.
- [77] K. Team und Contributors, *xmlhelpy: 0.14.0*, Version 0.14.0, Dez. 2024. DOI: 10.5281/zenodo.14513901.
- [78] Kadi4Mat Team and Contributors, *IAM-CMS/kadi-apy: Kadi4Mat API Library*, Version 0.23.0, Juni 2022. DOI: 10.5281/zenodo.6623518.
- [79] L. C. Griem, R. Thelen und M. Selzer, „Automated Documentation of Research Processes Using RDM“, in *Proceedings of the Conference on Research Data Infrastructure*, Bd. 1, 2023.
- [80] N. Brandt, „Strukturiertes Management von Forschungsdaten in den Ingenieurwissenschaften“, Diss., Karlsruher Institut für Technologie (KIT), 2024, 201 S. DOI: 10.5445/IR/1000168522.

- [81] D. Iglezakis, G. Lanza, S. Arndt, M. Fuhrmans, D. Terzijska, S. Leimer, J. Theissen-Lipp, S. Hachinger, J. Munke, B. Farnbacher, V. Sdralia und A. Wellmann, *Metadata4Ing: An ontology for describing the generation of research data within a scientific activity*, Version 1.3.1, März 2025.  
DOI: 10.5281/zenodo.14982558.
- [82] L. M. Anovitz und D. R. Cole, „Characterization and analysis of porosity and pore structures“, *Reviews in Mineralogy and geochemistry*, Jg. 80, Nr. 1, S. 61–164, 2015.
- [83] J. Conder, C. Marino, P. Novák und C. Villevieille, „Do imaging techniques add real value to the development of better post-Li-ion batteries?“, *Journal of Materials Chemistry A*, Jg. 6, Nr. 8, S. 3304–3327, 2018.
- [84] Y.-N. Wang, J. Wei, Q. She, F. Pacheco und C. Y. Tang, „Microscopic characterization of FO/PRO membranes—a comparative study of CLSM, TEM and SEM“, *Environmental science & technology*, Jg. 46, Nr. 18, S. 9995–10 003, 2012.
- [85] L. Vásárhelyi, Z. Kónya, Á. Kukovecz und R. Vajtai, „Microcomputed tomography–based characterization of advanced materials: a review“, *Materials Today Advances*, Jg. 8, S. 100 084, 2020.
- [86] N. E. Omori, A. D. Bobitan, A. Vamvakeros, A. M. Beale und S. D. Jacques, „Recent developments in X-ray diffraction/scattering computed tomography for materials science“, *Philosophical Transactions of the Royal Society A*, Jg. 381, Nr. 2259, S. 20 220 350, 2023.
- [87] P. J. Withers, C. Bouman, S. Carmignato, V. Cnudde, D. Grimaldi, C. K. Hagen, E. Maire, M. Manley, A. Du Plessis und S. R. Stock, „X-ray computed tomography“, *Nature Reviews Methods Primers*, Jg. 1, Nr. 1, S. 18, 2021.
- [88] C. J. Werth, C. Zhang, M. L. Brusseau, M. Oostrom und T. Baumann, „A review of non-invasive imaging methods and applications in contaminant hydrogeology research“, *Journal of contaminant hydrology*, Jg. 113, Nr. 1–4, S. 1–24, 2010.

- [89] Y. Ouyang, S. Luo, P. Ren, H. Wang, Y. Wu, Y. Fu und H. Wang, „Synchrotron X-ray computed tomography analysis of the morphological characterization of aluminum alloy powders produced by gas atomization“, *Powder Technology*, Jg. 429, S. 118–904, 2023.
- [90] R. Schulze, U. Heil, D. Groß, D. D. Bruellmann, E. Dranischnikow, U. Schwanecke und E. Schoemer, „Artefacts in CBCT: a review“, *Dentomaxillofacial Radiology*, Jg. 40, Nr. 5, S. 265–273, 2011.
- [91] T. Ridler, S. Calvard u. a., „Picture thresholding using an iterative selection method“, *IEEE trans syst Man Cybern*, Jg. 8, Nr. 8, S. 630–632, 1978.
- [92] A. Rosenfeld und P. De La Torre, „Histogram concavity analysis as an aid in threshold selection“, *IEEE Transactions on Systems, Man, and Cybernetics*, Nr. 2, S. 231–235, 1983.
- [93] J. N. Kapur, P. K. Sahoo und A. K. Wong, „A new method for gray-level picture thresholding using the entropy of the histogram“, *Computer vision, graphics, and image processing*, Jg. 29, Nr. 3, S. 273–285, 1985.
- [94] N. Ramesh, J.-H. Yoo und I. Sethi, „Thresholding based on histogram approximation“, *IEE Proceedings-Vision, Image and Signal Processing*, Jg. 142, Nr. 5, S. 271–279, 1995.
- [95] N. Otsu u. a., „A threshold selection method from gray-level histograms“, *Automatica*, Jg. 11, Nr. 285–296, S. 23–27, 1975.
- [96] C. Noiriél, „Resolving time-dependent evolution of pore-scale structure, permeability and reactivity using X-ray microtomography“, *Reviews in Mineralogy and Geochemistry*, Jg. 80, Nr. 1, S. 247–285, 2015.
- [97] P. Altschuh, „Skalenübergreifende Analyse Makroporöser Membranen im Kontext Digitaler Zwillinge“, 2020.
- [98] S. D. Yanowitz und A. M. Bruckstein, „A new method for image segmentation“, *Computer Vision, Graphics, and Image Processing*, Jg. 46, Nr. 1, S. 82–95, 1989.



- [99] A. P. Sheppard, R. M. Sok und H. Averdunk, „Techniques for image enhancement and segmentation of tomographic images of porous materials“, *Physica A: Statistical mechanics and its applications*, Jg. 339, Nr. 1-2, S. 145–151, 2004.
- [100] K. V. Mardia und T. Hainsworth, „A spatial thresholding method for image segmentation“, *IEEE transactions on pattern analysis and machine intelligence*, Jg. 10, Nr. 6, S. 919–927, 1988.
- [101] W. Oh und B. Lindquist, „Image thresholding by indicator kriging“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 21, Nr. 7, S. 590–602, 1999.
- [102] T. Pavlidis und Y.-T. Liow, „Integrating region growing and edge detection“, *IEEE transactions on pattern analysis and machine intelligence*, Jg. 12, Nr. 3, S. 225–233, 1990.
- [103] G. Sivewright und P. Elliott, „Interactive region and volume growing for segmenting volumes in MR and CT images“, *Medical informatics*, Jg. 19, Nr. 1, S. 71–80, 1994.
- [104] J. A. Sethian u. a., *Level set methods and fast marching methods*. Cambridge Cambridge UP, 1999, Bd. 98.
- [105] S. Osher und N. Paragios, *Geometric level set methods in imaging, vision, and graphics*. Springer Science & Business Media, 2003.
- [106] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz und D. Terzopoulos, „Image segmentation using deep learning: A survey“, *IEEE transactions on pattern analysis and machine intelligence*, 2021.
- [107] H. Seo, M. Badiei Khuzani, V. Vasudevan, C. Huang, H. Ren, R. Xiao, X. Jia und L. Xing, „Machine learning techniques for biomedical image segmentation: an overview of technical aspects and introduction to state-of-art applications“, *Medical physics*, Jg. 47, Nr. 5, e148–e167, 2020.

- 
- [108] O. Ronneberger, P. Fischer und T. Brox, „U-net: Convolutional networks for biomedical image segmentation“, in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, S. 234–241.
- [109] A. Marques, G. Racolte, E. M. De Souza, H. V. Domingos, R. K. Horota, J. G. Motta, D. C. Zanotta, C. L. Cazarin, L. Gonzaga und M. R. Veronez, „Deep learning application for fracture segmentation over outcrop images from UAV-based digital photogrammetry“, in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, IEEE, 2021, S. 4692–4695.
- [110] E. B. Alexakis und C. Armenakis, „Evaluation of UNet and UNet++ architectures in high resolution image change detection applications“, *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Jg. 43, S. 1507–1514, 2020.
- [111] J. McGlinchy, B. Johnson, B. Muller, M. Joseph und J. Diaz, „Application of UNet fully convolutional neural network to impervious surface segmentation in urban environment from high resolution satellite imagery“, in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2019, S. 3915–3918.
- [112] A. Koepe, F. Bamer und B. Markert, „An intelligent meta-element for linear elastic continua“, *PAMM*, Jg. 18, Nr. 1, e201800283, 2018. DOI: 10.1002/pamm.201800283.
- [113] A. Koepe, F. Bamer und B. Markert, „An intelligent nonlinear meta element for elastoplastic continua: Deep learning using a new Time-distributed Residual U-Net architecture“, *Computer Methods in Applied Mechanics and Engineering*, Jg. 366, S. 113 088, 1. Juli 2020. DOI: 10.1016/j.cma.2020.113088.
- [114] A. Koepe, F. Bamer und B. Markert, „An artificial intelligence approach to model nonlinear continua by intelligent meta-elements“, *PAMM*, Jg. 20, Nr. 1, e202000300, 2021. DOI: 10.1002/pamm.202000300.

- [115] A. Koeppe, D. Hesser, M. Mundt, F. Bamer, M. Selzer und B. Markert, „Mechanics 4.0. Artificial Intelligence for the analysis of mechanical systems“, in *Handbook Industry 4.0: Law, Technology, Society*, W. Frenz, Hrsg., Berlin, Heidelberg: Springer, 2022, S. 455–470. DOI: 10.1007/978-3-662-64448-5\_23.
- [116] A. B. Farimani, J. Gomes und V. S. Pande, „Deep Learning the Physics of Transport Phenomena“, *ArXiv170902432 Phys.*, 7. Sep. 2017.
- [117] J. Stuckner, B. Harder und T. M. Smith, „Microstructure segmentation with deep learning encoders pre-trained on a large microscopy dataset“, *npj Computational Materials*, Jg. 8, Nr. 1, S. 200, 2022.
- [118] S. Bellens, P. Vandewalle und W. Dewulf, „Deep learning based porosity segmentation in X-ray CT measurements of polymer additive manufacturing parts“, *Procedia CIRP*, Jg. 96, S. 336–341, 2021.
- [119] X. Xia und B. Kulis, „W-net: A deep model for fully unsupervised image segmentation“, *arXiv preprint arXiv:1711.08506*, 2017.
- [120] T. Wang und Q. Dai, „SURVS: A Swin-Unet and game theory-based unsupervised segmentation method for retinal vessel“, *Computers in Biology and Medicine*, Jg. 166, S. 107 542, 2023.
- [121] V. Thambawita, P. Salehi, S. A. Sheshkal, S. A. Hicks, H. L. Hammer, S. Parasa, T. d. Lange, P. Halvorsen und M. A. Riegler, „SinGAN-Seg: Synthetic training data generation for medical image segmentation“, *PloS one*, Jg. 17, Nr. 5, e0267976, 2022.
- [122] L. Lindner, D. Narnhofer, M. Weber, C. Gsaxner, M. Kolodziej und J. Egger, „Using synthetic training data for deep learning-based GBM segmentation“, in *2019 41st Annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, IEEE, 2019, S. 6724–6729.
- [123] C. Schlumberger und M. Thommes, „Characterization of hierarchically ordered porous materials by physisorption and mercury porosimetry—a tutorial review“, *Advanced Materials Interfaces*, Jg. 8, Nr. 4, S. 2 002 181, 2021.

- [124] C. Agarwal, S. Das und A. Pandey, „Study on pore size distributions of microporous polymer membranes having different physical architecture using capillary flow porometry“, *Materials Today Chemistry*, Jg. 23, S. 100 652, 2022.
- [125] K. S. Sing, „Adsorption methods for the characterization of porous materials“, *Advances in colloid and interface science*, Jg. 76, S. 3–11, 1998.
- [126] European Committee for Standardization, *EN ISO 1183-1:2024 – Plastics - Methods for determining the density of non-cellular plastics - Part 1: Immersion method, liquid pycnometer method and titration method*, Standard, European Standard (EN ISO 1183-1:2024), 2024.
- [127] C. L. Reedy und C. L. Reedy, „High-resolution micro-CT with 3D image analysis for porosity characterization of historic bricks“, *Heritage Science*, Jg. 10, Nr. 1, S. 83, 2022.
- [128] O. Nilsson und A. Söderström, *Euclidian distance transform algorithms: A comparative study*, 2007.
- [129] I. Ben-Noah, J. J. Hidalgo und M. Dentz, „Pore-space partitioning in geological porous media using the curvature of the distance map“, *Transport in Porous Media*, Jg. 152, Nr. 1, Dez. 2024. DOI: 10.1007/s11242-024-02142-4.
- [130] J. Alsayednoor, „Modelling and characterisation of porous materials“, Diss., University of Glasgow, 2013.
- [131] Y. Liang, P. Hu, S. Wang, S. Song und S. Jiang, „Medial axis extraction algorithm specializing in porous media“, *Powder technology*, Jg. 343, S. 512–520, 2019.
- [132] J. T. Gostick, „Versatile and efficient pore network extraction method using marker-based watershed segmentation“, *Physical Review E*, Jg. 96, Nr. 2, S. 023 307, 2017.
- [133] W. Kunz, „Datengetriebene Analyse der strukturabhängigen Flüssigkeitsausbreitung in porösen Membranen“, 2024.

- [134] M. Greenacre, P. J. Groenen, T. Hastie, A. I. d'Enza, A. Markos und E. Tuzhilina, „Principal component analysis“, *Nature Reviews Methods Primers*, Jg. 2, Nr. 1, S. 100, 2022.
- [135] S. R. Niezgodna, Y. C. Yabansu und S. R. Kalidindi, „Understanding and visualizing microstructure and microstructure variance as a stochastic process“, *Acta Materialia*, Jg. 59, Nr. 16, S. 6387–6400, 2011.
- [136] P. Altschuh, Y. C. Yabansu, J. Hötzer, M. Selzer, B. Nestler und S. R. Kalidindi, „Data science approaches for microstructure quantification and feature identification in porous membranes“, *Journal of Membrane Science*, Jg. 540, S. 88–97, 2017. DOI: <https://doi.org/10.1016/j.memsci.2017.06.020>.
- [137] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid u. a., „Fiji: an open-source platform for biological-image analysis“, *Nature methods*, Jg. 9, Nr. 7, S. 676–682, 2012.
- [138] A. Ley, P. Altschuh, V. Thom, M. Selzer, B. Nestler und P. Vana, „Characterization of a macro porous polymer membrane at micron-scale by Confocal-Laser-Scanning Microscopy and 3D image analysis“, *Journal of membrane science*, Jg. 564, S. 543–551, 2018.
- [139] K. Palágyi und A. Kuba, „A 3D 6-subiteration thinning algorithm for extracting medial lines“, *Pattern Recognition Letters*, Jg. 19, Nr. 7, S. 613–627, 1998.
- [140] R. Zhang, P. Isola, A. A. Efros, E. Shechtman und O. Wang, „The unreasonable effectiveness of deep features as a perceptual metric“, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, S. 586–595.
- [141] Z. Wang, A. C. Bovik, H. R. Sheikh und E. P. Simoncelli, „Image quality assessment: from error visibility to structural similarity“, *IEEE transactions on image processing*, Jg. 13, Nr. 4, S. 600–612, 2004.

- [142] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola und V. Lukin, „On between-coefficient contrast masking of DCT basis functions“, in *Proceedings of the third international workshop on video processing and quality metrics*, Scottsdale USA, Bd. 4, 2007.
- [143] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler und S. Hochreiter, „Gans trained by a two time-scale update rule converge to a local nash equilibrium“, *Advances in neural information processing systems*, Jg. 30, 2017.
- [144] E. Brophy, Z. Wang, Q. She und T. Ward, „Generative adversarial networks in time series: A systematic literature review“, *ACM Computing Surveys*, Jg. 55, Nr. 10, S. 1–31, 2023.
- [145] L. Van der Maaten und G. Hinton, „Visualizing data using t-SNE.“, *Journal of machine learning research*, Jg. 9, Nr. 11, 2008.
- [146] S. Brückmann, „Strukturen zur Energieumwandlung in Polyurethanschäum-Aluminium-Sandwich Bauweise unter Berücksichtigung spezifischer Versagensmodi“, Deutsches Zentrum für Luft- und Raumfahrt, Techn. Ber., Feb. 2017.
- [147] puren gmbh, *Produktdatenblatt puren-PIR NE 200 EU/CH*, <https://www.puren.com/de/industrielloesungen/puren-pir-ne-80-300/>, Stand der Technik 08/2018, 2018.
- [148] J. O. Akindoyo, M. H. Beg, S. Ghazali, M. R. Islam, N. Jeyaratnam und A. Yuvaraj, „Polyurethane types, synthesis and applications—a review“, *Rsc Advances*, Jg. 6, Nr. 115, S. 114 453–114 482, 2016.
- [149] L. Griem, A. Koeppel, A. Greß, T. Feser und B. Nestler, „Synthetic training data for CT image segmentation of microstructures“, *Acta Materialia*, S. 121 220, 2025. DOI: <https://doi.org/10.1016/j.actamat.2025.121220>.
- [150] J. Hötzer, A. Reiter, H. Hierl, P. Steinmetz, M. Selzer und B. Nestler, „The parallel multi-physics phase-field framework Pace3D“, *Journal of computational science*, Jg. 26, S. 1–12, 2018.

- [151] L. C. Griem, *Fiji Macro for Binary to Grayscale Image Translation*, Sep. 2024. DOI: 10.5281/zenodo.13711248.
- [152] J.-Y. Zhu, T. Park, P. Isola und A. A. Efros, „Unpaired image-to-image translation using cycle-consistent adversarial networks“, in *Proceedings of the IEEE international conference on computer vision*, 2017, S. 2223–2232.
- [153] L. C. Griem, *Reconstruction and Characterisation*, Sep. 2025. DOI: 10.5281/zenodo.17019210.
- [154] C. Stringer, T. Wang, M. Michaelos und M. Pachitariu, „Cellpose: a generalist algorithm for cellular segmentation“, *Nature methods*, Jg. 18, Nr. 1, S. 100–106, 2021.
- [155] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo u. a., „Segment anything“, in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, S. 4015–4026.
- [156] F. Aurenhammer, „Voronoi diagrams—a survey of a fundamental geometric data structure“, *ACM Computing Surveys (CSUR)*, Jg. 23, Nr. 3, S. 345–405, 1991.
- [157] M. G. Alinchenko, A. V. Anikeenko, N. N. Medvedev, V. P. Voloshin, M. Mezei und P. Jedlovsky, „Morphology of Voids in Molecular Systems. A Voronoi- Delaunay Analysis of a Simulated DMPC Membrane“, *The Journal of Physical Chemistry B*, Jg. 108, Nr. 49, S. 19 056–19 067, 2004.
- [158] Y. Du, H. Liang, D. Xie, N. Mao, J. Zhao, Z. Tian, C. Wang und L. Shen, „Design and statistical analysis of irregular porous scaffolds for orthopedic reconstruction based on voronoi tessellation and fabricated via selective laser melting (SLM)“, *Materials Chemistry and Physics*, Jg. 239, S. 121 968, 2020.
- [159] J. C. Tucker und A. D. Spear, „A Tool to Generate Grain-Resolved Open-Cell Metal Foam Models“, *Integrating Materials and Manufacturing Innovation*, S. 1–10, 2019.

- 
- [160] C. H. Park, S. Y. Lee, D. S. Hwang, D. W. Shin, D. H. Cho, K. H. Lee, T.-W. Kim, T.-W. Kim, M. Lee, D.-S. Kim u. a., „Nanocrack-regulated self-humidifying membranes“, *Nature*, Jg. 532, Nr. 7600, S. 480, 2016.
- [161] J. Alsayednoor und P. Harrison, „Evaluating the performance of microstructure generation algorithms for 2-d foam-like representative volume elements“, *Mechanics of Materials*, Jg. 98, S. 44–58, 2016.
- [162] S. Lloyd, „Least squares quantization in PCM“, *IEEE transactions on information theory*, Jg. 28, Nr. 2, S. 129–137, 1982.
- [163] S. Falco, J. Jiang, F. De Cola und N. Petrinic, „Generation of 3D polycrystalline microstructures with a conditioned Laguerre-Voronoi tessellation technique“, *Computational Materials Science*, Jg. 136, S. 20–28, 2017.
- [164] M. Reder, J. Holland-Cunz, P. Lorson, A. August und B. Nestler, „Simulative determination of effective mechanical properties for digitally generated foam geometries“, *Advanced Engineering Materials*, Jg. 25, Nr. 19, S. 2 300 340, 2023.
- [165] A. August, J. Ettrich, M. Rölle, S. Schmid, M. Berghoff, M. Selzer und B. Nestler, „Prediction of heat conduction in open-cell foams via the diffuse interface representation of the phase-field method“, *International Journal of Heat and Mass Transfer*, Jg. 84, S. 800–808, 2015.
- [166] M. Späth, C. Herrmann, N. Prajapati, D. Schneider, F. Schwab, M. Selzer und B. Nestler, „Multiphase-field modelling of crack propagation in geological materials and porous media with Drucker-Prager plasticity“, *Computational Geosciences*, Jg. 25, S. 325–343, 2021.
- [167] E. Coumans und Y. Bai, *PyBullet, a Python module for physics simulation for games, robotics and machine learning*, <http://pybullet.org>, 2016–2019.
- [168] M. Späth, „Phasenfeldmodellierung von Bruchbildungs-, Kristallisations- und Auflösungsvorgängen in hydrothermalen Umgebungen“, 38.04.04; LK 01, Diss., Karlsruher Institut für Technologie (KIT), 2022, 249 S. DOI: 10.5445/KSP/1000151417.



- [169] N. Prajapati, A. Abad Gonzalez, M. Selzer, B. Nestler, B. Busch und C. Hilgers, „Quartz cementation in polycrystalline sandstone: Insights from phase-field simulations“, *Journal of Geophysical Research: Solid Earth*, Jg. 125, Nr. 2, e2019JB019137, 2020.
- [170] F. Wang, P. Altschuh, L. Ratke, H. Zhang, M. Selzer und B. Nestler, „Progress report on phase separation in polymer solutions“, *Advanced Materials*, Jg. 31, Nr. 26, S. 1 806 733, 2019.
- [171] J. Holland-Cunz, A. Reiter, J. Hoetzer, A. August, M. Selzer und B. Nestler, „Numerical investigation of bubble dynamics in ageing foams using a phase-field model“, *Computational Materials Science*, Jg. 248, S. 113 557, 2025.
- [172] S. Vakili, I. Steinbach und F. Varnik, „Multi-phase-field simulation of microstructure evolution in metallic foams“, *Scientific reports*, Jg. 10, Nr. 1, S. 19 987, 2020.
- [173] M. Seiz, „Modelling and investigation of microstructural evolution during solidification and sintering with the phase-field method“, Englisch, Diss., Karlsruher Institut für Technologie (KIT), 2023, 256 S. DOI: 10.5445/IR/1000165361.
- [174] Y. Zhao, P. Altschuh, J. Santoki, L. Griem, G. Tosato, M. Selzer, A. Koeppe und B. Nestler, „Characterization of porous membranes using artificial neural networks“, *Acta Materialia*, Jg. 253, S. 118 922, 2023.
- [175] L. Mosser, O. Dubrule und M. J. Blunt, „Stochastic reconstruction of an oolitic limestone by generative adversarial networks“, *Transport in Porous Media*, Jg. 125, Nr. 1, S. 81–103, 2018.
- [176] P. C. Nguyen, N. N. Vlassis, B. Bahmani, W. Sun, H. Udaykumar und S. S. Baek, „Synthesizing controlled microstructures of porous media using generative adversarial networks and reinforcement learning“, *Scientific reports*, Jg. 12, Nr. 1, S. 9034, 2022.
- [177] S. Kench und S. J. Cooper, „Generating 3D structures from a 2D slice with GAN-based dimensionality expansion“, *arXiv preprint arXiv:2102.07708*, 2021.

- 
- [178] S. Kench, I. Squires, A. Dahari und S. J. Cooper, „MicroLib: A library of 3D microstructures generated from 2D micrographs using SliceGAN“, *Scientific Data*, Jg. 9, Nr. 1, S. 645, 2022.
- [179] X. Lyu und X. Ren, „Microstructure reconstruction of 2D/3D random materials via diffusion-based deep generative models“, *Scientific Reports*, Jg. 14, Nr. 1, S. 5041, 2024.
- [180] M. Seiz, H. Hierl, B. Nestler und W. Rheinheimer, „Revealing process and material parameter effects on densification via phase-field studies“, *Scientific Reports*, Jg. 14, Nr. 1, S. 5350, 2024.
- [181] S. Daubner, M. Weichel, P. W. Hoffrogge, D. Schneider und B. Nestler, „Modeling Anisotropic Transport in Polycrystalline Battery Materials“, *Batteries*, Jg. 9, Nr. 6, S. 310, 2023.
- [182] S. Daubner, M. Dillenz, L. F. Pfeiffer, C. Gauckler, M. Rosin, N. Burgard, J. Martin, P. Axmann, M. Sotoudeh, A. Groß u. a., „Combined study of phase transitions in the P2-type Na X Ni<sub>1/3</sub>Mn<sub>2/3</sub>O<sub>2</sub> cathode material: experimental, ab-initio and multiphase-field results“, *npj Computational Materials*, Jg. 10, Nr. 1, S. 75, 2024.
- [183] S.-Y. Park, J.-S. Lee, B.-H. Choi, I. H. Ahn und S. K. Moon, „Modeling and observation of compressive behaviors of closed cellular structures using central Voronoi tessellation concepts“, *International Journal of Precision Engineering and Manufacturing*, Jg. 16, S. 2459–2465, 2015.
- [184] Z. Sordo, E. Chagnon und D. Ushizima, *A Review on Generative AI For Text-To-Image and Image-To-Image Generation and Implications To Scientific Images*, 2025.
- [185] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser und I. Polosukhin, „Attention is all you need“, *Advances in neural information processing systems*, Jg. 30, 2017.
- [186] S. H. Hasanpour, M. Rouhani, M. Fayyaz und M. Sabokrou, *Lets keep it simple, Using simple architectures to outperform deeper and more complex architectures*, 2023.

- [187] L. C. Griem, *ML-based Generation of digital twins*, Sep. 2025. DOI: 10.5281/zenodo.17019360.
- [188] P. W. Hoffrogge, „Understanding Degradation Phenomena in Solid-Oxide Fuel-Cell Anodes by Phase-Field Modeling and Analytics“, Englisch, Diss., Karlsruher Institut für Technologie (KIT), 2024, 258 S. DOI: 10.5445/KSP/1000170482.
- [189] S. Daubner, „Modeling battery intercalation materials with the multiphase-field method“, Englisch, 38.02.01; LK 01, Diss., Karlsruher Institut für Technologie (KIT), 2023, 217 S. DOI: 10.5445/IR/1000164858.
- [190] N. Prajapati, „Computational modeling of syntaxial overgrowth cementation and fracture propagation in sedimentary rocks“, Englisch, Diss., Karlsruher Institut für Technologie (KIT), 2020, 154 S. DOI: 10.5445/IR/1000122917.
- [191] D. Schneider, „Phasenfeldmodellierung mechanisch getriebener Grenzflächenbewegungen in mehrphasigen Systemen“, Diss., Karlsruher Institut für Technologie (KIT), 2016, 177 S. DOI: 10.5445/IR/1000066948.
- [192] C. J. Herrmann, „Numerische Untersuchung von mechanischen und thermischen Eigenschaften von Gusseisen mit Lamellengraphit“, 38.04.04; LK 01, Diss., Karlsruher Institut für Technologie (KIT), 2021, 162 S. DOI: 10.5445/IR/1000139258.
- [193] Y. Wu, „Droplet wetting on chemically and mechanically structured surfaces“, Englisch, Diss., Karlsruher Institut für Technologie (KIT), 2021, 184 S. DOI: 10.5445/IR/1000139649.
- [194] F. Aurbach, F. Wang und B. Nestler, „Wetting phenomena of droplets and gas bubbles: Contact angle hysteresis based on varying liquid–solid and solid–gas interfacial tensions“, Englisch, *The Journal of Chemical Physics*, Jg. 161, Nr. 16, Art.–Nr. : 164708, 2024, 43.31.01; LK 01. DOI: 10.1063/5.0233997.

- [195] F. Wang, H. Zhang und B. Nestler, „Wetting Phenomena: Line Tension and Gravitational Effect“, Englisch, *Physical Review Letters*, Jg. 133, Nr. 24, S. 246 201, 2024, 43.31.01; LK 01. DOI: 10.1103/PhysRevLett.133.246201.
- [196] O. Abramova, D. Nugmanov, D. Schneider, A. Prahs, T. Mittnacht, J. Ivanisenko, B. Baretzky und B. Nestler, „Phase-field-simulation of the static recrystallization and grain growth kinetics of ultrafine-grained aluminum processed by high pressure torsion extrusion“, *Computational Materials Science*, Jg. 248, S. 113 553, 2025.
- [197] T. Kannenberg, L. Schöller, A. Prahs, D. Schneider und B. Nestler, „Investigation of microstructure evolution accounting for crystal plasticity in the multiphase-field method“, *PAMM*, Jg. 23, Nr. 3, e202300138, 2023.
- [198] J. Hötzer, M. Seiz, M. Kellner, W. Rheinheimer und B. Nestler, „Phase-field simulation of solid state sintering“, *Acta Materialia*, Jg. 164, S. 184–195, 2019.
- [199] D. Schneider, O. Tschukin, A. Choudhury, M. Selzer, T. Böhlke und B. Nestler, „Phase-field elasticity model based on mechanical jump conditions“, *Computational Mechanics*, Jg. 55, Nr. 5, S. 887–901, 2015.
- [200] A. G. Khachaturyan, *Theory of structural transformations in solids*. Courier Corporation, 2013.
- [201] L. C. Griem, *Workflow Simulation*, Sep. 2025. DOI: 10.5281/zenodo.17019442.
- [202] R. Bostanabad, Y. Zhang, X. Li, T. Kearney, L. C. Brinson, D. W. Apley, W. K. Liu und W. Chen, „Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques“, *Progress in Materials Science*, Jg. 95, S. 1–41, 2018.
- [203] Y. Liu, O. C. Esan, Z. Pan und L. An, „Machine learning for advanced energy materials“, *Energy and AI*, Jg. 3, S. 100 049, 2021.

- [204] J. Lin, Z. Liu, Y. Guo, S. Wang, Z. Tao, X. Xue, R. Li, S. Feng, L. Wang, J. Liu u. a., „Machine learning accelerates the investigation of targeted MOFs: performance prediction, rational design and intelligent synthesis“, *Nano Today*, Jg. 49, S. 101 802, 2023.
- [205] H. Danesh, M. Annamaraju, T. Brepols, S. Reese und S. R. Kalidindi, „Reduced-order structure-property linkages for stochastic metamaterials“, *arXiv preprint arXiv:2505.01283*, 2025.
- [206] W. Kunz, P. Altschuh, M. Bremerich, M. Selzer und B. Nestler, „Establishing structure–property linkages for wicking time predictions in porous polymeric membranes using a data-driven approach“, *Materials Today Communications*, Jg. 35, S. 106 004, 2023.
- [207] R. N. Saunders, K. Teferra, A. Elwany, J. G. Michopoulos und D. Lagoudas, „Metal AM process-structure-property relational linkages using Gaussian process surrogates“, *Additive Manufacturing*, Jg. 62, S. 103 398, 2023.
- [208] A. Gupta, A. Cecen, S. Goyal, A. K. Singh und S. R. Kalidindi, „Structure–property linkages using a data science approach: application to a non-metallic inclusion/steel composite system“, *Acta Materialia*, Jg. 91, S. 239–254, 2015.
- [209] A. Cecen, H. Dai, Y. C. Yabansu, S. R. Kalidindi und L. Song, „Material structure-property linkages using three-dimensional convolutional neural networks“, *Acta Materialia*, Jg. 146, S. 76–84, 2018.
- [210] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen und K.-R. Müller, *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer Nature, 2019, Bd. 11700.
- [211] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller und W. Samek, „On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation“, *PloS one*, Jg. 10, Nr. 7, e0130140, 2015.

- [212] G. Montavon, W. Samek und K.-R. Müller, „Methods for interpreting and understanding deep neural networks“, *Digital signal processing*, Jg. 73, S. 1–15, 2018.
- [213] A. Y.-T. Wang, R. J. Murdock, S. K. Kauwe, A. O. Oliynyk, A. Gurlo, J. Brgoch, K. A. Persson und T. D. Sparks, „Machine learning for materials scientists: an introductory guide toward best practices“, *Chemistry of Materials*, Jg. 32, Nr. 12, S. 4954–4965, 2020.
- [214] A. Koepe, „Deep learning in the finite element method“, Diss., Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, 2021, 2021.
- [215] L. C. Griem, *Workflow Database Analysis*, Sep. 2025. DOI: 10.5281/zenodo.17019477.
- [216] S. Schlabach, J. Wild, O. Petkau, M. Selzer und D. V. Szabó, „Using ELN Functionality of Kadi4Mat (KadiWeb) in a Materials Science Case Study of a User Facility“, *Data Science Journal*, Jg. 23, Nr. 1, 2024.
- [217] M. M. Flores-Leonar, L. M. Mejía-Mendoza, A. Aguilar-Granda, B. Sanchez-Lengeling, H. Tribukait, C. Amador-Bedolla und A. Aspuru-Guzik, „Materials Acceleration Platforms: On the way to autonomous experimentation“, *Current Opinion in Green and Sustainable Chemistry*, Jg. 25, S. 100 370, 2020.
- [218] S. P. Stier, C. Kreisbeck, H. Ihssen, M. A. Popp, J. Hauch, K. Malek, M. Reynaud, T. P. M. Goumans, J. Carlsson, I. Todorov u. a., „Materials acceleration platforms (MAPs): accelerating materials research and development to meet urgent societal challenges“, *Advanced Materials*, Jg. 36, Nr. 45, S. 2 407 791, 2024.



# Abbildungsverzeichnis

1.1	Allgemeiner Materialentwicklungskreislauf. Ausgehend von digitalen Zwillingen realer Strukturen werden Varianten mit gezielt veränderten Eigenschaften generiert. Ihre simulative Untersuchung liefert die zugehörigen makroskopischen Eigenschaften, aus deren Analyse sich Struktur–Eigenschaft–Beziehungen ableiten lassen. . . . .	2
2.1	Schematische Darstellung eines Multilayer Perceptron mit Eingabe-, versteckten und Ausgabeschichten. Jedes Perzeptron bildet eine gewichtete Summe und wendet eine Aktivierungsfunktion an, bevor die Ausgabe an die nächste Schicht weitergegeben wird. . . . .	15
2.2	Vergleich der in dieser Arbeit verwendeten Aktivierungsfunktionen: (a) Sigmoid, (b) tanh, (c) ReLU, (d) Leaky ReLU, (e) ELU und (f) SiLU. Die unterschiedlichen Verläufe verdeutlichen ihre jeweiligen Eigenschaften, etwa Sättigungseffekte bei Sigmoid und tanh, die Robustheit von ReLU-Varianten sowie den glatten, differenzierbaren Verlauf der SiLU. . . . .	16
2.3	Einfluss der Lernrate $\alpha$ auf das Konvergenzverhalten eines neuronalen Netzes. Dargestellt ist die Optimierung entlang einer vereinfachten, konvexen Verlustfunktion. Kleine Lernraten (orange) führen zu langsamer Annäherung, große Lernraten (rot) zu Überschwingen oder Divergenz. Eine optimal gewählte Lernrate (grün) ermöglicht schnelle und stabile Konvergenz zum Minimum. . . . .	22



2.4	Einfluss der Batchgröße auf den Optimierungspfad: Training mit allen Daten (blau) kann in lokalen Minima stecken bleiben, einzelne Datenpunkte (orange) oszillieren stark, während Minibatches (grün) einen stabileren Weg finden. . . . .	26
2.5	Typische Trainingsszenarien: (a) Underfitting: Modell zu simpel, hohe Verluste. (b) Guter Fit: Niedrige, ähnliche Verluste auf Trainings- und Validierungsdaten. (c) Overfitting: Sehr niedriger Trainingsverlust, aber hoher Validierungsverlust. . .	27
2.6	Schematische Darstellung verschiedener Normalisierungsmethoden. Die farbigen Markierungen kennzeichnen jeweils die Datenbereiche, über die die Normalisierung durchgeführt wird. Als Eingabedaten dienen Bilddaten der Form (N, C, H, W), wobei N die Batchgröße, C die Kanäle (Features) sowie H und W die Höhe und Breite des Bildes repräsentieren . . . . .	30
2.7	Schematische Darstellung der Faltung einer Eingabe $f$ mit einem Filter $k$ zur Erstellung einer Feature Map. . . . .	31
2.8	Vorwärtsdiffusionsprozess. Ein klares Bild wird iterativ in reines Rauschen umgewandelt. . . . .	35
2.9	Typischer Forschungsdatenlebenszyklus unterteilt in sechs Phasen.	40
3.1	Kadi4Mat Ökosystem. Kadi4Mat stellt verschiedene Module – KadiWeb, KadiStudio, KadiAPY, KadiAI, KadiFS – zur Verfügung, die unterschiedliche Funktionen implementieren. Durch definierte Schnittstellen können weitere Komponenten, angedeutet durch die grauen Formen, hinzugefügt werden. . . . .	51
3.2	Schematische Visualisierung des IPO-Prinzips. Definierte Eingaben parametrisieren einen Prozess, der anschließend ausgeführt wird. Die erzeugten Ergebnisse sind über Ausgaben verfügbar. Adaptiert von [68]. . . . .	53

3.3	Abstraktion eines Forschungsprozesses auf unterschiedlichen Abstraktionsebenen. Die mit Zahnrädern markierten Felder stellen Arbeitsschritte dar, während die weißen Boxen deren Parametrisierung zeigt. APR bezieht sich auf die Beschreibung dieser Aufgaben als Datenakquise, -verarbeitung und -weiterleitung (englisch acquisition, processing, routing). Durch die iterative Strukturierung eines Forschungsprozesses gemäß des IPO-Prinzips wird dieser letztlich über mehrere generische Werkzeuge definiert. Adaptiert von [68]. . . . .	54
3.4	Reduktion der Prozesskomplexität durch das iterative Anwenden des IPO-Prinzips. Auf jedem Abstraktionslevel wird der Prozess über atomistische Schritte beschrieben. Das Funktionslevel stellt eine allgemein anwendbare Abstraktion zur Modellierung beliebiger Forschungsprozesse dar. Adaptiert von [68]. . . . .	57
3.5	Übersicht über die verfügbaren Editoren: Oben ist die Desktopversion und unten die webbasierte Version gezeigt. Workflows werden durch das Hinzufügen und Verbinden von Knoten mittels eines Point-and-Click Mechanismus hinzugefügt. .	60
3.6	Verfügbare Knotentypen. Eingebaute Knoten sind grau, Umgebungsknoten grün und Werkzeugknoten blau. . . . .	61
3.7	Übersicht zur Knotenkonfiguration: (a) Parametrisierung eines <i>echo</i> -Knotens durch einen String-Quellknoten. (b) Erzeugter Befehl. . . . .	63
3.8	Workflow-Modellierung in KadiStudio: Mehrere Werkzeuge werden nach dem APR-Modell zu einem Workflow verknüpft. Das in (a) gezeigte Verbinden von stdout- und stdin-Ports leitet die Ausgabe des Werkzeugknotens an den File-Output-Knoten weiter und entspricht dem in (b) dargestellten CLI-Befehl. Adaptiert von [68]. . . . .	63
3.9	Verwendung von <i>UserInput</i> -Knoten: Während der Ausführung wird der Nutzer zur Eingabe aufgefordert, etwa zur Auswahl eines Bildausschnitts [68]. . . . .	64

3.10	Integration manueller Arbeitsschritte. Dem Nutzer werden alle notwendigen Parameter bereitgestellt und anschließend aufgefordert, die erzeugten Ergebnisse einzutragen. Adaptiert von [68].	65
3.11	Dialog zur Registrierung von Kadi4Mat-Instanzen. Registrierte Instanzen können während eines Workflows genutzt werden, um die Funktionalitäten des Repositoriums zu verwenden und Daten auszutauschen. . . . .	67
3.12	Konzept zur Strukturierung der Metadaten. Ausgehend von einem grundlegenden Metadatenschema werden mit jedem Arbeitspaket neue kontextspezifische Metadaten hinzugefügt. . . .	69
3.13	Hierarchische Strukturierung von Daten. Datenpunkte werden den Arbeitspaketen zugeordnet, so dass ihr Ursprung und ihre Abhängigkeiten nachvollziehbar sind. Ein Datenpunkt kann Teil mehrerer Arbeitsschritte sein. . . . .	70
3.14	Konzept zur Darstellung von Datenrelationen . . . . .	71
3.15	Konzeptioneller Aufbau von KadiWeb. KadiWeb stellt sowohl ein Repositorium zur strukturierten Datenablage, als auch ELN-typische Funktionalitäten zur Verfügung. Adaptiert von [80].	72
3.16	Ressourcen KadiWebs. Daten werden in Records abgelegt und mit Metadaten versehen. Mehrere Records können in eine Collection organisiert werden. Templates helfen Metadaten zu vereinheitlichen. Über Benutzer und Gruppen können gezielt Zugangsrechte vergeben werden. Adaptiert von [80]. . . . .	73
3.17	Angepasster Materialentwicklungsprozess. Für jeden Schritt im Entwicklungsprozess müssen generische Methoden entwickelt und in Kadi4Mat dokumentiert und automatisiert werden.	77
4.1	Funktionsprinzip des verwendeten Computertomographen. Eine stationäre Röntgenquelle sendet kegelförmige Röntgenstrahlen aus, die die rotierende Probe durchdringen und von einem dahinterliegenden Detektor erfasst werden. . . . .	86
4.2	U-Net Architektur. Der Encoder reduziert die Bildgröße und erweitert den Merkmalsraum. Der Decoder rekonstruiert daraus anschließend ein segmentiertes Bild. Skip connections verbinden den Encoder mit dem Decoder. . . . .	90

4.3	Exemplarische rekonstruierte Schaumstruktur mit den Kantenlängen $N_x$ , $N_y$ und $N_z$ . . . . .	91
4.4	Ausschnitt der rekonstruierten digitalen Schaumzwillinge als Voxelgitter, wobei der Porenraum als $P_p(x) = 0$ und der Strukturraum als $P_p(x) = 1$ abgebildet ist. . . . .	92
4.5	Euklidisches Distanzfeld und mediale Achse im Skelettierungs-Algorithmus. (a) zeigt eine exemplarische Schaumstruktur und (b) das EDF sowie die mediale Achse. . . . .	93
4.6	Ermittelte Wanddickenverteilung. In (a) ist die lokale Wanddicke entlang der medialen Achse als Kugelradien visualisiert; die Farbskala kodiert die Dicke. (b) zeigt die resultierende Wanddickenverteilung. . . . .	94
4.7	Unterschiede zwischen den ermittelten Poren (a) und den realen Poren (b). Jede Pore wird über mehrere Punkte beschrieben, was zu Ungenauigkeiten führt. . . . .	95
4.8	Zweipunktkorrelation (ZPK) einer exemplarischen Schaumstruktur: (a) binarisierte 3D-Mikrostruktur, (b) zugehörige ZPK, (c) ZPK-Verlauf entlang der Achsenrichtungen $r_x$ , $r_y$ und $r_z$ . . . . .	98
4.9	Entnahmepositionen der Schaumproben entlang der Aufschäumrichtung. Unterschiede in den Mikrostrukturen, die durch den Herstellungsprozess entstehen, werden durch die ausgewählten Proben abgedeckt. . . . .	101
4.10	CT-Aufnahme von <i>Probe B</i> . Die Kanten des Schaums und somit die Schaumkoordinaten (blau) fallen nicht mit den Bildkoordinaten (rot) zusammen. . . . .	104
4.11	Der erste Schritt des Datengenerierungsprozesses. CT-Daten werden mit dem Otsu-Schwellenwertverfahren binarisiert, rekonstruiert und geometrisch charakterisiert (4). Aus den geometrischen Eigenschaften werden mit geometriebasierten Methoden künstliche Schaumstrukturen synthetisiert (5), in gestapelte Bilder geschnitten (6) und verarbeitet, um CT-Daten mit bekannter Ground-Truth zu ähneln (7). Adaptiert von [149]. . . . .	106

4.12	Künstliche Trainingsdaten, generiert im ersten Schritt des zweistufigen Generierungsprozess. Bild a) zeigt ein Grauwertbild, das vom binären Bild b) abgeleitet wurde. . . . .	107
4.13	Der zweite Schritt des Datengenerierungsprozesses. CT-Scans werden unter Verwendung des im vorherigen Schritt unseres Verfahrens trainierten U-Net binarisiert. Anschließend wird ein GAN auf den binarisierten CT-Bildern trainiert, um realistische binäre Schaumstrukturen zu erzeugen. Die erhaltenen binären Bilder werden mit dem Watershed-Algorithmus verarbeitet, um fehlende Strukturmerkmale hinzuzufügen. Mit einem auf realen CT-Daten trainierten CycleGAN werden die binären Bilder in Graustufenbilder mit bekannter Ground-Truth umgewandelt. . . . .	109
4.14	Schematische Darstellung der GAN-Architektur. Der Generator erzeugt synthetische Bilder binärer Schaumstrukturen, die darauf abzielen, den Diskriminator zu täuschen. Der Diskriminator versucht wiederum, zwischen echten und gefälschten Bildern zu unterscheiden. . . . .	110
4.15	Bilder, die vom GAN-Generator erzeugt wurden. (a) Eine Ausgabe des GAN-Generators ohne weitere Nachbearbeitung. (b) Eine Mikrostruktur, die mit dem Watershed-Algorithmus nachbearbeitet wurde, um dünne strukturelle Elemente hinzuzufügen. . . . .	111
4.16	Schematische Darstellung der CycleGAN-Architektur. Zwei GANs werden gleichzeitig trainiert: eines transformiert Binär- in Grauwertbilder, das andere führt die inverse Abbildung durch. Darstellung adaptiert nach Zhu et al. [152]. . . . .	112
4.17	Vergleich zwischen CT-Bild und synthetischem Bild. (a) Ein reales CT-Bild. (b) Ein synthetisches Graustufenbild, das durch Hinzufügen von Rauschen zu einem vom GAN erzeugten Binärbild mit dem CycleGAN generiert wurde. . . . .	113
4.18	Einfluss der Watershed-Nachbearbeitung auf die Bildähnlichkeit (FID). . . . .	115

4.19	t-SNE-Visualisierung der reduzierten Bild-Embeddings. Die GAN-generierten Bilder gruppieren sich eng mit ihren entsprechenden CT-Datensätzen, während die algorithmisch erzeugten Bilder des ersten Generierungsschritts klar getrennte Cluster bilden. Die Ähnlichkeit wurde mittels Kosinus-Distanz berechnet. . . . .	116
4.20	Überlagerung ausgewählter Validierungsbilder mit den Modellvorhersagen. Grün kennzeichnet korrekt segmentierte Bereiche, Rot falsch-positive und Blau falsch-negative. . . . .	119
4.21	Einfluss der Watershed-Nachbearbeitung auf die Segmentierungsgenauigkeit. Für jede Probe verbessern Watershed-verarbeitete Bilder die erreichte Segmentierungsgenauigkeit. . . . .	120
4.22	Einfluss der Watershed-Nachbearbeitung auf die Segmentierung der verfügbaren CT-Bilder. Für jeden Datensatz werden die Ergebnisse der Otsu-Schwellenwertmethode, eines U-Nets, das auf algorithmischen (Schritt 1) Daten trainiert wurde, sowie eines U-Nets, das auf GAN-basierten (Schritt 2) Daten trainiert wurde, dargestellt. . . . .	121
4.23	RVE Analyse der Struktureigenschaften. Für die Porosität a), den Porenradius b) und die Wandstärke c) werden unterschiedliche RVE-Größen identifiziert. . . . .	123
4.24	Workflow zur Generierung künstlicher Trainingsdaten. Die Erstellung der Trainingsdaten benötigt zwei neuronale Netzwerke – Binary GAN und CycleGAN. Das Training dieser Netzwerke ist ebenfalls im Workflow implementiert. Welche Subprozesse des Workflows ausgeführt werden, wählt der Nutzer. . . . .	125
4.25	Workflow zum Training eines Segmentierungsnetzwerks in Form eines U-Nets. Der Workflow benötigt Trainingsdaten bestehend aus Grauwertbildern und binären Ziellabeln. Das trainierte U-Net wird abschließend auf Kadi4Mat hochgeladen. . .	126

4.26	Workflow zur Segmentierung. Neben einem trainierten U-Net Modell benötigt der Workflow die zu segmentierenden Grauwert CT-Bilder. Die segmentierten Bilder zu einer dreidimensionalen Struktur zusammengesetzt. Die segmentierten Bilder und die dreidimensionale Struktur werden auf Kadi4Mat hochgeladen. . . . .	127
4.27	Record der rekonstruierten Schaumprobe. Die Herstellerangaben stellen die Basismetadaten des Materials dar. . .	128
4.28	Workflow zur Strukturanalyse. Jede Strukturgröße wird in einem eigenen Unterprozess ermittelt. Der Nutzer entscheidet welche Werte ermittelt werden sollen. Die ermittelten Werte werden auf Kadi4Mat dokumentiert. . . . .	129
4.29	Dokumentation der Charakterisierung der rekonstruierten CT-Daten. Die Strukturparameter sind in den Metadaten, maschinenlesbar abgelegt. . . . .	130
4.30	Zusammenfassung der Strukturrekonstruktion und -analyse auf Kadi4Mat. Ausgehend von CT-Daten wird ein U-Net auf Grundlage künstlicher Trainingsdaten trainiert. Dieses U-Net dient zur Segmentierung der CT-Daten. . . . .	132
5.1	Voronoizerlegung eines dreidimensionalen Raums. Jeder Voronoipunkt definiert ein Volumen, das ihm am nächsten liegt. Die Flächen $B_{ij}$ bilden die Grenzflächen zwischen benachbarten Zellen. Grafik adaptiert von [97]. . . . .	143
5.2	Zentroidales Voronoigebiet. (a) Zufällig gewählte Voronoipunkte (schwarz) mit den Schwerpunkten der jeweiligen Voronoigeiete (orange). Der Lloyd-Algorithmus verschiebt die Punkte schrittweise in Richtung dieser Schwerpunkte. (b) Zentroidales Voronoidiagramm, bei dem die Voronoipunkte exakt in den Schwerpunkten ihrer Gebiete liegen.	144
5.3	Konstruktion eines Schaums aus der Voronoizerlegung. Entlang der Grenzflächen wird eine Strukturphase mit definierter Dicke aufgebaut. Bild adaptiert von [97]. . . . .	145
5.4	Vergleich eines realen Schaums (a) mit einem geometriebasiert generierten Schaum (b) vor der Nachbearbeitung.	146

5.5	Visualisierung der simulativen Nachbearbeitung. Die Poren werden als Flüssigkeitstropfen interpretiert und ihre innere Energie durch Formänderung optimiert. Ausgehen von polyedrischen Strukturen entsteht über die Simulation eine perfekte Kugel. . . . .	146
5.6	Entwicklung des Realismus der Schäume über die Simulationsdauer als Hauptkomponentenanalyse. Die Ausgangsstruktur ist in (a) gezeigt. Die nachfolgenden Grafiken (b), (c) und (d) zeigen den vierten, achten und zehnten Simulationsschritt. Die realen Daten sind in blau, die generierten Daten in orange dargestellt. . . . .	149
5.7	Entwicklung der Struktureigenschaften über die Nachbearbeitung mittels einer Simulation. . . . .	150
5.8	Eine Iteration der Nachbearbeitung der Struktur mittels Bildbearbeitungsmethoden. Die ursprüngliche Struktur wird mit einem Gaußfilter verschmiert und anschließend mit einem globalen Schwellwert $< 0,5$ binarisiert, wodurch diese breiter und runder werden. Ein erneuter Gaußfilter-Schritt mit anschließendem Schwellwert $> 0,5$ verdünnt die Struktur wieder. Das Ergebnis ist eine abgerundete Geometrie mit ähnlicher Porosität wie die Ausgangsstruktur. . . . .	152
5.9	Entwicklung des Realismus der Schäume über die Bildbearbeitungsschritte als Hauptkomponentenanalyse. Die Ausgangsstruktur ist in (a) gezeigt. Die nachfolgenden Grafiken (b), (c) und (d) zeigen den ersten, zweiten und zehnten Bearbeitungsschritt. Die realen Daten sind in blau, die generierten Daten in orange dargestellt. . . . .	153
5.10	Entwicklung der Struktureigenschaften über die Nachbearbeitung mittels Bildbearbeitungsmethoden. . . . .	154



5.11	Schematischer Ablauf eines Entrauschungsschrittes im eingesetzten DD-GAN. Der Generator erhält die verrauschte Struktur $\mathbf{x}_t$ , den Zeitschritt $t$ sowie die Konditionierung $\mathbf{c}$ und erzeugt daraus eine entrauschte Struktur $\mathbf{x}'_0$ . Diese wird mit Rauschen entsprechend dem vorherigen Zeitschritt $t - 1$ versehen, um die nächste Struktur $\mathbf{x}'_{t-1}$ zu erzeugen. Der Diskriminator bewertet sowohl den Realismus als auch die korrekte Umsetzung der Konditionierung. . . . .	157
5.12	Schematische Funktionsweise des Generators. Die Eingabe besteht aus einer verrauschten Struktur $\mathbf{x}_t$ , dem Zeitschritt $t$ sowie der Konditionierung $\mathbf{c}$ . Der Generator liefert daraus eine entrauschte Struktur $\mathbf{x}'_0$ . . . . .	158
5.13	Generator des DD-GANs. Die Architektur ist ein 3D-U-Net, das verrauschte Strukturen verarbeitet. Die hellgrauen Blöcke markieren die Einbindung der Konditionierung. Die angegebenen Werte beziehen sich auf <i>Kanäle:räumliche Größe</i> . In jeder Faltungsstufe wird zusätzlich ein Zeitembedding addiert. . . . .	160
5.14	Schematische Funktionsweise des Diskriminators. Eine verrauschte Struktur $\mathbf{x}_t$ sowie eine etwas entrauschte Struktur $\mathbf{x}_{t-1}$ oder $\mathbf{x}'_{t-1}$ werden dem Diskriminator gegeben. Der Diskriminator bewertet nun den Realismus der entrauschten Struktur und sagt die Konditionierung $\hat{\mathbf{c}}$ voraus. Ziel des Diskriminators ist es die Konditionierung möglichst genau vorherzusagen und reale und generierte Strukturen zuverlässig voneinander über den Realismus zu unterscheiden. . . .	162
5.15	Diskriminator des DD-GANs. Die Architektur ist weitgehend von [34] übernommen. Die Größe der verarbeiteten Eingänge ist mit „Kanäle:Größe“ neben den Blöcken notiert. Wie der Generator erhält auch der Diskriminator ein Zeitschrittembedding, um ihn im Diffusionsprozess verwenden zu können. . . . .	163
5.16	Vergleich des Informationsgehalts in den Subvolumen: (a) Original, (b) skaliertes Subvolumen. Die rote Box in (b) markiert die Position des ursprünglichen Volumens. . . . .	165

5.17	Verteilung der Struktureigenschaften in den Subvolumen aus <i>Probe A</i> und <i>Probe C</i> . Für jeden CT-Scan sind die Werte um einen Mittelwert normalverteilt. . . . .	168
5.18	Logik der Bildverarbeitung zum Schließen der Eigenschaftslücken. Es werden Subvolumen einer zufälligen Kantenlänge $l \in \{86; 96; 128\}$ Voxel ausgeschnitten. Durch Dilatation oder Erosion werden die Eigenschaften verändert. Abschließend werden die Subvolumen auf eine Kantenlänge von $l = 96$ skaliert. . . . .	169
5.19	In der Datenerweiterung erstellte Mikrostrukturen. (a) zeigt einen 86 Voxel Ausschnitt der erodiert wurde. (b) zeigt einen 96 Voxel Ausschnitt auf den Dilatation angewendet wurde. (c) zeigt einen 128 Voxel Ausschnitt nach einer Erosion. . . . .	170
5.20	Vergleich der initialen Datenverteilung mit den Ergebnissen der Datenvorverarbeitung. Während die initialen Daten um zwei Mittelwerte normalverteilt sind, deckt die erweiterte Datenverteilung einen größeren Wertebereich gleichmäßiger ab. . .	171
5.21	Vergleich eines realen Subvolumens (a) mit einer generierten Struktur (b). . . . .	173
5.22	Visualisierung des RMSE der Porosität. Die Werte sind gemittelt über $s$ und $\sigma_{PR}$ . . . . .	175
5.23	Visualisierung des RMSE der Porosität unter Verwendung optimaler Werte für $s$ und $\sigma_{PR}$ . . . . .	177
5.24	Optimale Parameterwahl für die Wandstärke. Ein klarer Trend ist zu erkennen. . . . .	178
5.25	Optimale Parameterwahl für die SD des Porenradius. Kein klarer Trend ist erkennbar. . . . .	179
5.26	Vergleich der Hauptkomponentenverteilung des Generators (orange) mit den realen Daten (blau). . . . .	180
5.27	Vergleich der Verteilung der vorhergesagten Eigenschaften (grün) mit den berechneten Eigenschaften (blau) des Testdatensatzes des Diskriminators. Bei jeder Eigenschaft werden die Vorhersagen des Zeitschritts mit dem niedrigsten durchschnittlichen Fehler verwendet. Zur Visualisierung wird eine Kerndichteschätzung (KDE) verwendet. . . . .	186

5.28	Netzwerk zur Erstellung der Bildkonditionierung. Es werden vier $64 \times 64$ große Ausschnitte als Eingabe verwendet. Durch mehrere Faltungsschichten wird ein Embedding mit 128 Einträgen erstellt, das dem Generator übergeben wird. . . . .	189
5.29	Vergleich einer realen Struktur (a) mit dem eigenschaftskonditionierten (b) und dem neuen bildkonditionierten Generator (c). . . . .	192
5.30	PCA der „schlechtesten“ bildgenerierten Strukturen. Das rote Kreuz markiert die zur Konditionierung verwendete Struktur. Die generierten Strukturen (orangene Punkte) verschieben sich in Richtung besser abgedeckter Datenbereiche. Die genauesten Strukturen sind mit einem roten Rand versehen. . . . .	193
5.31	PCA der besten bildgenerierten Strukturen. Das rote Kreuz markiert die Struktur die zur Konditionierung verwendet wurde. Die orangenen Punkte zeigen die davon generierten Strukturen. Es ist eine gute Übereinstimmung der Strukturen erkennbar. . . . .	194
5.32	PCA-Projektion der generierten Strukturen aus $G_{\text{Bild}}$ (orange) im Vergleich zu den realen Daten (blau) und den Ergebnissen von $G_{\text{Struktur}}$ (grau). Grundlage ist ein PCA-Modell auf den CT-Daten. . . . .	196
5.33	PCA-Vergleich der nicht augmentierten CT-Daten <i>Probe A</i> , <i>Probe C</i> und <i>Probe B</i> . Die Darstellung erfolgt im Raum der ersten beiden Hauptkomponenten. . . . .	198
5.34	Vergleich der PCA-Projektionen für <i>Probe B</i> : reale Daten (blau), Generator vor dem Finetuning (orange), Generator nach dem Finetuning (lila). . . . .	200
5.35	Workflow zum Training des DD-GANs. Das Netzwerk kann sowohl über Bilder als auch über Struktureigenschaften konditioniert werden. Das trainierte Netzwerk wird auf Kadi4Mat hochgeladen und mit den Trainingsdaten verlinkt. . . . .	201
5.36	Workflow zur Generierung von Schaumstrukturen mit einem DD-GAN Generator. Es können sowohl zufällige Strukturen als auch solche mit definierten Struktureigenschaften generiert werden.	202

5.37	Exemplarischer Record einer generierten Struktur. Die Metadaten umfassen nach der Generierung nur die Parametrisierungswerte. . . . .	203
5.38	Datengetriebene Generierung auf Kadi4Mat: Auf rekonstruierten CT-Daten trainiertes DD-GAN erzeugt synthetische digitale Zwillinge. . . . .	205
6.1	Simulationssetup: Auf den untersuchten Schaum wird eine Verschiebung entlang der y-Achse, analog zum modellierten Druckversuch, aufgebracht. Die Verschiebung wird über eine lineare Funktion langsam an das Maximum herangeführt, um den Druckversuch zu imitieren. . . . .	212
6.2	Bestimmung der RVE-Größe: Sobald die Abweichung unter 5 % liegt ist ein RVE erreicht. . . . .	213
6.3	Druckversuche des Schaumproben in Aufschäumrichtung. . . . .	215
6.4	Methode zur Bestimmung des Druckmoduls nach DIN EN:ISO 844. Der Kurvenbereich zwischen 25 % und 75 % der Maximalkraft wird in 6 gleich große Segmente geteilt. Die benachbarten Segmente, deren Summe ihrer Steigung am größten ist, definieren den Druckmodul des untersuchten Materials. Der Schnittpunkt einer um 0,2 % nach rechts verschobener Geraden mit der Steigung des Druckmoduls, definiert die Stauchgrenze. . . . .	216
6.5	Gegenüberstellung der experimentellen und simulativen Untersuchungen der Schäume. Die Simulation wird auf den Druckversuchen von Probe A parametrisiert. In (a) ist das Ergebnis der Parametrisierung gezeigt. Zur Validierung der Parametrisierung stellen (b) und (c) die Experimente den Simulationen der anderen Proben gegenüber. . . . .	217
6.6	Schematische Darstellung des Simulationsworkflows. Der Workflow implementiert die mechanische Belastung einer Struktur und die Auswertung der Ergebnisse, kann aber um andere Anwendungsbeispiele erweitert werden. . . . .	218

6.7	Analyse der Simulationsergebnisse. Blaue Punkte: Spannungs- und Stauchungswerte pro Zeitschritt; orange Kurve: Interpolation. Der lineare Verlauf der Kurve entspricht dem effektiven Druckmodul. Der Schnittpunkt einer um 0,2 % nach rechts verschobenen Geraden mit der Steigung des Druckmoduls mit der interpolierten Kurve bestimmt die Stauchgrenze $\sigma_p$ . . . . .	219
6.8	Record der die mechanischen und strukturellen Eigenschaften einer Mikrostruktur beschreibt. Die Metadaten wurden durch die Workflowanalysen automatisiert hinzugefügt. . . . .	220
6.9	Zusammenfassung der Simulationen. Ein auf Druckversuchen validierter Simulationsworkflow bestimmt die mechanischen Eigenschaften der Schäume. . . . .	222
7.1	Schematische Darstellung der LRP: Die Ausgabe wird schrittweise bis zu den Eingaben zurückgeführt. Adaptiert aus [212].	226
7.2	Projektion realer und generierter Strukturen im PCA-Raum, basierend auf der 2-Punkt-Korrelation. Die generierten Strukturen überlagern den Wertebereich der realen Strukturen weitgehend und erweitern diesen zudem. Gleichzeitig treten im Merkmalsraum Lücken zwischen einzelnen Clustern auf, in denen bislang keine Strukturen vorhanden sind. Diese Bereiche markieren gezielt zu ergänzende Regionen zur Vervollständigung der Datenbank. . . . .	228
7.3	Hauptkomponentenanalyse der Zweipunkt-Korrelation. Die Datenpunkte sind gemäß der Struktureigenschaften eingefärbt. . .	230
7.4	Heatmap der Generator-Parameterkombinationen. Jede Box zeigt Eigenschaftsbereiche, die ein gezieltes Einstellen des Generators ermöglichen. Graue Kreise markieren die Verteilung der Trainingsdaten, ihr Radius skaliert mit der Datenmenge.	231
7.5	PCA aller Schaumstrukturen. Die realen Schaumstrukturen sind blau, die zunächst synthetisierten in orange dargestellt. Die gezielte Generierung weiterer Strukturen erzeugt die grünen Datenpunkte. Der Datenbereich konnte auf diese Weise vollständig abgedeckt werden. . . . .	232

7.6	Datenbank in KadiWeb mit exemplarischem Record. Die Punktwolke zeigt die abgelegten Strukturen, die mit den zugehörigen Workflows verknüpft sind. Einheitlich strukturierte Metadaten ermöglichen Suche und Reproduktion. . . . .	233
7.7	Paritäts- und Residuendiagramm des auf der Porosität und der Wanddicke trainierten Netzwerks. . . . .	240
7.8	Vom neuronalen Netzwerk abgeleiteten Materialkarten. (a) zeigt den Druckmodul und (b) die Stauchgrenze. . . . .	241
7.9	Workflow zur Vorbereitung der Datenbank. Die zu untersuchende Collection wird ausgewählt. Aus den Metadaten der darin enthaltenen Records werden die Eingangs- und Ausgangswerte gewählt und in tfrecords geladen. . . . .	243
7.10	Workflow zum Training des Netzwerks zur Datenbankanalyse. Neben den Trainingsdaten, werden die Netzwerkparameter, der Trainingsschedule sowie die Eingangs- und Ausgangsdaten des Netzwerks definiert. Nach Abschluss des Trainings können durch Explainable AI die wichtigsten Eingabewerte identifiziert werden. . . . .	245
7.11	Workflow zur Erstellung der Materialkarten. Ein trainiertes Netzwerk wird geladen und Intervalle der Eingabedaten definiert. Die Vorhersagen des Netzwerks werden als Materialkarte visualisiert. . . . .	246
7.12	Zusammenfassung der Identifikation von Struktur-Eigenschaft-Beziehungen. Die Workflows zur Generierung und Simulation ermöglichen den Aufbau einer analysierbaren Datenbank. . . . .	248
8.1	Zusammenfassung des gesamten entwickelten Materialentwicklungsprozesses. Ausgehend von CT-Daten konnte eine Prozesskette bis zur Erstellung von Materialkarten entwickelt werden. . . . .	251

8.2	Langfristige Vision für die Materialentwicklung. Erweiterter, adaptiver Entwicklungszyklus mit Active Learning, Laboranbindung und Prozess–Struktur–Eigenschafts-Beziehungen, in dem die datengetriebene Analyse als zentrales Steuerungselement ein selbstoptimierendes Materialdesignsystem ermöglicht. . . . .	255
A.1	Visualisierung des RMSE des mittleren Porenradius unter Verwendung optimaler Werte für $s$ und $\sigma_{PR}$ . . . . .	303
A.2	Visualisierung des RMSE der Standarabweichung des Porenradius unter Verwendung optimaler Werte für $s$ und $\sigma_{PR}$ . . . .	304
A.3	Visualisierung des RMSE der Wanddicke unter Verwendung optimaler Werte für $s$ und $\sigma_{PR}$ . . . . .	305

# Tabellenverzeichnis

4.1	Ergebnisse der Pycnometrie zur Ermittlung der Dichte des Polyurethans. Die römischen Ziffern, zeigen die Ergebnisse der drei Messungen. . . . .	103
4.2	Effektive Dichte und Porosität jeder Probe. Durch Messung der effektiven Dichte kann die Porosität berechnet werden. . . . .	103
4.3	Bildgrößen und reale Abmessungen der tomographierten Polyurethanschaumproben bei einer Auflösung von $2,75 \mu\text{m}/\text{Px1}$ . . .	104
4.4	FID zwischen den realen und synthetischen Datensätzen. GAN bezeichnet die Trainingsdaten des zweiten Schritts und algorithmisch die des ersten. . . . .	114
4.5	Quantitative Ergebnisse des Finetunings. . . . .	118
4.6	Ergebnisse der Strukturanalysen. Werte in $\mu\text{m}$ und Voxel bei einer Voxelgröße von $2,75 \mu\text{m}/\text{Voxel}$ . . . . .	124
5.1	Vergleich der Eigenschaftsberechnungen von GPU und PACE3D bei verschiedenen Volumengrößen. Die Differenz ist definiert als $\text{Wert}_{\text{GPU}} - \text{Wert}_{\text{PACE}}$ . . . . .	167
5.2	Effizienz des Generators bei verschiedenen Batch-Größen auf einer <i>NVIDIA A100-SXM4</i> GPU mit 80 GB. OOM bedeutet, dass nicht genügend Arbeitsspeicher zur Verfügung stand. . . . .	182
5.3	Vergleich der Effizienz des Generator-Modells mit dem geometriebasierten Generator aus Unterabschnitt 5.3.1 bei unterschiedlichen Strukturgrößen. . . . .	183
5.4	Mittlerer absoluter prozentualer Fehler (MAPE) der Eigenschaftsvorhersage des Diskriminators je Zeitschritt $t \in [1;8]$ . Der Diskriminator erhält $(\mathbf{x}_{t-1}, \mathbf{x}_t)$ und sagt die Eigenschaften $\hat{\mathbf{c}}$ voraus. . . . .	185



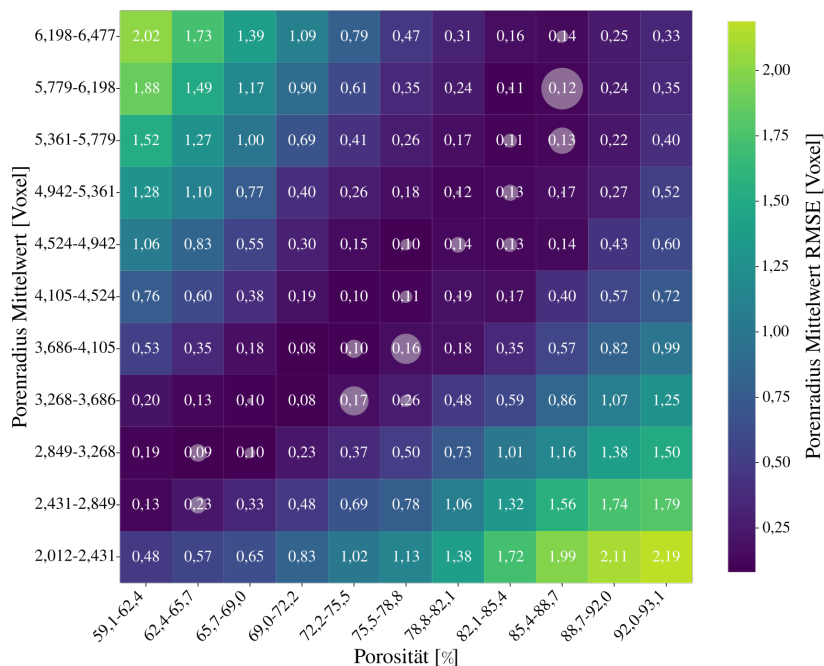
7.1	Relevanz der einzelnen Eingabewerte. Der Informationsbeitrag wurde über Layer-wise Relevance Propagation ermittelt. . . . .	236
7.2	Genauigkeit des trainierten neuronalen Netzwerks. Als Eingangsparameter dienen die Porosität und die Wandstärke. . . . .	237
7.3	Resultate der Kreuzvalidierung bezogen auf den Druckmodul. . . .	238
7.4	Resultate der Kreuzvalidierung bezogen auf die Stauchgrenze. . . .	239

# Abkürzungsverzeichnis

API	Application Programming Interface
BPMN	Business Process Model and Notation
CNN	Convolutional Neural Network
CT	Computertomographie
DD-GAN	Denoising Diffusion-Generative Adversarial Network
DDPM	Denoising Diffusion Probabilistic Model
EDF	Euklidisches Distanzfeld
ELN	Elektronisches Laborbuch
ELU	Exponential Linear Unit
FAIR	Findable, Accessible, Interoperable, Reusable
FDM	Forschungsdatenmanagement
FID	Fréchet Inception Distance
GAN	Generative Adversarial Network
KDE	Kerndichteschätzung
LRP	Layer-wise Relevance Propagation
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NN	Neuronales Netzwerk
PCA	Hauptkomponentenanalyse
ReLU	Rectified Linear Unit

RVE	Repräsentatives Volumenelement
SGD	Stochastic Gradient Descent
SiLU	Sigmoid Linear Unit
t-SNE	t-distributed Stochastic Neighbour Embedding
VAE	Variational Autoencoder
ZPK	Zweipunktkorrelation

# A Anhang



**Abbildung A.1:** Visualisierung des RMSE des mittleren Porenradius unter Verwendung optimaler Werte für  $s$  und  $\sigma_{PR}$ .

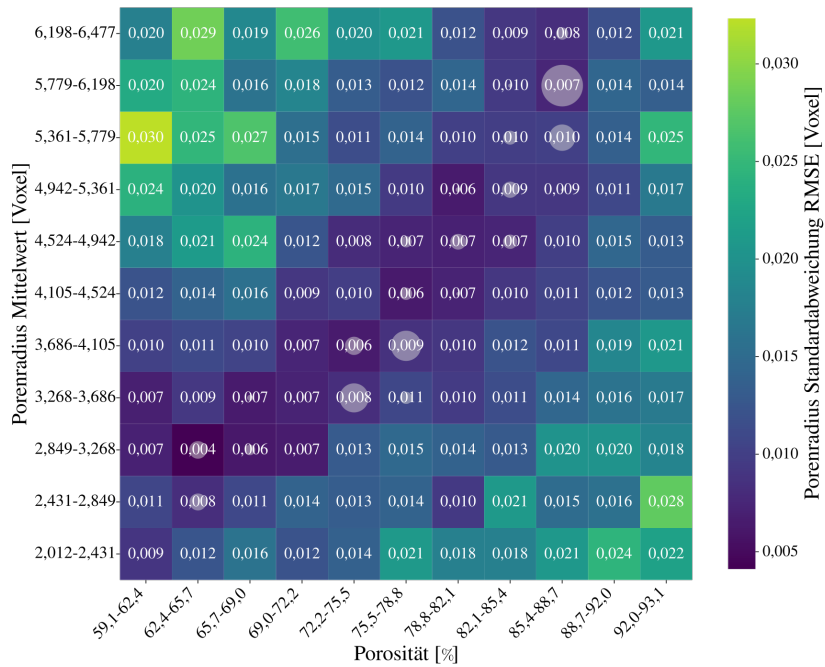
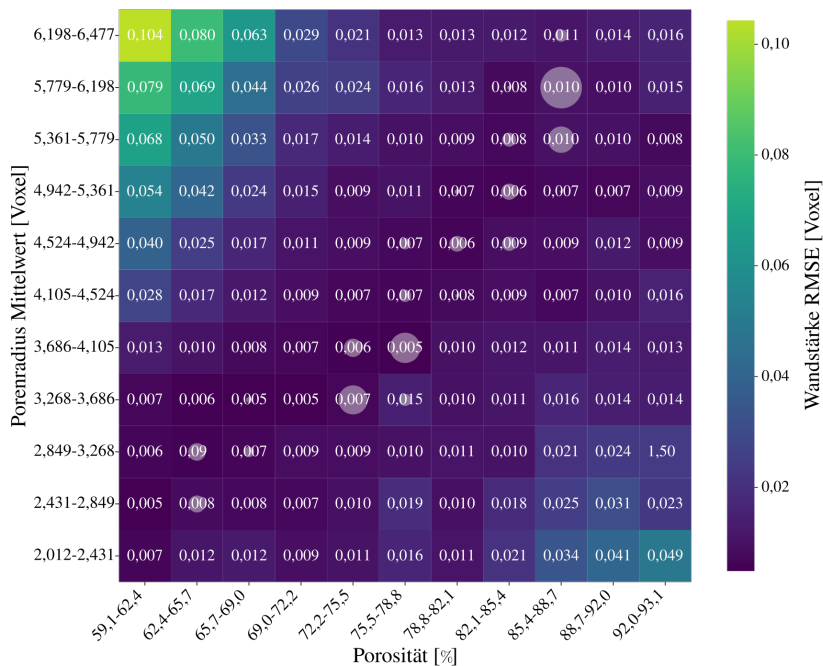


Abbildung A.2: Visualisierung des RMSE der Standardabweichung des Porenradius unter Verwendung optimaler Werte für  $s$  und  $\sigma_{PR}$ .



**Abbildung A.3:** Visualisierung des RMSE der Wanddicke unter Verwendung optimaler Werte für  $s$  und  $\sigma_{PR}$ .



## B Veröffentlichungen

### Beiträge in Printmedien

- [1] **L. Griem**, P. Zschumme, M. Laqua, N. Brandt, E. Schoof, P. Altschuh, M. Selzer "KadiStudio: FAIR Modelling of Scientific Research Processes", Data Science Journal, 21(1), p.16, September 2022  
<https://doi.org/10.5334/dsj-2022-016>
- [2] **L. Griem**, A. Koeppe, A. Greß, T. Feser, B. Nestler, "SSynthetic training data for CT image segmentation of microstructures", Acta Materialia, 296, September 2025 121220.  
<https://doi.org/10.1016/j.actamat.2025.121220>
- [3] **L. Griem**, R. Thelen, M. Selzer, Automated Documentation of Research Processes using RDM, Proceedings of the Conference on Research Data Infrastructure. Vol. 1. 2023  
<https://doi.org/10.52825/cordi.v1i.411>
- [4] N. Brandt, **L. Griem**, C. Herrmann, E. Schoof, G. Tosato, Y. Zhao, P. Zschumme, M. Selzer, "Kadi4Mat: A research data infrastructure for materials science", Data Science Journal, 20(1), p.8., Januar 2021  
<https://doi.org/10.5334/dsj-2021-008>
- [5] Y. Ji, A. Koeppe, P. Altschuh, **L. Griem**, D. Rajagopal, B. Nestler, (2024). A U-Net-based self-stitching method for generating periodic grain structures. Physica Scripta, 99(7), 076010.  
<https://doi.org/10.1088/1402-4896/ad52cf>



- [6] B. Nestler, F. Finsterwalder, L. Wallat, **L. Griem**, W. Kunz, M. Reder, J. Holland-Cunz, F. Jamshidi, M. Selzer (2023). Neue Materialien mit poröser Mikrostruktur: Beschleunigte Entwicklung durch Simulationen und Datenanalyse. *Athene – Magazin der Heidelberger Akademie der Wissenschaften*, 2, 21–24. <https://www.hadw-bw.de/sites/default/files/documents/Athene-2-2023.pdf>
- [7] Y. Zhao, P. Altschuh, J. Santoki, **L. Griem**, G. Tosato, M. Selzer, A. Koeppe, B. Nestler (2023). Characterization of porous membranes using artificial neural networks. *Acta Materialia*, 253, 118922. <https://doi.org/10.1016/j.actamat.2023.118922>
- [8] R. Al-Salman, C. Teixeira, P. Zschumme, S. Lee, **L. Griem**, J. Aghassi-Hagmann, C. Kirchlechner, M. Selzer (2023.) KadiStudio use-case workflow: Automation of data-processing for in situ micropillar compression tests. *Data Science Journal*, 22(1). <https://doi.org/10.5334/dsj-2023-021>
- [9] **L. Griem**, B. Nestler (2025). Künstliche Materialzwillinge: Strukturdesign mit generativer KI. *Forschung aktuell*, 2025, 76–79. Hochschule Karlsruhe – Technik und Wirtschaft. <https://www.h-ka.de/forschung-aktuell>

## Konferenzbeiträge

- [1] **L. Griem**, A. Koeppe, M. Selzer, ”Kadi4Mat: AI Driven Extraction of Structure-Property Linkages from Material Databases”, *Materials Process Applications Seminar* (2024), Stuttgart, Deutschland, Vortrag
- [2] **L. Griem**, R. Thelen, M. Selzer, ”Automated Documentation of Research Processes Using RDM”, *Conference on Research Data Infrastructure* (2023), Karlsruhe, Deutschland, Poster

- [3] **L. Griem**, A. Koeppe, T. Feser, M. Selzer, E. Beeh, B. Nestler, "Revealing Structure-Property Linkages using Explainable AI", Helmholtz AI Konferenz (2023), Hamburg, Deutschland, Vortrag
- [4] **L. Griem**, A. Koeppe, T. Feser, M. Selzer, E. Beeh, B. Nestler, "Explainable Prediction of Mechanical Properties in Porous Microstructures", 1st International Seminar on Modelling, Simulation and Machine Learning for the rapid development of porous materials (2023), Köln, Deutschland, Vortrag
- [5] **L. Griem**, A. Greß, P. Altschuh, T. Feser, A. Koeppe, M. Selzer, B. Nestler, "Identifying mechanical properties of foam structures using machine learning", MSE (2022) Materials Science and Engineering, Darmstadt, Deutschland, Vortrag
- [6] **L. Griem**, A. Greß, P. Altschuh, A. Koeppe, T. Feser, M. Selzer, E. Beeh, B. Nestler, "Predicting mechanical properties of foam structures using machine learning", Helmholtz AI Konferenz (2022), Dresden, Deutschland, Poster
- [7] **L. Griem**, A. Koeppe, P. Altschuh, E. Schoof, N. Brandt, P. Zschumme, M. Selzer, B. Nestler, "Membrane Characterisation using Machine Learning", Euromembrane (2021), Kopenhagen, Dänemark, Poster
- [8] **L. Griem**, A. Koeppe, M. Selzer, B. Nestler, "Kadi4Mat: Automated Extraction of Structure-Property Linkages using AI", FAIR-DI (2024) Karlsruhe, Deutschland, Vortrag  
<https://events.fairmat-nfdi.eu/event/9/contributions/208/>

## Dataset

- [1] Griem, L. (2025). Reconstruction and Characterisation. Zenodo. <https://doi.org/10.5281/zenodo.17019209>
- [2] Griem, L. C. (2025). ML-based Generation of digital twins. Zenodo. <https://doi.org/10.5281/zenodo.17019359>

- [3] Griem, L. C. (2025). Workflow Simulation. Zenodo. <https://doi.org/10.5281/zenodo.17019441>
- [4] Griem, L. C. (2025). Workflow Database Analysis. Zenodo. <https://doi.org/10.5281/zenodo.17019476>
- [5] Griem, L. C. (2024). Fiji Macro for Binary to Grayscale Image Translation. Zenodo. <https://doi.org/10.5281/zenodo.13711247>

## Betreute studentische Arbeiten

- [1] V. Rossknecht, „Phase-field simulation of a droplet impacting on solid and liquid surfaces using PACE3D“, Bachelorarbeit, Hochschule Karlsruhe 2023.
- [2] M. Hornberger, „Entwicklung eines Workflows zur Analyse und Generierung von Schaumstrukturen“, Bachelorarbeit, KIT 2023.
- [3] B. Herold, „Integration makroskopischer Wickinguntersuchungen in Kadi-Studio und deren Anwendung zur automatisierten Analyse von Diagnostikmembranen“, Bachelorarbeit, KIT 2024.
- [4] X. Chen, „Multi-Label UNet Segmentation of CT Images“, Seminararbeit, KIT 2024.
- [5] L. Dietrich, „Generierung von Schaumstrukturbildern mit Diffusionsmodellen“, Projektarbeit, Hochschule Karlsruhe, 2024.
- [6] S. Pfefferkorn, „Dynamische Erzeugung von Docker-Images für Processing Tools“, Projektarbeit, Hochschule Karlsruhe, 2025.
- [7] M. Falk, „Generierung digitaler Mikrostrukturen mittels generativer KI“, Bachelorarbeit, Hochschule Karlsruhe, 2025.
- [8] Z. Huang, „Geometriebasierte Generierung realistischer Schaumstrukturen“, Projektarbeit, Hochschule Karlsruhe, 2025.

- [9] S. Pohsner, „Erzeugung von Partikelpackungen“, Projektarbeit, Hochschule Karlsruhe, 2025.