

---

**Vorgehensmodell und Methoden zur datengetriebenen  
Optimierung von Entwicklungsprozessen in der  
Auslegung crashbelasteter Systeme am Beispiel der  
passiven Fahrzeugsicherheit**

---

**Dissertation**

zur Erlangung des akademischen Grades des  
*Doktors der Ingenieurwissenschaften (Dr.-Ing.)*  
der Naturwissenschaftlich-Technischen Fakultät  
der Universität des Saarlandes

vorgelegt von

**Janis Noah Mathieu**  
aus St. Ingbert

Saarbrücken, 2025

Tag des Kolloquiums	3. März 2026
Dekan Fakultät NT	Univ.-Prof. Dr.-Ing. Dirk Bähre
Vorsitz	Univ.-Prof. Dr. rer. nat. Andreas Schütze
1. Berichterstatter	Univ.-Prof. Dr.-Ing. Michael Vielhaber
2. Berichterstatter	Univ.-Prof. Dr. rer. nat. Jochen Garcke
Akad. Mitarbeiter	Dr.-Ing. Niklas König

# Kurzzusammenfassung

In der Fahrzeugentwicklung wird sicherheitsrelevanten Systemen eine hohe Priorität zugewiesen. Die Auslegung dieser – bei Eintritt eines Unfalls – crashbelasteten Systeme ist jedoch komplex und kostenintensiv. Entsprechend werden in dieser Arbeit vier Ansätze erarbeitet, um Entwicklungsprozesse durch Daten und intelligente Methoden zu optimieren. Zur Schaffung bestmöglicher Rahmenbedingungen wird das Vorgehensmodell *RAMEn* entwickelt, das eine zielgerichtete Prozessintegration neuer und bestehender datengetriebener Methoden realisiert. *DrAGOS* ist ein Reinforcement-Learning-basierter Optimierungsansatz, der durch Vortraining auf Metamodellen komplexes Systemverhalten innerhalb einer Stunde erlernt und dieses in der Anwendung auf Finite-Elemente-Simulationsmodelle überträgt. Das ermöglicht eine Optimierung in weniger als zehn Schritten sowie den Transfer auf ungesehene Problemstellungen. *DAYSi* ermöglicht die Analyse von Crashesensordaten. Die Erkennung von Abweichungen bildet die Basis für die Rekonstruktion einer Eventkette jedes Datenpunkts. In drei Use Cases konnten Auffälligkeiten und mögliche Ursachen in wenigen Minuten automatisch identifiziert werden. *ExPLAIN* bezieht sämtliche Datentypen einer Crashsimulation ein, um mittels interpretierbarem Machine Learning relevante Zusammenhänge zu erklären. Anhand von drei Use Cases wird gezeigt, dass neu generierte Einblicke das Verständnis und die Nachvollziehbarkeit bisher intransparenter Prozesse erhöhen.

Alle vier Ansätze tragen dazu bei, die Entwicklung neuer Fahrzeuge effizienter, transparenter und wissensbasierter zu gestalten.



# Abstract

In vehicle development, safety-critical systems are assigned as a high priority. However, the design of these systems—subjected to crash loads in the event of an accident—is complex and cost-intensive. Accordingly, this work presents four approaches to optimize development processes through data and intelligent methods.

To establish a systematic framework, the *RAMEn* process model is introduced, enabling the targeted process integration of new and existing data-driven methods. *DrAGOS* is a reinforcement-learning-based optimization approach that, through pre-training on metamodels, learns complex system behavior within one hour and transfers this knowledge to finite-element simulation models. This enables optimization in fewer than ten steps as well as transfer to previously unseen problems. *DAYSi* facilitates the analysis of crash-sensor signals. The detection of anomalies forms the basis for reconstructing an event chain for each datapoint. Across three use cases, anomalies and potential root causes were automatically identified within minutes. *ExPLAIN* incorporates all data types of a crash simulation to explain relevant relationships using interpretable machine learning. Three use cases demonstrate that newly generated insights improve understanding and transparency in otherwise black-box processes.

Collectively, all four approaches contribute to making the development of new vehicles more efficient, transparent, and knowledge-driven.



# Danksagung

Diese Dissertation wurde während meiner Tätigkeit bei der Porsche Engineering Services GmbH in Bietigheim-Bissingen und am Lehrstuhl für Produktentstehung der Universität des Saarlandes in Saarbrücken verfasst.

Mein außerordentlicher Dank gilt meinem Doktorvater Prof. Dr. Michael Vielhaber – Professor für Produktentstehung an der Universität des Saarlandes – für seine hervorragende Betreuung und die wertvollen Impulse. Seine Expertise und sein stets offenes Ohr haben maßgeblich dazu beigetragen, dass diese Arbeit einen erfolgreichen Abschluss findet.

Ebenso möchte ich mich aufrichtig bei Prof. Dr. Jochen Garcke – Professor für numerische Simulation an der Universität Bonn und Gruppenleiter für numerische datengetriebene Vorhersage am Fraunhofer SCAI – bedanken. Seine Unterstützung, gerade in Bezug auf die mathematischen Fragestellungen dieser Arbeit, haben einen sehr wichtigen Beitrag zu meiner Forschung geleistet. Auch den Austausch am Fraunhofer SCAI mit Dr. Rodrigo Iza Teran, Dr. Daniela Steffes-Lai und Tom Klein habe ich als äußerst bereichernd empfunden.

Seitens der Porsche Engineering möchte ich mich außerdem bei Michael Di Roberto und Dr. Joachim Schaper für das entgegengebrachte Vertrauen, die Freiheiten in der Umsetzung meiner Ideen, die fachliche und organisatorische Unterstützung sowie die mir zur Verfügung gestellten Ressourcen herzlich bedanken.

Mein aufrichtiger Dank gilt allen involvierten Kolleginnen und Kollegen der Porsche Engineering, der Dr. Ing. h.c. F. Porsche AG und des Volkswagen Konzerns. Dies gilt auch für das ganze Team des Lehrstuhls für Produktentstehung. Der stets sehr gute fachliche und überfachliche Austausch in großer und kleiner Runde sowie die unermüdliche Hilfsbereitschaft hat einen maßgeblichen Beitrag zum Erfolg meiner Forschung geleistet. Hierzu gehörten insbesondere Daniel Sommer, Dominik Neumann, Dr. Jana Hahlweg, Johannes Pfahler, Kristian König, Nicolas Scherer, Dr. Simon Mörsdorf, Stefan Kronwitter und Dr. Tobias Häfele.

Abschließend gilt mein besonderer Dank meinen Freunden und meiner Familie sowie meiner Lebenspartnerin Anna Jung. Ihr Beistand und ihre Geduld waren stets eine Quelle der Motivation. Meinen Eltern Tanja Bieg-Mathieu und Josef Mathieu – die beiden größten Unterstützer meines Lebens – widme ich diese Arbeit.

Saarbrücken, im Dezember 2025

Janis Noah Mathieu



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Problemstellung . . . . .	3
1.3	Aufbau der Arbeit . . . . .	4
1.3.1	Zielkriterien . . . . .	4
1.3.2	Zielsetzung . . . . .	5
1.3.3	Einführung der Forschungsfragen . . . . .	5
1.3.4	Eingrenzung . . . . .	6
<b>2</b>	<b>Theoretische Grundlagen</b>	<b>9</b>
2.1	Produktentwicklung . . . . .	9
2.1.1	Produktentwicklungsprozesse . . . . .	9
2.1.1.1	Abstrakte Prozessmodelle . . . . .	11
2.1.1.2	Prozedurale Prozessmodelle . . . . .	11
2.1.1.3	Analytische Prozessmodelle . . . . .	13
2.1.1.4	Management- und Operations-Research-Prozessmodelle . . . . .	13
2.1.2	Produktgenerationsentwicklung . . . . .	14
2.1.3	Situative Methodenauswahl . . . . .	15
2.2	Künstliche Intelligenz . . . . .	17
2.2.1	Maschine Learning . . . . .	17
2.2.1.1	Supervised Learning . . . . .	18
2.2.1.2	Unsupervised Learning . . . . .	22
2.2.1.3	Reinforcement Learning . . . . .	26
2.2.2	Interpretierbares Machine Learning . . . . .	27
2.3	Auslegung crashbelasteter Systeme . . . . .	33
2.3.1	Passive Fahrzeugsicherheit . . . . .	33
2.3.1.1	Anforderungen . . . . .	33
2.3.1.2	Anthropomorphe Testpuppen . . . . .	34
2.3.2	Virtuelle Systemauslegung . . . . .	35
2.3.2.1	Grundlagen der Finite-Elemente-Methode . . . . .	35
2.3.2.2	Anwendung der Finite-Elemente-Methode . . . . .	37
2.3.2.3	Optimierung von mechanischen Systemen . . . . .	38
2.4	Begriffsbestimmungen . . . . .	41

<b>3</b>	<b>Stand der Technik</b>	<b>43</b>
3.1	Datengetriebene Produktentwicklung . . . . .	44
3.1.1	Datengetriebene Produktentwicklungsprozesse . . . . .	45
3.1.2	Situative Auswahl von Machine-Learning-Methoden . . . . .	47
3.2	Datentypen in der Finite-Elemente-Methode . . . . .	50
3.3	Datengetriebene Optimierung . . . . .	53
3.3.1	Optimierung der Strukturintegrität . . . . .	53
3.3.1.1	Metamodellbasierte Optimierungsverfahren . . . . .	53
3.3.1.2	Graphen- und Heuristik-basierte Optimierungsverfahren . . . . .	54
3.3.1.3	Reinforcement-Learning-basierte Optimierungsverfahren . . . . .	54
3.3.2	Optimierung der Rückhaltesysteme . . . . .	57
3.4	Datengetriebene Analyse . . . . .	60
3.4.1	Skalare Daten . . . . .	60
3.4.2	1D-Zeitseriendaten . . . . .	61
3.4.3	Finite-Elemente-Daten . . . . .	63
3.4.3.1	Vorverarbeitung von Finite-Elemente-Daten . . . . .	63
3.4.3.2	Analyse von Finite-Elemente-Daten . . . . .	64
<b>4</b>	<b>Handlungsbedarf</b>	<b>69</b>
4.1	Defizite . . . . .	69
4.1.1	Datengetriebene Produktentwicklung . . . . .	69
4.1.2	Datengetriebene Optimierung . . . . .	74
4.1.3	Datengetriebene Analyse . . . . .	77
4.2	Konkretisierung der Forschungsfragen . . . . .	81
4.3	Abgrenzung der Arbeit . . . . .	82
<b>5</b>	<b>Einführung</b>	<b>85</b>
5.1	<i>RAMEn</i> – Methodenauswahl . . . . .	87
5.1.1	Orientierung in komplexen Schnittfeldern . . . . .	87
5.1.2	Vorgehensmodell . . . . .	88
5.1.2.1	Ziele und Ergebnisse . . . . .	89
5.1.2.2	Aktivitäten . . . . .	89
5.1.2.3	Anwendung . . . . .	90
5.1.2.4	Dokumentation von Methoden . . . . .	91
5.1.2.5	Einordnung und Auswahl von Methoden . . . . .	93
5.2	<i>DrAGOS</i> – Datengetriebene Systemauslegung . . . . .	94
5.2.1	Kurzbeschreibung und Schaubild . . . . .	94
5.2.2	Prozessintegration . . . . .	95

---

5.2.3	Ein- und Ausgaben . . . . .	96
5.2.4	Arbeitsschritte und Ablauf . . . . .	96
5.3	<i>DAYSi</i> – Analyse von Crashsensordaten . . . . .	99
5.3.1	Kurzbeschreibung und Schaubild . . . . .	99
5.3.2	Prozessintegration . . . . .	100
5.3.3	Ein- und Ausgaben . . . . .	101
5.3.4	Arbeitsschritte und Ablauf . . . . .	102
	5.3.4.1 Analyse von Zeitbereichen . . . . .	104
	5.3.4.2 Abstandsmaß zur Ausreißerkennung . . . . .	104
5.4	<i>exPLAIN</i> – Erklärung von Systemverhalten . . . . .	107
5.4.1	Kurzbeschreibung und Schaubild . . . . .	107
5.4.2	Prozessintegration . . . . .	109
5.4.3	Ein- und Ausgaben . . . . .	110
5.4.4	Arbeitsschritte und Ablauf . . . . .	111
	5.4.4.1 System-SHAP . . . . .	114
	5.4.4.2 Differenz-SHAP . . . . .	114
5.5	Zusammenfassung . . . . .	116
<b>6</b>	<b>Evaluierung</b> . . . . .	<b>117</b>
6.1	<i>RAMEn</i> – Methodenauswahl . . . . .	117
6.1.1	Use Case I: Einordnung von Methoden . . . . .	117
6.1.2	Use Case II: Auswahl von Methoden . . . . .	120
6.1.3	Strukturierte Verfügbarkeit von Wissen . . . . .	121
6.1.4	Generalisierbarkeit . . . . .	124
6.2	<i>DrAGOS</i> – Datengetriebene Systemauslegung . . . . .	126
6.2.1	Rückhaltesystemauslegung . . . . .	126
6.2.2	Anpassung der Architektur . . . . .	127
6.2.3	Vortraining mittels Metamodell . . . . .	129
	6.2.3.1 Auswahl des Metamodells . . . . .	129
	6.2.3.2 Trainingsprozess des Reinforcement-Learning-Agenten . . . . .	130
	6.2.3.3 Robustheit des Reinforcement-Learning-Agenten . . . . .	133
6.2.4	Benchmark mit globalem Optimierungsalgorithmus . . . . .	138
6.2.5	Anwendung <i>DrAGOS</i> auf Crashsimulationen . . . . .	139
	6.2.5.1 Use Case I: Aufgabenübergreifender Einsatz . . . . .	139
	6.2.5.2 Use Case II: Produktgenerationsübergreifender Einsatz . . . . .	140
6.3	<i>DAYSi</i> – Analyse von Crashsensordaten . . . . .	141
6.3.1	Use Case I: Bewertung einer Entwicklungsmaßnahme . . . . .	141

6.3.2	Use Case II: Erkennung von Ausreißern . . . . .	144
6.3.3	Use Case III: Bewertung einer Modellierungsmaßnahme . . . . .	148
6.3.4	Analyse Verfahrensparameter . . . . .	151
6.3.4.1	Auswahl und Anzahl Datenpunkte . . . . .	151
6.3.4.2	Auswahl und Anzahl Sensoren . . . . .	153
6.3.4.3	Zeitlags und Korrelation zwischen den Abweichungen . . . . .	157
6.3.4.4	Abstandsmaß zur Berechnung von Ausreißern . . . . .	158
6.3.4.5	Skalierung der Daten . . . . .	162
6.3.4.6	Zeitfenstergröße und Filterung . . . . .	164
6.4	<i>ExPLAIN</i> – Erklärung von Systemverhalten . . . . .	165
6.4.1	Use Case I: Lösungsvergleich in der Strukturoptimierung . . . . .	165
6.4.2	Use Case II: Einflussgrößen auf die Insassenbelastungen . . . . .	168
6.4.3	Use Case III: Erklärung von Einbettungsräumen . . . . .	173
6.4.3.1	Analyse der PCA-Einbettung . . . . .	174
6.4.3.2	Analyse von Clusterzugehörigkeiten . . . . .	177
6.4.3.3	Differenz-SHAP im 3D-Einbettungsraum . . . . .	180
6.4.4	Analyse Verfahrensparameter . . . . .	181
6.4.4.1	Vergleich Metamodelle . . . . .	181
6.4.4.2	SHAP-Werte für unterschiedlich große Datensätze . . . . .	183
6.4.4.3	SHAP-Werte für unterschiedliche Metamodelle . . . . .	187
6.4.4.4	Vergleich globaler und lokale Sensitivitätsmaße . . . . .	191
6.4.4.5	Umgang mit korrelierten Eingaben . . . . .	193
6.4.4.6	Vergleich Dimensionsreduktionsmethoden . . . . .	197
6.4.4.7	Vergleich Einbettungsräume . . . . .	203
6.5	Zusammenfassung . . . . .	207
<b>7</b>	<b>Diskussion</b>	<b>209</b>
7.1	<i>RAME<sub>n</sub></i> – Methodenauswahl . . . . .	209
7.1.1	<i>RAME<sub>n</sub></i> im Kontext bestehender Literatur . . . . .	209
7.1.2	Beantwortung der Forschungsfrage . . . . .	211
7.1.3	Einfluss auf Bewertungskriterien . . . . .	212
7.1.4	Schwachstellen und zukünftige Potenziale . . . . .	212
7.2	<i>DrAGOS</i> – Datengetriebene Systemauslegung . . . . .	214
7.2.1	<i>DrAGOS</i> im Kontext bestehender Literatur . . . . .	214
7.2.2	Beantwortung der Forschungsfrage . . . . .	215
7.2.3	Einfluss auf Bewertungskriterien . . . . .	216
7.2.4	Schwachstellen und zukünftige Potenziale . . . . .	217

---

7.3	<i>DAYSi</i> – Analyse von Crashesensorsignalen . . . . .	218
7.3.1	<i>DAYSi</i> im Kontext bestehender Literatur . . . . .	218
7.3.2	Beantwortung der Forschungsfrage . . . . .	221
7.3.3	Einfluss auf Bewertungskriterien . . . . .	222
7.3.4	Schwachstellen und zukünftige Potenziale . . . . .	222
7.4	<i>ExPLAIN</i> – Erklärung von Systemverhalten . . . . .	224
7.4.1	<i>ExPLAIN</i> im Kontext bestehender Literatur . . . . .	224
7.4.2	Beantwortung der Forschungsfrage . . . . .	226
7.4.3	Einfluss auf Bewertungskriterien . . . . .	227
7.4.4	Schwachstellen und zukünftige Potenziale . . . . .	227
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>231</b>
	<b>Literatur</b>	<b>235</b>
<b>A</b>	<b>Anhang</b>	<b>257</b>
A.1	<i>DrAGOS</i> – Metamodelle und RL-Agent . . . . .	257
A.1.1	Architektur des Metamodells zum Vortraining . . . . .	257
A.1.2	Hyperparameter der Metamodelle . . . . .	259
A.1.3	Hyperparameter RL-Agent . . . . .	260
A.1.4	Hyperparameter Differential Evolution . . . . .	261
A.2	<i>DAYSi</i> – Robustheit Verfahrensparameter . . . . .	262
A.2.1	Zeitfenstergröße . . . . .	262
A.2.2	Filterung . . . . .	266
A.3	<i>ExPLAIN</i> – Hyperparameter Metamodelle . . . . .	272



# Abkürzungen

**a3ms** maximale Brustbeschleunigung über einem Intervall von 3 ms.

**ATD** Anthropomorphe Testpuppe.

**CAD** Computer-Aided Design.

**CAE** Computer-Aided Engineering.

**CRISP-DM** Cross-Industry Standard Process for Data Mining.

**DAYSi** Datengetriebene Analyse von auffälligem SYstemverhalten in CrashsensorSignalen.

**DiffMaps** Diffusion Maps.

**DrAGOS** Datengetriebene Aufgaben- und Generationsübergreifende Optimierung von Systemen.

**DRM** Design Research Methodology.

**ECE UN-R** Economic Commission for Europe United Nations Regulations.

**EuroNCAP** European New Car Assessment Programme.

**ExPLAIN** Datengetriebene Erklärung und Vergleich von Systemverhalten.

**FE** Finite Elemente.

**FEM** Finite-Elemente-Methode.

**FMVSS** Federal Motor Vehicle Safety Standard.

**FN-56** Frontalaufprall 0° gegen eine starre Wand mit 56 km/h.

**H3** Hybrid III 50 % ATD-Typ.

**HF** Hybrid III 5 % ATD-Typ.

**HIC** Head Injury Criterion.

**IIHS** Insurance Institute for Highway Safety.

**(i-o)** Eingabe-Ausgabe.

**ISO-MME** Standardformat für 1D-Zeitreihendaten.

**KI** Künstliche Intelligenz.

**KL-Divergence** Kullback-Leibler-Divergenz.

**LIME** Local Interpretable Model-agnostic Explanations.

**LLM** Large Language Model.

**logPCA** Logistic Principal Component Analysis.

**MAD** Absolute Distanz zum Median.

**MCD** Abstand zum Hauptclustermittelpunkt.

**ML** Machine Learning.

**MSE** Mean Square Error.

**NHTSA** National Highway Traffic Safety Administration.

**(o-o)** Ausgabe-Ausgabe.

**PCA** Principal Component Analysis.

**PPO** Proximal Policy Optimization.

**R<sup>2</sup>** Bestimmtheitsmaß.

**RA** Restabstand ATD-Kopf zu Instrumententafel.

**RAMEn** VoRgehensmodell zur situativen Auswahl und Prozessintegration datengetriebener Methoden in Entwicklungsprozesse zur Auslegung crashbelasteter Systeme.

**RL** Reinforcement Learning.

**RMSE** Root Mean Square Error.

**Schweller\_PCA-1\_Tend** Hauptkomponente I im letzten Zeitschritt.

**SHAP** SHapley Additive Explanations.

**SwM** Optimierungszielwert Sitzbreitenminderung und Masse.

**TabPFN** Tabular Prior-Data Fitted Network.

**UMAP** Uniform Manifold Approximation and Projection.

**XGBoost** Extreme Gradient Boosting.

# Formelzeichen

## Allgemein

$\theta$	Parameter eines ML-Modells
$\mathcal{D}$	Datensatz
$\mathcal{X}$	Merkmalsraum der Eingaben
$\mathcal{Y}$	Ausgaberaum
$x_i$	$i$ -ter Eingabedatenpunkt
$y_i$	$i$ -ter Ausgabedatenpunkt
$\mathcal{L}$	Verlustfunktion eines Optimierungs- oder Lernproblems
$f$	Modellfunktion bzw. funktionaler Zusammenhang
$\mathbb{E}[\cdot]$	Erwartungswertoperator
$\hat{\cdot}$	Geschätzte oder approximierte Größe, z. B. $\hat{y}_i$
$\Sigma$	Kovarianzmatrix multivariater Zufallsgrößen
$p(\cdot)$	Wahrscheinlichkeitsdichte
$P(A   B)$	Wahrscheinlichkeit von Ereignis $A$ unter Bedingung $B$

## *DrAGOS*

$\gamma \in [0, 1]$	Diskontierungsfaktor zukünftiger Belohnungen
$\beta_{\mathcal{H}} \in \mathbb{R}^+$	Gewichtung des Entropieterms im PPO-Verlust
$\eta$	Lernrate des RL-Agenten
$\pi$	Strategie (Policy) des RL-Agenten
$a \in \mathcal{A} \subset \mathbb{R}^M$	Aktion des RL-Agenten und Aktionsraum
$\delta \in \mathbb{R}$	Schrittweite einer diskreten Aktion
$s \in \mathcal{S} \subset \mathbb{R}^M$	Umgebungszustand und Zustandsraum
$T, \hat{T}$	Systemmodell und dessen Approximation
$o \in \mathcal{O} \subset \mathbb{R}^D$	Systemantwort von $T, \hat{T}$
$R(o)$	Belohnungsfunktion des RL-Agenten
$r \in \mathbb{R}$	Belohnung des RL-Agenten
$q_{\pi}(s, a)$	Q-Funktion
$n \in \mathbb{N}$	Anzahl der Iterationen im Training oder Vortraining
$\cdot^*$	Optimale Lösung, z. B. $x^*$

***DAYSi***

$\tau \in \mathbb{R}$	Zeitlag in Millisekunden (ms)
$\rho \in \mathbb{R}$	Pearson-Korrelationskoeffizient
$M \in \mathbb{N}$	Anzahl der Sensoren
$N \in \mathbb{N}$	Anzahl der FE-Simulationen bzw. Versuche
$T \in \mathbb{N}$	Anzahl Signalwerte über die Zeit (Merkmale)
$S \in \mathbb{R}^{M \times N \times T}$	Sensorsignale
$\mu \in \mathbb{R}^T$	Clusterzentrum
$K \in \mathbb{N}$	Anzahl der Clusterzentren pro Signal
$J \in \mathbb{R}$	Varianz innerhalb der Cluster
$q \in \mathbb{R}^N$	Silhouette eines Clusters
$d \in \mathbb{R}^{M \times N}$	Globaler Abweichungskoeffizient
$D \in \mathbb{R}^{M \times N \times T}$	Lokaler Abweichungskoeffizient
$\mathcal{R} \in \mathbb{R}^{M \times M \times T \times N}$	Kreuzkorrelationsmatrix
$r \in \mathbb{R}^{M \times M \times N}$	Maximale Pearson-Korrelation zwischen allen Sensoren
$l \in \mathbb{R}^{M \times M \times N}$	Zeitlag bei maximaler Korrelation
$\mathcal{T} \in \mathbb{R}^{M \times M}$	Zugelassener Bereich der Zeitlags
$\bar{l} \in \mathbb{R}^{M \times N}$	Durchschnittlicher Zeitlag

***ExPLAIN***

$t$	Zeitindex mit $T =  t $ Anzahl der Zeitschritte
$N \in \mathbb{N}$	Anzahl der FE-Simulationen
$P \in \mathbb{N}$	Anzahl der Designparameter
$E \in \mathbb{N}$	Anzahl der extrahierten Merkmale 1D-Zeitserien
$F \in \mathbb{N}$	Dimension der assemblierten Metamodelleingabe
$\mathcal{M} \subset \mathfrak{M}$	Subsysteme als Teil eines Systems
$\mathcal{G}$	Graphenrepräsentation von FE-Daten
$\xi$	Attribute des FE-Graphen (z. B. Knoten-/Elementattribute)
$\Phi$	Abbildung vom FE-Graphen zum Voxelgrid
$\mathcal{V} \in \mathbb{R}^{N \times X \times Y \times Z \times C}$	Voxelgrid für ein Subsystem
$\phi \in \mathbb{R}^{N \times F}$	SHAP-Werte für das ML-Modell $f$
$r$	Zufallsstartwert (Random Seed)
$\alpha$	Korrekturfaktor Punktdichteverzerrung (DiffMaps)
$\epsilon$	Skalierungsparameter des Gauß-Kerns (DiffMaps)

# Abbildungsverzeichnis

1.1	Aufbau der Arbeit . . . . .	6
2.1	Produktentwicklung im Kontext des Produktlebenszyklus . . . . .	10
2.2	Lerntypen im Machine Learning . . . . .	18
2.3	Seitenaufprall Crashversuch Porsche Taycan . . . . .	35
2.4	FEM-Prozess . . . . .	38
2.5	Verschiedene Arten der Strukturoptimierung . . . . .	39
2.6	Strukturoptimierung mit Metamodellen . . . . .	40
3.1	Überblick und Aufbau Stand der Technik . . . . .	43
3.2	Datentypen in der Finite-Elemente-Methode . . . . .	50
3.3	Aufbau eines ISO-MME Kanalcodes . . . . .	52
5.1	Eingeführte Methoden im Kontext des Stands der Technik . . . . .	86
5.2	Erweiterter Problemlösezyklus zur Orientierung in Schnittfeldern . . . . .	88
5.3	Schaubild <i>RAMEn</i> : Vorgehensmodell . . . . .	89
5.4	Integration und Auswahl von Methoden im Vorgehensmodell <i>RAMEn</i> . . . . .	92
5.5	Schaubild <i>DrAGOS</i> : Arbeitsschritte und Ablauf . . . . .	95
5.6	Schaubild <i>DAYSi</i> : Arbeitsschritte und Ablauf . . . . .	100
5.7	Analyse von Zeitbereichen am Beispiel von Brustbeschleunigungskurven . . . . .	105
5.8	Silouhette Score und Abstandsmaß MCD zur Ausreißererkenntnis . . . . .	106
5.9	Schaubild <i>ExPLAIN</i> : Arbeitsschritte und Ablauf . . . . .	108
5.10	Voxelisierung von Subsystemen am Beispiel eines Airbags . . . . .	113
6.1	Einordnung von Methoden aus dem Stand der Technik . . . . .	120
6.2	<i>RAMEn</i> : Auswahl von Methoden im Vorgehensmodell . . . . .	121
6.3	Umfrage zu KI in der passiven Fahrzeugsicherheit – Frage I . . . . .	122
6.4	Umfrage zu KI in der passiven Fahrzeugsicherheit – Frage II . . . . .	123
6.5	Umfrage zu KI in der passiven Fahrzeugsicherheit – Frage III . . . . .	123
6.6	<i>RAMEn</i> Generalisierung: Leichtbau und datengetriebene Methoden . . . . .	125
6.7	<i>DrAGOS</i> Use Case: Rückhaltesystemauslegung . . . . .	126
6.8	<i>DrAGOS</i> Formulierung für die Rückhaltesystemauslegung . . . . .	128
6.9	<i>DrAGOS</i> Robustheit: Trainingsprozess . . . . .	131
6.10	<i>DrAGOS</i> Robustheit: Trainings- und Anwendungsprozess . . . . .	132

6.11	<i>DrAGOS</i> Robustheit: Trainingsprozess – Netzwerkgröße . . . . .	134
6.12	<i>DrAGOS</i> Robustheit: Trainingsprozess – Belohnung . . . . .	136
6.13	<i>DrAGOS</i> Robustheit: Trainingsprozess – Schrittweite . . . . .	137
6.14	<i>DrAGOS</i> Benchmark: Metamodellbasierte Optimierung . . . . .	138
6.15	<i>DrAGOS</i> Anwendung vortrainierter Strategie auf FE-Simulationen . . . . .	140
6.16	<i>DAYSi</i> Use Case I: Problemstellung . . . . .	141
6.17	<i>DAYSi</i> Use Case I: Überblick Datensatz . . . . .	142
6.18	<i>DAYSi</i> Use Case I: Lokale Ausreißer und Eventkette . . . . .	143
6.19	<i>DAYSi</i> Use Case II: Problemstellung . . . . .	145
6.20	<i>DAYSi</i> Use Case II: Überblick Datensatz . . . . .	145
6.21	<i>DAYSi</i> Use Case II: Lokale Ausreißer und Eventkette I . . . . .	146
6.22	<i>DAYSi</i> Use Case II: Lokale Ausreißer und Eventkette II . . . . .	147
6.23	<i>DAYSi</i> Use Case III: Problemstellung . . . . .	148
6.24	<i>DAYSi</i> Use Case III: Überblick Datensatz . . . . .	149
6.25	<i>DAYSi</i> Use Case III: Lokale Ausreißer und Eventkette . . . . .	150
6.26	<i>DAYSi</i> Robustheit: Use Case II mit weniger Datenpunkten I . . . . .	152
6.27	<i>DAYSi</i> Robustheit: Use Case II mit weniger Datenpunkten II . . . . .	153
6.28	<i>DAYSi</i> Robustheit: Use Case II mit zusätzlichen Sensoren I . . . . .	154
6.29	<i>DAYSi</i> Robustheit: Use Case II mit zusätzlichen Sensoren II . . . . .	155
6.30	<i>DAYSi</i> Robustheit: Use Case II mit zusätzlichen Sensoren III . . . . .	156
6.31	<i>DAYSi</i> Robustheit: Use Case I mit anderen Zeitlags und Korrelationen . . . . .	157
6.32	<i>DAYSi</i> Robustheit: Use Case I mit verschiedenen Abstandsmaßen . . . . .	159
6.33	<i>DAYSi</i> Robustheit: Use Case I mit MAD . . . . .	160
6.34	<i>DAYSi</i> Robustheit: Use Case I mit PCA+MAD . . . . .	161
6.35	<i>DAYSi</i> Robustheit: Use Case I mit signalweise Skalierung I . . . . .	162
6.36	<i>DAYSi</i> Robustheit: Use Case I mit signalweise Skalierung II . . . . .	163
6.37	<i>ExPLAIN</i> Use Case I: Problemstellung . . . . .	165
6.38	<i>ExPLAIN</i> Use Case I: Training Metamodell . . . . .	166
6.39	<i>ExPLAIN</i> Use Case I: Verteilung SwM und SHAP-Werte auf Systemebene . . . . .	167
6.40	<i>ExPLAIN</i> Use Case I: SHAP-Werte auf Merkmalsebene . . . . .	167
6.41	<i>ExPLAIN</i> Use Case I: Optimierung und Differenz-SHAP-Werte . . . . .	168
6.42	<i>ExPLAIN</i> Use Case II: Problemstellung . . . . .	169
6.43	<i>ExPLAIN</i> Use Case II: Abhängigkeiten und Verteilung der Daten . . . . .	170
6.44	<i>ExPLAIN</i> Use Case II: Training Metamodell . . . . .	171
6.45	<i>ExPLAIN</i> Use Case II: SHAP-Werte auf Subsystem- und Merkmalsebene . . . . .	171
6.46	<i>ExPLAIN</i> Use Case II: SHAP-Werte auf Merkmalsebene . . . . .	172
6.47	<i>ExPLAIN</i> Use Case III: Problemstellung . . . . .	173

---

6.48	<i>ExPLAIN</i> Use Case III: Voxelgrid Seitenschweller . . . . .	174
6.49	<i>ExPLAIN</i> Use Case III: Training Metamodell I . . . . .	175
6.50	<i>ExPLAIN</i> Use Case III: SHAP-Werte auf Subsystem und Merkmalsebene .	176
6.51	<i>ExPLAIN</i> Use Case III: Differenz-SHAP Ausreißer . . . . .	177
6.52	<i>ExPLAIN</i> Use Case III: Cluster im 3D-Einbettungsraum . . . . .	178
6.53	<i>ExPLAIN</i> Use Case III: Training Metamodell II . . . . .	179
6.54	<i>ExPLAIN</i> Use Case III: Globale und lokale Sensitivität . . . . .	179
6.55	<i>ExPLAIN</i> Use Case III: Vergleich Simulationen im 3D-Einbettungsraum .	181
6.56	<i>ExPLAIN</i> Robustheit: Use Case III mit abweichender Datensatzgröße I . .	184
6.57	<i>ExPLAIN</i> Robustheit: Use Case III mit abweichender Datensatzgröße II .	185
6.58	<i>ExPLAIN</i> Robustheit: Use Case III mit abweichender Datensatzgröße III .	186
6.59	<i>ExPLAIN</i> Robustheit: Use Case III mit abweichender Datensatzgröße VI .	187
6.60	<i>ExPLAIN</i> Robustheit: Use Case III mit verschiedenen Metamodellen I . .	188
6.61	<i>ExPLAIN</i> Robustheit: Use Case III mit verschiedenen Metamodellen II . .	189
6.62	<i>ExPLAIN</i> Robustheit: Use Case III mit verschiedenen Metamodellen III .	189
6.63	<i>ExPLAIN</i> Robustheit: Use Case III mit verschiedenen Metamodellen IV .	190
6.64	<i>ExPLAIN</i> Robustheit: Use Case I mit zwei globalen Sensitivitätsmaßen . .	191
6.65	<i>ExPLAIN</i> Robustheit: Use Case I mit zwei lokalen Sensitivitätsmaßen . .	192
6.66	<i>ExPLAIN</i> Robustheit: Use Case II mit korrelierten Eingaben global . . . .	195
6.67	<i>ExPLAIN</i> Robustheit: Use Case II mit korrelierten Eingaben lokal . . . .	196
6.68	<i>ExPLAIN</i> Robustheit: Use Case III mit logPCA-Einbettung . . . . .	198
6.69	<i>ExPLAIN</i> Robustheit: Use Case III mit DiffMaps-Einbettung . . . . .	199
6.70	<i>ExPLAIN</i> Robustheit: Use Case III mit UMAP-Einbettung I . . . . .	200
6.71	<i>ExPLAIN</i> Robustheit: Use Case III mit UMAP-Einbettung II . . . . .	201
6.72	<i>ExPLAIN</i> Robustheit: Use Case III mit Laplace-Beltrami-Einbettung . . .	202
6.73	<i>ExPLAIN</i> Robustheit: Use Case III Vergleich 1D-Einbettungsräume global	204
6.74	<i>ExPLAIN</i> Robustheit: Use Case III Vergleich 1D-Einbettungsräume lokal .	206
A.1	<i>DrAGOS</i> Prognosemodell zum Vortraining der Optimierungsstrategie . . .	258
A.2	<i>DAYSi</i> Robustheit: Use Case I mit anderer Zeitfenstergröße I . . . . .	263
A.3	<i>DAYSi</i> Robustheit: Use Case I mit anderer Zeitfenstergröße II . . . . .	264
A.4	<i>DAYSi</i> Robustheit: Use Case I mit anderer Zeitfenstergröße III . . . . .	265
A.5	<i>DAYSi</i> Robustheit: Use Case I mit anderer Filterung I . . . . .	266
A.6	<i>DAYSi</i> Robustheit: Use Case I mit anderer Filterung II . . . . .	267
A.7	<i>DAYSi</i> Robustheit: Use Case I mit anderer Filterung III . . . . .	268
A.8	<i>DAYSi</i> Robustheit: Use Case I mit anderer Filterung III . . . . .	269
A.9	<i>DAYSi</i> Robustheit: Use Case I mit anderer Filterung IV . . . . .	270

A.10 <i>DAYSi</i> Robustheit: Use Case I mit anderer Filterung V . . . . .	271
--	-----

# Tabellenverzeichnis

2.1	Verschiedene Darstellungsformen des Produktentstehungsprozesses . . . . .	12
2.2	Überblick modellagnostische Interpretationsmethoden . . . . .	28
4.1	Einordnung der Literatur zu datengetriebener Produktentwicklung . . . . .	73
4.2	Einordnung der Literatur zu datengetriebener Optimierung . . . . .	76
4.3	Einordnung der Literatur zu datengetriebener Analyse . . . . .	80
6.1	Metamodelle zur Prognose von ATD Belastungen im Frontalaufprall . . . . .	130
6.2	<i>ExPLAIN</i> Vergleich Metamodelle für Use Cases I-III . . . . .	182
7.1	<i>RAMEn</i> im Kontext bestehender Literatur . . . . .	209
7.2	<i>DrAGOS</i> im Kontext bestehender Literatur . . . . .	214
7.3	<i>DAYSi</i> im Kontext bestehender Literatur . . . . .	218
7.4	<i>ExPLAIN</i> im Kontext bestehender Literatur . . . . .	224
A.1	<i>DrAGOS</i> Vortraining: Hyperparameter TabPFN . . . . .	259
A.2	<i>DrAGOS</i> Vortraining: Hyperparameter XGBoost . . . . .	259
A.3	<i>DrAGOS</i> Vortraining: Hyperparameter Support Vector Regression . . . . .	259
A.4	<i>DrAGOS</i> Vortraining: Hyperparameter Multilayer Perceptron . . . . .	260
A.5	<i>DrAGOS</i> Vortraining: Hyperparameter Decision Trees . . . . .	260
A.6	<i>DrAGOS</i> Vortraining: Hyperparameter Lasso Regression . . . . .	260
A.7	<i>DrAGOS</i> Benchmark: Differential Evolution Hyperparameter . . . . .	261
A.8	<i>ExPLAIN</i> Metamodelle: Hyperparameter TabPFN . . . . .	272
A.9	<i>ExPLAIN</i> Metamodelle: Hyperparameter XGBoost . . . . .	272
A.10	<i>ExPLAIN</i> Metamodelle: Hyperparameter Support Vector Regression . . . . .	272
A.11	<i>ExPLAIN</i> Metamodelle: Hyperparameter Multilayer Perceptron . . . . .	273
A.12	<i>ExPLAIN</i> Metamodelle: Hyperparameter Decision Trees . . . . .	273
A.13	<i>ExPLAIN</i> Metamodelle: Hyperparameter Lasso Regression . . . . .	273



# 1 Einleitung

Das erste Kapitel beschreibt die Motivation der vorliegenden Arbeit in Abschnitt 1.1, legt die grundlegende Problemstellung in Abschnitt 1.2 ab S. 3 dar und führt Ziele sowie allgemeine Bewertungskriterien ein. Anschließend wird der Aufbau in Abschnitt 1.3 ab S. 4 anhand der Design Research Methodology (DRM) erläutert. Abschließend erfolgt die Eingrenzung des Forschungsfeldes.

## 1.1 Motivation

Die Zahl der weltweiten Verkehrstoten wird im Jahr 2021 auf etwa 1,19 Millionen beziffert [Wor23]. Dies entspricht – trotz der Verdopplung des globalen Fahrzeugbestands und der erheblichen Ausweitung der Straßennetze – einem Rückgang von 5 % gegenüber dem Jahr 2010. Obwohl die bislang ergriffenen Maßnahmen damit Wirksamkeit zeigen, reichen sie nicht aus, um das von den Vereinten Nationen formulierte Ziel zu erreichen, die Zahl der Verkehrstoten bis 2030 im Vergleich zu 2010 zu halbieren. Neben tragischen Schicksalen durch verlorene Menschenleben entstehen zudem enorme sozial-ökonomische Kosten, die weltweit pro Jahr über 65 Milliarden US-Dollar betragen [BHT22]. Neben der Initiierung von Programmen durch Organisationen, wie die Vereinten Nationen oder Regierungen, kann der Schutz von Verkehrsteilnehmern auch durch verbesserte Sicherheitssysteme in neuen Fahrzeugen erhöht werden [Gon+21, S. 1074]. Zu diesen zählen Fahrzeugstrukturen und Sicherheitsgurte aber auch der Spurhalteassistent oder das Antiblockiersystem.

In der Entwicklung von neuen Fahrzeugen wird dementsprechend sicherheitsrelevanten Komponenten und Systemen eine hohe Priorität zugewiesen, um die Fahrzeugsicherheit zu gewährleisten [Gon+21, S. 1074, 1155]. Zur Verbesserung der passiven Fahrzeugsicherheit werden im Rahmen der Strukturauslegung Fahrzeugkomponenten wie Seitenschweller, Längs- oder Querträger so optimiert, dass diese in verschiedenen Crashlastfällen Energie durch plastische Deformation absorbieren, dabei aber dennoch die strukturelle Integrität wahren. Diese Entwicklung findet im Zusammenspiel mit der Auslegung von Rückhaltesystemen statt, die die Bewegung eines Insassen während eines Aufpralls kontrollieren, um eine bestmögliche Körperposition und Energieabsorption sicherzustellen. Integrale Bestandteile des Rückhaltesystems sind die Sicherheitsgurte, die Airbags und die Sitze. Das übergeordnete Ziel dieser Entwicklungsaufwände ist entsprechend der Schutz von Insassen, aber auch von anderen Verkehrsteilnehmern, wie beispielsweise Fußgängern. Regularien, anhand derer die Verletzungsschwere und die strukturelle Integrität bewertet werden,

sind durch marktspezifische Gesetze durch den Federal Motor Vehicle Safety Standard (FMVSS) für Nordamerika oder die Economic Commission for Europe United Nations Regulations (ECE UN-R) für Europa sowie durch Verbraucherschutzinstitute wie das European New Car Assessment Programme (EuroNCAP) oder das Insurance Institute for Highway Safety (IIHS) vorgegeben. Im Rahmen der Fahrzeugentwicklung sind die dort aufgeführten Bewertungskriterien von großer Bedeutung, um die Zulassung und Akzeptanz der Fahrzeuge auf den entsprechenden Märkten zu gewährleisten.

Das Verhalten mechanischer Fahrzeugstrukturen unter Crashbelastung ist aufgrund von großen Deformationen, nichtlinearem Materialverhalten, Materialversagen sowie Kontaktinteraktionen hochgradig komplex [TSZ24]. Die Entwicklung einzelner Komponenten und Systeme sowie die Systemintegration im Gesamtfahrzeug sind daher zeit- und kostenaufwändig. In heutigen Entwicklungsprozessen werden viele Untersuchungen virtuell mittels numerischer Finite Elemente (FE)-Crashsimulationen durchgeführt. Diese ermöglichen eine realitätsnahe Abbildung des dynamischen Systemverhaltens und sind im Vergleich zu zerstörenden physischen Prüfverfahren deutlich ressourceneffizienter [BRK21]. Durch Crashsimulationen können daher bereits in frühen Phasen des Entwicklungsprozesses virtuell Konzepte bewertet sowie Schwachstellen identifiziert und verbessert werden. Damit die FE-Modelle der Realität entsprechen, ist der Abgleich von Simulationen mit physischen Crashversuchen unerlässlich. Darüber hinaus sind physische Crashversuche auch heute noch für eine Endabnahme notwendig. Bestrebungen hin zu virtuellen Abnahmen, beispielsweise für den EuroNCAP, sind jedoch absehbar [Klu+23]. Der hohe Detailgrad der Crashsimulationen, beispielsweise durch aufwändige Material- und Versagensmodellierung [TSZ24], geht allerdings mit einem hohen Modellaufbau- und Rechenaufwand einher [BRK21]. Berechnungszeiten können selbst auf Rechenclustern über 30 Stunden erreichen, was Kosten von mehreren Hundert bis Tausend Euro pro Simulation zur Folge hat. Durch die Vielzahl an betrachteten Crashlastfällen, entstehen zehntausende Crashsimulationen pro Fahrzeugprojekt. Die hohe Komplexität sowie der beträchtliche Zeit- und Kostenaufwand weisen auf ein erhebliches technisches und wirtschaftliches Potenzial zur Optimierung von Entwicklungsprozessen in der Auslegung crashbelasteter Systeme hin.

Ein vielversprechender Ansatz, um dieses Optimierungspotenzial zu heben ist, der Einsatz von Daten und intelligenten Datenverarbeitungsmethoden, wie dem Machine Learning (ML). ML ist ein Teilbereich des Feldes Künstliche Intelligenz (KI). ML-Algorithmen optimieren sich auf Basis von Daten und gesammelten Erfahrungen selbst und sind daher in der Lage, Zusammenhänge zu erlernen [PH20, S. 45]. So können sie Entscheidungen und Vorhersagen treffen, ohne explizit dafür programmiert worden zu sein. Die Reduktion von Rechenzeiten im Kontext der Crashsimulation kann beispielsweise den Einsatz von ML-Modellen, die erlerntes mechanisches Systemverhalten für neue und ungesehene

Systemkonfigurationen prognostizieren, signifikant reduziert werden [GW22; Büt+23]. Durch die hohe Anzahl generierter Simulationsdaten, die nach der Evaluation keinem weiteren Zweck mehr dienen, besteht ein hohes technisches und wirtschaftliches Potenzial in der Wieder- und Weiterverwendung dieser Daten [Koh+21]. Um Herausforderungen durch stetig steigende Simulationsdatenumengen zu begegnen, wurden datengetriebene Analysemethoden vorgestellt, die dabei helfen, auffälliges Systemverhalten automatisch zu detektieren und zu analysieren [Die+18; Kra+23]. Darüber hinaus können intelligente Agenten das mechanische Systemverhalten so optimieren, dass im Crashszenario Energie effizienter absorbiert werden kann [TSZ24; BGS24a]. Neben den Methoden in der Literatur sind auch erste kommerzielle Lösungen, wie beispielsweise SimExplore [Fra24] zur Analyse von Simulationsdaten oder ClearVu-Analytics [Div24] zur prädiktiven Modellierung und Optimierung, verfügbar. Die genannten Ansätze tragen dazu bei, Entwicklungsprozesse zur Auslegung crashbelasteter Systeme mit Hilfe von Daten und Algorithmen zu optimieren und leisten dadurch einen Beitrag zur Gestaltung sicherer Produkte.

## 1.2 Problemstellung

Durch hohen Konkurrenzdruck und immer komplexer werdende Systeme aufgrund von Kundenanforderungen, aber auch gesetzlichen Regularien in einer Vielzahl von Märkten, bedarf es auch weiterhin dem Vorantreiben der Entwicklung digitaler und automatisierter Entwicklungsmethoden und -prozesse [Can+20]. Der zielgerichtete und situative Einsatz von intelligenten Methoden entlastet involvierte Ingenieure von repetitiven Tätigkeiten und befähigt diese, effizientere und effektivere Entwicklungsaufgaben zu bewältigen. Des Weiteren lassen sich so Fehlerquellen minimieren und einheitliche Rahmenbedingungen schaffen. Neben den Entwicklungsmethoden selbst bedarf es auch projektübergreifend anwendbarer Umsetzungsstrategien und Vorgehensempfehlungen, da aufgrund des Neuheitsgrades wenig Erfahrungswerte vorhanden sind. Weitere Potenziale können durch den verbesserten Einsatz neuer, aber auch bestehender datengetriebener Methoden gehoben werden, da dort klassisch ausgebildeten Ingenieuren meist die Expertise fehlt [Son+24b]. Die vorliegende Arbeit ist dementsprechend im Forschungsfeld der datengetriebenen Optimierung von Entwicklungsprozessen in der Auslegung crashbelasteter Systeme verortet. Motiviert wird dies durch eine direkte Relevanz in der Anwendung – der passiven Fahrzeugsicherheit im Automobilbau. Trotz rapider Fortschritte im Feld der KI allgemein [Bom+24; PB25] lassen sich Herausforderungen und bisher unbetrachtete Aspekte, gerade in Bezug auf das betrachtete Forschungsfeld, identifizieren. Im Bereich der Produktentwicklungsmethodik, die den Rahmen für den effizienten Einsatz datengetriebener Methoden schaffen soll, fehlen beispielsweise anwendungsspezifische Vorgehensmodelle. Schwachstellen lassen sich

zudem im Einsatz von intelligenten Methoden zur generationsübergreifenden Optimierung komplexer Systeme finden. Ansätze für spezifische, aber essenzielle Problemstellungen im betrachteten Feld, wie beispielsweise die Analyse von Zeitseriendaten – wie sie durch eine Anthropomorphe Testpuppe (ATD) zur Belastungsbewertung ausgegeben werden – sind ebenfalls nicht hinreichend betrachtet worden. Dies ist darin begründet, dass der Fokus auf dem kontextbasierten Vergleich der Zeitserien in wenigen Datenpunkten liegt, was in der klassischen Zeitseriendatenanalyse eher selten der Fall ist. Ebenso sind die neuesten Fortschritte der KI-Forschung, beispielsweise im interpretierbaren ML, das sich ideal zur Analyse komplexer Zusammenhänge in Simulationsdaten eignet, oder auch die hohen Prognosegüten von In-Context-Learning-Algorithmen, nicht weitreichend genug evaluiert und in das Forschungsfeld übertragen worden.

Eine weitergehende Konkretisierung der Problemstellung – aufbauend auf Erkenntnissen aus dem Stand der Technik – erfolgt in Kapitel 4 ab S. 69.

## 1.3 Aufbau der Arbeit

Um einen Beitrag zur Lösung der Problemstellung zu leisten, werden im Rahmen dieser Dissertation neue Ansätze erarbeitet, die Entwicklungsprozesse in der Auslegung crash-belasteter Systeme durch den Einsatz von Daten und Algorithmen optimieren. Um die wissenschaftlichen Fragestellungen dabei systematisch zu beantworten wird die DRM nach Blessing und Chakrabarti [BC09] zur Strukturierung dieser Arbeit herangezogen. Das Vorgehen nach der DRM ist in die Phasen I: *Klärung des Forschungsgegenstandes*, II: *Deskriptive Studie I*, III: *Präskriptive Studie* und IV: *Deskriptive Studie II* untergliedert. Eine grafische Übersicht des Aufbaus ist Abbildung 1.1 auf S. 6 zu entnehmen. Bevor dieser genauer erläutert wird, werden zunächst Zielkriterien und Ziele der Arbeit formuliert.

### 1.3.1 Zielkriterien

Damit bestehende und im Rahmen dieser Arbeit neu erarbeiteten Ansätze besser evaluiert und diskutiert werden können, erfolgt nun die systematische Einführung übergeordneter Zielkriterien. Dies erfolgt anhand von [VDI 2221-1; Alb+15] im Kontext der Produktentwicklungsmethodik. Verwendet werden hier die Erhöhung der *technischen Wertigkeit* der Systeme in Bezug auf die passive Fahrzeugsicherheit, die Einsparung von *Entwicklungszeit*, die Einsparung von *Entwicklungskosten*, die Erhöhung der (*Prozess- und Produkt-)*Qualität, die Erhöhung des *Innovationsgrades* sowie die Erhöhung von *Wissen* beteiligter Ingenieure und Organisationen.

### 1.3.2 Zielsetzung

Im Rahmen dieser Arbeit werden entwicklungsunterstützende Maßnahmen, darunter ein Vorgehensmodell und drei Methoden, eingeführt, evaluiert und diskutiert.

Das Ziel des Vorgehensmodells ist es, die Auswahl und Prozessintegration von datengetriebenen Methoden in der Entwicklung zu verbessern. Die Verknüpfung datengetriebener Methoden im Anwendungsfeld mit der Produktentwicklungsmethodik trägt dazu bei, eine neuartige Sichtweise auf das betrachtete Forschungsfeld zu ermöglichen und dabei die Ingenieure sowie das Management in der Entscheidungsfindung systematisch und anwendungsorientiert zu unterstützen. Durch die Schaffung neuer Rahmenbedingungen soll Wissen strukturiert verfügbar gemacht und erhöht werden.

Die drei eingeführten Methoden unterstützen die Durchführung von Entwicklungsaktivitäten direkt. Mit dem Ziel, die Entscheidungsfindung zu automatisieren und den Wissenstransfer zu verbessern, wird eine Methode zur Optimierung von generations- und aufgabenstellungsübergreifender Systemauslegung eingeführt. Dabei sollen insbesondere Entwicklungszeit und -kosten durch erhöhte Effizienz reduziert und die Spezifikation technisch besserer Optima realisiert werden. Mit dem Ziel, Fehler und mögliche Ursachen in Crashsensordaten automatisch auffinden zu können, wird eine datengetriebene Ausreißerererkennung entwickelt. Neben gesteigerter Effizienz soll zudem die Modell- und Prozessqualität sichergestellt werden. Mit dem Ziel, tiefere Einblicke in das Systemverhalten zu erlangen und Informationen bereitzustellen, die bisher nicht verfügbar waren, wird eine Analyseverfahren auf Basis modularer Datenverarbeitung und interpretierbarem ML zur Erklärung beliebiger Zusammenhänge in Simulationsdaten vorgeschlagen. Das Wissen und Verständnis über die betrachteten Systeme und deren Verhalten soll so für Personen und Organisationen erhöht werden.

### 1.3.3 Einführung der Forschungsfragen

Im Aufbau der Arbeit in Abbildung 1.1 werden jeder Phase Forschungsfragen und Ergebnisse sowie das zugehörige Kapitel zugeordnet. In der *Klärung des Forschungsgegenstandes* wird das grundlegende Forschungsziel der Arbeit definiert. Außerdem werden neben der Beschreibung der Ausgangssituation Kriterien zur Bewertung des Forschungsvorhabens festgelegt. Diese Aspekte werden in Kapitel 1 behandelt. In der *Deskriptiven Studie I* wird das Forschungsvorhaben zur Zielerreichung präzisiert. Unterstützt wird das durch zielgerichtete Literaturrecherchen und die Identifikation relevanter Einflussfaktoren, die die Bewertung von Methoden und Prozessen während der Forschung maßgeblich beeinflussen. In der vorliegenden Arbeit werden diese Aspekte in Kapitel 2, Kapitel 3 und Kapitel 4 abgehandelt. In der *Präskriptiven Studie* wird auf Basis des in Phase II definierten Ist-Zustands der Übergang in einen gewünschten Zielzustand gestaltet. Dabei

werden identifizierte Einflussfaktoren aus verschiedenen Szenarien berücksichtigt und in entwicklungsunterstützende Maßnahmen überführt. Im Rahmen der vorliegenden Arbeit wird das in Kapitel 5 durchgeführt. In der *Deskriptiven Studie II* werden die in der vorangegangenen Phase entwickelten Ansätze und Methoden auf ihre Eignung zur Erreichung des Zielzustands geprüft. Die Eignung kann über eine Evaluation mittels einfacher funktionaler Tests bis hin zu umfangreichen Studien festgestellt werden. In dieser Arbeit erfolgt das in Kapitel 6 unter Verwendung von sechs verschiedenen Use Cases aus der Entwicklung passiver Sicherheitssysteme im Automobilbau und einer Umfrage.

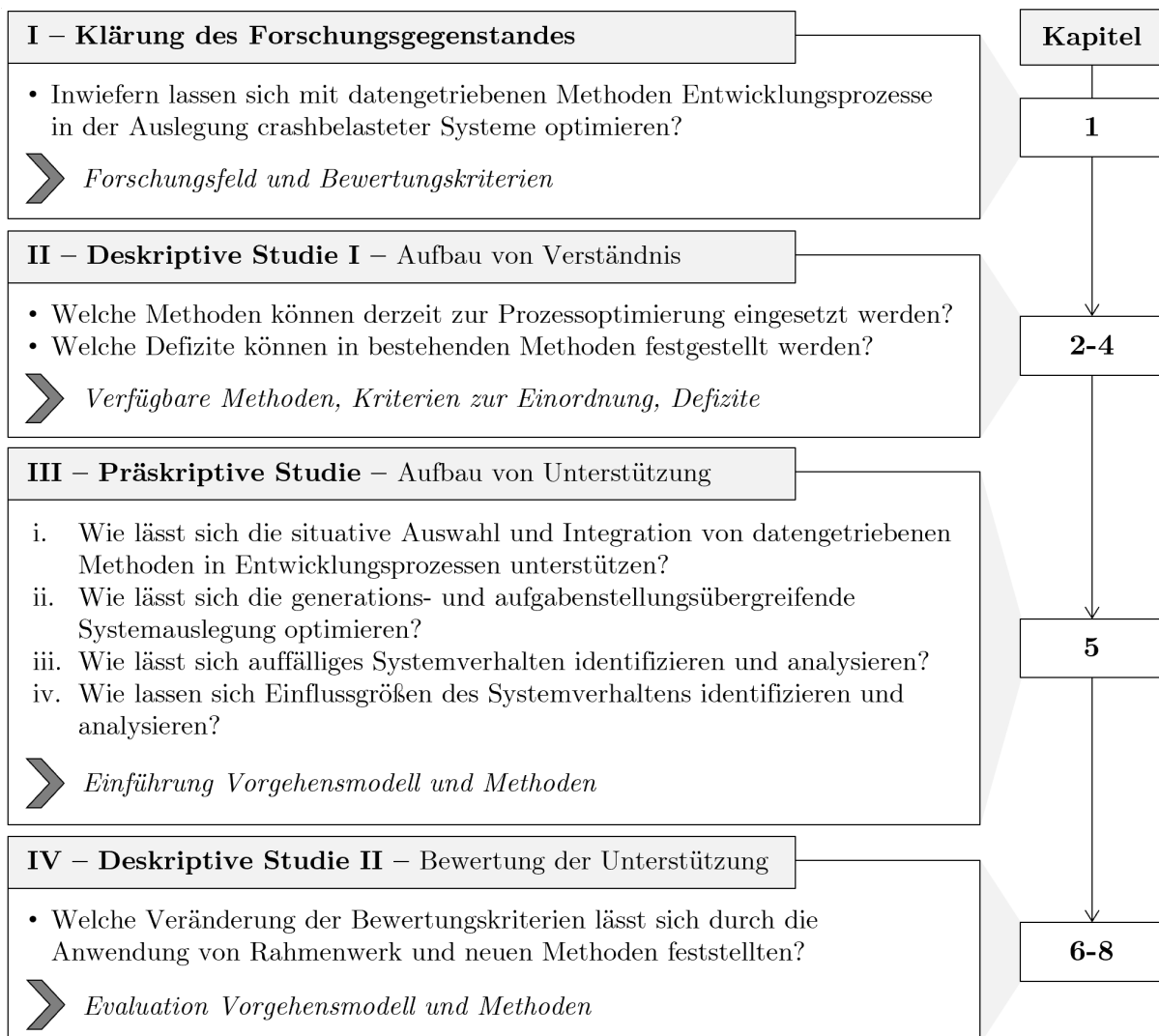


Abbildung 1.1: Aufbau der Arbeit mittels der DRM, nach [BC09]

### 1.3.4 Eingrenzung

In Bezug auf das bereits eingeführte Forschungsfeld der datengetriebenen Optimierung von Entwicklungsprozessen in der Auslegung crashbelasteter Systeme mit dem Anwendungsfeld

---

der passiven Fahrzeugsicherheit, erfolgen an dieser Stelle weitere Eingrenzungen. Als Grundlage, insbesondere für die Methoden, dienen in dieser Arbeit die im Entwicklungsprozess anfallenden Daten, mit denen auch die Ingenieure selbst alltäglich interagieren. Andere datengetriebene Methoden, wie sie beispielsweise in den Arbeiten von Thel et al. [The+24] oder Böhringer et al. [Böh+23] diskutiert werden, die den Berechnungsprozess auf der Ebene des Löfers innerhalb der Finite-Elemente-Methode (FEM) optimieren, sind dabei außerhalb des Betrachtungsrahmens. Ebenfalls außerhalb des Betrachtungsrahmens ist die Entwicklung der KI- und ML-Algorithmen selbst. Der Fokus liegt daher auf der Erarbeitung neuer anwendungsorientierter Ansätze, unter Einbindung bestehender Algorithmen. Ein wissenschaftlicher Mehrwert entsteht daher insbesondere durch den generalisierten Betrachtungsrahmen und die Kombination einzelner Algorithmen zu neuen Entwicklungsmethoden für die industrielle Anwendung.



## 2 Theoretische Grundlagen

Im Grundlagenkapitel werden die für das Verständnis dieser Arbeit wichtigsten Aspekte im Bereich Produktentwicklung in Abschnitt 2.1, KI und insbesondere ML in Abschnitt 2.2 ab S. 17 sowie der simulativen Auslegung von crashbelasteten Systemen im Kontext der passiven Fahrzeugsicherheit in Abschnitt 2.3 ab S. 33 erläutert. Diese Überlegungen bilden die Grundlage für den darauffolgenden Stand der Technik des betrachteten Forschungsfeldes in Kapitel 3 ab S. 43.

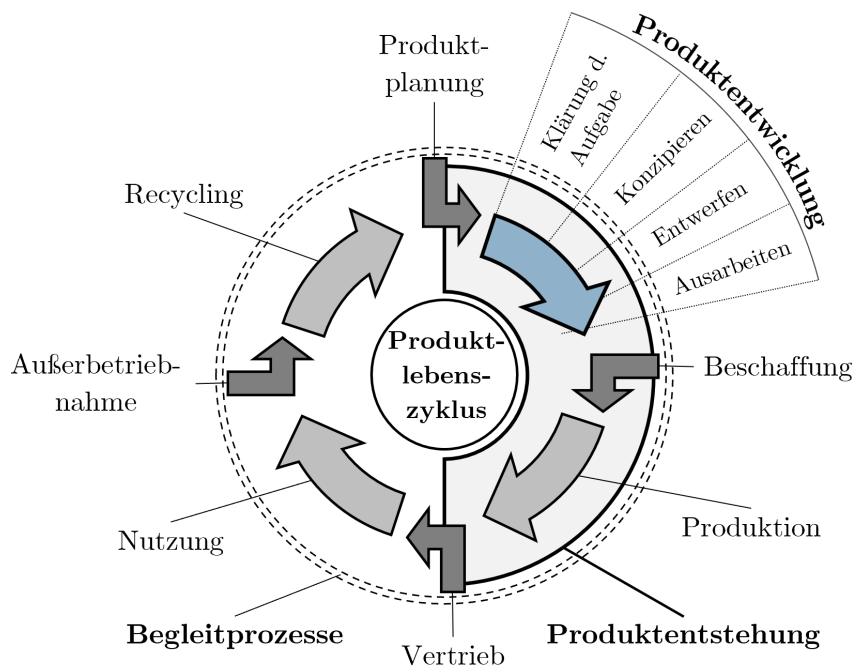
### 2.1 Produktentwicklung

Die *Produktentwicklung* ist ein zentraler Bestandteil der *Produktentstehung* und bildet entsprechend auch die Grundlage des *technischen Produktlebenszyklus* in Abbildung 2.1. Auf Basis von Rahmenbedingungen, insbesondere der Inhalts- sowie Zeit- und Kostenplanung des Produkts, die in der *Produktplanung* definiert werden, erfolgt in der Produktentwicklung die Erarbeitung von neuen Lösungen [Ger+21b, S. 27–28]. Hierbei wird eine Unterteilung in vier Hauptarbeitsphasen *Klärung der Aufgabe*, *Konzipieren*, *Entwerfen* und *Ausarbeiten* vorgenommen [Ger+21a, S. 66–70].

Grundsätzlich lässt sich das Vorgehen in der Produktentwicklung aus verschiedenen Perspektiven betrachten [Ger+21a, S. 57–59]. Diese heben jeweils einzelne Aspekte hervor, was zum besseren Verständnis beiträgt und die Ableitung von methodischem Vorgehen ermöglicht. Beispielsweise ist eine Darstellung als Problemlösung, Informationsverarbeitung, iterativer Prozess oder auch als Koevolution von Problem und Lösung möglich. Auch wenn die Aktivitäten der Produktentstehung oft sequenziell dargestellt werden, durchlaufen sie in der Praxis meist iterative und parallelisierte Prozesse [VDI 2221-1]. Neben den technischen Prozessen vervollständigen *Begleitprozesse*, wie beispielsweise Projektmanagement, Organisationsentwicklung, Patentwesen oder Risikomanagement, die Prozesslandschaft in einem Unternehmen [Ger+21a; ISO 15288, S. 66–70]. Auf der operativen Ebene werden entsprechend fachliche und organisatorische Schnittstellen benötigt.

#### 2.1.1 Produktentwicklungsprozesse

Produktentwicklungsprozesse beschreiben alle systematischen Abläufe, die mit der Produktentwicklung gemäß Abbildung 2.1 assoziiert werden. Unterschiedliche Informationsbedarfe verschiedener Stakeholder motivieren verschiedene Prozessmodelle, die auf verschiedenen Betrachtungsperspektiven des Entstehungs- bzw. Entwicklungsprozesses fußen [Ger+21a,



**Abbildung 2.1:** Produktentwicklung im Kontext des technischen Produktlebenszyklus, nach [Ger+21a, S. 59]

S. 57–58]. Betrachtungsperspektiven können dabei die Prozessvisualisierung, Prozessplanung oder auch die Prozessverbesserung sein [BFN06].

Gericke und Blessing [GB12] differenziert zwischen phasen- und aktivitätsbasierten Prozessmodelle. Aktivitätsbasierte Prozessmodelle gliedern den Entwicklungsprozess in einzelne, flexible Tätigkeiten wie Anforderungsanalyse, oder Ideengenerierung, die iterativ und in variabler Reihenfolge durchgeführt werden. Dadurch wird eine anpassungsfähige und praxisnahe Steuerung des Entwicklungsablaufs ermöglicht. Phasenbasierte Prozessmodelle hingegen strukturieren den Entwicklungsprozess in klar abgegrenzte, aufeinanderfolgende Abschnitte mit definierten Zielen und Ergebnissen, wie Klären der Aufgabe oder Entwerfen. Sie bieten damit eine übersichtliche und kontrollierbare Vorgehensweise, die besonders für Projekte mit klaren Anforderungen geeignet ist.

Eine Klassifikation und Einordnung verschiedener Prozessmodelle wird von Wynn und Clarkson [WC18] ausführlich diskutiert. Im Rahmen dieser Arbeit wird ein Überblick mit jeweils zwei Beispielen bereitgestellt. Die Klassifikation differenziert, wie in Tabelle 2.1 abgebildet, zwischen vier Modelltypen und drei Ebenen. Die *Mikroebene* setzt den Fokus auf einzelne Prozessschritte und deren unmittelbare Kontexte. Die *Mesoebene* bezieht sich auf den durchgängigen Ablauf der Aufgaben und Aktivitäten während der Entwicklung. Die *Makroebene* betrachtet die Projektstrukturen und oder den Entwicklungsprozess im übergeordneten Kontext. Die vier Modelltypen *abstrakte*, *prozedurale* und *analytische*

Modelle sowie Modelle der *Managementwissenschaften* und des *Operations Research* werden in den folgenden Subkapiteln kurz an repräsentativen Beispielen erläutert.

### 2.1.1.1 Abstrakte Prozessmodelle

Abstrakte Modelle vermitteln theoretische und konzeptionelle Erkenntnisse. Obwohl sie bedeutende theoretische Einsichten liefern und praktische Ansätze inspirieren, bieten sie nicht immer eine direkte Anleitung für Anwender [WC18].

Auf der **Mikroebene** befindet sich das erweiterte Function-Behaviour-Structure-Modell (FBS) [GK14], das sich auf situative Aspekte des Entwicklungsprozesses fokussiert. Das FBS basiert auf der Annahme, dass alle Systeme als Funktion (wofür ist es), Verhalten (was tut es) und Strukturen (was ist es) beschrieben werden können. Das Modell unterteilt den Entwicklungsprozess in 20 Transformationsprozesse, die zwischen den FBS-Domänen in der Außenwelt, der Interpretation des Ingenieurs und dem angestrebten Endprodukt verlaufen. Ein weiteres Beispiel auf der Mikroebene wird von Braha und Reich [BR03] vorgeschlagen, die die Entwicklung mit formalen mathematischen Operationen beschreiben. Auf der **Mesoebene** befindet sich die Theory of Domains [AHC15]. Diese ist ein Vorgehensmodell, das den mechanischen Entwurf in vier Domänen beschreibt: (1) Transformationsprozesse, die das Produkt in der Anwendung ermöglicht, (2) Funktionen, die die Transformation realisieren (3) die Organe, die diese Funktionen physisch ausführen, und (4) die physischen Teile des Produkts. In den Domänen werden Methoden und Modelle auf Basis der beiden Dimensionen Abstraktions- und Detailgrad eingeordnet. Ein weiteres Beispiel ist das Characteristics-Properties Modelling/Property-Driven Development Modell (CPM/PDD-Modell) [Web14], das die Integration von Computerwerkzeugen in den Entwicklungsprozess betrachtet. Auf der **Makroebene** ist das integrierte Produktentstehungsmodell (iPeM) [AB11] verortet, das einen Problemlösungszyklus mit einer phasenbasierten Sichtweise der Produktentwicklung kombiniert. Das Modell beschreibt ein Operationssystem, das Ziel- und Anforderungssysteme in Objektsysteme transformiert, und bietet eine Grundlage zur Ableitung situationsspezifischer Modelle für Produktentwicklungsprozesse unter Berücksichtigung von Dynamik und Einzigartigkeit. Beispielsweise ist eine Formulierung im Kontext der Produktgenerationsentwicklung möglich [Alb+16]. Die Capability Maturity Model Integration [CMM10] bietet ein Vorgehensmodell zur Beschreibung der Prozessbereiche in generischen Entwicklungsprojekten.







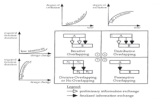
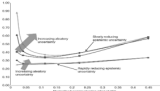
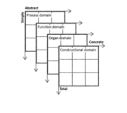

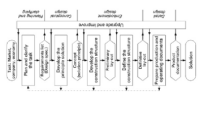

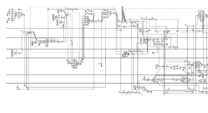
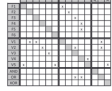
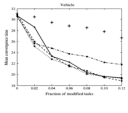
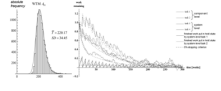
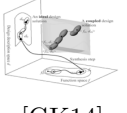
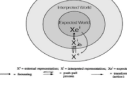

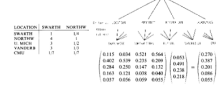
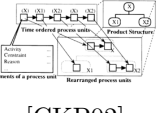
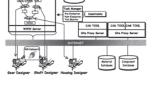
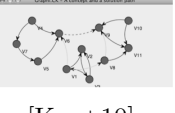
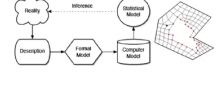
### 2.1.1.2 Prozedurale Prozessmodelle

Prozedurale Modelle dienen als Leitfäden für bewährte Praktiken und sollen reale Situationen unterstützen [WC18].

Auf der **Mikroebene** zählt hierzu hier der Plan-Do-Check-Act Zyklus. Dieser beschreibt einen iterativen Prozess, der aus einer Problemanalyse und Lösungsplanung (Plan), der

Umsetzung einer Lösung (Do), dem Einholen von Feedback (Check) sowie der Anpassung der Lösung (Act) besteht [MN10]. Auf derselben Ebene ist der Analytic-Hierarchy-Process [Saa87] verortet, der systematisch die Entscheidungsfindung unterstützt. Auf der **Mesoebene** befinden sich das im Maschinenbau weit verbreitete Phasenmodell nach Pahl et al. [Pah+07], das spezifische Arbeitsschritte für jede Phase vorgibt. Das Modell beschreibt die schrittweise Konkretisierung von abstrakten Anforderungen hin zu detaillierten technischen Lösungen und betont dabei die Bedeutung von Feedbackschleifen, um unerwünschte Nacharbeit zu minimieren. Ein weiteres Beispiel ist das Phasenmodell aus [VDI2221-2], in dem einzelne Ingenieursaktivitäten systematisch zeitlich aufgeplant und strukturiert werden.

**Tabelle 2.1:** Verschiedene Darstellungsformen des Produktentstehungsprozesses anhand von Betrachtungsebene und Modelltyp, nach [WC18]

	<b>Abstrakt</b>	<b>Prozedural</b>	<b>Analytisch</b>	<b>MS/OR*</b>
<b>Makro</b>	<p>[CMM10]</p>  <p>[AB11]</p> 	<p>[Pra96]</p>  <p>[VDI 2206]</p> 	<p>[Coo80]</p>  <p>[PT06]</p> 	<p>[KEW97]</p>  <p>[ST12]</p> 
<b>Meso</b>	<p>[AHC15]</p>  <p>[Web14]</p> 	<p>[Pah+07]</p>  <p>[VDI 2221-2]</p> 	<p>[WCC14]</p>  <p>[KL11]</p> 	<p>[BB07]</p>  <p>[SDS13]</p> 
<b>Mikro</b>	<p>[BR03]</p>  <p>[GK14]</p> 	<p>[MN10]</p>  <p>[Saa87]</p> 	<p>[TK99]</p>  <p>[CKP02]</p> 	<p>[ZY09]</p>  <p>[Kaz+10]</p> 

\* Managementwissenschaft/Operations Research

Auf der **Makroebene** ist das V-Modell [VDI2206] verortet. Dieses betont die Zerlegung eines komplexen Systems in einzeln entwickelte Subsysteme. Anschließend erfolgt die

Systemintegration, wobei die Verifikation und Validierung in Bezug auf Anforderungen auf verschiedenen Ebenen zentrale Bestandteile sind. Die Concurrent-Engineering Wheels [Pra96] betonen eine simultane, multidisziplinäre Arbeitsweise, um Schnittstellenprobleme frühzeitig zu identifizieren. Sie unterstützen die ganzheitliche Betrachtung des Entwicklungsprozesses, indem sie technische, wirtschaftliche und produktionstechnische Aspekte parallel integrieren.

### 2.1.1.3 Analytische Prozessmodelle

Analytische Modelle konzentrieren sich auf die Analyse einer spezifischen Situation, indem sie die Details abbilden, um Einblicke, Verbesserungen oder Unterstützung zu ermöglichen [WC18].

Auf der **Mikroebene** zählt hierzu die Engineering History Base [TK99]. Diese Wissensdatenbank ermöglicht die Begründung von Designattributen durch den Bezug zu Zielen und Einschränkungen vorheriger Entscheidungsprozesse. Ein weiteres Beispiel auf derselben Ebene ist das Manufacturing Integration and Design Automation System (MIDAS) [CKP02]. Dieses führt Ingenieure dynamisch durch den Entwurfsprozess, überführt Aufgaben auf hoher Abstraktionsebene schrittweise in detailliertere Prozessabläufe und schlägt alternative Strategien vor. Auf der **Mesoebene** sind Aufgabennetzwerke [KL11] verortet. Diese integrieren explizit Prozessmodelle, die den übergreifenden Ablauf von Aufgaben mit detaillierten Informationen zu anderen Domänen verknüpfen. Das reduziert Kommunikationsaufwände oder kann überflüssige Iterationen aufzeigen. Wynn, Caldwell und Clarkson [WCC14] beschreiben ein Prozessmodell, bei dem zentrale Eigenschaften von Aufgaben anhand von systematischen Regeln definiert werden. Dabei wird die Unsicherheit in Bezug zu Designinformationen mitberücksichtigt. Auf der **Markoebene** ist die Arbeit von Parry und Turner [PT06] verortet, die den Einsatz von visuellen Management Dashboards zur Darstellung des Arbeitsfortschritts und zur Identifikation von Prioritäten sowie Engpässen in Prozessen hervorhebt. Auf gleicher Ebene ist der Rework Cycle [Coo80] angeordnet, der die Auswirkungen von Einflussfaktoren und Rahmenbedingungen auf einen Entwicklungsprozess darstellt. Dieser zeigt, wie qualitative Probleme und der Einfluss externer Faktoren zu Verzögerungen und Produktivitätsverlusten führen können.

### 2.1.1.4 Management- und Operations-Research-Prozessmodelle

Modelle der Managementwissenschaft oder im Operations Research nutzen mathematische oder computergestützte Analysen von repräsentativen Fällen, um allgemein gültige Erkenntnisse zu Fragestellungen zu entwickeln [WC18].

Auf der **Mikroebene** zählt hierzu der Ansatz von Zeng und Yao [ZY09]. Die Autoren verwenden Computersimulationen, um die Auswirkungen verschiedener Designstrategien innerhalb ihrer axiomatischen Theorie zu untersuchen. Anhand der Generierung von FE-

Netzen wird gezeigt, dass unterschiedliche Ansätze zur Problemreformulierung, Synthese und Sequenzierung von Aufgaben zu unterschiedlichen Lösungen führen können. Kazakci et al. [Kaz+10] stellen ein Computermodell vor, das Entwicklungsprozesse simuliert, wobei Konzepte und Wissen durch mathematische Graphen repräsentiert werden. Der Fokus auf Konzeptentwicklung liefert schnellere Lösungen, während die Betonung der Wissensentwicklung robustere Ergebnisse ermöglicht. Auf der **Mesoebene** betrachten Braha und Bar-Yam [BB07] modellbasiert Informationsflüsse in Aufgaben-Netzwerken bei paralleler Bearbeitung unter der Berücksichtigung von interdependenten Aufgaben. Stark verbundene Aufgaben sind besonders anfällig für Iterationen, was die Konvergenz eines Designs erschweren kann. Auf gleicher Ebene betrachten Schlick, Duckwitz und Schneider [SDS13] ein spektrales Modell, das dynamische Komplexitäten in Projektprozessen abbildet und so Einblicke in die Stabilität iterativer Entwicklungsabläufe ermöglicht. Auf der **Makroebene** ist die Arbeit von Krishnan, Eppinger und Whitney [KEW97] verortet. Die Autoren entwickelten mathematische Strategien, um den optimalen Grad der Überlappung von Phasen und Informationsweitergabe zu bestimmen. Suss und Thomson [ST12] stellt mit dem Collaborative Process Model (CoPM) einen simulationsbasierten Ansatz vor, der Entwicklungsprozesse unter Berücksichtigung von Stage-Gate-Strukturen und Teamaktivitäten auf mehreren Ebenen modelliert. Dabei wird gezeigt, dass agile Ansätze wie Scrum [Sch04] bei hoher Unsicherheit effektiver sind als traditionelle, phasenbasierte Prozesse.

### 2.1.2 Produktgenerationsentwicklung

Die meisten Produkte und Systeme der heutigen Zeit werden in *Generationen* entwickelt [ABW15; AR22]. Typischerweise existieren die Produkte bereits auf dem Markt und zukünftige Generationen sind optimierte Versionen in Bezug auf spezifische Eigenschaften. Drei Grundprinzipien lassen sich hier unterscheiden: Die *Wiederverwendung* bestehender Lösungen aus Referenzprodukten, um den Entwicklungsaufwand zu minimieren; die *Variation* der Lösungsprinzipien zur Verbesserung des Produkts; und die *Einführung* grundlegend neuer und innovativer Lösungsprinzipien. Anteile dieser drei Grundprinzipien können zur Planung neuer Produkte verwendet werden [ABW15]. Folglich wird zur Entwicklung neuer Produkte ein Referenzprodukt, auf dem dieses aufbaut, benötigt. Dieses Referenzprodukt dient als Ausgangspunkt für die Weiterentwicklung und umfasst frühere Produkte des eigenen Unternehmens, Konkurrenzprodukte, deren Komponenten sowie relevante Konzepte, Teilmodelle und Produktdokumentationen. Teilsysteme dieses Produkts können wiederum Teile mehrerer bereits existierender Produkte sein. Bei der Übernahme in das zu entwickelnde Produkt ist es wichtig, Abhängigkeiten zwischen Referenzen miteinzubeziehen, da diese ihren Ursprung in unterschiedlichen Domänen haben können [Alb+19].

### 2.1.3 Situative Methodenauswahl

Auf Basis von allgemeinen Merkmalen, die Methoden in der Produktentwicklung beschreiben und kategorisieren, schlägt Zanker [Zan99] eine individuelle Anpassung und Anordnung der Methoden je nach Situation im Entwicklungsprozess vor. Die sechs eingeführten Merkmale der Methoden sind die (1) Struktur (verwendete Elemente und Gruppen), die Art der (2) Formulierung (Interaktionsform) und (3) Darstellung, benötigte (4) Hilfsmittel, das (5) Dokumentationsmedium sowie der (6) Anwenderkreis. Die Anpassung der Methoden bezieht sich auf die Variation der Methodenmerkmale. Zum Beispiel kann für die Kreativitätstechnik Brainstorming eine Anpassung des Teilnehmerkreises aufgrund der Verfügbarkeit von Personen vorgenommen werden. Ebenso kann durch die Variation der Merkmale auch eine Weiterentwicklung von Methoden realisiert werden. Durch die Veränderung des Trägermediums (sechs statt einem zentralen Arbeitsdokument) und einer Änderung der Struktur von Arbeitsschritten (fünf Weiterentwicklungsphasen statt einer Lösungsfindungsphase) entsteht aus dem konventionellen Brainstorming die Methode 6-3-5. Die Erweiterung dieser Grundüberlegungen, insbesondere um den Aspekt der Abhängigkeit in Bezug auf den Anwendungsfall, ist Teil der Arbeit von Braun und Lindemann [BL04]. Dabei werden fünf Anwendungskriterien aufgeführt: (1) Ressourcen (Informationen, Qualifizierung und Motivation von Beteiligten, Software, Zeit, etc.), (2) Situation (Priorität, Größe des Unternehmens, Partner, etc.), (3) Aufgabe (Anforderungen, Schwierigkeit, Ziele, etc.), (4) Transfer (Schulungs- und Beratungsaufwand, Vorbereitungszeit, etc.) und (5) weitere Charakteristiken (Modularität, Interaktion, Effektivität, etc.). In der Wechselwirkung zwischen Methode und Anwendungsfall gilt es sicherzustellen, dass die Abweichung bezüglich des Ziels, das durch die Anwendung der Methode erreicht wird, gegenüber den angestrebten Ergebnissen in erlaubten Grenzen bleibt. Ebenso muss die Abweichung zwischen den für die Methode erforderlichen Eingaben und den im Entwicklungsprozess bereitgestellten Informationen begrenzt sein.

Weitere Herangehensweisen, um Methoden auszuwählen, werden von Franke und Deimel [FD04] beschrieben. Die erste klassifiziert Methoden anhand ihrer grundlegenden Aktivitäten (vgl. [Zan99]: Kombinieren, Priorisieren, Analysieren, etc.) und ordnet diese anschließend den Arbeitsschritten zu. Arbeitsschritte können beispielsweise die Definition einer Funktionsstruktur oder die Suche nach Wirkprinzipien im Schritt des Konzipierens (vgl. Abbildung 2.1 auf S. 10) sein. Der zweite Ansatz stützt sich auf die strukturellen Eigenschaften einer Methode. Die Auswahl erfolgt durch den Abgleich der Merkmale einer Methode (vgl. [Zan99]) mit den Anforderungen eines Anwendungsfalls.

Albers et al. [Alb+15] diskutieren eine Auswahl von Entwicklungsmethoden im Kontext des in Abschnitt 2.1.1.1 auf S. 11 bereits erwähnten integrierten Produktentstehungsmodells (iPeM) [AB11]. Die Situation im Produktentstehungsprozess, um die der Ansatz

zentriert ist, wird auf Basis der durchzuführenden Aktivität, der angestrebten Ziele durch Anwendung der Methode (Kostenreduktion, Senkung der Fehlertoleranz, etc.) und der bereitgestellten Ergebnisse (Funktionsdokumente, Ressourcenplanung, etc.) festgelegt. Zur interaktiven Auswahl werden entsprechend der Situation und der Charakteristik des Anwendungsfalls alle hinterlegten Methoden aus der Produktentwicklung, dem Wissens- und Zukunftsmanagement durch einen Kongruenzwert bewertet. Dieser beschreibt die Eignung. Darüber hinaus wird eine Grobstruktur aus neun Punkten für Methodensteckbriefe vorgeschlagen: (1) Zusammenfassung, (2) Vorteile/Nachteile, (3) Kurzbeschreibung, (4) Input/Output, (5) wesentliche Arbeitsschritte, (6) Hilfsmittel, (7) alternative Methoden, (8) Quellen und Literatur sowie ein (9) verantwortlicher Experte. Neben Methodensteckbriefen können auch Kontrollfragen, Checklisten oder Prozesserklärungen Auswahlprozesse vereinfachen und Ingenieure in ihrer täglichen Arbeit unterstützen.

Oft unterscheiden sich Entwicklungsprozesse verschieden großer Organisationen sowie Branchen [VDI 2221-2]. Hinzu kommen individuelle Erfahrungen und Fachkenntnisse beteiligter Personen. Zur breiten Akzeptanz neuer Methoden müssen diese Aspekte miteinbezogen werden [Jän07].

## 2.2 Künstliche Intelligenz

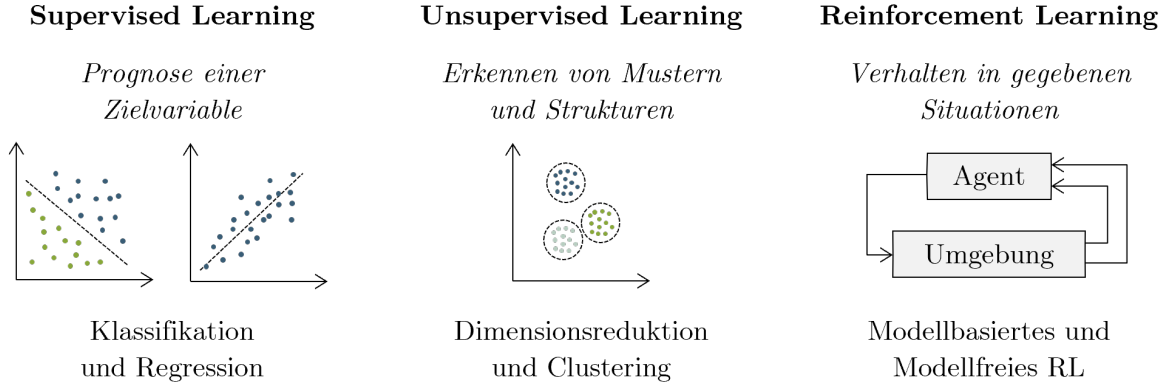
Seit den 1950er-Jahren gewinnt die KI in vielen Lebensbereichen zunehmend an Bedeutung. Im industriellen Kontext birgt sie große Potenziale zur Steigerung der Wettbewerbsfähigkeit, indem sie komplexe Prozesse optimiert, das Arbeitsleben erleichtert und bei wichtigen Entscheidungen unterstützt. Die dynamische Entwicklung des Forschungsfeldes, gerade in den letzten Jahren, lässt sich unter anderem auf die steigende Rechenleistung von Computersystemen und die Verfügbarkeit großer Datenmengen zurückführen [DTK20, S. 3, 24]. In der Wissenschaft gibt es bis heute noch keine einheitliche Definition von KI [Get+22, S. 8], wobei der Grundgedanke *Maschinen zu entwickeln, die sich intelligent verhalten und Aufgaben übernehmen, die Menschen zur Zeit noch besser können* an mehreren Stellen in der Literatur wiederzufinden ist [McC+55, S. 1–2][Ric83, S. 1–2][Get+22, S. 5–7]. Unterschieden wird dabei zwischen *schwacher* und *starker* KI [HBS22, S. 14]. Eine schwache KI ist auf einen klar abgegrenzten Anwendungsfall ausgerichtet, sodass ihre Übertragung auf andere Problemstellungen nur durch gezielte Anpassungen möglich ist. Eine starke KI hingegen ist in der Lage, sich selbst darüber hinaus weiterzuentwickeln. Darüber hinaus unterteilen Humm, Buxmann und Schmidt [HBS22, S. 15–18] das Feld anhand von den sechs folgenden menschlichen Fähigkeiten: (1) *Wahrnehmen*, (2) *Kommunizieren*, (3) *Lernen*, (4) *Wissen*, (5) *Denken*, (6) *Handeln*. Beispielsweise ist es möglich, durch Bildverarbeitungstechniken im Feld Computer Vision Tiere in einem Video zu klassifizieren (Wahrnehmen) oder basierend auf logisch programmierten Regeln Anfragen von Nutzern zu beantworten (Denken).

### 2.2.1 Maschine Learning

ML ist ein Teilgebiet der KI und betrachtet *Computersysteme, die die Fähigkeit besitzen aus vorhandenen Erfahrungen und Daten zu lernen ohne explizit dafür programmiert worden zu sein* [Gér19, S. 4]. Die Grundlage für ML bildet ein *Datensatz*  $\mathcal{D}$ , bestehend aus einzelnen *Datenpunkten*, mit denen die ML-Algorithmen interagieren. Unterschieden wird typischerweise zwischen strukturierten und unstrukturierten Daten [Tri24, S. 29–31]. Strukturierte Daten lassen sich tabellarisch auffassen. Entsprechend weisen alle Datenpunkte eine Konfiguration aller Merkmale auf. Ein *Merkmal* bezeichnet dabei eine Repräsentation des anzulernenden Sachverhalts und kann aus der Transformation von Rohdaten gewonnen werden. Ein Beispiel hierfür ist die Rückhaltekraft eines Sicherheitsgurts von einem Fahrzeug. Unstrukturierten Daten, beispielsweise Bilder oder Graphen, fehlt diese tabellarische Struktur. In einem Graphen können sich die Anzahl von Knoten und Kanten unterscheiden.

Abhängig davon, wie aus den Daten gelernt wird, werden nach [ISO 23053] im ML verschiedene Lerntypen unterschieden: *Supervised Learning*, *Unsupervised Learning* und

Reinforcement Learning (RL), die in den folgenden drei Abschnitten näher erläutert werden. Neben diesen drei Grundtypen, die zur Übersicht in Abbildung 2.2 abgebildet sind, existieren Untertypen, wie *Semi-Supervised-* oder *Ensemble-Learning*. Diese Formen stellen jedoch eine Mischform der beschriebenen Haupttypen dar. Für weitere Informationen wird der Leser an die Literatur verwiesen [Gér19; SB20; HBS22].



**Abbildung 2.2:** Lerntypen im Machine Learning, nach [ISO 23053]

### 2.2.1.1 Supervised Learning

Beim Supervised Learning werden dem Algorithmus *gelabelte* Daten zur Verfügung gestellt. Entsprechend gibt es, wie in Gleichung (2.1) aufgezeigt,  $N$  Eingabe-Ausgabe-Paare, bei denen die Eingaben  $x_i$  aus dem Merkmalsraum  $\mathcal{X}$  und die Ausgaben  $y_i$  aus dem Zielraum  $\mathcal{Y}$  stammen.  $M$  und  $O$  beschreiben dabei die Dimension der Ein- bzw. Ausgabe. Ein Beispiel für eine klassische Supervised-Learning-Problemstellung ist die *Prognose* des Kaufpreis eines Hauses auf Basis von dessen Lage und Größe.

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N, \quad x_i \in \mathcal{X} \subset \mathbb{R}^M, \quad y_i \in \mathcal{Y} \subset \mathbb{R}^O \quad (2.1)$$

Beim überwachten Lernen stehen dem Algorithmus Trainingsdaten  $\mathcal{D}^{\text{Training}} \subseteq \mathcal{D}$  zum Anlernen zur Verfügung. Nach Gleichung (2.2) besteht der Trainingsvorgang darin, einen Zusammenhang durch die Transformation  $f^S(x)$  zwischen Eingaben  $x$  und Ausgaben  $y$  durch Minimierung des Erwartungswerts einer Verlustfunktion  $\mathcal{L}$  herzustellen. Relevante Informationen werden so im ML-Modell gespeichert, was die Prognosen für ungesehene Eingaben ermöglicht. Danach wird das Modell mit ungesesehenen Daten  $\mathcal{D}^{\text{Validierung}} \subseteq \mathcal{D}$ , mit  $\mathcal{D}^{\text{Training}} \cap \mathcal{D}^{\text{Validierung}} = \emptyset$ , validiert. Oft werden zur Evaluierung sogar zwei ungesehene Datensätze verwendet, was die Anpassung von Hyperparametern des ML-Modells im Trainingsvorgang ermöglicht [ISO 23053].

$$f^S = \arg \min_f \mathbb{E}_{x,y} [\mathcal{L}(y, f(x))] \quad (2.2)$$

Im Fall einer Klassifikation weist die Ausgabe nur eine begrenzte Anzahl an festen Werten auf, wohingegen diese bei einer Regression einen beliebigen numerischen Wert annehmen kann.

Zur Evaluierung der Genauigkeit der ML-Modelle in Bezug auf die Prognosen werden Fehlerfunktionen eingeführt. Wird die Fehlerfunktion zur Optimierung des Modells verwendet, wird typischerweise der Begriff Verlustfunktion verwendet. Für Regressionsprobleme lassen sich der Mean Square Error (MSE) nach Gleichung (2.3), der Root Mean Square Error (RMSE) nach Gleichung (2.4) oder das Bestimmtheitsmaß ( $R^2$ ) nach Gleichung (2.5) nennen. Hierbei ist  $N$  die Anzahl betrachteter Datenpunkte,  $y$  der tatsächliche,  $\bar{y}$  der mittlere und  $\hat{y}$  der durch das ML-Modell prognostizierte Wert. Aufgrund des quadratischen Fehlers im Root Mean Square Error (RMSE), fallen starke Ausreißer höher ins Gewicht. Das Ergebnis lässt sich durch die Angabe in derselben Einheit wie die Zielvariable einfacher interpretieren. Das Bestimmtheitsmaß  $R^2 \in ]-\infty, 1]$  entspricht dem Anteil der Varianz der Ausgabe, die durch das ML-Modell erklärt wird. Ist  $R^2 < 0$  erklärt das ML-Modell weniger als die Mittelwertvorhersage.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N \|y_i - \hat{y}_i\|^2 \quad (2.3)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|y_i - \hat{y}_i\|^2} \quad (2.4)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N \|y_i - \hat{y}_i\|^2}{\sum_{i=1}^N \|y_i - \bar{y}\|^2} \quad (2.5)$$

Ebenfalls werden am Rande dieser Arbeit Klassifikationsprobleme betrachtet, deren Fehler im Fall von mehreren Klassen beispielsweise mit dem multinomialen Kreuzentropie-Fehler nach Gleichung (2.6) bestimmt werden kann. Hier steht  $f_k(x_i)$  für die Rohmodell-Ausgabe (Logit) der Klasse ( $k$ ). Dieser wird anschließend über die Softmax-Funktion in eine Wahrscheinlichkeitsvorhersage für eine bestimmte Klasse umgerechnet.

$$\text{MKL} = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K y_{ik} \log \left( \frac{\exp(f_k(x_i))}{\sum_{j=1}^K \exp(f_j(x_i))} \right) \quad (2.6)$$

Für Klassifikationsprobleme existieren zudem weitere Evaluationsmetriken. Die Accuracy in Gleichung (2.7) misst den Anteil korrekt vorhergesagter Datenpunkte an allen Vorhersagen. Hierfür werden die vier Größen TP (True Positives), TN (True Negatives), FP (False Positives) und FN (False Negatives) benötigt. Die Precision in Gleichung (2.8) gibt an, welcher Anteil der als positiv klassifizierten Datenpunkte tatsächlich positiv ist. Der Recall in Gleichung (2.9) beschreibt den Anteil korrekt erkannter positiver Daten-

punkte. Der F1-Score in Gleichung (2.10) ist das harmonische Mittel aus Precision und Recall. Für Problemstellungen mit mehreren Klassen lassen sich diese Kennzahlen durch ein One-vs-Rest-Schema für jede Klasse bestimmen und anschließend über alle Klassen zusammenfassen.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2.7)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.8)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.9)$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.10)$$

**Supervised-ML-Modelle**, die zum Verständnis der Arbeit relevant sind, werden im Folgenden kurz erläutert.

- **Extreme Gradient Boosting (XGBoost)** nach [CG16] basiert auf dem allgemeinen Prinzip des *Boostings*. Dabei werden mehrere schwache Modelle  $f_t \in \mathcal{F}$  (hier Decision Trees) sequentiell zu einem starken Modell kombiniert [FS99]. Die Definition erfolgt rekursiv unter Verwendung der Lernrate  $\eta$  nach

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i). \quad (2.11)$$

Für einen Datensatz  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$  ergibt sich die Gesamtprognose durch Addition der  $T$  schwachen Modelle:

$$\hat{y}_i = \hat{y}_i^{(0)} + \sum_{t=1}^T \eta f_t(x_i). \quad (2.12)$$

Beim *Gradient Boosting* werden die Modelle  $f_t$  so gewählt, dass sie den negativen Gradienten der Verlustfunktion approximieren. Die in XGBoost [CG16] eingeführten Erweiterungen steigern Stabilität und Effizienz des Verfahrens. In der Verlustfunktion nach Gleichung (2.13) wird eine explizite Regularisierung  $\Omega$  verwendet, die Modellkomplexität und Überanpassung kontrolliert:

$$\mathcal{L} = \sum_{i=1}^N l(y_i, \hat{y}_i) + \sum_{t=1}^T \Omega(f_t), \quad \Omega(f) = \gamma J + \frac{1}{2} \lambda \|w\|^2. \quad (2.13)$$

Hierbei entspricht  $J$  der Blattanzahl und  $w$  den Blattwerten.  $\gamma$  und  $\lambda$  sind die Regularisierungsparameter. Der Term  $\gamma J$  bestraft zu komplexe Bäume mit vielen Blättern, während der Term  $\lambda \|w\|^2$  große Blattwerte einschränkt.

In XGBoost werden zudem optimale Blattwerte  $w_j^*$  unter Einbezug der Ableitungen der Verlustfunktion bestimmt:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}. \quad (2.14)$$

Dabei sind  $g_i = \frac{\partial}{\partial \hat{y}_i} l(y_i, \hat{y}_i^{(t-1)})$  und  $h_i = \frac{\partial^2}{\partial \hat{y}_i^2} l(y_i, \hat{y}_i^{(t-1)})$  die erste bzw. zweite Ableitung des Einzelfallsverlusts für jeden Datenpunkt. Hierbei bezeichnet  $I_j$  die Menge der Trainingsdatenpunkte, die dem Blatt  $j$  zugeordnet sind. XGBoost nutzt für die Aktualisierung eine Taylor-Approximation zweiter Ordnung der Verlustfunktion, um die Änderung des Verlusts durch den neuen Baum lokal um die aktuelle Vorhersage  $\hat{y}_i^{(t-1)}$  abzuschätzen. So lassen sich optimale Blattwerte und die Verbesserung der Verlustfunktion (Gain) effizient berechnen. Diese Ableitungen sowie die Regularisierungsparameter dienen außerdem zur Berechnung des Gain, der angibt, wie stark die Verlustfunktion durch einen Split verbessert wird.

Die Modellarchitektur von XGBoost ist unabhängig von der Art der Aufgabe identisch; Unterschiede entstehen ausschließlich durch die Wahl der Verlustfunktion. Für Regression kann der Mean Square Error (MSE) nach Gleichung (2.4) und für Klassifikation die Kreuzentropie nach Gleichung (2.6) verwendet werden.

- Das **Tabular Prior-Data Fitted Network (TabPFN)** ist ein vortrainiertes Transformer-basiertes Modell, das in erster Linie für tabellarische Daten entwickelt wurde [Hol+25]. Im Unterschied zu klassischen ML-Modellen wird TabPFN nicht auf einen einzelnen Datensatz trainiert, sondern datensatzübergreifend auf synthetisch generierten Daten, die unterschiedliche Verteilungen und Prognoseaufgaben abbilden. Die zentrale Idee beruht auf dem Konzept des In-Context Learnings, bei dem Vorhersagen nicht für einzelne Datenpunkte, sondern unter Berücksichtigung eines Kontextes  $\mathcal{D}_C$  aus wenigen Beispieldatenpunkten getroffen werden [Don+24]. Formal kann das als bedingte Wahrscheinlichkeitsverteilung ausgedrückt werden:

$$P(y_{\text{new}} \mid x_{\text{new}}, \mathcal{D}_C) \approx f_{\theta}(x_{\text{new}}, \mathcal{D}_C), \quad (2.15)$$

wobei  $x_{\text{new}}$  ein neuer Eingabedatenpunkt,  $\mathcal{D}_C = \{(x_i, y_i)\}_{i=1}^K$  der bereitgestellte Kontext aus  $K$  Beispielen und  $f_{\theta}$  das vortrainierte Modell mit gespeicherten Parametern  $\theta$  darstellt. Während der Inferenz erhält TabPFN einen Datensatz bestehend aus gelabelten und ungelabelten Datenpunkten:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \cup \{x_j\}_{j=N+1}^{N+M}. \quad (2.16)$$

Das Modell nutzt in einem einzigen Durchlauf, dem sog. Forward-Pass, den Kontext der gelabelten Daten, um gleichzeitig die Vorhersagen für die ungelabelten Datenpunkte zu treffen:

$$\hat{y}_j = \arg \max_y P_\theta(y | x_j, \{(x_i, y_i)\}_{i=1}^N), \quad (2.17)$$

wobei dies einer Maximum-a-posteriori Vorhersage entspricht.

Der Forward-Pass übernimmt dabei die Rolle eines impliziten Bayes'schen Inferenzschrittes. Statt explizit die Posteriorverteilung

$$P(\theta | \mathcal{D}) \propto P(\mathcal{D} | \theta) P(\theta) \quad (2.18)$$

zu berechnen, wurde das Modell im Vorfeld so trainiert, dass es diese Inferenz direkt approximiert. Der Attention-Mechanismus in der Transformer-Architektur ermöglicht dabei die selektive Gewichtung von Beispielen aus dem Kontext  $\mathcal{D}_C$ . Weitere Details sind [Hol+23; Hol+25] zu entnehmen.

- Für alle weiteren ML-Algorithmen, die weniger zentraler Bestandteil der Arbeit sind und in erster Linie für zusätzlichen Kontext in der Evaluation verwendet werden, wird der Leser an die Literatur verwiesen. Hierzu zählen **Support Vector Regression** [Dru+96], **Multilayer Perceptron** [RHW86], **Decision Trees** [Bre+84] und **Lasso Regression** [Tib96].

### 2.2.1.2 Unsupervised Learning

Im Gegensatz zum Supervised Learning, werden zum Unsupervised Learning Datensätze *ohne explizite Labels* verwendet:  $\mathcal{D} = \{x_i\}_{i=1}^N$ ,  $x_i \in \mathcal{X} \subset \mathbb{R}^M$ . Ziel ist es *Strukturen* wie intrinsische Muster oder Cluster in des Datensatzes zu finden. Folglich kann hier keine Fehlerfunktion bezüglich einer Ausgabegröße definiert werden. Nach [ISO 23053] zählen Dimensionsreduktions- und Clusteringmethoden zu den Unsupervised-Learning-Methoden. Ein Beispiel ist das Separieren von Knickrichtungen des Längsträgers eines Fahrzeugs im Frontalaufprall.

Ziel einer **Dimensionsreduktion** nach Gleichung (2.19), ist es, durch eine Transformation  $f^U$ , die Daten in einen Raum mit geringer Dimension zu transformieren. Stark korrelierende oder redundante Merkmale werden so aus den Daten herausgerechnet. Allerdings gilt: Je geringer die Dimension dieser niedrigdimensionalen Repräsentation  $z$  ist, desto schwieriger ist es den Informationsverlust gering zu halten. Des Weiteren ist eine intuitive Interpretierbarkeit der Daten oft nicht mehr gegeben.  $f^U$  kann hierbei eine lineare oder nichtlineare Transformation sein.

$$z = f^U(x), \quad f^U : \mathcal{X} \rightarrow \mathcal{Z}, \quad x \in \mathcal{X} \subset \mathbb{R}^M, \quad z \in \mathcal{Z} \subset \mathbb{R}^K, \quad K < M. \quad (2.19)$$

Dimensionsreduktionsmethoden, die zum Verständnis der Arbeit relevant sind, werden im Folgenden kurz erläutert.

- **Principal Component Analysis (PCA)** [Jol02] ist ein lineares Verfahren zur Dimensionsreduktion, das darauf abzielt, die Varianz der Daten  $\mathcal{D} = \{x_1, \dots, x_N\}$  in einem niedrigdimensionalen Raum zu maximieren. Hierbei wird eine orthogonale Transformation konstruiert, die die ursprünglichen Merkmale auf neue, unkorrelierte Variablen, die sogenannten Hauptkomponenten, abbildet. Die Eigenwerte quantifizieren den Anteil der durch die jeweilige Hauptkomponente erklärten Varianz. Die  $k$ -te Hauptkomponente  $z_k$  ergibt sich als lineare Projektion der Daten auf den Eigenvektor  $u_k$  der Kovarianzmatrix  $\Sigma$ :

$$z_k = x^\top u_k, \quad \text{mit} \quad \Sigma = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^\top, \quad (2.20)$$

wobei  $u_k$  den  $k$ -größten Eigenwert von  $\Sigma$  trägt. Da die PCA die Daten auf lineare Achsen projiziert, die die maximale Varianz erklären, werden insbesondere die Hauptvariationsrichtungen der Daten in einem niedrigdimensionalen Raum abgebildet. Zur Vermeidung von Skalierungs- und Heteroskedastizitätseffekten kann zunächst eine logarithmische Transformation der Daten vorgenommen werden. Wird die PCA im logarithmierten Raum durchgeführt, wird dies als Logistic Principal Component Analysis (logPCA) bezeichnet [LL20].

- **Diffusion Maps Diffusion Maps (DiffMaps)** [CL06] ist eine nichtlineare Methode zur Dimensionsreduktion, die die zugrundeliegende Geometrie von Daten durch einen Diffusionsprozess auf einem Graphen erfasst. Gegeben sei ein Datensatz  $\mathcal{D} = \{x_1, \dots, x_N\}$ , der als Knoten eines gewichteten Graphen betrachtet wird. Die Gewichte  $w_{ij}$  beschreiben dabei die Ähnlichkeit zwischen Punkten  $x_i$  und  $x_j$ , beispielsweise über eine Gauß-Kernel-Funktion

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\epsilon}\right), \quad (2.21)$$

mit dem Skalenparameter  $\epsilon > 0$ . Aus der Gewichtsmatrix  $W$  wird die Markov-Übergangsmatrix  $P$  gebildet, die die Wahrscheinlichkeit beschreibt, von Punkt  $x_i$  zu  $x_j$  zu „diffundieren“:

$$P = D^{-1}W, \quad \text{mit} \quad D_{ii} = \sum_j w_{ij}. \quad (2.22)$$

Die Eigenvektoren und Eigenwerte von  $P$  definieren die Diffusionskoordinaten, die eine niedrigdimensionale Einbettung der Daten liefern und die intrinsische Geometrie der Datenstruktur bewahren. DiffMaps sind insbesondere geeignet, um nichtlineare Zusam-

menhänge und glatte mannigfaltige Strukturen in hochdimensionalen Datensätzen zu erkennen. Ein klassisches Beispiel hierfür ist die Swiss-Roll, deren intrinsisch zweidimensionale Struktur korrekt wiederhergestellt werden kann. Mannigfaltige Strukturen sind dabei niedrigdimensionale Formen in hochdimensionalen Daten, die die lokalen Nachbarschaften und Zusammenhänge der Daten so weit wie möglich erhalten. Beispielsweise bildet die Oberfläche einer Kugel im dreidimensionalen Raum eine zweidimensionale Mannigfaltigkeit, da sie lokal wie eine flache Ebene erscheint, global jedoch gekrümmt ist.

- **Uniform Manifold Approximation and Projection (UMAP)** [MHM18] ist ebenfalls eine nichtlineare Methode zur Dimensionsreduktion, das die zugrunde liegende mannigfaltige Struktur hochdimensionaler Daten modelliert. Gegeben sei ein Datensatz  $\mathcal{D} = \{x_1, \dots, x_N\}$ , auf dessen Basis ein gewichteter Graph konstruiert wird, der lokale Nachbarschaften der Datenpunkte repräsentiert. Die Kanten des Graphen werden mit Wahrscheinlichkeiten  $p_{ij}$  gewichtet, die die Ähnlichkeit zwischen Punkten  $x_i$  und  $x_j$  im hochdimensionalen Raum quantifizieren:

$$p_{ij} = \exp\left(-\frac{\|x_i - x_j\| - \rho_i}{\sigma_i}\right), \quad (2.23)$$

Dabei ist  $\rho_i$  der Abstand zum nächsten Nachbarn und  $\sigma_i$  ein adaptiver Skalenparameter. Anschließend wird über eine Verlustfunktion nach Gleichung (2.24) die niedrigdimensionale Einbettung so optimiert, dass die Struktur der lokalen Nachbarschaften aus dem hochdimensionalen Raum in der Einbettung möglichst gut erhalten bleiben:

$$\mathcal{L} = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) + (1 - p_{ij}) \log\left(\frac{1 - p_{ij}}{1 - q_{ij}}\right). \quad (2.24)$$

Die lokalen Nachbarschaften werden in der Einbettung durch

$$q_{ij} = \frac{1}{1 + \|z_i - z_j\|} \quad (2.25)$$

beschrieben. Die resultierenden Dimensionen der UMAP-Einbettung sind im Gegensatz zur PCA oder den DiffMaps nicht orthogonal oder unabhängig voneinander, sondern spiegeln die Optimierung der Erhaltung lokaler Nachbarschaften und der globalen Struktur wider.

Für die meisten klassischen Dimensionsreduktionsmethoden muss die Dimension des Einbettungsraums manuell festgelegt werden. Bei linearen Verfahren wie PCA erfolgt dies typischerweise durch die Analyse des Eigenwertspektrums der modellinduzierten Koordinaten: Ein deutlicher Abfall der Eigenwerte weist auf die intrinsische Dimensionalität der Daten hin. Hohe Eigenwerte repräsentieren dominante Strukturen, während niedrig

bewertete Dimensionen meist Rauschen oder lokale Variabilität erfassen. Die Wahl der Einbettungsdimension sollte daher so erfolgen, dass wesentliche Zusammenhänge bewahrt werden, ohne rauschbehaftete Informationen einzubeziehen.

Ein **Clustering** der Daten  $x$  in  $\mathcal{C} = \{1, \dots, K\}$  Gruppen mittels einer Transformation  $f^G$  nach Gleichung (2.26), kann ebenfalls aufschlussreich sein, um Strukturen in den Daten zu identifizieren. Diejenigen Datenpunkte mit ähnlichen Merkmalen und oder geringen Abständen im betrachteten Raum, werden automatisch einem Cluster zugeordnet. Im Gegensatz zur Klassifikation, die zu den Supervised-Learning-Methoden zählt, sind die Gruppen vor Ablauf des Verfahrens nicht bekannt. Die Ergebnisse sind jedoch davon abhängig, wie viele Cluster  $K$  berücksichtigt werden. Für Verfahren, wie das k-Means Clustering, müssen diese im Vorhinein definiert werden. Eine Möglichkeit das Clustering zu bewerten ist beispielsweise der Silhouette-Score [Rou87], der eine Aussage über die Abgrenzung der einzelnen Cluster trifft.

$$c = f^G(x) \quad f^G : \mathcal{X} \rightarrow \mathcal{C}. \quad (2.26)$$

Für das Verständnis der Arbeit ist insbesondere das k-Means Clustering relevant und wird daher im Folgenden kurz erläutert:

- **K-Means Clustering** dient zur Partitionierung eines Datensatzes  $\mathcal{D} = \{x_1, \dots, x_N\}$  in  $K$  disjunkte Cluster  $\{C_1, \dots, C_K\}$ . Ziel ist die Minimierung der Summe der quadrierten Abstände der Punkte  $x_i \in C_k$  zu den jeweiligen Clusterzentren  $\mu_k$  nach Gleichung (2.27). Dies entspricht der Minimierung der Varianz innerhalb der jeweiligen Cluster.

$$J = \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - \mu_k\|^2, \quad (2.27)$$

wobei  $\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$  das arithmetische Mittel der Punkte im Cluster  $C_k$  darstellt. Der Algorithmus arbeitet iterativ in zwei Schritten: Zunächst werden die Punkte den nächstgelegenen Clusterzentren zugewiesen, anschließend werden die Clusterzentren anhand der aktuellen Punktzugewisungen aktualisiert. Dieser Prozess wird wiederholt, bis eine Konvergenz der Zuordnungen oder Clusterzentren erreicht ist. Das k-Means Clustering eignet sich insbesondere für kugelförmige, gleichmäßig dichte Cluster, da die Minimierung der quadratischen Abstände implizit runde Cluster annimmt. Abweichungen von dieser Annahme, etwa stark unterschiedlich geformte oder dichte Cluster, können zu suboptimalen Partitionierungen führen.

### 2.2.1.3 Reinforcement Learning

RL beschäftigt sich mit Algorithmen, die, wie in Abbildung 2.2 angedeutet, *Entscheidungen* auf Basis von äußeren Einflüssen einer Umgebung treffen. Ein Beispiel ist das Spielen eines Brett- oder Videospiele wie Go oder StarCraft.

Durch das Handeln werden neue Informationen generiert und es kann so eine Aktionsstrategie ausgebildet werden, mit der die Anzahl an numerischen Belohnungen maximiert wird [SB20, S. 6–7, 58–60]. Die Grundlage für einen RL-Algorithmus bildet ein *Agent*, der mit einer Umgebung interagiert. Die Umgebung, welche eine Menge  $\mathcal{S}$  an *Zuständen*  $s$  annehmen kann, erhält *Aktionen*  $a \in \mathcal{A}$  von dem Agenten, worauf dieser den Zustand  $s_t$  observiert und eine *Belohnung*  $r_t \in \mathbb{R}$  für den  $t$ -ten Schritt erhält. Die Belohnungen werden typischerweise durch eine auf die Umgebung und die Problemstellung angepasste Funktion vergeben und können beispielsweise einer Punktzahl in einem Computerspiel entsprechen. Damit der Agent nun die Anzahl der Belohnungen maximieren kann, verfolgt dieser eine *Strategie*  $\pi \in \Pi$ , die kontinuierlich im Training verbessert wird. Mathematisch betrachtet ist die Strategie  $\pi$  nach Gleichung (2.28) und Gleichung (2.29) die Wahrscheinlichkeit für die Aktion  $A_t = a$  unter der Bedingung, dass der Zustand  $S_t = s$  vorliegt.

$$\pi : \mathcal{A} \times \mathcal{S} := \{(a, s) \mid a \in \mathcal{A}, s \in \mathcal{S}\} \rightarrow [0, 1] \quad (2.28)$$

$$\pi(a, s) = P(a|s) \quad (2.29)$$

Der RL-Agent schätzt wie gut die Durchführung von Aktion  $a$  in Zustand  $s$  unter Strategie  $\pi$  ist über die Aktionswertfunktion  $q_\pi(s, a)$  nach Gleichung (2.30). Die Aktionswertfunktion entspricht dabei der erwarteten Belohnung, die der RL-Agent versucht durch das Finden einer optimalen Strategie  $\pi^*$  nach Gleichung (2.30) und Gleichung (2.31) zu maximieren. Der Diskontierungsfaktor  $\gamma$  gewichtet dabei zukünftige Belohnungen. Ist die Anzahl der Schritte  $T$  endlich, liegt ein episodisches Problem vor – hier eignet sich  $\gamma = 1$ . Im Fall eines kontinuierlichen Problems ist  $T = \infty$  und es muss  $\gamma < 1$  gelten, damit die Reihe konvergiert. Für  $\gamma \rightarrow 1$  wird der Agent weitsichtiger, da zukünftige Belohnungen stärker zum Tragen kommen.

$$q_\pi(s, a) = \mathbb{E}_\pi \left[ \sum_{k=0}^T \gamma^k \cdot r_{t+k+1} \mid S_t = s, A_t = a \right] \quad \text{mit} \quad 0 \leq \gamma \leq 1, s \in \mathcal{S}, a \in \mathcal{A} \quad (2.30)$$

$$q_*(s, a) = \max_{\pi^* \in \Pi} q_{\pi^*}(s, a) \quad (2.31)$$

Die durch den RL-Agenten verfolgte Strategie versucht durch Observationen einen Zusammenhang zwischen dem herrschenden Zustand der Umgebung und den durchgeführten

Aktionen herzustellen. Die Bestimmung der nächsten Aktion wird in den meisten Fällen über ein neuronales Netzwerk realisiert, dem die Wahrscheinlichkeit, wann welche Aktion durchzuführen ist, antrainiert wird. Differenziert wird zwischen einer *Online*-Strategie und einer *Offline*-Strategie. Bei der Online-Strategie wird nur die letzte erlernte Strategie herangezogen. Bei der Offline-Strategie werden alle Erfahrungen, die der Agent gesammelt hat, in einem Zwischenspeicher abgelegt und zur Bestimmung der bestmöglichen Strategie mit verwendet.

Für das Verständnis der Arbeit ist insbesondere der Proximal Policy Optimization (PPO) RL-Algorithmus relevant und wird daher im Folgenden kurz erläutert:

- **Proximal Policy Optimization (PPO)** ist ein Policy-Gradient-Algorithmus, der entwickelt wurde, um eine stabile Optimierung von parametrischen Strategien  $\pi_\theta(a|s)$  in diskreten Aktionsräumen zu ermöglichen. Im Gegensatz zu klassischen Policy-Gradient-Methoden begrenzt PPO explizit die Änderung der Strategie pro Trainingsupdate, um divergentes Verhalten und Instabilitäten zu vermeiden. Die Optimierung erfolgt über die clipped-surrogate-objective-Funktion:

$$L_t^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[ \min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t) \right], \quad (2.32)$$

wobei  $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$  das Verhältnis der Wahrscheinlichkeiten für eine Aktion der aktuellen und der vorherigen Strategie angibt,  $\hat{A}_t$  den geschätzten Vorteil repräsentiert und  $\epsilon$  als Hyperparameter die maximale zulässige Strategie-Änderung steuert. Durch das Clipping werden große Sprünge der Strategie vermieden, während die Lernrate beibehalten wird, was zu höherer Stabilität in diskreten Aktionsräumen und komplexen, stochastischen Umgebungen führt.

Die Exploration wird durch einen skalierbaren Entropie-Term  $\beta H[\pi_\theta]$  in das Optimierungsziel integriert. Dieser Term belohnt eine höhere Unsicherheit der probabilistischen Strategie, wodurch die Auswahl von Aktionen nicht zu deterministisch erfolgt.

## 2.2.2 Interpretierbares Machine Learning

Ein Modell, das keine Interpretation der zugrundeliegenden Zusammenhänge ermöglicht, wird auch als *Blackbox* bezeichnet [Mol+22]. Der Begriff Interpretation bezieht sich hier auf die Nachvollziehbarkeit des Modellverhaltens und der Entscheidungsprozesse ausgehend von den Eingaben. Bei einfachen ML-Modellen, wie Decision Trees, ist eine inhärente Interpretationsfähigkeit durch die Modellstruktur gegeben. Sind die ML-Modelle hingegen komplex, sind die abgebildeten Zusammenhänge  $f$  nicht mehr ersichtlich. Das ist beispielsweise bei tiefen neuronalen Netzwerken der Fall. Warum die Nachvollziehbarkeit der Modelle relevant ist, zeigt ein Beispiel von Ribeiro, Singh und Guestrin [RSG16]: Ein

Klassifikationsmodell, das zwischen Wölfen und Hunden differenzieren soll, stützt seine Entscheidung auf Schnee im Hintergrund anstatt auf das Erscheinungsbild der Tiere selbst. Gerade für den produktiven Einsatz ist das Offenlegen solcher Zusammenhänge essenziell, um Verständnis und Akzeptanz zu schaffen [Stein+22].

Ein Überblick über verschiedene *modellagnostischen Interpretationsmethoden*, mit denen Einblicke in das Entscheidungsverhalten von ML-Modellen generiert werden können, ist in Tabelle 2.2 dargestellt. Diese lassen sich in zwei zentrale Kategorien unterteilen: Während *globale* Ansätze das Verhalten eines ML-Modells über den gesamten Datenraum hinweg beschreiben, konzentrieren sich *lokale* Ansätze auf die Analyse einzelner Prognosen. Darüber hinaus untersuchen *Merkmal-Effekt*-Methoden wie stark sich eine Änderung eines Eingabemerkmals auf die Prognose auswirkt, indem sie den funktionalen Zusammenhang zwischen Eingabe und Ausgabe aufzeigen. Im Gegensatz dazu quantifizieren *Merkmal-Wichtigkeit*-Methoden welchen Beitrag ein Eingabemerkmal zur Güte oder zur Varianz der Prognose einer Zielvariable leistet, ohne dabei die Richtung oder Art des Einflusses genauer darzustellen.

**Tabelle 2.2:** Überblick modellagnostische Interpretationsmethoden, nach [Mol+22]

Merkmal	Lokal	Global
<b>Effekt</b>	<ul style="list-style-type: none"> <li>• Individual Conditional Importance (ICI)</li> <li>• Local Interpretable Model-Agnostic Explanations (LIME)</li> <li>• Counterfactuals</li> <li>• Shapley Values</li> <li>• SHapley Additive exPlanations (SHAP)</li> </ul>	<ul style="list-style-type: none"> <li>• Partial Dependence Plots (PDP)</li> <li>• Accumulated Local Effects (ALE)</li> </ul>
<b>Wichtigkeit</b>	<ul style="list-style-type: none"> <li>• Individual Conditional Importance (ICI)</li> </ul>	<ul style="list-style-type: none"> <li>• Partial Importance</li> <li>• Permutation Feature Importance (PFI)</li> <li>• Shapley Additive Global Explanations (SAGE)</li> </ul>

Auf was bei der Verwendung von modellagnostischen Interpretationsmethoden zu achten ist, kann dem detaillierten Überblick von Molnar et al. [Mol+22] entnommen werden. Grundlegend zu beachten ist die korrekte Wahl eines ML-Modells, das die Datenstruktur adäquat abbildet. Im zweiten Schritt sind die Einschränkungen oder Begrenzungen der Interpretationsmethoden selbst zu berücksichtigen. In beiden Fällen müssen Abhängigkeiten in den Daten einbezogen werden.

Die genannten Methoden des interpretierbaren ML in Tabelle 2.2 können allgemein auch im Feld der *Sensitivitätsanalyse* eingeordnet werden [Stein+22]. Die Sensitivitätsanalyse befasst sich mit der Messung von Unsicherheiten in der Ausgabe basierend auf Änderungen in der Eingabe. In diesem Kontext lassen sich Methoden aus Tabelle 2.2 wie SHapley Additive Explanations (SHAP) aber auch die inhärent interpretierbaren ML-Modelle als *modellbasierte* Ansätze auffassen. Einblicke in die Ein- und Ausgabebeziehung werden dann über das trainierte ML-Modell generiert. Darüber hinaus lassen sich drei weitere Ansätze nennen. Hierzu zählen (1) *Varianz-basierte Methoden*, denen die Sobol Indizes [Sob01] oder der Fourier Amplitude Sensitivity Test (FAST) [Cuk+73] zugeordnet werden können. Ebenfalls zählen (2) *Ableitungsbasierte Methoden*, wie die Derivative-based Global Sensitivity Measures (DGSM) [SK10], die als Verallgemeinerung der ebenfalls dort verorteten Morris-Methode [Mor91] gesehen werden kann, dazu. Zuletzt sind (3) *Dichtebasierte Methoden* zu nennen, zu denen Derivative-based Global Sensitivity Measure (DELTA) [PBS13] und Probability and Weighted Normal (PAWN) [PW15] gehören. Dichtebasierte Sensitivitätsanalyse-Methoden betrachten die gesamte Wahrscheinlichkeitsdichtefunktion der Ausgabe, um die Sensitivität der Eingaben und deren Wechselwirkungen zu berechnen. Diese sind besonders geeignet, um Einschränkungen von varianzbasierter Analyse bei abhängigen Eingaben zu überwinden, können aber bei hoher Eingabedimensionalität oder langen Rechenzeiten schnell unpraktikabel werden. Varianzbasierte Methoden zerlegen die Varianz der Daten und treffen somit die Annahme, dass diese die Unsicherheiten vollständig beschreibt.

Neben der Analyse der Relevanz einzelner Merkmale betrachten Methoden der Sensitivitätsanalyse, wie die Sobol Indices [SK10], auch Interaktionseffekte. Die Bestimmung der Interaktionseffekte von Eingaben beliebiger Ordnung wurde für SHAP [LL17] beispielsweise erst vor kurzer Zeit erweitert [Mus+24].

Modellagnostische Methoden und Methoden der Sensitivitätsanalyse, die zum Verständnis der Arbeit relevant sind, werden im Folgenden kurz erläutert.

- **SHapley Additive exPlanations (SHAP)** ist eine theoretisch fundierte und modellagnostische Methode zur Erklärbarkeit von ML-Modellen, die auf den Shapley-Werten aus der kooperativen Spieltheorie basiert. Shapley-Werte geben den fairen Beitrag jedes Spielers zu einem kooperativen Spiel an; übertragen auf ML quantifizieren sie den Einfluss einzelner Eingabemerkmale  $x_j$  auf eine Ausgabe  $\hat{y}_i$ . Gegeben sei die Menge aller Merkmale  $F$  und ein ML-Modell  $f$ , das auf einem Datenpunkt  $x$  ausgewertet wird. Der Shapley-Wert eines Merkmals  $j$  ergibt sich aus dem durchschnittlichen marginalen Beitrag dieses Merkmals über alle möglichen Teilmengen  $S \subseteq F \setminus \{j\}$ :

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|! (|F| - |S| - 1)!}{|F|!} (v(S \cup \{j\}) - v(S)). \quad (2.33)$$

Die Funktion  $v(S)$  bezeichnet dabei die sogenannte *Wertfunktion*. Diese gibt an, welche erwartete Modellausgabe die Merkmalskoalition  $S$  erzielt. In der Modellagnostik wird  $v(S)$  typischerweise als marginale Erwartung über die fehlenden Merkmale definiert:

$$v(S) = \mathbb{E}_{X_{\bar{S}}} [f(x_S, X_{\bar{S}})], \quad (2.34)$$

wobei  $X_{\bar{S}}$  die Merkmale außerhalb von  $S$  bezeichnet. Diese Wertfunktion wird in **KernelSHAP** [LL17] über einen Hintergrunddatensatz approximiert. Aufgrund der exponentiellen Kosten  $\mathcal{O}(2^{|F|})$  bei der Auswertung aller Subsets schätzt KernelSHAP die Shapley-Werte durch eine gewichtete lineare Regression auf den Subset-Vorhersagen. Für eine Stichprobe von Subsets  $S$  wird das folgende gewichtet lineare Problem gelöst:

$$\min_{\phi_0, \dots, \phi_M} \sum_{S \subseteq F} w(S) (f_S(x_S) - \phi_0 - \sum_{j \in S} \phi_j)^2, \quad (2.35)$$

wobei  $w(S)$  die Shapley-Kernel-Gewichtung und  $\phi_0$  die erwartete Modellausgabe darstellen.  $f_S(x_S)$  bezeichnet die Modellvorhersage, bei der nur die Merkmale  $S$  gesetzt werden und alle übrigen Merkmale durch Hintergrunddaten ersetzt werden. KernelSHAP liefert dadurch effiziente, aber approximative Schätzungen der Shapley-Werte.

**TreeSHAP** [Lun+20] nutzt die Struktur von Entscheidungsbäumen, um Shapley-Werte exakt in polynomialer Zeit berechnen. Für einen Baum  $T$  beträgt die Zeitkomplexität  $\mathcal{O}(L D^2)$ , wobei  $L$  die Anzahl der Blätter und  $D$  die maximale Tiefe des Baums darstellt. Die Wahrscheinlichkeit, dass ein Pfad im Baum durch ein Merkmal  $j$  beeinflusst wird, wird rekursiv über alle Knoten aggregiert. Für einen Baum  $T$  mit Knoten  $v$  ergibt sich der Beitrag  $\phi_j$  als

$$\phi_j = \sum_{v \in \text{Knoten}(T)} \sum_{p \in \text{Pfade}(v)} P(p | j) \cdot \Delta v. \quad (2.36)$$

$P(p | j)$  gibt dabei die gewichtete Wahrscheinlichkeit an, dass das Merkmal  $j$  entlang des Pfades  $p$  aktiv ist.  $\Delta v$  ist die Änderung der Ausgabe am Knoten  $v$ .

Sowohl Kernel- als auch TreeSHAP ermöglichen eine transparente Attribution einzelner Merkmalsbeiträge. KernelSHAP ist universell für beliebige ML-Modelle  $f$  einsetzbar, TreeSHAP hingegen nur für Baumstrukturen. Dafür ist TreeSHAP besonders effizient und exakt. Grundlegend wird für beide Verfahren die Unabhängigkeit der Merkmale angenommen.

- **Sobol Indizes** [Sob01] sind ein globales Sensitivitätsmaß, das den Beitrag einzelner Eingangsmerkmale und deren Wechselwirkungen zur Varianz einer Modellprognose quantifiziert. Gegeben sei ein Modell  $f$  mit unabhängigen Eingangsvariablen  $x_j$ , dann ergeben sich die Anteile der totalen Varianz der Modellprognose durch die Sobol-Zerlegung nach

$$\text{Var}(y) = \sum_{j=1}^M V_j + \sum_{j < k} V_{jk} + \cdots + V_{1\dots M}. \quad (2.37)$$

Hierbei ist  $V_j = \text{Var}_{x_j}(\mathbb{E}[y | x_j])$  der Beitrag von Variable  $j$  zur Varianz.  $V_{jk}$  bezeichnen den Interaktionseffekt zwischen Variablen  $j$  und  $k$ .

Die **First-Order Sobol Indizes**  $S_j^I$  für Variable  $j$  werden definiert als

$$S_j^I = \frac{V_j}{\text{Var}(y)}, \quad (2.38)$$

und quantifiziert den Anteil der Varianz, der allein durch  $x_j$  erklärt wird. Der **Total Sobol-Index**  $S_j^T$  berücksichtigt zusätzlich alle Wechselwirkungen, an denen  $x_i$  beteiligt ist:

$$S_j^T = 1 - \frac{\text{Var}_{x_{\sim j}}(\mathbb{E}[y | x_{\sim j}])}{\text{Var}(y)}. \quad (2.39)$$

$x_{\sim j}$  entspricht dabei allen Merkmalen außer  $x_j$ . Im Vergleich zu SHAP geben Sobol Indizes eine globale Wichtigkeit an und keine lokalen Effekte.

- **Local Interpretable Model-agnostic Explanations (LIME)** [RSG16] ist eine modellagnostische Methode zur lokalen Erklärbarkeit von ML-Modellen. Ziel ist es, die Vorhersage eines komplexen Blackbox-Modells  $f$  für eine einzelne Instanz  $x_0$  durch ein einfaches, interpretierbares Modell  $g$  zu approximieren. LIME erzeugt dazu eine Menge von Perturbationen  $s \in \mathcal{S}$  um einen Datenpunkt  $x_0$ . Anschließend wird  $g$  trainiert, um die Vorhersagen von  $f$  in der Nähe von  $x_0$  zu reproduzieren. Dabei wird die Verlustfunktion

$$\mathcal{L}(f, g, w_{x_0}) + \Omega(g) \quad (2.40)$$

verwendet, die die Abweichung zwischen  $f(s)$  und  $g(s)$  durch die Nähe von  $s$  zu  $x_0$  mit  $w_{x_0}(s)$  gewichtet.  $\Omega(g)$  entspricht der Regularisierung, die die Interpretierbarkeit von  $g$  fördert. Formal wird die Optimierung über die lokal gewichteten Punkte  $s$  durchgeführt:

$$\min_{g \in \mathcal{G}} \sum_{s \in \mathcal{S}} w_{x_0}(s) (f(s) - g(s))^2 + \Omega(g). \quad (2.41)$$

Dabei ist  $\mathcal{G}$  die Menge der einfachen Modelle, wie beispielsweise Decision Trees. LIME liefert somit für jede einzelne Vorhersage eine lokale Merkmalsattribution, die die wichtigsten Einflussfaktoren auf die Entscheidung des Modells identifiziert. Durch die lokale Approximation auf  $x_0$  werden globale Nichtlinearitäten handhabbar, während die Interpretierbarkeit erhalten bleibt. Im Vergleich zu den Sobol Indizes liegen hier lokale Merkmal-Effekte vor. Für eine globale Attribution wie bei SHAP fehlt LIME jedoch das theoretische Fundament.

## 2.3 Auslegung crashbelasteter Systeme

Zur späteren Diskussion von datengetriebenen Ansätzen für die Optimierung der Entwicklungsprozesse in diesem Kontext, werden nun die Grundlagen im Bereich der Auslegung crashbelasteter Systeme besprochen. Der Fokus liegt hier insbesondere auf der passiven Fahrzeugsicherheit. Für das Verständnis der Arbeit sind die generellen Anforderungen und Lastfälle aber auch die eingesetzten virtuellen Auslegungsmethoden wie die FEM bedeutsam.

### 2.3.1 Passive Fahrzeugsicherheit

Die Fahrzeugsicherheit ist ein breites Feld, das sich in *aktive*, *integrale* sowie *passive Fahrzeugsicherheit* unterteilen lässt [Gon+21, S. 1074–1076]. Zur aktiven Sicherheit zählen die Systeme, die dazu dienen Unfälle zu verhindern. Dazu zählen beispielsweise der Spurhalteassistent und das Elektronische Stabilitätsprogramm (ESP). Für die passive Sicherheit sind die Systeme relevant, die die Unfallfolgen abmildern, wenn ein Aufprall nicht mehr vermieden werden kann. Hierzu zählen auch Systeme, die nach dem Unfall beispielsweise einen Notruf absetzen. Die integrale Fahrzeugsicherheit ist das Schnittfeld zwischen aktiver und passiver Sicherheit. Hierzu zählt beispielsweise die Vorkonditionierung von Gurten oder Sitzen sowie der Notbremsassistent.

*Passive Sicherheitssysteme*, die den *Insassenschutz* sicherstellen, sind im besonderen Fokus dieser Arbeit. Hierzu gehören nach [Gon+21, S. 1089–1093] die folgenden Systeme:

- Die *Fahrzeugstruktur* deformiert bei frontalen, seitlichen und rückwärtigen Kollisionen sowie Überschlägen plastisch und nimmt damit die Energie des Aufpralls auf.
- Die *Insassenzelle* beinhaltet die Sicherheitszelle des Fahrzeugs, die im Crashfall für bestmöglichen Schutz integer bleibt, sowie das darin befindliche Interieur, mit dem die Insassen im Crashfall in Kontakt kommen können.
- *Rückhaltesysteme* beinhalten Systeme, die dafür sorgen, dass der Insasse während dem Crash bestmöglich auf seiner Position verweilt und dabei möglichst gleichmäßig abgebremst wird. Hierzu zählen insbesondere Sicherheitsgurte, Airbags und Sitze.

Bezogen auf den *Partnerschutz* für Radfahrer oder Fußgänger gibt es weitere Sicherheitssysteme, wie beispielsweise eine aktive Motorhaube. Diese werden im Rahmen der vorliegenden Arbeit nicht näher betrachtet, auch wenn sie grundsätzlich nach derselben Vorgehensweise ausgelegt werden.

#### 2.3.1.1 Anforderungen

In der Entwicklung neuer Fahrzeuge und deren passiver Sicherheitssysteme gilt es, verschiedene, typischerweise widersprüchliche Anforderungen aus Design-, Package- sowie Gewichtsvorgaben mit den Sicherheitsanforderungen zu vereinen. Hierbei ist das Design

in erster Linie von den Herstellern selbst festzulegen. Im Gegensatz dazu werden die Sicherheitsanforderungen durch marktspezifische Gesetze oder Verbraucherschutzregularien, die durch physische Crashtests nachzuweisen sind, vorgegeben. Gesetzliche Regelungen werden durch den FMVSS oder die ECE UN-R für den amerikanischen bzw. europäischen Markt spezifiziert. Zu den Verbraucherschutzorganisationen zählen das IIHS oder das EuroNCAP. Letztere Regelungen sind nicht bindend, jedoch wird die Sicherheit nach einem Buchstaben- bzw. Sternesystem bewertet, was Kunden bei der Kaufentscheidung unterstützen soll. Unterschiedliche *Lastfälle* werden in standardisierten Testverfahren abgeprüft und dienen dabei der reproduzierbaren Nachbildung des realen Unfallgeschehens unter Laborbedingungen. Das ermöglicht auch den Vergleich verschiedener Fahrzeuge diverser Hersteller. Betrachtet werden Kollisionen mit deformierbaren und starren Barrieren sowie Pfählen in verschiedenen Winkeln und Überdeckungen. Eine für die Praxis gute Übersicht diverser Regularien kann dem Safety Companion [Reu25] entnommen werden.

### 2.3.1.2 Anthropomorphe Testpuppen

Zur Bewertung der Verletzungswahrscheinlichkeit für reale Insassen werden in den Labortests ATD verwendet. Diese physischen Modelle bilden in Proportionen, Masse, Größe und Bewegungskinetik einen menschlichen Körper ab. Verschiedene ATD-Typen – optimiert nach Kollisionsrichtung, Geschlecht, Größe und Alter – ermöglichen eine bestmögliche Biofidelität in Abhängigkeit des betrachteten Lastfall. Die Regularien geben dabei einen entsprechenden ATD-Typ vor. Während für den Frontalaufprall gegen eine starre Wand nach FMVSS 208 auf Fahrer- und Beifahrerposition sowohl der Hybrid III 50 % ATD-Typ (H3) als auch die Hybrid III 5 % ATD-Typ (HF) gefordert werden, muss für den Seitenaufprall nach EuroNCAP mit deformierbarer Barriere der WorldSid 50 % Mann auf der Fahrerposition sowie der Q6 (sechs Jahre altes Kind) und der Q10 (10.5 Jahre altes Kind) auf der Rückbank im Fahrzeug positioniert werden. Zur realitätsnahen Abbildung und Bewertung wurden im Laufe der Zeit zahlreiche ATD mit immer besserer Biofidelität entwickelt.

Die ATDs enthalten Sensormesstechnik, um Belastungen während des Crashes in Form von Kräften, Momenten, Wegen und Beschleunigungen aufzuzeichnen. Diese müssen für eine erfolgreiche Zulassung des Fahrzeugs unterhalb der in den Regularien festgelegten Grenzwerte der Verletzungskriterien liegen [Gon+21]. Die Bestimmung biomechanisch fundierter Verletzungskriterien wie dem Head Injury Criterion (HIC) oder dem Neck Injury Criterion (nij) erfolgt in internationalen Forschungsgruppen, wie beispielsweise der International Research Council on Biomechanics of Injury [IRCOBI98] [Gon+21, S. 1077]. Beispiele für die gesetzlichen Grenzwerte der Verletzungskriterien beim Frontalaufprall in Europa nach ECE UN-R94 und -R137 sind für die Kopfbeschleunigung 80 g sowie ein HIC von 1000. Zur Verdeutlichung für die in Abschnitt 2.3.1.1 und Abschnitt 2.3.1.2

diskutierten Anforderungen und ATDs ist in Abbildung 2.3 ein Seitenaufprall mit einer deformierbaren Barriere – hier 1400 kg schwer und 60 km/h schnell – dargestellt.



**Abbildung 2.3:** Seitenaufprall Porsche Taycan und zugehöriger Versuchsaufbau, nach [EuroNCAP] und [Reu25, S. 26]

### 2.3.2 Virtuelle Systemauslegung

Auch wenn zur Endabnahme, wie in Abbildung 2.3 gezeigt, physische Crashtests erforderlich sind, findet die Entwicklung vorwiegend im Virtuellen statt [Sch20, S. 21, 40–43]. Numerische FE-Simulationen bieten in der Entwicklung eine Plattform, um Konzepte schnell zu evaluieren und zu optimieren. Darüber hinaus können gleichzeitig Zeit- und Ressourcenaufwand für teure physische Crashtests an einzelnen Bauteilen oder vollständigen Fahrzeugprototypen reduziert werden. Dementsprechend dienen die physischen Versuche auch dazu die Prognosegüte der FE-Simulationen sicherzustellen. Aufgrund detailliert modellierter und großer FE-Modelle, betragen die Rechenzeiten in der Praxis zwischen vier und 33 Stunden [BRK21], was Kosten zwischen 400 US-Dollar bis hin zu mehreren tausend US-Dollar pro Simulation verursacht [SHT09]. Hinzu kommen Kosten für Aufbau und Verwaltung solcher virtuellen Modelle. Im Kontext der Kosten eines physischen Crashtests von 300,000 US-Dollar [SHT09] sind die Simulationskosten vergleichsweise gering, dennoch aber nicht vernachlässigbar, was das Bestreben nach weiterer, oft datengetriebener, Prozessoptimierung deutlich macht [Koh+21]. Die Methoden zur virtuellen Systemauslegung und -Entwicklung werden gegenwärtig mit dem Überbegriff Computer-Aided Engineering (CAE) zusammengefasst [Kle15, S. 3].

#### 2.3.2.1 Grundlagen der Finite-Elemente-Methode

Der simulativen Systemauslegung liegt die FEM zugrunde. Diese ist ein *Näherungsverfahren* zur Lösung physikalischer und ingenieurwissenschaftlicher Problemstellungen mit Hilfe

mathematischer Modelle [Wag22, S. 1–2]. Die Grundidee der Methode ist die *Zerlegung kontinuierlicher Körper* in endliche (finite) Elemente [Sch20, S. 22–24]. Das eigentlich kontinuierliche Verschiebungsfeld wird durch die geometrische Diskretisierung approximiert. Für jedes einzelne Element lässt sich mit Hilfe einer Ansatzfunktion und einer Materialformulierung ein integraler Ausdruck über das Gebiet des FEs aufstellen. Diese lokalen Gleichungssysteme werden anschließend zu einer Systemmatrix assembliert, da die Kontinuitätsbedingung durch gemeinsame Knoten benachbarter Elemente erfüllt ist. Die Systemmatrizen werden im nächsten Schritt zum Aufbau eines globalen Gleichungssystems verwendet, das durch den FE-Löser im Berechnungsprozess gelöst wird. Lösen bedeutet in diesem Kontext die Ermittlung der unbekanntenen Zustandsvariablen, insbesondere der Knotenverschiebung. Die Dehnungen im FE wiederum werden aus den räumlichen Ableitungen des über die Ansatzfunktionen interpolierten Verschiebungsfeldes ermittelt. Anschließend ermöglichen die Dehnungen, unter Anwendung der Materialgesetze, die Berechnung der mechanischen Spannung im FE.

Die für diese Arbeit relevante *nichtlineare dynamische Strukturanalyse* ist die Systemgleichung (allgemeine Bewegungsgleichung) in Gleichung (2.42) formuliert. Hierbei ist  $u$  der Systemverschiebungsvektor der Knoten,  $M$  die Massenmatrix,  $D$  die Dämpfungsmatrix,  $K$  die Steifigkeitsmatrix und  $P^{\text{ext}}$  beschreibt die äußeren Kräfte [Wag22, S. 256–281]. Nichtlinearitäten im Systemverhalten können materiell und geometrisch oder strukturell begründet sein [Mil13, S. 37–40]. Die Größen sind aufgrund der dynamischen Problemstellung von der Zeit  $t$  und im Fall von Nichtlinearität zusätzlich von  $u$  abhängig. Entsprechend ist die Bilanzierung auf dem undeformierten Zustand nicht mehr hinreichend. Für komplexe Strukturen ist die Dämpfungsmatrix nicht exakt bekannt und wird daher geschätzt.

$$M\ddot{u}_t + D\dot{u}_t + Ku_t = P_t^{\text{ext}} \quad (2.42)$$

Die zeitabhängige Lastaufbringung erfolgt durch Diskretisierung des Zeitverlaufs, was als Zeitintegration bezeichnet wird. Diese kann *implizit* oder *explizit* erfolgen. Der Unterschied liegt darin, wie der Übergang von einem Zeitpunkt  $t$ , an dem alle Systemgrößen als bekannt vorausgesetzt werden, zu einem darauf folgenden Zeitpunkt  $t + \Delta t$ , an dem die Systemgrößen gesucht werden, ermittelt wird. Im expliziten Zeitintegrationsverfahren werden für den Übergang ausschließlich Systemmatrizen zum Zeitpunkt  $t$  verwendet. Die Beschleunigungen  $\ddot{u}$  und Geschwindigkeiten  $\dot{u}$  werden an einem Punkt innerhalb eines Zeitinkrements als konstant angenommen. So können Beschleunigungen und Geschwindigkeiten für jedes Zeitinkrement  $\Delta t$  als Größe der Verschiebung ausgedrückt werden. Dies erfolgt für explizite Verfahren in LS-DYNA [Ans25] über die zentrale Differenzenmethoden nach Gleichung (2.43) und Gleichung (2.45) [Kle15, S. 236–237].

$$\dot{u}_t \approx \frac{1}{2\Delta t}(u_{t+\Delta t} - u_{t-\Delta t}) \quad (2.43)$$

$$\ddot{u}_t \approx \frac{1}{\Delta t^2}(u_{t+\Delta t} - 2u_t + u_{t-\Delta t}) \quad (2.44)$$

Anschließendes Einsetzen in Gleichung (2.42) bei Vernachlässigung der Dämpfung, liefert:

$$\ddot{u}_{t+\Delta t} = \Delta t^2 \cdot M^{-1} \cdot (P_t^{\text{ext}} - Ku_t) + 2u_t - u_{t-\Delta t} \quad (2.45)$$

Um den Ressourcenaufwand weiter zu reduzieren, wird die Massenmatrix zwecks leichterer Invertierbarkeit typischerweise diagonalisiert. Damit das explizite Zeitintegrationsverfahren stabil ist, ist erforderlich, dass das Zeitinkrement  $\Delta t$  kleiner als die Zeit ist, die eine elastische Spannungswelle braucht, um die Strecke entlang der kürzesten Elementkante zurückzulegen [Kle15, S. 237–240].

Im Gegensatz dazu sind implizite Zeitintegrationsverfahren unbedingt stabil, da in jedem Zeitschritt das Kräftegleichgewicht iterativ hergestellt werden muss. Dies ist auf die Abhängigkeit der Systemmatrizen von der gesuchten Größe  $u_{t+\Delta t}$  zurückzuführen. In jeder Iteration ist daher eine rechenintensive Invertierung, insbesondere der Steifigkeitsmatrix  $K_{t+\Delta t}$ , notwendig. Auch wenn implizite Verfahren die Strukturantwort tendenziell besser abbilden, sind diese aus wirtschaftlicher Sicht für Crashsimulationen nur bedingt geeignet. Die Approximation des Gleichgewichtspfades mit expliziter Zeitintegration ist typischerweise hinreichend genau. Hinzu kommen potenzielle Instabilitäten in impliziten Verfahren bei hochgradiger Nichtlinearität, wie sie in Crashsimulationen auftreten.

### 2.3.2.2 Anwendung der Finite-Elemente-Methode

In der Anwendung lässt sich der Simulationsprozess nach Abbildung 2.4 in drei Hauptprozessschritte – *Preprocess*, *FEM-Löser* und *Postprocess* – untergliedern. Im *Preprocess* wird das FE-Modell aufgebaut. Hier werden die Systemmatrizen durch Diskretisierung der Geometrie und Zuweisung von Materialien aufgestellt. Darüber hinaus werden Rand- und Kontaktbedingungen sowie Ausgabedaten und -formate und die Art der Analyse (implizit oder explizit) oder auch der FEM-Löser definiert. Anforderungen oder Richtlinien aus dem Entwicklungsprozess können Grenzwerte sein, die für den *Postprocess* relevant sind, oder maximale Elementgrößen. Nach dem Lösen des Problems – hier mit LS-DYNA [Ans25] – liegen die Daten typischerweise binär vor und können mit Hilfe von Analysetools im Rahmen vom *Postprocess* visualisiert werden. Abschließend werden die Simulationsdaten in einem Datenmanagementsystem archiviert und dienen als Referenzen zu neu durchgeführten Simulationen. Hierbei liegt der hohe Rechenaufwand in erster Linie im FEM-Löser, die größte Ingenieursarbeit in Form von Personenstunden jedoch in *Pre-* und *Postprocess*

[Kle15, S. 7]. Unterstützt wird der Einsatz von der FEM durch die in Abbildung 2.4 dargestellten Begleitprozesse.

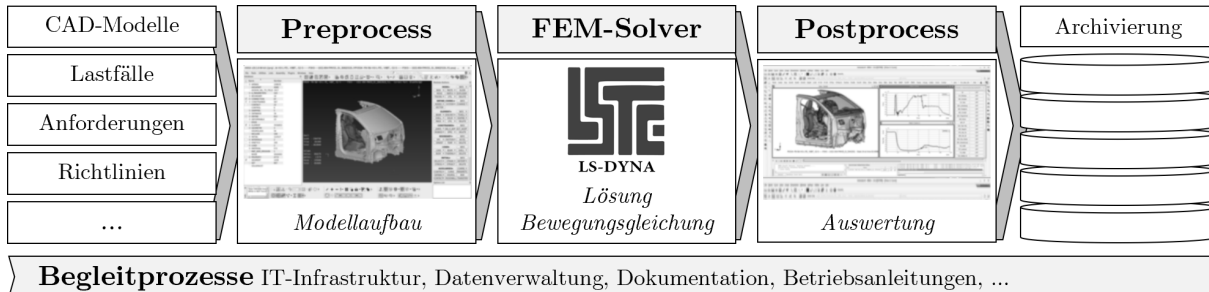


Abbildung 2.4: FEM-Prozess, nach [Kle15, S. 4–7] und [GI15]

Allgemein ist in Bezug auf die FEM jedoch zu beachten, dass die Lösung nur so exakt wie die zugrunde liegenden Annahmen ist, die durch die Ingenieure festgelegt werden. Hierzu zählen unter anderem verwendete Materialmodelle, die Kinematik, Elementgrößen, Elementtypen sowie die zeitliche Diskretisierung [Kle15, S. 8–11]. Phänomene, die nicht vorher explizit modelliert worden sind, werden nicht berücksichtigt. Gerade diese Aspekte sind auf der Datenebene, auf der sich diese Arbeit in erster Linie fokussiert, ebenfalls zu berücksichtigen.

Neben der für die vorliegende Arbeit in erster Linie relevante nichtlineare dynamische Strukturanalyse, lassen sich auch eine Vielzahl weiterer strukturmechanischer Problemstellungen lösen. Hierzu gehören beispielsweise Steifigkeits- oder Schwingungsanalysen, aber auch gekoppelte Problemstellungen, wie die zusätzliche Berücksichtigung des Wärmeeinflusses.

### 2.3.2.3 Optimierung von mechanischen Systemen

Im mathematischen Sinne bezieht sich der Begriff **Optimierung** auf die Ermittlung von einer Eingabe  $x$  eines Systems, durch die eine festgelegte *Zielfunktion minimiert* bzw. maximiert wird. Sollen, wie in Gleichung (2.46) dargestellt, die Werte von  $K$  Zielfunktionen  $f$  gleichzeitig optimiert werden, liegt ein mehrkriterielles Optimierungsproblem vor [Pie17, S. 44–45]. Die Menge der zulässigen Lösungen, bezeichnet durch  $\mathcal{X}$ , erfüllt die Nebenbedingungen in Form von Ungleichungen  $g_i$  und Gleichungen  $h_j$ . Sofern für keine andere zulässig Lösung  $x \in \mathcal{X}$   $f(x) \leq f(x^*)$  gilt, wird die Lösung  $x^* \in \mathcal{X}$  als pareto optimal bezeichnet. Die Definition von Nebenbedingungen hat großen Einfluss auf das Ergebnis, da durch eine (zu starke) Beschränkung die Möglichkeit besteht kein Optimum mehr aufzufinden.

$$\min_{x \in \mathbb{R}^M} f(x) = (f_1(x), f_2(x), \dots, f_K(x))$$

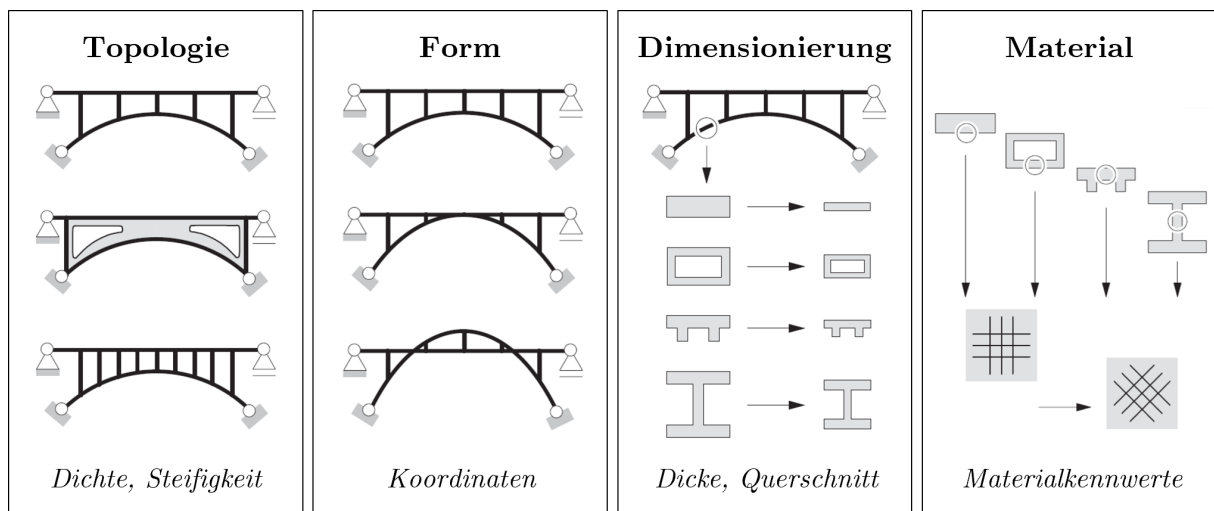
sodass  $x \in \mathcal{X}$ , mit

(2.46)

$$\mathcal{X} := \{x \in \mathbb{R}^M : g_i(x) \leq 0, i = 1, 2, \dots, h_j(x) = 0, j = 1, 2, \dots\} \subset \mathbb{R}^M.$$

Mit Bezug auf die passive Fahrzeugsicherheit (vgl. Abschnitt 2.3.1 auf S. 33) sind solche Nebenbedingungen beispielsweise durch die im Gesetz festgelegten Grenzwerte für die Insassenbelastung gegeben. Werden die Grenzwerte erfüllt, gilt es die Belastungen so weit wie möglich weiter zu reduzieren. Dementsprechend werden Nebenbedingungen auch als harte und die Zielfunktion als weiche Kriterien bezeichnet.

Typen der mechanischen **Strukturoptimierung** werden nach Art der Eingaben unterteilt, da so auch die Lösungsstrategien auszuwählen sind [Sch20, S. 4–5]. Ein Überblick ist in Abbildung 2.5 gezeigt. Bei der *Topologieoptimierung* wird der optimale konzeptionelle Aufbau einer Struktur durch Lage, Anzahl und Anordnung einzelner Elemente ermittelt. Parameter sind hier die Dichte oder die Steifigkeit der Struktur. Im Fall der *Formoptimierung* werden die inneren und äußeren Berandungskanten bzw. -flächen einer bereits festgelegten Topologie optimiert. Zu den Parametern gehören insbesondere die Koordinaten der Struktur. Die Optimierung der *Dimensionierung* ist ein Sonderfall der Formoptimierung und fokussiert sich auf die Bestimmung von Querschnitten und Wandstärken. Eine *Materialoptimierung* bezieht sich auf die Auswahl von Materialkennwerten, aber auch die des Lagenaufbaus im Fall von Kompositmaterialien.



**Abbildung 2.5:** Verschiedene Arten der mechanischen Strukturoptimierung inklusive Beispiele für Entwurfsvariablen, nach [Sch20, S. 4–5] und [Büt22, S. 54]

Zur automatischen Optimierung crashbelasteter mechanischer Strukturen in praktischen Anwendungsfällen existieren in der Literatur verschiedene Ansätze. Eine Topologieoptimie-

rung ist beispielsweise mit Graphen- und Heuristikbasierten Ansätzen möglich und kann sowohl für zwei- und dreidimensionale Strukturen herangezogen werden [Sch20; TSZ24; SB25, S. 290–294]. Des Weiteren gibt es Ansätze wie die *Response Surface Methode*, die iterativ anhand mehrerer Stützstellen Subräume des betrachteten Designraums analysiert und so den Pfad zu einem Optimum finden kann [Thi+06; Hor17]. Hierbei kann es jedoch dazu kommen, dass das Verfahren in lokalen Optima verbleibt. Im Rahmen dieser Arbeit ist vor allem die **metamodellbasierte Strukturoptimierung**, wie sie in der Arbeit von [Büt22; Büt+23] zur Dimensionierung erweitert und angewendet wird, von Bedeutung. Grundlegende Prozessschritte sind in Abbildung 2.6 dargestellt. Basierend auf Simulationsdaten, die systematisch durch ein Sampling im Designraum verteilt werden, wird ein Metamodell (hier: ML-Modell) trainiert, um relevante Zielgrößen auf Basis definierter Eingaben zu prognostizieren. Der Optimierungsalgorithmus interagiert anschließend mit dem Metamodell, um durch Variation der Eingaben und unter Berücksichtigung der Nebenbedingungen ein Optimum zu finden. Wichtig ist hierbei die Verifikation des Optimums durch weitere FE-Simulationen, da das Metamodell ebenfalls Ungenauigkeiten aufweisen kann. Solche Ansätze können einerseits in der Strukturauslegung der passiven Fahrzeugsicherheit (Insassenzelle und Fahrzeugstruktur [Büt22]) aber auch in der Auslegung von Rückhaltesystemen [Hor17; Hor21] gefunden werden. Die Details der genannten Ansätze werden im Stand der Technik in Abschnitt 3.3 ab S. 53 erneut aufgegriffen. Für diese bildet das hier beschriebene grundlegende Vorgehen die Voraussetzung.

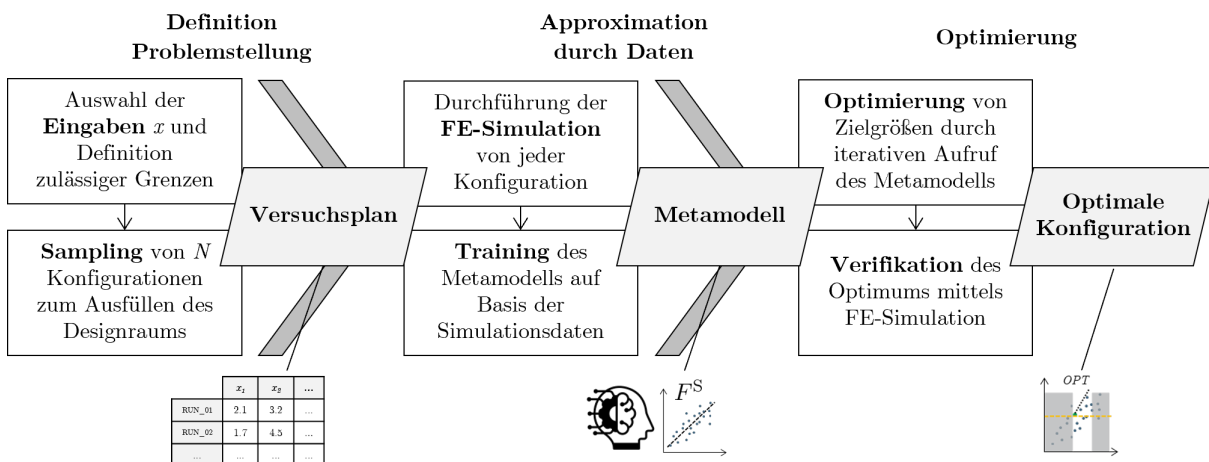


Abbildung 2.6: Strukturoptimierung mit Metamodellen

## 2.4 Begriffsbestimmungen

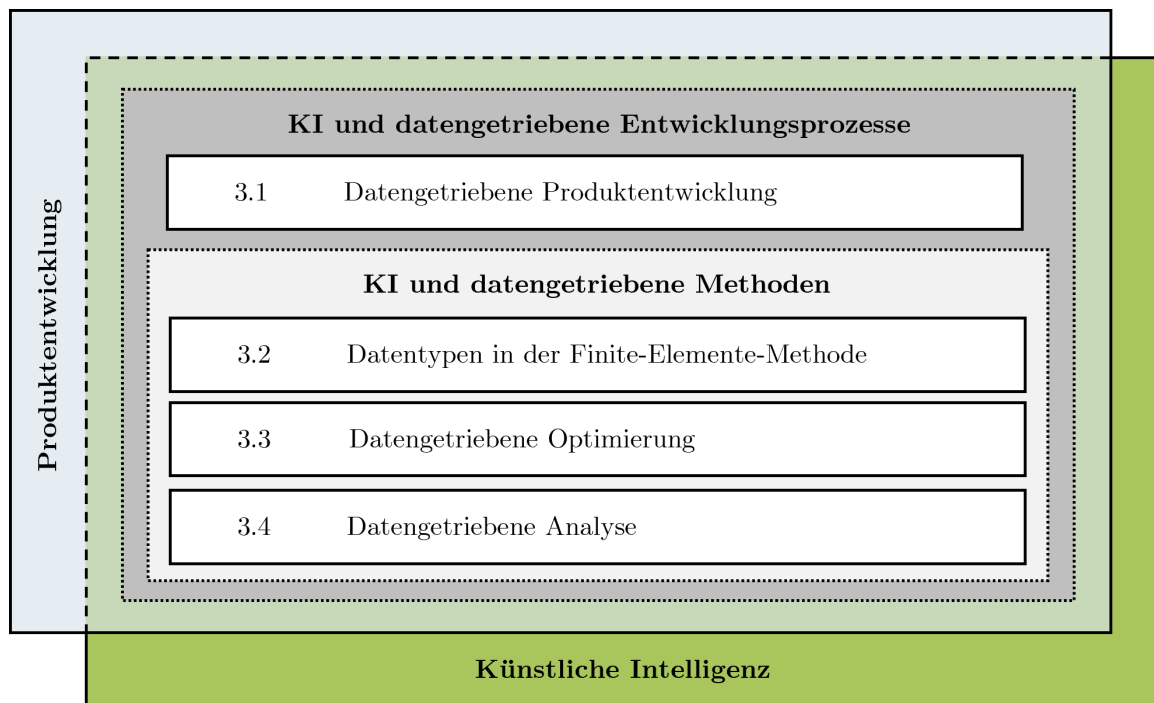
Abschließend werden nun Begriffe für ein allgemeines Verständnis der Arbeit bestimmt.

- Ein **System** ist nach [DIN 60050] eine Menge miteinander in Beziehung stehenden Elemente, die in einem definierten Kontext als Ganzes betrachtet und von ihrer Umgebung durch eine gedachte Grenze getrennt werden, um eine bestimmte Funktion oder ein Ziel zu erfüllen. Verschiedene Systemebenen (vgl. [VDI 2206]) beschreiben hierbei hierarchische Stufen innerhalb des Systems auf der bestimmte Elemente und deren Beziehungen betrachtet werden, um das Gesamtsystem besser zu strukturieren und zu verstehen.
- Ein **Prozess** umfasst nach [DIN 60050; DIN 9000] die Gesamtheit aufeinander einwirkender Vorgänge in einem System, mit dem Ziel Eingabegrößen durch zeitliche Wert- und Informationsänderungen in interpretierbare Ausgangsgrößen überführen. Die Begriffe Methode, Organisation, System, Information und Daten sind dem Prozessbegriff zur Beschreibung der inhaltlichen Aktivitäten unterzuordnen [BMV07].
- Eine **Methode** ist nach Illmer [Ill22, S. 11] eine systematische Vorgehensweise zur Lösung einer definierten Problemstellung durch die Anwendung zweckmäßiger aufeinander aufbauender Schritte [BMV07; Ham08, S. 57].
- **Datengetriebene Methoden** sind Methoden, deren zentraler Bestandteil die intelligente Verarbeitung und oder Nutzung von Daten ist. Nach Mehlstäubl et al. [Meh+21] können diese auf Algorithmen des ML und des Data Minings aufbauen.
- Ein **Ansatz** beschreibt im Rahmen dieser Arbeit – übergeordnet zum Begriff der Methode – eine grundlegende Herangehensweise an eine Problemstellung.
- **Wissen** ist nach Illmer [Ill22, S. 15–16] ist die Vernetzung von Informationen, die aus Daten in einem kontextspezifischen Zusammenhang abgeleitet und durch Zeichen ausgedrückt werden.
- Ein **Rahmenwerk** beschreibt hier im Sinne des englischen Begriffs „Framework“ [JF88] eine kohärente Struktur, die zur Standardisierung und Organisation von Prozessen, Methoden oder Systemen dient, um Aufgaben systematisch zu lösen oder zu gestalten.
- Im Kontext eines Rahmenwerks organisiert ein **Vorgehensmodell** die zeitliche Entwicklung des geplanten Vorhabens der Lösung einer Aufgabe als Problemlösungsstrategie [VDI 2221-1]. Das Vorgehensmodell legt dabei fest, in welchen Entwicklungsschritten welche Aufgaben mit welchen Mitteln und Methoden, zu welchem Zeitpunkt, unter welchen Randbedingungen, basierend auf welchen Anforderungen, zum Erzielen welcher Ergebnisse und von wem durchgeführt werden [Goe06, S. 54]. Dementsprechend erlaubt ein Vorgehensmodell, Methoden und Methodenkombinationen in Tätigkeitsphasen zu organisieren [Ill22].



# 3 Stand der Technik

Der Stand der Technik baut auf den bereits in Kapitel 2 ab S. 9 abgehandelten theoretischen Grundlagen auf. Hier werden nun die wichtigsten Arbeiten vorgestellt und diskutiert, auf denen die in dieser Arbeit verfolgte Strategie und die entwickelten Methoden basieren. Abbildung 3.1 zeigt den Aufbau dieses Kapitels im Kontext des betrachteten Schnittfelds aus Produktentwicklung (vgl. Abschnitt 2.1 ab S. 9) und KI (vgl. Abschnitt 2.2 ab S. 17). Die weitere Einordnung der Arbeiten aus dem Stand der Technik findet anschließend im Ableitung des Handlungsbedarfs in Kapitel 4 ab S. 69 statt.



**Abbildung 3.1:** Überblick und Aufbau Stand der Technik

In Abschnitt 3.1 wird einerseits auf die Integration von KI, insbesondere datengetriebenen Methoden, in konventionelle Entwicklungsprozessen eingegangen (vgl. Abschnitt 3.1.1). Andererseits wird in Abschnitt 3.1.2 die situative Auswahl und Bereitstellung von datengetriebenen Methoden betrachtet. Diese Aspekte bilden die Grundlage für das später erörterte Vorgehensmodell und ermöglichen den Methoden, die im Stand der Technik in Abschnitt 3.3 und Abschnitt 3.4 betrachtet werden, im richtigen Kontext verstanden zu werden. Anschließend wird in Abschnitt 3.2 der aktuelle Stand der Technik der Simulationsdaten diskutiert. Darauf aufbauend wird in Abschnitt 3.3 beschrieben, wie durch die Verwendung von Daten mechanische Strukturen optimal ausgelegt werden können. Ab-

schließlich beschreibt Abschnitt 3.4 aktuelle Ansätze, um Simulationsdaten automatisiert zu analysieren und so Ingenieure im Umgang mit komplexen und großen Datenmengen zu unterstützen.

Insbesondere das Feld der KI ist äußerst dynamisch. Neue, immer leistungsfähigere KI-Modelle werden oft mehrfach innerhalb eines Jahres vorgestellt [Bom+24; PB25]. Für den grundlegenden Überblick in Bezug auf KI und die Entwicklung passiver Sicherheitssysteme wurde im Vorfeld eine systematische Literaturrecherche nach dem Preferred Reporting Items for Systematic Review and Meta-Analysis Protocols (PRISMA-P)-Schema [Moh+15] von Mathieu und Vielhaber [MV24] durchgeführt. Systematisch definierte Suchstrings werden verwendet, um drei Datenbanken – IEEE-Explore, Science Direct und Web of Science – nach Einträgen zu durchsuchen. Aus 2,285 Treffern wurden 105 Beiträge als relevant für das Thema eingestuft. Weitere 21 Beiträge wurden durch ergänzende Recherchen auf Basis der zuvor identifizierten Literatur, insbesondere über Google Scholar, aufgenommen. 78% der identifizierten Literatur fokussieren sich auf Optimierungsansätze entlang des FEM-Simulationsprozesses (vgl. Abbildung 2.4 auf S. 38), was sich auf den generellen Fokus hinsichtlich virtueller Entwicklung und Digitalisierung im Bereich der Produktentwicklung zurückführen lässt. Eine ausführliche Betrachtung zur Identifikation relevanter Literatur im Schnittfeld FE-Simulation und ML wird durch die Arbeit von Nath et al. [Nat+24] bereitgestellt.

Für den Bereich KI und datengetriebene Produktentwicklung wird ein grundlegender Überblick des Feldes über die Arbeiten von [Can+20; SCF24] aufgebaut. Auf dieser Basis wurden weitere Ansätze identifiziert.

## 3.1 Datengetriebene Produktentwicklung

Die fortschreitende Digitalisierung und die immer zentralere Bedeutung von Daten führen in sämtlichen Bereichen von Wirtschaft und Gesellschaft zu einer allumfassenden Transformation. Von dieser ist entsprechend auch die Entwicklung von Produkten nicht ausgeschlossen [Can+20]. Traditionelle Entwicklungsmethoden werden ergänzt oder angepasst, sodass sie beispielsweise auf intelligente Datenanalysen und Echtzeitinformationen zurückgreifen können. Datengetriebene Produktentwicklung befasst sich mit der Integration und Anwendung von datengetriebenen Methoden in den Produktentwicklungsprozess im Rahmen der Produktentstehung nach Abbildung 2.1 auf S. 10. Hierzu zählen insbesondere die auf Daten basierenden Ansätze des ML und Data Minings [Meh+21]. Neben der Integration in den Produktentwicklungsprozess, der im klassischen Sinne bereits im Grundlagenkapitel in Abschnitt 2.1 auf S. 9 abgehandelt wurde, werden nun auch Ansätze zur Auswahl datengetriebener Methoden diskutiert.

### 3.1.1 Datengetriebene Produktentwicklungsprozesse

Während für das Feld der Datenwissenschaften bereits allgemeine Prozessmodelle, wie der Cross-Industry Standard Process for Data Mining (CRISP-DM) [WH00], Obtain, Scrub, Explore, Model, and iNterpret (OSEMN) [MW10] oder die Analytics Solutions Unified Method (ASUM) [IBM16] existieren, ist dieser Aspekt in der Produktentwicklung weniger stark fokussiert worden [Kle+18]. Einer der ersten Ansätze, einen *Datenanalyseprozess auf die Produktentwicklung anzupassen*, wird in [Kle+18] betrachtet. Während der als Basis verwendete CRISP-DM aus Sicht der Datenwissenschaften den industriellen Einsatz datengetriebener Methoden in die Phasen Geschäftsverständnis, Modellierung, Datenverständnis und Evaluation unterteilt, schlagen [Kle+18] die vier Hauptschritte Zieldefinition, Datenvorbereitung, Modellierung und Ergebnisnutzung vor. Damit werden spezifische Bedingungen in der Produktentwicklung durch die Nähe des Ingenieurs zur Problemstellung und den Daten sowie die iterative Natur des Vorgehens adressiert.

Mehlstäubl et al. [Meh+21] verwenden ebenfalls den CRISP-DM-Prozess als Grundlage und spezifizieren Voraussetzungen, die für den Einsatz datengetriebener Methoden erfüllt sein müssen, um datengetriebene Methoden wirksam in Entwicklungsprozesse zu integrieren. Den genannten Phasen des CRISP-DM werden verschiedene technische, organisatorische und menschliche Voraussetzungen zugeordnet. Zur Erfüllung dieser Voraussetzungen werden im zweiten Schritt Stellhebel abgeleitet zu denen verfügbare und qualitativ hochwertige Daten, Managementunterstützung, methodische und technische Infrastruktur sowie qualifizierte und interdisziplinär arbeitende Anwender gehören. Des Weiteren erfordert die verstärkte Digitalisierung eine Anpassung der Herangehensweisen zur Erfassung und Bewertung von Entwicklungsprozessen [Ger+21c]. Die Autoren entwickeln auf Basis einer Literaturrecherche eine Methode zur ganzheitlichen Prozessanalyse unter der Berücksichtigung wirtschaftlicher Aspekte. Das Ziel besteht darin, gerade für kleine und mittelgroße Unternehmen Optimierungspotenziale offenzulegen, die mit dem Einsatz datengetriebener Methoden gehoben werden können. Ein dreiteiliger Erfassungsansatz (Mitarbeiter-, Daten- und Managementsicht) realisiert ein niedrig abstrahiertes Prozessmodell, in dem klassische Prozessaspekte mit digital geprägten Elementen kombiniert werden. Zu diesen Elementen zählen die verwendeten Softwaretools, Dateiformate und Speicherorte. Eine erste Validierung des Ansatzes wird anhand der Konstruktion und FE-Simulation einer Welle durch Interviews mit zwei Mitarbeitern sowie einem Manager durchgeführt. Dabei liegt der Fokus auf Anwendbarkeit, Realitätsnähe, Vollständigkeit und dem Potenzial zur weiteren datenbasierten Analyse.

Die umgekehrte Herangehensweise – also die *Anpassung von Prozessmodellen der Produktentwicklung* – zeigt sich beispielsweise in der Erweiterung des V-Modells [VDI 2206]

von Rudolph [Rud24] oder des Phasenmodells aus [VDI 2221-1; VDI 2221-2] in eigenen Arbeiten [MV24]. Das von Rudolph [Rud24] vorgestellte maschinenausführbare V-Modell im Kontext des Model-Based Systems Engineering (MBSE) basiert auf einer einheitlichen Graph-based Design Language (GBDL), in der alle Modellräume – von verbalen Ontologien über boolesche Konfigurationsbäume bis hin zu reellen Gleichungssystemen und physikalischen Computer-Aided Design (CAD)-Strukturen – als Knoten und gerichtete Transformationen repräsentiert werden. Innerhalb dieser Architektur können Entwicklungsaufgaben im V-Modell mittels KI-Methoden optimiert werden, beispielsweise durch Partikelschwarmoptimierer zur Layout-Minimierung beim Packaging, Simulated Annealing zur Reduktion von Rohrlängen bei vorgegebenem Volumen sowie einen modifizierten A\*-Algorithmus zur Findung der kürzesten Verbindung in 3D-Räumen einer Konstruktion. Use Cases aus der Industrie belegen Einsparungen von bis zu 20 % Flächeneffizienz beim Packaging, 10 % Gewichts- und Längenreduktion bei Rohrleitungssystemen sowie optimierte Verdrahtungslösungen im Automobilssektor. Die Erweiterung des GBDL-Vokabulars vergrößert den Funktions- und Einsatzrahmen des Systems, wobei ein Large Language Model (LLM) als besonders vielversprechende Methode zur dynamischen Vokabularerweiterung gilt. Dadurch können KI-Methoden sowohl die Generierung von Designalternativen als auch Phasen von Simulation, Analyse und erster Designbewertung unterstützen. Das ermöglicht eine durchgängige Automatisierung und Effizienzsteigerung des gesamten V-Modell-Prozesses. Analog zu Rudolph [Rud24] befasst sich auch Ullrich et al. [Ull+24] mit einem datengestützten V-Modell und integriert systematisch virtuelle und reale Daten in den Entwicklungs- und Validierungsprozess. Der Fokus liegt jedoch insbesondere auf der Entwicklung von sicherheitskritischen und KI-gestützten Systemen, wie sie für autonome Fahrzeuge benötigt werden.

Das MNOZUKURI AI Framework [Noz+17] beschreibt ein digitales Lösungskonzept auf Basis einer Cloud-Architektur zur Integration von KI in den Produktentwicklungsprozess. Insbesondere der Aspekt der Beschaffung von Daten sowie der Speicherung und Ausführung von ML-Modellen wird von den Autoren hervorgehoben. Die Integration in bestehende Softwaretools der virtuellen Entwicklung wird vorgeschlagen und ermöglicht den interdisziplinären Einsatz in der Entwicklung von Mechanik und Elektronik. Die Autoren weisen auf die Notwendigkeit von standardisierten Daten hin, da Ingenieursdaten nicht von vornherein als Ressource für ML-Methoden gedacht sind.

Inwiefern verschiedenste Daten im Rahmen des Produktentwicklungsprozesses durch semantische Beziehungen besser verknüpft und an der korrekten Stelle bereitgestellt werden, wird von Pickel et al. [Pic+24] anhand von Ontologien diskutiert. Die Integration der Daten in eine Ontologie erfolgt durch ein systematisches Klassifikationsschema, das je nach Datentyp unterschiedliche Schnittstellenlösungen vorsieht. Dazu gehört beispielsweise der Einsatz

von Plugins wie Cellfie für Excel-Daten oder KI-basierte Verfahren für unstrukturierte Texte. Zusätzlich wird erläutert, wie weitere Datenformate, etwa Tabellen, Bilder, Grafiken und Simulationsdaten, über Transformations- und Mapping-Ansätze in semantische Strukturen überführt werden können. Diese Ansätze sind relevant für Wissensmanagement und Begleitprozesse im Entwicklungsprozess. Der Aspekt des Wissensmanagements wird von Schweitzer, Bitzer und Vielhaber [SBV23] in Bezug auf datengetriebene Entscheidungsfindung und Verknüpfung von internen und externen Datenquellen diskutiert. Die Autoren schlagen einen Engineering Graph vor, der Ingenieursobjekte und deren Beziehungen abbildet, die in den frühen Entwicklungsphasen eine fundierte Unterstützung von Entscheidungen bieten, indem sie system- und kontextbezogene Informationen zusammenführen und in Beziehung setzen.

Damit die datengetriebenen Entwicklungsprozesse effizient realisiert und durchlaufen werden können, sind strukturelle Änderungen in Bezug auf Rollen beteiligter Ingenieure [Bic+19] und Unternehmensstrukturen [ZRD19, S. 18f] vonnöten. Von Bickel et al. [Bic+19] werden Veränderungen in den Tätigkeitsfeldern von Konstruktions- und Simulationsingenieuren diskutiert: Erstere übernehmen zunehmend Aufgaben in der Simulation, während Simulationsingenieure sich auf komplexe Spezialfälle und die Pflege von Simulationsdatenbanken konzentrieren. Dadurch entstehen neue, interdisziplinäre Rollen, die sowohl Domänenwissen in der Produktentwicklung als auch Datenkompetenz vereinen. Von Ziegler, Rossmann und Danzer [ZRD19, S. 18f] werden drei grundlegende Organisationsstrukturen zur Verankerung von KI in Unternehmen diskutiert. Erstens eine dezentrale Programmstruktur, bei der KI-Initiativen innerhalb bestehender Strukturen durch ein zentrales Programmmanagement koordiniert werden, das jedoch vor allem beratend tätig ist. Zweitens ein virtuelles Center of Excellence, das zentrale Steuerungsaufgaben übernimmt, während die Umsetzung in den Fachbereichen bleibt. Drittens ein zentralisiertes Cluster of Excellence mit direkter Anbindung an die oberste Führungsebene der Organisation, das sowohl strategisch als auch operativ die vollständige Verantwortung für die Aktivitäten trägt. Zur Umsetzung wird dabei auf dedizierte Produktteams und KI-Botschafter in den Fachbereichen zurückgegriffen.

### 3.1.2 Situative Auswahl von Machine-Learning-Methoden

Neben den konzeptionell-methodischen Modellen, die zur Erfassung der datengetriebenen Produktentwicklungsprozesse im vorherigen Abschnitt thematisiert wurden, ergibt sich im Entwicklungsprozess oft die spezifische Frage, *wann welche Methode wie eingesetzt werden soll*. In Bezug auf konventionelle Entwicklungsmethoden wurde dieser Aspekt bereits in Abschnitt 2.1.3 auf S. 15 im Rahmen der Grundlagen diskutiert. Für datengetriebene Methoden ist dieser Aspekt besonders relevant, da klassisch ausgebildeten

Ingenieuren im Bereich der KI und des ML notwendige Expertise fehlt [GGW23; Son+24b].

Kriterien, die die Bewertung und Auswahl von ML-Algorithmen ermöglichen, werden in einem zweistufigen Prozess von Riesener et al. [Rie+20] vorgestellt. Zunächst werden auf Basis einer systematischen Literaturrecherche die elf am häufigsten vorkommenden ML-Algorithmen aus den drei Lernverfahren identifiziert. Hierzu zählen beispielsweise k-Means Clustering oder neuronale Netzwerke und Q-Learning. Anschließend werden auf Basis weiterer Literatur und Expertendiskussionen neun übergreifende Bewertungskriterien abgeleitet. Zu diesen zählen der Rechenaufwand, die Genauigkeit und Nachvollziehbarkeit bzw. Transparenz. Durch die Bewertung der identifizierten ML-Algorithmen anhand der Kriterien lässt sich eine Matrix aufspannen, aus der für gegebene Lernverfahren und Aufgaben ein Score berechnet wird. Dieser ermöglicht im Produktentwicklungsprozess dann den objektiven Vergleich. Wie genau diese Kriterien angewendet werden, wird jedoch nicht näher diskutiert [Son+23].

Während von Riesener et al. [Rie+20] alle drei Lerntypen betrachtet werden, liegt der Fokus in [Lic+21] vor allem auf den Klassifikationsalgorithmen. Hierfür stellen die Autoren ein kriterienbasiertes Verfahren zur algorithmischen Vorauswahl bereit, das insbesondere unerfahrenen Nutzern eine systematische Orientierung bieten soll. Neben den in [Rie+20] verwendeten Kriterien, wie der Transparenz und dem Rechenaufwand, werden auch Aspekte der verwendeten Daten, wie Größe, Typ und Dimensionalität, hinzugezogen. Die Kriterien und ML-Algorithmen werden in eine übersichtliche Tabelle eingeordnet, um eine gezielte Vorauswahl zu ermöglichen. Als Beispiel zur Validierung wird die Wiederaufbereitung von Autoteilen aus der Rückführungslogistik verwendet. Ziel ist es mit ML – basierend auf Merkmalen wie dem Rückgabedatum, dem Lieferanten oder der Teilenummer – den wahrscheinlichen Zustand eines Teils vorherzusagen. Ein Bayes-Classifer wird für die gegebene Problemstellung als beste Option identifiziert.

Während in den ersten Arbeiten von Sonntag et al. [Son+23] Pattern Languages verwendet werden, um systematisch ML-Algorithmen zu beschreiben und auszuwählen, werden in Folgearbeiten [Son+24b] Aspekte der multikriteriellen Entscheidungsfindung herangezogen. In [Son+23] werden konkrete Problemmerkmale wie Problemtyp, Datenart, Trainingsanforderungen und Nachvollziehbarkeit standardisiert erfasst. Durch den Abgleich dieser Merkmale mit bestehenden Mustern lässt sich die bestmögliche ML-Algorithmus für eine neue Problemstellung identifizieren. Der in [Son+24b] vorgestellte Ansatz stützt sich auf ein dreistufiges Entscheidungsunterstützungskonzept. Dieses beginnt mit der Einordnung der Problemstellung anhand eines referenzierten Entwicklungsprozesses zur Ableitung der benötigten Eigenschaften der ML-Algorithmus. In der zweiten Phase erfolgt die Bewertung verbleibender Algorithmen mittels einer mehrkriteriellen Bewertungsmatrix, wobei

die Priorisierung der Alternativen mittels Preference Ranking Organisation Method for Enrichment Evaluations (PROMETHEE) vorgenommen wird. Die Gewichtung der Kriterien erfolgt dabei auf Basis der „SIMOS“-Methode, wodurch individuelle Präferenzen der Entscheidungsträger strukturiert und nachvollziehbar berücksichtigt werden können. Zur Validierung wird beispielhaft die Aufgabe der Informationsextraktion aus technischen Dokumenten herangezogen. Der neuere Ansatz aus [Son+24b] ist damit dem Entwicklungsprozess näher und ermöglicht darüber hinaus die Evaluation der Robustheit einer gefundenen Lösung.

Um Wissen über datengetriebene Methoden strukturiert zu erfassen und verfügbar zu machen, wird von Gerschütz, Schleich und Wartzack [GSW21] ein Ansatz vorgestellt, der auf der Erstellung eines semantischen Klassendiagramms basiert. Hierzu wird ein Klassenmodell nach den Prinzipien der Unified Modeling Language (UML) entwickelt, das zentrale Elemente jeder ML-Methode – Kategorie, Konzept, Methode und Tool – systematisch beschreibt und über semantische Eigenschaften miteinander verknüpft. Neben den reinen Algorithmen, wie sie in bisher vorgestellten Arbeiten fokussiert worden sind, ermöglichen die Autoren die Einbindung (kommerzieller) Softwaretools und den Algorithmen übergeordneten ML-Methoden. Die Umsetzung erfolgt in einem Semantic Media Wiki, das kollaborative Bearbeitung, semantische Annotationen und gezielte Abfragen ermöglicht, um ML-Methoden zu identifizieren. Ein stärkerer Fokus auf die Integration von datengetriebenen Methoden in Entwicklungsprozesse wird in späteren Arbeiten mithilfe eines semantischen Ansatzes auf Basis einer Ontologie vorgeschlagen [GGW23]. Diese Ontologie basiert auf vier zentralen Bausteinen: (1) Menschen (Ingenieure, die mit den ML-Methoden arbeiten), (2) (Entwicklungs-)Prozessen, (3) (ML-)Methoden und (4) Anwendung (spezifischer Einsatz der Methoden in ingenieurtechnischen Aktivitäten). Diesen Bausteinen sind sogenannte Konzepte, beispielsweise ein Teilprozess oder eine datengetriebene Methode, zugeordnet, und sie werden semantisch miteinander verknüpft. Erstellt wird die Ontologie mit dem Software-Tool Protégé unter Verwendung der standardisierten Web Ontology Language (OWL). Die Evaluation des Ansatzes erfolgt anhand von zwei Szenarien. Das erste Szenario zeigt, wie bereits existierende datengetriebene Methoden systematisch beschrieben und dokumentiert werden können. Dabei werden zentrale Informationen wie der Entwicklungsprozess, eingesetzte Werkzeuge, Methoden sowie Ein- und Ausgabedaten formal abgebildet und miteinander verknüpft. Das zweite Szenario verdeutlicht, wie auf Basis einer strukturierten Problembeschreibung – etwa Engpässe in einem Simulationsprozess – geeignete datengetriebene Methoden, Werkzeuge und Aufgaben identifiziert werden können. Hierzu werden Kriterien wie Datenformate, Speichersysteme, verfügbare Hardware und Schnittstellen berücksichtigt.

## 3.2 Datentypen in der Finite-Elemente-Methode

Ein zentraler Aspekt der vorliegenden Arbeit ist die maschinelle und automatisierte Verarbeitung von Simulationsdaten. Ausgehend von den Grundlagen zum Thema FEM im Kontext der virtuellen Systemauslegung in Abschnitt 2.3.2 ab S. 35 werden nun aktuelle Datenformate aus der Praxis eingeführt, mit denen Ingenieure im Rahmen ihres Alltags interagieren. Dies erlaubt dem Leser die anschließend diskutierten Optimierungs- und Analysemethoden besser verstehen zu können. Eine grafische Übersicht ist Abbildung 3.2 zu entnehmen.

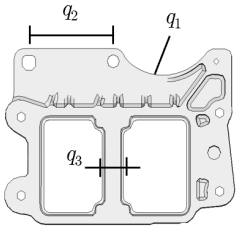
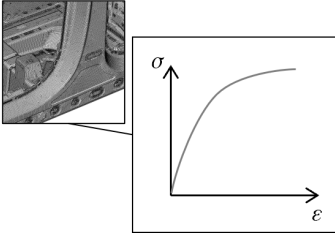
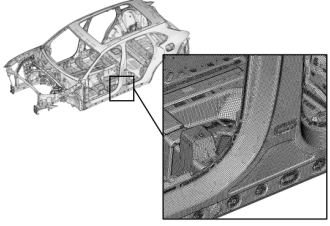
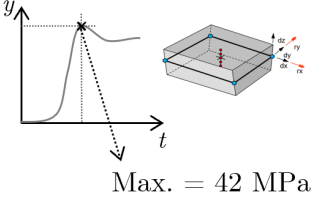
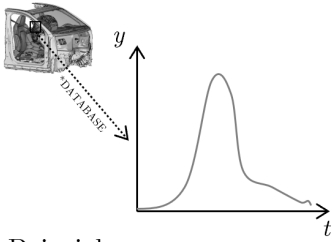
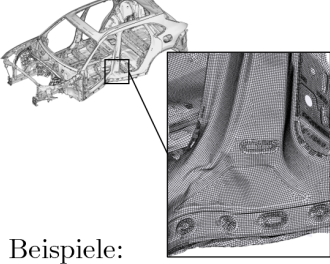
Metadaten Zeitschritt, Dämpfung, Parallelisierung...			
<b>Typ</b>	<p><b>Skalar</b></p> <ul style="list-style-type: none"> <li>• Skalar <math>q \in \mathbb{R}</math></li> <li>• Strukturiert</li> </ul>	<p><b>Seriell</b></p> <ul style="list-style-type: none"> <li>• 1D (Zeit-)Serie <math>s \in \mathbb{R}^T</math></li> <li>• Strukturiert</li> </ul>	<p><b>Finite Elemente</b></p> <ul style="list-style-type: none"> <li>• Graph <math>\mathcal{G}(t, \xi)</math></li> <li>• Unstrukturiert</li> </ul>
<b>Simulationseingabe</b>	 <p>Beispiele:</p> <ul style="list-style-type: none"> <li>• <i>Wandstärke</i></li> <li>• <i>Airbagauslass</i></li> </ul>	 <p>Beispiele:</p> <ul style="list-style-type: none"> <li>• <i>Spannung/Dehnung</i></li> <li>• <i>Triaxialität</i></li> </ul>	 <p>Beispiele:</p> <ul style="list-style-type: none"> <li>• <i>Initiale Form B-Säule</i></li> <li>• <i>Vorspannung Türblech</i></li> </ul>
<b>Simulationsergebnis</b>	 <p>Max. = 42 MPa</p> <p>Beispiele:</p> <ul style="list-style-type: none"> <li>• <i>Min. Türintrusion</i></li> <li>• <i>Max. Elementdehnung</i></li> </ul>	 <p>Beispiele:</p> <ul style="list-style-type: none"> <li>• <i>Schnittkraft am Gurt</i></li> <li>• <i>Kopfbeschleunigung</i></li> </ul>	 <p>Beispiele:</p> <ul style="list-style-type: none"> <li>• <i>Deformation B-Säule</i></li> <li>• <i>Dehnungen Tür</i></li> </ul>
<b>Format</b>	LS-DYNA key-File, csv, markdown	LS-DYNA key-File/ binout0000, ISO-MME	LS-DYNA key-File/d3plot

Abbildung 3.2: Datentypen in der Finite-Elemente-Methode

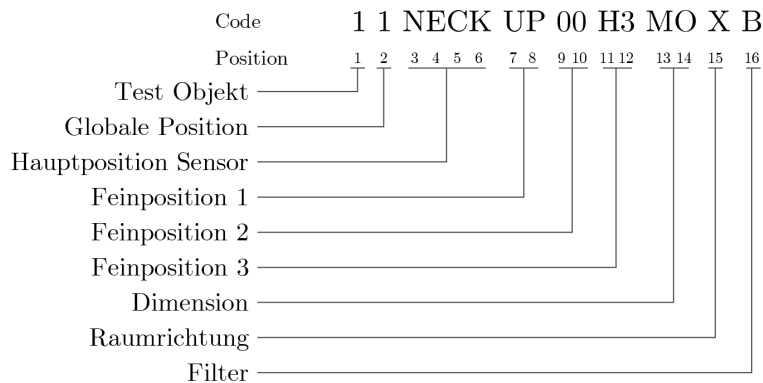
Der einfachste Datentyp sind die *skalaren Daten*, oft auch als Entwurfsvariablen im Kontext der Optimierung [Sch20, S. 2–3] oder als Designparameter im Kontext der Produktentwick-

lung bezeichnet, die Eigenschaften des mechanischen Systems beschreiben. Wandstärken einer Blechstruktur werden beispielsweise durch solche Daten in der Simulationseingabe beschrieben. Skalare Daten können aber auch in Bezug auf die Simulationsergebnisse relevant sein. Maximalwerte, die aus Spannungsfeldern oder aus Beschleunigungskurven abgeleitet werden, können insbesondere für die Analyse interessant sein. In Bezug auf die in Abschnitt 2.2.1 ab S. 17 definierten Datentypen handelt es sich um strukturierte Daten, für die zur Speicherung beispielsweise ein Excel-CSV Datenformat verwendet werden kann. Der zweite relevante Datentyp sind *1D serielle Daten*, die im Berechnungsprozess selbst durch virtuelle Sensoren generiert werden. In LS-DYNA [Ans25] lässt sich das mit \*DATABASE-Karten im Preprocess definieren. Nach Durchführung der Simulation liefern Beschleunigungswerte an Knotenpunkten, Drücke oder Schnittkräfte im Postprocess wertvolle Informationen über das zeitliche Verhalten des betrachteten Systems. 1D serielle Daten sind aber auch in der Simulationseingabe vorhanden. Beispielsweise können Spannungs-Dehnungskurven bei verschiedenen Dehnraten oder auch Triaxialitätskurven zur Beschreibung des Materialverhaltens hinterlegt werden. Bei einem Vergleich ist auf identische physikalische Einheiten ( $y$ -Achse) und Abstraten ( $x$ -Achse) zu achten. In Bezug auf die in Abschnitt 2.2.1 definierten Datentypen handelt es sich unter Berücksichtigung identischer Einheiten und Abstraten ebenfalls um strukturierte Daten. Im Fall von LS-DYNA [Ans25] werden diese Daten als ASCII im key-Format eingegeben und binär als binout0000 ausgegeben.

Der grundlegende Datentyp der FE-Simulation sind die finiten Elemente selbst – hier allgemein als *FE-Daten* bezeichnet (vgl. Abbildung 3.2). Mathematisch können diese als Graph betrachtet werden, der eine Zeitabhängigkeit  $t$  aufweist und von weiteren Attributen  $\xi$  abhängt, die weitere Informationen über Elemente und Knoten enthalten können. FE-Daten sind unstrukturiert, und erfordern, je nach eingesetztem ML-Algorithmus, eine Vorverarbeitung [Kra24; Kne+25]. In Bezug auf die Simulationseingabe beschreiben die FE die geometrische Form der Komponenten und enthalten Informationen über verwendete Materialien sowie Randbedingungen. Als Teil der Simulationsergebnisse beschreiben die FE das Deformationsverhalten über die Zeit. Plastische Dehnungen, Spannungen und Versagensinformation sind ebenfalls enthalten und ermöglichen detaillierte Einblicke in das komplexe Verhalten crashbelasteter Systeme. Vorliegende Datenformate in Bezug auf LS-DYNA [Ans25] sind einerseits die Simulationseingaben im key-Format aber auch die Ausgaben im binären d3plot-Format.

Ebenfalls nennenswert sind jegliche Metadaten der Simulation, wie Zeitschritte, Dämpfung, Genauigkeit der Gleitkommazahlen und Parallelisierung. Diese sind für die vorliegende Arbeit zweitrangig, aus Zwecken der Vollständigkeit in Abbildung 3.2 aber dennoch dargestellt.

Austauschformate, die eine FE-Löser-unabhängige sowie versuchs- und simulationsunabhängige Darstellung der Daten ermöglichen und dabei baustands- und baureihenübergreifend einsetzbar sind, erleichtern die algorithmische Verarbeitung erheblich. Für diese Arbeit ist das Standardformat für 1D-Zeitreihendaten (ISO-MME) [ISO 13499]. ISO-MME ermöglicht eine einheitliche und allgemeine Benennung mit Kanalcodes. Abbildung 3.3 zeigt ein Beispiel. Die dem ISO-MME-Format zugrundeliegenden Daten können über virtuelle Sensoren in einer Crashtestsimulation (vgl. Abbildung 3.2), aber auch durch physische Sensoren in einem Crashtest erzeugt werden. Als Beispiel für die einheitliche Darstellung von Simulationsdaten, inklusive FE-Daten, lässt sich der VMAP-Standard [VMAP25] nennen.



Beispiel: Testobjekt 1, Position (1) Fahrzeug vorne links, Fahrer Nacken, oben, Hybrid III 50% ATD, Moment, X, Filterung (B) cfc600

**Abbildung 3.3:** Aufbau eines ISO-MME Kanalcodes, nach [ISO 13499]

## 3.3 Datengetriebene Optimierung

Die Optimierung des Verhaltens von crashbelasteten Strukturen in der Entwicklung passiver Sicherheitssysteme ist insbesondere aufgrund der Vielzahl an Regularien aus Gesetzen und dem Verbraucherschutz (vgl. Abschnitt 2.3.1.1 auf S. 33) eine herausfordernde Entwicklungsaufgabe. Die Grundlagen zum Verständnis dieser Sektion wurden bereits im Rahmen der allgemeinen Optimierung von mechanischen Systemen in Abschnitt 2.3.2.3 ab S. 38 näher erläutert. Angelehnt an die in Abschnitt 2.3.1 ab S. 33 erwähnten Systeme (Fahrzeugstruktur, Insassenzelle, Rückhaltesysteme), untergliedert sich diese Sektion in *Struktur-* und *Rückhaltesystemoptimierung*. Der Fokus liegt insbesondere auf den Optimierungsverfahren mit RL, da dieser Ansatz auch im Rahmen dieser Arbeit betrachtet und erweitert wird.

### 3.3.1 Optimierung der Strukturintegrität

Drei für diese Arbeit relevante Ansätze können der Literatur entnommen werden, um mechanische Strukturen zu optimieren: (1) *metamodellbasierte Methoden*, (2) *Heuristiken* und (3) *RL-basierte Methoden*. Umfangreiche Betrachtung von Methoden zur Optimierung mechanischer Strukturen allgemein kann Schumacher [Sch20] entnommen werden. Ziele der strukturellen Optimierung können beispielsweise die Maximierung der Energieabsorption oder die Reduzierung von Beschleunigungsspitzen während der Crashlast sein.

#### 3.3.1.1 Metamodellbasierte Optimierungsverfahren

Die Grundidee von metamodellbasierten Ansätze, wie sie in [Cra+02; CDP05; Büt22; Büt+23] verfolgt werden, wurde bereits in Abschnitt 2.3.2.3 auf S. 38 anhand von Abbildung 2.6 auf S. 40 erläutert. Die grundlegende Herangehensweise ist somit bereits einige Zeit bekannt und ist bereits viele Jahre in den Entwicklungsprozessen verankert. Zur Modellierung des Zusammenhangs zwischen Eingabeparametern und Zielvariablen verwenden Cadete, Dias und Pereira [CDP05] einen probabilistischen Gaußschen Regressionsprozess. Die Autoren betrachten in ihrer Arbeit die Optimierung von einem Unterraum eines Zugs, an dem Energieabsorption, Verzögerungen und Deformationen durch Anpassung geometrischer Größen für den Aufprall auf eine starre Wand optimiert werden sollen. Das zugrundeliegende Simulationsmodell basiert auf einer Mehrkörpersimulation. Zur Optimierung wird ein genetischer Optimierungsalgorithmus verwendet. Büttner et al. [Büt+23] basieren ihre Erweiterung einer multidisziplinäre Optimierungsstrategie in der Karosserieentwicklung, ähnlich wie in [Cra+02], auf einem Metamodellansatz. Hierfür wird eine automatisierte ML-Methode herangezogen, die das für den Anwendungsfall beste ML-Modell durch Optimierung von Hyperparametern und Durchführung von Kreuzvalidierung zur Beschreibung der Zusammenhänge ermittelt. Verschiedene Metamodelle, wie lineare

Modelle, Support Vector Machines oder neuronale Netzwerke, stehen dabei zur Auswahl. Ebenfalls wird ein heuristisches Maß für die Vorhersageunsicherheit der Metamodelle vorgeschlagen, das für alle – auch nicht probabilistische Modelle – berechnet werden kann. Die Optimierung selbst, wird hier durch einen evolutionären Algorithmus durchgeführt. Die Vielzahl an Eingabeparametern und Zielgrößen über verschiedene Disziplinen hinweg wird mit Hilfe einer globalen Sensitivitätsmatrix strukturiert. Validiert wird dieser Ansatz in der Optimierung eines Rohbaus unter Berücksichtigung von sechs Lastfällen (fünf Crashlastfälle und eine Frequenzanalyse). In beiden Fällen [CDP05; Büt+23] wird nach Abbildung 2.5 auf S. 39 die Optimierung der Dimensionierung – Höhe des H-Profiles bzw. Wandstärken der Rohbaubleche – durchgeführt.

### 3.3.1.2 Graphen- und Heuristik-basierte Optimierungsverfahren

Mittels der Graphen- und Heuristik-basierten Optimierungsmethoden lassen sich crashbelasteten Strukturen in Bezug auf Topologie, Form und Dimensionierung (vgl. Abbildung 2.5 S. 39) optimieren. Die Ansätze resultieren aus Schwachstellen Voxel-basierter Optimierungsverfahren für die Optimierung unter Crashbelastung [Sch20, S. 290].

Beispiele für die Optimierung von zwei- und dreidimensionalen Strukturen wie Querschnitten von Strangpressprofilen und Crashbox-Layout können [Ort+21; BSS21] entnommen werden. Das eigentliche Optimierungsproblem ist in zwei ineinander verschachtelte Optimierungsschleifen unterteilt: In der äußeren Optimierungsschleife verändern Heuristiken, die aus Expertenwissen abgeleitet wurden, die Topologie basierend auf Systemantworten aus Crashsimulationen. In der inneren Optimierungsschleife werden konventionelle Optimierer für die Anpassung von Form- und Dimension eingesetzt, um die von der äußeren Schleife vorgeschlagenen Strukturen zu bewerten und zu verbessern. Die Geometrie wird dabei durch einen mathematischen Graphen beschrieben. Zu den Heuristiken zählen beispielsweise das Abstützen schnell deformierender Wände und das Ausgleichen (Verbinden) von Bereichen mit hoher und geringer Energiedichte [Ort+21].

[Ort+21; BSS21] betrachten die Optimierung einer akademischen Crashbox und Rahmenstruktur, die beispielsweise in [BSS21] hinsichtlich Minimierung von Intrusion und Kontaktkraft ausgelegt werden. Diese auf Heuristiken basierten Methoden können dadurch, dass maschinell menschliche intelligente Entscheidungsfindung nachgeahmt wird, auch dem Feld der KI zugeordnet werden (vgl. Abschnitt 2.2, S. 17).

### 3.3.1.3 Reinforcement-Learning-basierte Optimierungsverfahren

Durch intelligente Entscheidungsfindung mit RL-Agenten, beispielsweise in dem Spiel Go [SB20, S. 441–450], liegt die Idee nahe, diese Strategie auch zur Optimierung mechanischer Strukturen einzusetzen.

Die Optimierung von ebenen Stabstrukturen und Stahlrahmen unter statischen Spannungs- und Verformungsrandbedingungen mit dem Ziel der Volumenminimierung ist Teil der Arbeiten von Hayashi und Ohsaki [HO20; HO22]. Analog zu den bereits erwähnten Arbeiten [Ort+21; BSS21], wurde auch hier eine Grapheneinbettung verwendet, um eine strukturierte Datenrepräsentation zur Anwendung von Faltungsoperatoren zu ermöglichen. Nativ können Faltungsoperationen aufgrund der unregelmäßigen Konnektivität nicht auf die diskreten Strukturen angewendet werden. Die Systemantwort der Strukturen wird mittels FE-Simulation in jedem Schritt bestimmt, was den Trainingsprozess in [HO20] bzw. [HO22] zwischen 3.5 Stunden und mehreren zehn Stunden dauern lässt. Dieser Aspekt wird als Limitierung der Methode aufgeführt, jedoch muss dieser Prozess nur einmal durchgeführt werden. Die Generalisierbarkeit der RL-Agenten wird an Strukturen unterschiedlicher Größen und unvorhergesehener Randbedingungen erfolgreich nachgewiesen. Der trainierte RL-Agent ist in der Lage einen genetischen Algorithmus bei der Optimierung der Stabstrukturen hinsichtlich der benötigten Rechenzeit zu übertreffen. Im Benchmark mit einem genetischen Optimierungsalgorithmus werden suboptimale Konfigurationen aufgefunden [HO20]. In der Folgearbeit [HO22] werden im Vergleich mit einem Partikelschwarm-Optimierer in kürzerer Zeit auch qualitativ bessere Lösungen gefunden. Ähnliche Ergebnisse sind auch in [Bro+22] zu finden, die zeigen, dass der Einsatz eines RL-Agenten in der Topologieoptimierung einer ebenen Schalenstruktur unter statischer Last, in fünf von sechs Lastenfällen eine bessere Lösungsqualität als der Benchmark mit einem gradientenbasierten Optimierer liefert. Die Generalisierung der Strategie wird durch die Betrachtung verschieden großer planarer Strukturen nachgewiesen. Die betrachtete viereckige Struktur aus quadratischen Elementen mit je vier Knoten wird direkt mit einem Convolutional Neural Network verarbeitet. Dieses analysiert mehrere sog. Kanäle gleichzeitig und kann damit Spannungsfelder, Randbedingungen und Belastungspunkte gemeinsam zu verarbeiten. Eine Einschränkung der Methode ist der Rechenaufwand, da die Ausführung von bis zu 144 FE-Simulationen innerhalb einer Trainingsepisode erforderlich ist, da die Spannungen unter Last bestimmt werden müssen. Die Belohnung des RL-Agenten wird mittels der Verformungsenergie und den entfernten Elementen berechnet.

Um dem aufwändigen Trainingsvorgang, der in den vorangegangenen Arbeiten unter anderem als Limitierung aufgeführt wurde, schlägt [Mat23] die Verwendung von Metamodellen zum Training des RL-Agenten vor. Borse, Gulakala und Stoffel [BGS23; BGS24a; BGS24b] und eigene Nachfolgearbeiten [Mat+24] vertiefen diesen Ansatz. Borse, Gulakala und Stoffel [BGS23; BGS24a; BGS24b] betrachten in ihren Arbeiten die Optimierung von geometrischen Größen einer akademischen Crashbox, die durch Deformation Energie einer aufprallenden Masse aufnehmen soll. Während [BGS23; BGS24a] ausschließlich die Wandstärke der Crashbox optimieren, erweitern die Autoren die Anzahl der Eingabeparameter

in der Optimierung auf zwei – Wandstärke und Breite – in ihrer Folgearbeit [BGS24b]. Zielgrößen sind die absorbierte Energie, die initiale maximale Kontaktkraft, die mittlere Kraft sowie die Länge im deformierten Zustand. Der RL-Agent interagiert mit einem Regressionsmodell, das neben den für die Optimierung relevanten Eingabeparametern auch Geschwindigkeitsunterschiede, Länge und Höhe der Crashbox in Bezug auf die Zielgrößen mit abbildet. Dadurch wird die Generalisierung der erlernten Strategie anschließend evaluiert. Die zum Training des Regressionsmodells bereitgestellten Daten werden mit Hilfe eines Deep Convolutional Generative Adversarial Network künstlich augmentiert. In den Arbeiten werden verschiedene ML-Modelle für die Regression (vgl. Abschnitt 2.2.1.1, S. 18) verwendet. Während [BGS23] ein neuronales Netzwerk trainieren, wird von [BGS24a] ein polynombasiertes Modell und von [BGS24b] ein Gradient-Boosting-Tree-Modell herangezogen. Alle Metamodelle sind in der Lage den RL-Agenten zu trainieren. In den früheren beiden Arbeiten werden verschiedene RL-Agenten – Deep-Q-Network, Proximal Policy Optimization und Reinforce – miteinander verglichen. Deep-Q-Network- und Proximal Policy Optimization-Agenten konnten die Struktur der Crashbox während des Trainings optimieren. Im Gegensatz dazu war die Trainingsdauer für den Reinforce-Agenten mit über einer Stunde signifikant länger. Zudem konnte dieser optimale Lösungen nicht zuverlässig finden. Der Deep-Q-Network-Agent trainiert in [BGS24a] lediglich etwa 3 min und wird daher ausschließlich in [BGS24b] verwendet. Der in wenigen Minuten trainierte Deep-Q-Network-Agent findet für unterschiedliche Startpunkte innerhalb weniger Sekunden optimale Lösungen für Breite und Dicke der betrachteten Struktur. Ein Benchmark mit konventionellen Optimierern oder eine Anwendung direkt auf eine FE Simulationsumgebung wird nicht durchgeführt.

Die Erweiterung der bereits diskutierten Graphen- und Heuristik-basierten Optimierungsmethoden in Abschnitt 3.3.1.2 auf S. 54 um eine RL-basierte Heuristik betrachtet [TSZ22; TSZ24; Tri24]. Ziel ist es, die Querschnitte von Rahmen- und Querträgerstrukturen unter dynamischer Last lokal durch die Einbringung von Aussteifungen zu optimieren. Auch hier wird, wie bei der Graphen- und Heuristik-basierten Optimierung grundsätzlich, aber auch bereits von [HO20] in Bezug auf RL-basierte Strukturoptimierung eingeführt, eine Grapheneinbettung verwendet. Der Aktionsraum des RL-Agenten enthält eine Identifikationsnummer, die auf die Positionierung der Kante, die hinzugefügt werden soll, in einer Adjazenzmatrix der Graphenrepräsentation referenziert. Da der RL-Agent lokale Änderungen einbringen soll, betrachtet dieser immer nur einzelne Zellen innerhalb der Graphenrepräsentation des Querschnitts. Die Observation, die der RL-Agent vor der Definition einer Aktion tätigt, enthält Größen zur Beschreibung der Struktur und Antworten der mit der FEM berechneten Struktur. Zu den Observationsgrößen zählen beispielsweise die Länge der Kanten und Wandstärken der Struktur, Fertigbarkeit, Knotenverschiebungen sowie

ein Bild im deformierten und undeformierten Zustand. Ähnlich der Methode von [Bro+22] wird ein Convolutional Neural Network zur Bildverarbeitung der Observation verwendet. Die Belohnung des Agenten wird auf Basis von einer Ähnlichkeit des Flächenmaßes im deformierten und undeformierten Zustand bestimmt. Für drei, vier und fünfseitige Zellen wird jeweils ein RL-Agent trainiert. Ausführlich validiert wird der Ansatz in [TSZ24] anhand von zwei akademischen Beispielen, wobei bei einer Aktivierung der RL-basierten Heuristik die auftretenden Verschiebungen stärker reduziert wurden als ohne. Die Optimierte Lösung unter Einbezug der RL-Heuristik weist im Fall der Rahmenstruktur 7.5 mm statt 14.7 mm y-Verschiebung auf. Die unoptimierte Struktur zeigt eine y-Verschiebung von 72.4 mm.

### 3.3.2 Optimierung der Rückhaltesysteme

Zur Reduktion der Insassenbelastungen ist das Rückhaltesystem essenziell, da dieses während des Crashes in direktem Kontakt mit dem Insassen steht. Die hierfür zum Verständnis nötige Theorie wurde in Abschnitt 2.3.1 auf S. 33 bereits abgehandelt. Durch die Fortschritte im Bereich des automatisierten Fahrens werden zudem alternative Sitzposition relevant, da auch diese einen Einfluss auf die Belastung hat. Beispielsweise muss dann auch die Absicherung beim Rückwärtsfahren sichergestellt sein [Hua+15; YH23]. Durch verschiedene Fahrzeugvarianten, ATDs, und Lastfälle entspricht diese Entwicklungsaufgabe einer Mehrzieloptimierung unter diversen Randbedingungen [Thi+06]. Auch in Bezug auf die hier fokussierten Rückhaltesysteme finden sich oben bereits erwähnte Strategien wieder: Während [Thi+06], [Hor17], [JGK21] oder [HCK24] den Metamodellansatz verwenden, setzen [Xue+12; Hua+15] auf vereinfachte Starrkörpermodelle. In eigenen vorangegangenen Arbeiten [Mat+24] wurde bereits eine RL-basierte Auslegung von Rückhaltesystemen vorgeschlagen, die später im Rahmen dieser Arbeit weiter vertieft wird.

Xue et al. [Xue+12] vergleichen zwei genetische (MOGA-II, NSGA-II), einen gradientenbasierten (MIPSQP) und einen Simulated-Annealing-Optimierungsalgorithmus (MOSA). Acht Eingabeparameter, die durch den Optimierer angepasst werden können – unter anderem die Adaptivität von Lenksäule, Airbag und Gurtkraft sowie die Art des Airbags und eine Gurtvorstraffung – werden berücksichtigt. Die betrachtete Zielgröße ist hier der sog. Relative Risk Score nach der National Highway Traffic Safety Administration (NHTSA) unter Berücksichtigung von 32 Nebenbedingungen, die Grenzwerte für einzelne Belastungswerte wie die Oberschenkel oder Nackenkraft festsetzen. Xue et al. [Xue+12] verwendet FE-Starrkörpermodelle, die in modernen Fahrzeugentwicklungsprozessen eher selten Anwendung finden. Die Ergebnisse zeigen, dass die genetischen Algorithmen die global beste Lösungsqualität mit der geringsten Anzahl an Iterationen auffinden. Im Gegensatz dazu erweist sich das gradientenbasierte Optimierungsverfahren aufgrund lokaler Konvergenzei-

enschaften als ungeeignet für die vorliegende Problemstellung. Dass Gradientenverfahren in nichtlinearen Crashlastfällen ungeeignet sind, wurde auch von Büttner [Büt22] in Bezug auf die Strukturauslegung hervorgebracht. Der Simulated-Annealing-Algorithmus weist mit über 4,000 benötigten Simulationen die langsamste Konvergenz auf. Im Gegensatz dazu benötigt der NSGA-II nur etwa 2,000 Simulationen bis zur Konvergenz. Ebenfalls wurden mit ihm die meisten 5-Sterne Bewertungen aufgefunden. Grundsätzlich ist die Anzahl an benötigten Simulationen, gerade wenn detaillierte Simulationsmodelle wie in [Thi+06; JGK21] verwendet werden, aus wirtschaftlicher Sicht kaum mehr praktikabel.

Horii [Hor17] kombiniert die Verwendung von globalen Optimierern mit der von Metamodellen, analog zu den bereits diskutierten Ansätzen in Abschnitt 3.3.1.1 ab S. 53. Der Autor verwendet ein auf einem Gaußprozess basierendes Regressionsmodell und kann Pareto-optimale Lösungen für das Kopfverletzungskriterium HIC15 und die maximale Brustbeschleunigung unter Verwendung eines evolutionären Algorithmus spezifizieren. Eingabeparameter werden auch hier für Fahrerairbag, Knieairbag und Gurt definiert und enthalten Zündzeiten, Kraftlimits, Massenflussrate und einen Steifigkeitsfaktor. In [Thi+06; JGK21] wird neben der automatischen Findung von Konfigurationen durch einen globalen Optimierer auch die manuelle bzw. interaktive Spezifizierung durch den Ingenieur genannt. Ein Vergleich verschiedener Metamodelle, einschließlich Lasso Regression und neuronalen Netzwerken, zur Modellierung der Zusammenhänge zwischen Insassenbelastung und Rückhaltesystem ist [JGK21] zu entnehmen. Die Autoren trainieren die verschiedenen ML-Modelle auf Basis von 450 detailliert modellierten Schlittensimulationen, wie sie auch in der Industrie verwendet werden. Die Konfigurationen des Rückhaltesystems werden mit 16 Parametern beschrieben und sind mit Latin Hypercube Sampling definiert worden. Schlittenmodelle im Insassenschutz bilden das Gesamtfahrzeug, gerade in Frontalaufprallszenarien, reduziert ab und enthalten daher nur einen Rohbau mit relevanten Teilen des Interieurs, die Rückhaltesysteme und die ATD. Die Verzögerung aus einem Versuch oder eine Gesamtfahrzeugsimulation wird auf den Rohbau aufgebracht. So kann Rechenzeit eingespart werden, wobei eine Simulation trotzdem zehn bis 20 Stunden auf einem Hochleistungscluster dauern kann [Mat+24]. Ein Ensemble-ML-Modell, das den Mittelwert der Prognosen aus Lasso Regression, neuronalen Netzwerke, Support Vector Regression und Random Forest bildet, konnte die höchste Genauigkeit ( $R^2 = 0.72$ ) liefern. Die Lasso Regression, zeigte eine angemessene Genauigkeit von ( $R^2 = 0.70$ ), bei einem einfach anzupassenden Hyperparameter. Für die übrigen Modelle wurden ebenfalls Hyperparameter auf Basis einer Kreuzvalidierung optimiert. Einige der diskutierten Ansätze für die Rückhaltesystemauslegung werden von Heo, Cho und Kim [HCK24] zur Verbesserung der EuroNCAP-Bewertung kombiniert. Es wird ein detailliertes Simulationsmodell mit 14 Parametern verwendet, wobei hier in den Rückhaltesystemparametern auch die Position

---

von D-Ring und Gurtschloss und Endbeschlag und die Skalierung der Fahrzeugverzögerung berücksichtigt werden. Ein genetischer Optimierungsalgorithmus legt das System durch den Aufruf von 11 neuronalen Netzwerken, die für den EuroNCAP relevante Verletzungswerte vorhersagen, aus. Das definierte Optimum erreicht die höchstmögliche Anzahl Sterne in der Bewertung. Im Vergleich zur Basisvariante werden in allen Belastungswerten bis auf das Kopfverletzungskriterium HIC15, die Kopfbeschleunigung und die Kraft am Acetabulum geringere Belastungswerte erreicht. Darüber hinaus verwenden Heo, Cho und Kim [HCK24] SHAP, um globale Sensitivität im Designraum zu ermitteln. Beispielsweise wird so gezeigt, dass die Massenflussrate des Airbags auf die Brusteingdrückung eine hohe Relevanz aufweist.

## 3.4 Datengetriebene Analyse

Die datengetriebene Analyse von FE-Simulationen fokussiert sich darauf, Ingenieure mit intelligenten Methoden zu unterstützen, um die Qualität und ein schnelleres Verständnis der betrachteten Systeme und Modelle zu gewährleisten. Dies ist in erster Linie eine Erweiterung der Simulationswerkzeuge und befreit die Ingenieure nicht von der Prüfung physikalisch sinnvoller Modellierung. Diese muss im Vorhinein sichergestellt sein, da die in diesem Abschnitt diskutierten Methoden erst auf den nach dem Simulationsablauf anfallenden Daten aufsetzen. Für die vorliegende Arbeit sind in erster Linie Methoden für transiente Crashsimulationen im Fokus. Einige Methoden, die als Mehrwert für diese Arbeit gesehen werden, betrachten Daten aus statischen Struktursimulationen. Um den Abschnitt zu strukturieren, wird die Literatur anhand der in Abbildung 3.2 auf S. 50 drei eingeführten Datentypen, die maßgeblich für die Methode sind, untergliedert. In Bezug auf die verschiedenen Lernverfahren in Abbildung 2.2 auf S. 18 sind sowohl Supervised- als auch Unsupervised-Learning-Methoden vertreten.

### 3.4.1 Skalare Daten

Ackermann et al. [Ack+08] nutzt die PCA, um die Dimension der Knoten- und Elementinformationen einer akademischen U-Struktur unter dynamischer Last zu reduzieren. Im zweiten Schritt wird eine lineare Korrelation zwischen den ersten 13 Hauptkomponenten und 24 skalaren Eingabeparametern, die Eigenschaften der U-Struktur wie Dicke, Material sowie Reibungs- und Dämpfungskoeffizienten beschreiben, berechnet. Die Auswertung der Korrelationen ermöglicht Einblicke welcher der Eingabeparameter das globale Verhalten der Struktur, das mit den Hauptkomponenten beschrieben wird, beeinflusst. Dieser Ansatz verdeutlicht jedoch ein grundsätzliches Vorgehen, das später auch in Diez et al. [Die+18] in Kombination mit einer Supervised-Learning-Methode verwendet wird.

Die dimensionsreduzierte Repräsentation einer Stoßstange von einem Gesamtfahrzeugmodell im Frontalaufprall gegen eine starre Wand wird von Diez et al. [Die+18] untersucht. Die geometrische Dimensionsreduktionsmethode wird in früheren Arbeiten (vgl. [DHS16; Die+16]) näher beleuchtet und bestimmt komponentenweise eine Distanzmatrix, die dann durch ein hierarchisches Clusterverfahren Gruppen ähnlichen Systemverhaltens bildet. Die Komponenten der niedrigdimensionalen Repräsentation werden oft auch als Deformationsmoden bezeichnet, da sie typischerweise charakteristisches Verhalten, wie eine Verschiebung oder Starrkörperrotation, repräsentieren. Eine direkte Zuweisung dieses Verhaltens oder eine Interpretation ist jedoch nicht möglich. Analog dem Vorgehen in [Ack+08] wird ein Zusammenhang zwischen skalaren Eingabeparametern, hier Wandstärke der Stoßstange und Neigung der starren Wand, und den Clusterzugehörigkeiten (Stoß-

stange beult mittig oder nicht mittig) über einen Decision Tree hergestellt. Auf Basis dieser Supervised-Learning-Methode werden Teilräume im Entwurfsraum isoliert und Regeln aus dem trainierten Modell in einem Wenn-Dann-Format extrahiert. Ein Beispiel hierfür ist: Wenn die Neigung der Wand kleiner als  $2.3^\circ$  ist, dann beult die Stoßstange nicht links. Ingenieure können sich auf diese Regeln beziehen, um das Systemverhalten zu verstehen und zu verbessern. Die Robustheit der vorgeschlagenen Methode in Bezug auf die Datenverfügbarkeit wird ebenfalls untersucht. Die Regelfindung und der relative Fehler stabilisieren sich, sobald mehr als 200 Simulationen (20 % des verwendeten Datensatzes) verwendet werden.

Die Grundidee tiefere Einblicke in das Systemverhalten zu erhalten, was ein bestimmtes Simulationsergebnis verursacht, wird auch von Conti und Kaijima [CK20] verfolgt. Die Autoren betrachten parametrische Daten aus Struktursimulationen, die einen akademischen rechteckigen Balken unter statischer Biegebelastung abbilden. Als ML-Modell werden Bayesian Networks verwendet, die ähnlich zu dem Decision Tree aus [Die+18] von sich selbst aus interpretierbar sind. Bei Decision Trees zeigen die Abzweigungen, welche Eingabeparameter zur Entscheidungsfindung verwendet werden. Die Reihenfolge der Abzweigungen erlaubt Rückschlüsse auf die relative Wichtigkeit. Bei Bayesian Networks hingegen ergibt sich die Interpretierbarkeit aus der Graphenstruktur, die Abhängigkeiten offenlegt, und den bedingten Wahrscheinlichkeiten zwischen den Variablen. Zwei Eingabeparameter – Höhe und Länge des Rechteckprofils – werden mit Hilfe einer pseudoregulären Sequenz im Designraum verteilt. Die zugehörige Verschiebung unter der Belastung von 400 N wird mittels der FEM bestimmt. Das Bayesian Network wird auf die Eingabe-Ausgabe-Beziehung, die auch durch einen Graphen repräsentiert wird, trainiert. Das trainierte ML-Modell wird dann verwendet, um Ursache-Wirkungs-Beziehungen zu untersuchen, wenn ein oder beide Eingabeparameter fixiert werden. Die resultierenden Wahrscheinlichkeiten helfen Ingenieuren bei der Analyse und dem Verständnis des Systems.

### 3.4.2 1D-Zeitseriendaten

Die Verarbeitung von 1D-Zeitseriendaten, hier von sog. Crashsensoren, zu Analyse Zwecken wird in eigener vorangegangener Arbeit [MDV23] sowie der von Iza Teran et al. [Iza+23] und Pakiman, Garcke und Schumacher [PGS23] betrachtet. Im Kontext der Crashsimulation besteht der Vorteil im Vergleich zur Verwendung von FE-Daten insbesondere in der fehlenden Abhängigkeit von der Diskretisierung durch das FE-Netz, was die Datenmenge und Komplexität signifikant reduziert (vgl. Abbildung 3.2 auf S. 50). Trotzdem lassen sich Informationen über das zeitliche Verhalten extrahieren. Mathieu, Di Roberto und Vielhaber [MDV23] bringen den Aspekt der Disziplinen in der Fahrzeugentwicklung hervor, da in der Entwicklung von Insassenzelle und Rückhaltesystem ein besonders starker Fokus auf den

1D-Zeitseriendaten liegt. Dort sind die ATDs zentraler Gegenstand der Betrachtung, deren Ausgaben – sowohl physisch als auch virtuell – Sensorsignale für Kraft, Beschleunigungen oder Wege sind.

Iza Teran et al. [Iza+23] verwendet den Laplace-Beltrami-Operator analog zu weiteren Arbeiten in der niedrigdimensionalen Repräsentation von dreidimensionalen FE-Daten [IG19]. Eine Verallgemeinerung dieses Ansatzes von dreidimensionalen Oberflächen auf eindimensionale Zeitserien aus Sensordaten ist möglich. Hierfür werden Abstände – sog. Segmentlängen – entlang der Kurve gemessen. Unter der Annahme, dass Verformungen der Kurven ebenfalls abstandserhaltend sind, kann die mit dem Operator berechnete Eigenvektorbasis verwendet werden, um zahlreiche Kurven darauf zu projizieren. Da die Hauptänderungen in nur wenigen Koeffizienten beschrieben werden, kann dieses Verfahren zur Dimensionsreduktion eingesetzt werden. Für das Clustering im niederdimensionalen Raum werden von den Autoren ein dichtebasiertes sowie ein hierarchisches Clustering vorgeschlagen. Eine interaktive Visualisierung ermöglicht die Interaktion der Nutzer mit den Ergebnissen der Analyse und erlaubt die Identifizierung von auffälligen Simulationen. Die Validierung erfolgt anhand von Energien aus Strukturcrashsimulationen und Kopfbeschleunigungen aus Fußgängerschutzsimulationen. Die in [MDV23] vorgestellte Methode identifiziert automatisch Ausreißer in den Crashsensordaten und bezieht das zeitliche Verhalten durch lokale Betrachtung weiter mit ein. Im Rahmen dieser Arbeit wird die Methode später vertieft behandelt und um einige Aspekte erweitert.

Pakiman, Garcke und Schumacher [PGS23] leiten aus den globalen Energien von Komponenten eines crashbelasteten Gesamtfahrzeugs drei Merkmale ab, die in einer heterogenen Graphdatenbank die explorative Analyse verschiedener Entwicklungsphasen, Lastfälle und Bauteile ermöglichen. Zu den aus der Energie abgeleiteten Größen zählen der Zeitpunkt des Energieabsorptionsbeginns, die absolute maximale interne Energie sowie die Zeit bis zum Erreichen dieser maximalen Energie. Die Autoren zeigen, dass Graphalgorithmen und die semantische Anreicherung durch physikalisch interpretierbare Merkmale ein neues, intuitives und intelligentes Werkzeug zur Wissensgenerierung in der Crashsimulation darstellen. Dadurch entsteht eine datengetriebene und kollaborative Alternative zur klassischen statischen Berichterstattung. In Folgearbeiten [PGS25] dient der Wissensgraph als Basis für die intelligente Suche und den Vergleich von Simulationen. Hierbei werden Ähnlichkeiten von Simulationen über ein Ähnlichkeitsmaß prognostiziert. Das verwendete Ähnlichkeitsmaß wird auf Basis von SimRank spezifisch für Simulationsdaten entworfen. Explizit auf Crashsensordaten wird auch in Koch, Wang und Bäck [KWB18] und Koch und Bäck [KB18] verwiesen. Diese Arbeiten fokussieren sich insbesondere auf die Prognose von beschädigten Fahrzeugkomponenten und der Identifikation von einem von 8 Kollisionspunkten rund um das Fahrzeug mit Supervised-Learning-Methoden. Nach

einer automatischen Extraktion von Features aus den Signalen, werden die wichtigsten automatisch identifiziert und zu finalen Training des Modells herangezogen. In [KB18] wird dazu die Boruta-Methode verwendet, die unter anderem das erste Maximum und die Standardabweichung als relevant herausstellt. In Koch, Wang und Bäck [KWB18] wird für jedes Bauteil ein Random Forest trainiert. Diese weisen eine Prognosegenauigkeit in der Klassifikationsaufgabe von bis zu 94 % auf. Koch und Bäck [KB18] verwenden Decision Trees, die eine Prognosegüte von 76 % im Rahmen einer k-fold Cross Validation erreichen.

### 3.4.3 Finite-Elemente-Daten

Zur Analyse der Systeme ist insbesondere die Vergleichbarkeit der Datentypen zwischen verschiedenen Simulationen von großer Relevanz. Während ein maschineller Vergleich von Wandstärken oder seriellen Daten zwischen verschiedenen Bauständen einfach realisiert werden kann, ist das mit FE-Daten weitaus komplexer. Das Fortschreiten der Entwicklung hat einen Einfluss auf die Eigenschaften der FE, beispielsweise durch Veränderung von Form oder Material, und damit auch auf Anzahl und Reihenfolge der FE im Simulationsmodell. Ein direkter automatisierter bzw. maschineller Vergleich ist entweder nur mit speziellen Methoden oder durch eine Transformation in ein strukturiertes Datenformat möglich. Dementsprechend werden in Abschnitt 3.4.3.1 zunächst aktuelle Vorverarbeitungsverfahren betrachtet und erst in Abschnitt 3.4.3.2 auf die Analysemethoden selbst eingegangen.

#### 3.4.3.1 Vorverarbeitung von Finite-Elemente-Daten

Einige Methoden beschäftigen sich mit einer **direkten Verarbeitung oder Angleichung der unstrukturierten FE-Daten**. Existiert in der Betrachtung mehrerer Simulationen eine Referenzsimulation, was in der Entwicklung typischerweise gegeben ist, eignet sich ein *Mapping* der übrigen Simulationen auf die Struktur der Referenzsimulation [GPP17]. Die Informationen der FE liegen somit in voller Auflösung vor und können unabhängig von Geometrieänderungen verarbeitet werden. Wie groß geometrische Änderungen sein dürfen, damit das Mapping noch funktioniert, wird nicht näher diskutiert. Wird das FE als *dreidimensionale Punktwolke* aufgefasst, eignet sich PointNet [Qi+17], welches durch die Architektur die Permutationsinvarianz einzelner Punkte sicherstellt. Ebenfalls können die FE auch als *mathematischer Graph* betrachtet werden [Pfa+21; Kne+25]. Mit Graph Convolutional Neural Networks können Informationen aus mathematischen Graphen mit Hilfe eines Faltungsoperators, ähnlich zur Bildverarbeitung, gelernt werden [KW17] und beispielsweise auch für die Ergebnisprognose von Crashsimulationen verwendet werden [Kne+25]. Hahner und Garcke [HG22] und Hahner, Kerkhoff und Garcke [HKG22] transformieren unregelmäßige FE in *semi-reguläre FE*, die ebenfalls Anwendung von Faltungsoperationen ermöglichen. Dies wird durch adaptives Vernetzen einer groben Basis-FE, die an die ursprüngliche Oberfläche angepasst wird, realisiert. Ein Resampling in Kombi-

nation mit einem *Coherent-Point-Drift-Algorithmus*, um verschiedene FE-Konfigurationen eines zweidimensionalen Bauteils in eine einheitliche Knoten- und Elementstruktur zu überführen wird von Greve und van de Weg [GW22] diskutiert.

Eine Alternative zur direkten Verarbeitung von unstrukturierten FE-Daten ist die **Transformation in ein strukturiertes Datenformat**. In [DHS16; Die+16; Die+18] werden FE-Daten auf eine eindimensionale *Schwerpunktlinie* projiziert. Das eignet sich insbesondere für die Verarbeitung von profilierten Bauteilen wie einem Längsträger im Vorderwagen eines Fahrzeugs und wird in [Die+16] näher erläutert. Allerdings ist mit diesem Ansatz die Verarbeitung von flächigen oder komplexen Bauteilen nur bedingt möglich [Kra+23; Kra24]. Ein weiterer Ansatz, ist die Projektion der FE im dreidimensionalen Raum auf eine zweidimensionale *Kugeloberfläche*, was in [SW14; SSW17; Spr+20; BGW23] vorgeschlagen und vertieft wird. Eine zweidimensionale Repräsentation der FE kann dann mit Ansätzen aus der Bildverarbeitung, wie Convolutional Neural Networks, weiterverarbeitet werden. Ein ähnlicher Ansatz wird von Steinhäuser [Ste24] verfolgt, der aus den Simulationsmodellen mehrere einheitliche *Schnittbilder* mit einer Einfärbung nach Materialeigenschaften ableitet, die mittels Convolutional Neural Networks zur Prognose von Belastungswerten herangezogen werden. Ein möglicher Nachteil dieser bildbasierten Verfahren ist jedoch die detaillierte Abbildung von Hinterschnitten, vor allem bei komplexen Komponenten [Kra24]. Alternativ ist die Projektion der FE in ein dreidimensionales *Voxelgitter*, wie in [KGS20; Kra+23; Koh+21] behandelt, möglich. Die Gestalt der Komponenten bleibt somit erhalten, was eine visuelle Interpretation der strukturierten Repräsentation vereinfacht. In Bezug auf Datengröße ermittelt [Kra24] im Vergleich von Projektions- und Voxelmethode, dass der Speicherbedarf, abhängig von der Pixel- bzw. Voxelgröße, für die Projektionsmethode bis zu elf Mal höher ist. Darüber hinaus wird gezeigt, dass der Informationsgehalt, aufgrund weniger starker Verzerrung, höher ist. Von Hahner, Iza Teran und Garcke [HIG20] werden Bounding-Boxes zur Analyse der räumliche Orientierung von Komponenten in Crashesimulationen vorgeschlagen, die ebenfalls eine netzunabhängige strukturierte Repräsentation der FE darstellen. Die Transformation in ein strukturiertes Datenformat ermöglicht die vereinfachte Verarbeitung der FE-Daten, allerdings sind zusätzlicher Speicher- und Rechenaufwand sowie ein gewisser Informationsverlust relevante Abwägungen, die berücksichtigt werden müssen [Kra24].

### 3.4.3.2 Analyse von Finite-Elemente-Daten

Die meisten Methoden in der Literatur können zur Analyse der FE-Daten mittels **Unsupervised Learning** gefunden werden. Ältere Arbeiten gehen auf den Anfang der Nullerjahre zurück [TM03]. In umfangreicheren Folgearbeiten betrachtet [Tho+10] einen Datensatz von Crashesimulationen eines Gesamtfahrzeugs im Frontalaufprall und identifizieren mit ihrer Methode auf Basis einer PCA diejenigen Komponenten, die einen

dominanten Unterschied in den Ergebnissen verursachen. Die Methode besteht aus einer Difference-PCA und einem Correlation Clustering. Für die Difference-PCA wird für jede Komponente in jedem Zeitschritt eine PCA durchgeführt. Um Komponenten zu identifizieren, deren Verhalten zusammenhängen könnte, werden die Kovarianzmatrizen für zwei verschiedene Komponenten und Zeitschritte betrachtet. Unter Einbezug von Streuung und Korrelationen können Einflüsse der Komponenten aufeinander ermittelt werden, was das Finden potenzieller Ursachen ermöglicht. Durch diese Herangehensweise konnte im betrachteten Datensatz das Verhalten der Firewall als sekundärer Effekt identifiziert werden. Das Correlation Clustering aggregiert hochkorrelierte Daten und eignet sich daher vor allem für die interaktive Analyse.

Während die PCA eine lineare Dimensionsreduktionsmethode ist, betrachtet [Boh+13] nichtlineare Ansätze und heben deren Vorteile hervor. In ihrer Arbeit wird zunächst ein Clustering der Daten durchgeführt, um ähnliche Simulationen zu ermitteln, beispielsweise zur Findung identischen Knickverhaltens des Längsträgers. Hierfür werden k-Means und ein Sparse-Grid Spectral Clustering verwendet. Beide Methoden sind in der Lage Cluster zu identifizieren. Das Sparse-Grid Spectral Clustering, das auch nicht konvexe Strukturen identifizieren kann, ist hingegen in der Lage, im frühen Zeitbereich der Simulation sinnvollere Ergebnisse zu liefern. Aus den Ergebnissen des Clusterings geht hervor, dass der Träger von vorne die Deformation beginnt. Grundsätzlich sind aber die Ergebnisse aus beiden Methoden, gerade wenn global der gesamte Zeitbereich berücksichtigt wird, in guter Übereinstimmung mit der visuellen Beobachtung des Crashverhaltens im Gesamtfahrzeugmodell eines Pick-Up Trucks im Frontalaufprall. Im nächsten Schritt wird mit drei nichtlinearen Dimensionsreduktionsmethoden – Lokale Lineare Approximation basierend auf Tangent Space Estimation, DiffMaps- sowie Principal Manifold Learning – die Dimensionalität der Einbettung auf  $\mathbb{R}^2$  zwecks besserer Interpretierbarkeit von Unterschieden verringert. Für das betrachtete Modell liefern die DiffMaps die zur Bifurkation des Längsträgers passendste Einbettung. Basierend auf Local Tangent Space Alignment und Principal Manifold Learning konnte gezeigt werden, dass im Vergleich zu einer linearen PCA weniger Komponenten für eine gute Rekonstruktion der Daten benötigt werden.

Die Verwendung von DiffMaps zur Dimensionsreduktion zum besseren Verständnis der Variation in den betrachteten FE-Modellen wird ebenfalls von Iza Teran [Iza14] diskutiert. Die Methode wird auf FE-Simulationen aus drei Anwendungsfeldern – Umformtechnik, Crash und Vibrationsmoden – angewendet und hebt die generelle Anwendbarkeit des Vorgehens in der Analyse von Simulationsdaten hervor. Es wird gezeigt, dass die DiffMaps-Einbettung eine Identifikation der langsam veränderlichen Variablen (Slow Variables) in stochastischen dynamischen Systemen ermöglicht, die das wesentliche Verhalten des Systems beschreiben. Diese Auffindung solcher Trends wird am Beispiel der Bifurkation des Längsträgers in

der betrachteten Crashsimulation gezeigt. Darüber hinaus wird eine Nyström-Methode beschrieben, mit der neue, ursprünglich nicht verfügbare Datenpunkte effizient in die bestehende DiffMaps-Einbettung überführt werden können. In einer weiterführenden Arbeit stellen Iza Teran und Garcke [IG19] einen alternativen Ansatz zur niederdimensionalen Repräsentation hochdimensionaler Crashsimulationen in  $\mathbb{R}^3$  vor. Ausgangspunkt der Methode ist die Annahme, dass die verschiedenen Simulationsergebnisse durch isometrische Transformationen miteinander verbunden sind. Dabei werden Translationen, Rotationen sowie lokale und globale Deformationen berücksichtigt. Zur mathematischen Modellierung dieser Zusammenhänge wird ein diskreter Laplace-Beltrami- und Fokker-Planck-Operator auf dem Oberflächennetz der FE-Daten konstruiert. Dies ermöglicht eine spektrale Zerlegung der Geometrieinformationen aller Simulationen und liefert eine projektionstaugliche Orthonormalbasis, die gegenüber den betrachteten Transformationen invariant ist. Durch die Projektion der hochdimensionalen Oberflächeninformationen auf diese Eigenbasis gelingt eine kompakte und interpretierbare Darstellung der Deformationsverläufe über die Zeit. Im Vergleich zu etablierten Verfahren wie der PCA und DiffMaps, die auch von [Boh+13] betrachtet wurden, zeigt die vorgeschlagene Methode eine überlegene Strukturierung des resultierenden Einbettungsraums. Insbesondere lassen sich unterschiedliche physikalische Verhaltensweisen, vor allem im Hinblick auf ihr zeitliches Auftreten, deutlich besser voneinander trennen.

T-distributed Stochastic Neighbor Embedding (t-SNE) ist eine weitere nichtlineare Dimensionsreduktionsmethode, die in [KGS20] mit der linearen PCA verglichen wird. Während die PCA eine erste gute Schätzung über die Unterschiede in den Daten liefert, ist t-SNE in der Lage, lokale Effekte, insbesondere bei kleinen Perplexitäten, deutlich präziser aufzulösen. Ein Nachteil von t-SNE sind jedoch die nicht-deterministischen Eigenschaften sowie die Abhängigkeit von der Perplexität, die als Unsicherheitsquelle hervorgehoben wird, da sie zu variierenden und nicht unbedingt optimalen Ergebnissen führt. Wie bereits von Iza Teran und Garcke [IG19] angesprochen, kann die Zeit für die Dimensionsreduktion entweder reduziert oder beibehalten werden. Kracker, Garcke und Schumacher [KGS20] zeigen an mehreren Beispielen, dass sich eine Simulation im eingebetteten Raum entweder als Linie (OLioS – One Line One Simulation) oder als Punkt (OPioS – One Point One Simulation) darstellen lässt. Das Ziel der dimensionsreduzierten Repräsentation der Simulationsdaten ist es, interaktiv durch Interaktion mit dem Ingenieur Cluster, Ähnlichkeiten oder sog. Ausreißer zu identifizieren. Diese Simulationen weisen grundlegend anderes Verhalten auf, beispielsweise durch Fehler, aber auch durch günstige bzw. ungünstige Konfigurationen betrachteter Eingabeparameter. Kracker et al. [Kra+23] erweitern die Ausreißerkennung um einen automatischen Prozess, der kein manuelles Suchen mehr erfordert. Die Methode verwendet eine PCA zur Reduktion der Dimension von FE-Daten

einzelner Fahrzeugkomponenten für jeden Zeitschritt. Die Ausreißerkennwerte werden auf Basis des Abstands zum  $k$ -ten nächsten Nachbarn bestimmt. Zur Definition, welche Ausreißer als auffällig bezeichnet werden sollen, wird der absolute Abstand zum Median verwendet, der sich im Vergleich zur Verwendung von Interquartilsbereichen in verschieden stark kontaminierten Datensätzen als vorteilhaft erwiesen hat.

Die bisher diskutierte Literatur fokussiert sich auf lineare und nichtlineare Dimensionsreduktionsmethoden, die – entsprechend Abschnitt 2.2.1 auf S. 17 – dem Unsupervised Learning zugeordnet werden können. Methoden zur Analyse von FE-Daten, die auf **Supervised Learning** basieren, werden in [SW14; SSW17; Spr+20; HIG20; BGW23] vorgestellt.

In der Arbeit von Bickel, Goetz und Wartzack [BGW23], die auf die Arbeiten [SW14; SSW17; Spr+20] aufbaut, fokussieren sich die Autoren auf die Analyse statischer Simulationen strukturmechanischer Problemstellungen. Ihre Methode nutzt bereits durchgeführte Simulationen, um die Plausibilität neuer Simulationen vorherzusagen, indem die FE-Daten, die Randbedingungen und die Ergebnisse berücksichtigt werden. Zur Vorverarbeitung der Daten wird die bereits angesprochene Projektionstechnik verwendet, die die Verwendung von Convolutional Neural Networks ermöglicht. Ein Multi-Label Classifier wird verwendet, um zu prognostizieren, ob die Diskretisierung, die geometrische Form oder die Lastwerte plausibel sind. Der Ansatz wird mit Daten aus fünf verschiedenen Simulationsaufbauten validiert. Dabei werden statische Lastszenarien an einem Bremshebel, einer Kurbelwelle und einer Wippe verwendet. Insgesamt besteht der Datensatz aus etwa 60,000 FE-Simulationen. Verschiedene ML-Modellarchitekturen werden verglichen, wobei für alle die Geometric-Mean-Metrik über 0.94 liegt. Alle einbezogenen Architekturen weisen somit eine gute Eignung auf. Aufgrund der Differenzierung von drei Klassen erhalten Ingenieure Ideen, warum eine Simulation möglicherweise nicht plausibel ist; allerdings werden keine Gründe angeführt, warum eine bestimmte Klasse zugeordnet wird. Eine Supervised-Learning-Methode auf Basis von Long Short-Term Memory Autoencodern wird in [HIG20] vorgestellt. Diese ermöglicht es, auch das zeitliche Verhalten mitzulernen, was gerade für die crashbelasteten Strukturen relevant ist. Der Ansatz nutzt zwei Decoder: einen zur Rekonstruktion und einen zur Vorhersage zukünftiger Zeitpunkte der analysierten Sequenz. Das in der Einbettung durchgeführte Clustering ermöglicht Einblicke in 192 Crashtsimulationen, in denen Materialeigenschaften variiert wurden. Trotz der vergleichsweise einfachen Geometrierepräsentation können Muster zuverlässig erkannt werden.



## 4 Handlungsbedarf

Nach der Diskussion des Stands der Technik in Kapitel 3 ab S. 43 wird nun der Grundstein für die im Rahmen dieser Arbeit entwickelten Ansätze gelegt. Hierzu werden Defizite in Bezug auf die Fragestellungen der Präskriptiven Studie nach Abbildung 1.1 auf S. 6 herausgearbeitet. Aus diesen lässt sich der Handlungsbedarf anschließend systematisch ableiten. Zur *Bewertung des Defizits* in Bezug auf die Forschungsfragen werden in Abschnitt 4.1 jeweils spezifische Kriterien zur Einordnung bestehender Arbeiten eingeführt. Diese sind ergänzend zu den übergeordneten Zielkriterien in Abschnitt 1.3.1 auf S. 4 zu sehen. Abschließend erfolgt in Abschnitt 4.2 ab S. 81 die *Abgrenzung* und Konkretisierung der *Beiträge* dieser Arbeit in Vorbereitung auf die Präskriptive Studie.

### 4.1 Defizite

Entsprechend der Orientierung an den Fragestellungen der Präskriptiven Studie nach Abbildung 1.1 auf S. 6 teilt sich dieser Abschnitt in drei Sektionen: datengetriebene Produktentwicklung, datengetriebene Optimierung crashbelasteter Systeme und datengetriebene Analyse von FE-Simulationsdaten. Während Abschnitt 4.1.1 und Abschnitt 4.1.2 sich auf die Forschungsfragen i) und ii) in Abbildung 1.1 auf S. 6 beziehen, werden Defizite und Handlungsbedarf für die Analysemethoden (vgl. iii) und iv)) in Abschnitt 4.1.3 gemeinsam betrachtet. Zur schnellen Übersicht ist die Einordnung relevanter bestehender Arbeiten in Tabelle 4.1 bis Tabelle 4.3 zusammengefasst. Der dort eingetragene Erfüllungsgrad liefert keinen Rückschluss auf die Qualität, sondern lediglich über den Betrachtungsrahmen bezogen auf das eingeführte Kriterium.

#### 4.1.1 Datengetriebene Produktentwicklung

Tabelle 4.1 auf S. 73 beinhaltet die Einordnung relevanter Literatur im Feld der datengetriebenen Produktentwicklung. Die in Abschnitt 3.1 ab S. 44 diskutierten Ansätze werden anhand der Kriterien *Ebene* und *Typ des Prozessmodells*, *Auswahl* und *Prozessintegration von Methoden*, *Einfache Anwendung*, *Generalisierbarkeit* sowie *Bezug zu einem Anwendungsfeld* im Kontext der Fragestellung *i. Wie lässt sich die situative Auswahl und Integration von datengetriebenen Methoden in Entwicklungsprozessen unterstützen?* eingeordnet. Der Erfüllungsgrad sowie die Einordnung in das von Wynn und Clarkson [WC18] vorgestellte Schema sind Inhalt von Tabelle 4.1.

Die **Ebene des Prozessmodells** bezieht sich dabei auf Mikro-, Meso- und Makroebene (vgl. Abschnitt 2.1.1 auf S. 9) und beschreibt daher den grundlegenden Betrachtungsrahmen des Modells. Auf der Mikroebene sind Arbeiten hinsichtlich Datenanalyseprozessen [Kle+18], Algorithmen- und Methodenauswahl [Rie+20; Lic+21; GSW21; GGW23; Son+23], Voraussetzungen [Meh+21] oder Prozesserfassung [Ger+21c] verortet, die spezifische, vor allem lokale, Problemstellungen im Prozess angehen. Im Auswahlmechanismus von [Son+24b] werden Entwicklungsphasen einbezogen, was die Betrachtungen in den Kontext höherer Ebene setzt. Auf der Mesoebene legen [Bic+19; Pic+24] ihren Überlegungen Phasenmodelle zugrunde. Durch das Einbeziehen von übergeordneten Aspekten wie Anwendung, Personen und Prozess in der Ontologie, lässt sich [GGW23] auf der Makroebene einsortieren. Selbiges gilt für das MNOZUKURI AI Framework [Noz+17], das sich auf die Verknüpfung von Nutzer, Cloud und Tools, die für den Entwicklungsprozess fokussiert, die als Rahmenbedingungen relevant sind. Das V-Modell, das nach [WC18] ebenfalls auf der Makroebene verortet ist, bildet die Ausgangsbasis für [Ull+24; Rud24]. Folglich sind alle Ebenen vertreten, wobei die Literatur zum Thema Auswahl datengetriebener Methoden vorrangig auf den unteren Ebenen zu finden ist.

Hinsichtlich dem **Typ des Prozessmodells** liegen vorwiegend prozedurale Modelle vor. Die Arbeiten [Meh+21; Ger+21c], die sich mit Voraussetzungen und Prozessanalyse beschäftigen, und daher den analytischen Modellen zugeordnet werden können, sind entsprechend die Ausnahme. Defizite hinsichtlich dem Typ des Prozessmodells im betrachteten Feld, sind folglich die fehlende Betrachtung von abstrakten Prozessmodellen aber auch von Management- und Operations-Research-Modellen. Eine Erklärung könnte der Neuheitsgrad datengetriebener Methoden sein, bei denen vorerst die Überführung in die praktische Anwendung im Fokus steht.

Prozessmodelle können auch hinsichtlich **Phasen** und oder **Aktivitäten** differenziert werden [GB12]. Beispiele für phasenbasierte Modelle sind in [Kle+18; Meh+21; Ull+24; Rud24] zu finden. Diese dienen zur klaren Strukturierung der Reihenfolge der Phasen auf Basis des CRISP-DM oder des V-Modells. Aktivitäten, die zeitlich auch mehrfach auftreten können, werden in [Ger+21c; Lic+21; Pic+24] betrachtet. Das ist insbesondere für die Auswahl von Methoden relevant. Allerdings fehlt dann ein klarer Bezug zum zeitlichen Verwendung einer Methode. Einige Arbeiten wie [Rie+20; GSW21; Son+23] weisen keinen direkten Bezug zu Aktivitäten oder Phasen des Produktentwicklungsprozesses auf, da sie den Fokus auf ML-Algorithmen oder die Struktur von Datenbanken setzen. Zudem gibt es Ansätze, wie [GGW23; Son+24b], die versuchen beide Sichtweisen zu vereinen. Beide Arbeiten sind in Bezug auf Aktivitäten und Phasen jedoch wenig spezifisch, da sehr allgemeingültige Vorgehensmodelle zur Verfügung gestellt werden.

Die **Auswahl von Methoden** ist im Rahmen der datengetriebenen Prozessmodelle nach Tabelle 4.1 wenig fokussiert worden. Mehlstäubl et al. [Meh+21] betont die Wichtigkeit der Auswahl bestmöglicher Methoden, schlägt jedoch kein konkretes Auswahlverfahren vor. Entsprechend liegt dort ein geringer Erfüllungsgrad vor. In Bezug auf die Ansätze, die sich mit der Auswahl beschäftigen, fokussieren sich [Rie+20; Lic+21; Son+23; Son+24b] vor allem auf die Auswahl bestmöglicher ML-Algorithmen. Im Kontext von Methoden im Betrachtungsrahmen dieser Arbeit, können die Algorithmen Teilschritte einer Methode sein. Entsprechend greifen die genannten Ansätze dahingehend etwas kurz, da ausbleibt, wie mit der Kombination von Algorithmen zu einer Methode, trainierten Metamodellen für projektübergreifenden Einsatz oder einer Datenvorverarbeitung umzugehen ist. In [GSW21; GGW23] wird eine Trennung zwischen Methode und Tool vorgenommen. Ein Vorgehen für den kombinierten Einsatz oder die Spezifikation von Kriterien anhand derer ein Einsatz einer Methode erfolgt bleibt aber auch hier unberücksichtigt. Defizite ergeben sich neben der Notwendigkeit für einen niedrighwelligeren Ansatz im Vergleich zu den Ontologien auch in Bezug auf die Berücksichtigung von kombinierten Methoden, Schnittstellen und Daten(vor)verarbeitung. Gerade letztere Aspekte sind mit starkem Bezug zu ML-Algorithmen selbst zu feingranular für das klassische Ingenieursfeld. Ebenfalls bieten LLMs in der heutigen Zeit vielfältige Möglichkeiten für eine einfache Interaktion und könnten potenziell beteiligte Ingenieure bei der Methodenauswahl unterstützen.

Der Aspekt der **Prozessintegration von Methoden** bezieht sich auf die Bereitstellung eines systematischen Vorgehens zur bestmöglichen Verwendung im Prozess. Das fokussiert sich einerseits auf Merkmale der Situation im Entwicklungsprozess, andererseits auch auf die Bereitstellung entsprechender Schnittstellen, um Daten und Informationen weiterzuverarbeiten. Beispiele hierfür sind der Anwendungszeitpunkt (frühe Phase oder späte Phase) und Voraussetzungen hinsichtlich Rechenkapazität, um eine Weiterverarbeitung zu ermöglichen. Interessanterweise wird dieser Aspekt auch in den datengetriebenen Entwicklungsprozessen, beispielsweise durch Konzepte für Schnittstellen und Benutzeroberfläche [Noz+17] oder Voraussetzungen [Meh+21], zumindest in Teilen beleuchtet. Die beispielhafte Erläuterung anhand von Use Cases wie in [Bic+19; Rud24; Ull+24] ist ebenfalls ein erster Schritt in Richtung Prozessintegration. Hinsichtlich der Ansätze zur situativen Auswahl bezieht [Son+24b] Entwicklungsphasen ein und [GGW23] führt eine Differenzierung zwischen Aktivität und Anwendung ein. Anhand welcher Kriterien die Anwendung stattfindet, wird allerdings nicht hinreichend beleuchtet. Die übrigen Arbeiten [Rie+20; Lic+21; GSW21; Son+23] legen den Fokus insbesondere auf Eigenschaften von ML-Algorithmen selbst. Entsprechend ist hier ein reduzierter Erfüllungsgrad hinsichtlich der Prozessintegration in Tabelle 4.1 eingetragen. Defizite sind folglich in Bezug auf die Verknüpfung von Methoden und Schnittstellen im Prozess vorhanden. Darüber hinaus

fehlen die klare Formulierung globaler Ziele und Ergebnisse, die durch den Einsatz von datengetriebenen Methoden erreicht werden sollen. Dies kann beteiligten Ingenieuren dabei helfen, neue Methoden zu akzeptieren und zu verwenden.

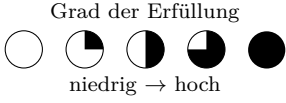
Die vorgestellten Ansätze unterscheiden sich deutlich im Aufwand und im erforderlichen Expertenwissen, die für einen erfolgreichen Einsatz in der Praxis notwendig sind. Entsprechend wird die **Einfache Anwendung** ebenfalls qualitativ bewertet. Ein vergleichsweise einfacher Einsatz ohne viel Expertenwissen ist mit den Ansätzen aus [Kle+18; Meh+21; Ger+21c] im Bereich datengetriebener Entwicklungsprozesse und [Rie+20; Lic+21; Son+23; Son+24b] im Bereich Methodenauswahl möglich. Entsprechend ist hier ein hoher Erfüllungsgrad in Tabelle 4.1 eingetragen. [Noz+17; Rud24; Ull+24] hingegen erfordern durch die höhere Betrachtungsebene ein etwas globaleres Verständnis. Der hohe Datenbezug in [Pic+24] und Berücksichtigung von Organisationsstrukturen in [Bic+19] machen auch diese Ansätze etwas komplexer in einer möglichen Umsetzung. Ontologiebasierte Ansätze wie in [GSW21; GGW23] ermöglichen einen hohen Detailgrad hinsichtlich Auswahl und Anwendung, allerdings ist hier ein erhöhter Aufwand vonnöten, um das System auf dem neuesten Stand halten, insbesondere wenn Fortschritte im Feld der KI auch in Zukunft schnell vonstattengehen werden. Defizite ergeben sich entsprechend aber auch dahingehend solche komplexen Ansätze – sofern diese für den betrachteten Anwendungsfall einen Mehrwert bieten – zu unterstützen. Ein Beispiel könnte die LLM-basierte Vorauffüllung der Datenstrukturen in [GSW21; Son+23] sein.

Datengetriebene Methoden und Produktentwicklung spannen auf übergeordneter Ebene betrachtet ein komplexes Schnittfeld auf. Entsprechend liegt die Überlegung nahe, Ansätze so zu gestalten, dass eine **Generalisierbarkeit** hinsichtlich weiterer Schnittfelder gegeben ist. Beispiele könnten der Einsatz von datengetriebenen Methoden in Bezug auf R-Strategien nach [KRH17] oder – noch allgemeingültiger – Leichtbaumethoden zur Unterstützung von Teilbereichen des Produktentstehungsprozesses nach Abbildung 2.1 auf S. 10 sein. Auch wenn diese Überlegungen grundsätzlich möglich sind und auch in Ontologien (vgl. [GGW23]) umsetzbar sein könnten, werden in keinem der betrachteten Ansätze Überlegungen zur Generalisierung durchgeführt. Lediglich der intrinsische Bezug zum Schnittfeld KI und Produktentwicklung ist gegeben, weshalb überall ein geringer Erfüllungsgrad gegeben ist. Da sich Erkenntnisse übertragen lassen, fehlt an dieser Stelle die Analyse des Potenzials einer Verknüpfung mit anderen Domänen und Bereichen.

Der **Bezug zu einem Anwendungsfeld** soll eine Aussage hinsichtlich der Berücksichtigung branchen- oder fachgebietsspezifischer Bedürfnisse treffen. Das Herausstellen dieser Bedürfnisse, beispielsweise durch spezifische Use Cases, lässt sich oft auf ähnliche Anwendungsfälle generalisieren und wird in allgemeingültigen Ansätzen wie in [Rie+20; Lic+21; Son+24b] nicht direkt thematisiert. In [GGW23] wird hingegeben ein Platzhalter

in Form eines Task Clusters definiert. Während [Ull+24] einen Bezug zur Entwicklung intelligenter und autonomer Systeme herstellt, fokussiert sich [Rud24] auf Anwendungsfälle im Flugzeug- und Fahrzeugbau und [Lic+21] auf Logistik. In [Bic+19] wird ein Beispiel aus der Plausibilitätsprüfung von FE-Simulationen verwendet. Grundlegende Ableitungen, wie das Anwendungsfeld davon grundsätzlich profitieren kann, ist bisher jedoch noch kein direkter Fokus.

**Tabelle 4.1:** Einordnung der Literatur zu datengetriebener Produktentwicklung

		Ebene des Prozessmodells	Typ des Prozessmodells	Aktivitäten- vs. Phasenbasiert	Auswahl von Methoden	Prozessintegration von Methoden	Einfache Anwendung	Generalisierbarkeit	Bezug zu einem Anwendungsfeld
		Grad der Erfüllung  niedrig → hoch							
Datengetriebene Produktentwicklungsprozesse Abschnitt 3.1.1	Klement et al. [Kle+18]	MI	PR	P	○	○	◐	◑	○
	Mehlstäubl et al. [Meh+21]	MI	AN	P	○	◐	◑	◑	○
	Gerschütz et al. [Ger+21c]	MI	AN	A	○	○	◐	◑	○
	Ullrich et al. [Ull+24]	MA	PR	P	○	◐	◑	◑	◐
	Rudolph [Rud24]	MA	PR	P	○	◐	◑	◑	◐
	Nozaki et al. [Noz+17]	MA	PR	-	○	◐	◑	◑	○
	Pickel et al. [Pic+24]	ME	PR	A	○	○	◐	◑	○
	Bickel et al. [Bic+19]	ME	PR	P	○	◐	◑	◑	◐
Situative Auswahl von ML-Methoden Abschnitt 3.1.2	Riesener et al. [Rie+20]	MI	PR	-	◐	◐	◑	◑	○
	Lickert et al. [Lic+21]	MI	PR	A	◐	◐	◑	◑	◐
	Sonntag et al. [Son+23]	MI	PR	-	◐	◐	◑	◑	○
	Sonntag et al. [Son+24b]	MI	PR	A+P	◐	◐	◑	◑	○
	Gerschütz et al. [GSW21]	MI	PR	-	◐	◐	◑	◑	○
	Gerschütz et al. [GGW23]	MA	PR	A+P	◐	◐	◑	◑	○

MI = mikro, ME = meso, MA = makro, PR = prozedural, AN = analytisch, AB = abstrakt, A=aktivitätenbasiert, P=phasenbasiert

### 4.1.2 Datengetriebene Optimierung

Eine Einordnung relevanter Literatur im Feld der datengetriebenen Optimierung crash-belasteter Systeme ist in Tabelle 4.2 auf S. 76 zu finden. Die in Abschnitt 3.3 ab. S. 53 diskutierten Ansätze werden anhand der Kriterien *Verwendung von RL-Agenten*, *Optimierer steuert Metamodell* und *Optimierer steuert FE-Simulation*, *Wissenstransfer im Entwicklungsprozess* und *Integration in den Entwicklungsprozess* sowie *industrienahes Evaluationsszenario* eingeordnet. Der Erfüllungsgrad der genannten Aspekte im Kontext der Forschungsfrage *ii. Wie lässt sich die generations- und aufgabenstellungsübergreifende Systemauslegung optimieren?* ist Inhalt von Tabelle 4.2.

Die **Verwendung von RL-Agenten** liegt vier der 15 betrachteten Ansätze als Optimierungsmethode zugrunde. Die übrigen fokussieren auf klassische Optimierung, beispielsweise mit evolutionären Algorithmen [Hua+15; Büt+23] oder Expertenwissen durch Heuristiken [Ort+21]. Auch eine manuelle Optimierung anhand eines interaktiven Ergebnisraums ist möglich [Thi+06; JGK21]. Im Bereich der Rückhaltesystemoptimierung liegen bisher keine RL-basierten Ansätze vor. Hier ergeben sich dementsprechend auch Defizite hinsichtlich dem betrachteten Anwendungsfeld.

In den Ansätzen aus dem Stand der Technik lässt sich differenzieren, ob der **Optimierer** (RL-Agent, klassischer (evolutionärer) Optimierungsalgorithmus oder auch interaktiv durch beteiligte Ingenieure) mit dem **FE-Modell** selbst oder einem **Metamodell interagiert**. In den Arbeiten [BGS23; BGS24a; BGS24b] interagiert der RL-Agent ausschließlich mit einem Metamodell, analog zu [Büt22; Büt+23] im Kontext der multidisziplinären Optimierung. Für die Metamodelloptimierung in [Thi+06; Büt+23] wird ebenfalls eine sukzessive Reduktion des Designraums durchgeführt, um erforderliche Simulationsdaten zu reduzieren. Alle übrigen RL-basierten Ansätze interagieren jedoch direkt mit FE-Simulationen, die entweder auf Basis der Aktionen generiert und berechnet werden [HO20; HO22; Bro+22] oder aus einer Datenbank gezogen werden [TSZ22; TSZ24]. Da Joodaki, Gepner und Kerrigan [JGK21] keinen automatischen Optimierer verwendet, sondern nur darauf verweist, wie mit den trainierten Modellen weiter verfahren werden kann, ist hier der Erfüllungsgrad nur halb eingetragen. Defizite ergeben sich hier vor allem durch die Kombination der Ansätze, beispielsweise ein RL-Agent der mit Metamodellen und FE-Simulationen interagiert.

Hinsichtlich der in der Forschungsfrage aufgegriffenen Produktgenerationsentwicklung ist relevant, wie **Wissenstransfer im Entwicklungsprozess** realisiert werden kann. Die metamodellbasierten Ansätze erlauben es, lokal – für eine spezifische Problemstellung – Abschätzungen zu treffen, ohne dass eine rechenintensive FE-Simulation durchgeführt werden muss. Tritt eine neue Problemstellung auf, muss solch ein Prozess, wie beispielsweise in [CDP05; Thi+06; Büt+23], erneut durchlaufen werden. Neben den Modellen selbst,

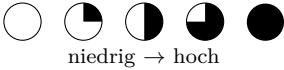
sind auch die Optimierungsstrategien, beispielsweise von evolutionären Algorithmen wie in [Hua+15; Hor17], nicht übertragbar. Die Effizienzsteigerung kann hier in erster Linie durch die Automatisierung generiert werden.

Auf Basis der trainierten Strategie konnten beispielsweise Strukturen mit unterschiedlichen Maßen und Randbedingungen (vgl. [Bro+22; HO22]) optimiert werden. Entsprechend kann bei erneuter Anwendung von bereits antrainierten Strategie ein Auslegungsprozess mittels maschinell abgebildetem Wissen in Form einer Entscheidungsstrategie optimiert werden. Indirekter Wissenstransfer ist über die Ermittlung und Visualisierung von Zusammenhängen, durch Plots [Thi+06; JGK21] oder Sensitivitäten [Büt+23; HCK24] möglich. Diese müssen jedoch erst wieder durch einen Ingenieur manuell interpretiert werden, weshalb in Tabelle 4.2 ein reduzierter Erfüllungsgrad angegeben wird. Entsprechend ergeben sich in Bezug auf den Wissenstransfer noch Potenziale für zukünftige Arbeiten, einerseits auf der konzeptionellen Ebene durch passende Prozessmodelle, aber auch in den Methoden selbst. Intelligente Agenten könnten wiederkehrende Probleme in der Dimensions- oder Formoptimierung durch den Einsatz von Strategien effizient lösen. Ingenieure würden in diesem Szenario nur noch eine überwachende Rolle spielen.

Eine Grundlage für den Wissenstransfer, ist die bestmögliche **Integration in den Entwicklungsprozess** der einzelnen Methoden. Defizite ergeben sich hier beispielsweise in Bezug auf die Betrachtung von Konzepten wie der Produktgenerationsentwicklung (vgl. Abschnitt 2.1.2 auf S. 14), die die Grundlage für die große Mehrheit der heute am Markt verfügbaren Produkte bildet. Die erfolgreiche Integration kann einen Beitrag zur verbesserten Bereitstellung und Auswahl von Methoden in der praktischen Anwendung ermöglichen. Das kann ebenfalls durch die Durchführung von Benchmarks mit konventionellen Optimierern wie in [HO20; HO22; Bro+22; TSZ24] unterstützt werden, um Grenzen zu identifizieren. Zur Integration muss ebenfalls der Umgang mit hohen Rechenzeiten komplexer FE-Simulationen, auf die in [HO20; HO22; Bro+22] lediglich verwiesen wird, detailliert beleuchtet werden. Eine Möglichkeit dieser zu begegnen ist die in eigener vorangegangenen Arbeit [Mat23] vorgeschlagene und in [BGS23; Mat+24; BGS24a; BGS24b] vertiefte Möglichkeit Metamodelle zum Training der RL-Agenten heranzuziehen. Eine erneute Anwendung der vortrainierten Strategie wird hingegen nur in eigenen Arbeiten [Mat23] vorgeschlagen, jedoch nicht validiert. Trotz der Durchführung erster Benchmarks, sollten zur vertieften Bewertung komplexere und disziplinsübergreifende Use Cases auf höheren Systemebenen betrachtet werden. Bei der Auswahl eines geeigneten Verfahrens kann das die Entscheidungsfindung unterstützen.

Um die Eignung für die Praxis bestmöglich zu zeigen, muss ein **industrienahes Evaluationsszenario** verwendet werden. Methodische Grenzen, wie beispielsweise Limitierungen aufgrund der Modellgröße und Komplexität lassen sich so herausstellen.

**Tabelle 4.2:** Einordnung der Literatur zu datengetriebener Optimierung

		Verwendung von RL-Agenten	Optimierer steuert Metamodell	Optimierer steuert FE-Simulation	Wissenstransfer im Entwicklungsprozess	Integration in den Entwicklungsprozess	Industrienahes Evaluationszenario
Grad der Erfüllung  niedrig → hoch							
Optimierung der Strukturintegrität Abschnitt 3.3.1	Büttner et al. [Büt22; Büt+23]						
	Craig et al. [Cra+02]						
	Cadete et al. [CDP05]						
	Beyer et al. [BSS21]						
	Ortmann et al. [Ort+21]						
	Hayashi und Ohsaki [HO20; HO22]						
	Brown et al. [Bro+22]						
	Borse et al. [BGS23; BGS24a; BGS24b]						
	Trilling et al. [TSZ22; TSZ24]						
Optimierung der Rückhaltesysteme Abschnitt 3.3.2	Huang et al. [Hua+15]						
	Heo et al. [HCK24]						
	Thiele et al. [Thi+06]						
	Horii [Hor17]						
	Xue et al. [Xue+12]						
	Joodaki et al. [JGK21]						

Während in [BGS23; BGS24a; BGS24b] oder [TSZ22; TSZ24] vergleichsweise kleine Simulationsmodelle verwendet werden, um die Methoden zu validieren, sind die Standardmodelle in der Industrie weitaus größer. Diese werden in den Arbeiten von [Cra+02; CDP05; Thi+06; JGK21; Büt+23; HCK24] verwendet. In Tabelle 4.2 liegt für diese entsprechend ein hoher Erfüllungsgrad vor. Defizite ergeben sich hier insbesondere bei den RL-basierten

Ansätzen, die ausschließlich auf akademische Modelle, wie eine einfache Crashbox [BGS24b], zurückgreifen.

### 4.1.3 Datengetriebene Analyse

Eine Einordnung relevanter Literatur im Feld der datengetriebenen Analyse von FE-Simulationsdaten ist in Tabelle 4.3 auf S. 80 zu finden. Die in Abschnitt 3.4 ab S. 60 diskutierten Ansätze werden anhand der Kriterien *Simulationstyp*, *Lernverfahren*, *Datenverarbeitung (FE-Daten, (Zeit-)Seriendaten und Skalare)*, *Analysezweck (Cluster-/Ausreißer-, Kausalitäts- und Einfluss-/Beitrags-Analyse)* sowie *automatischer Prozess* eingeordnet. Der Erfüllungsgrad der genannten Aspekte im Kontext der beiden Fragen *iii. Wie lässt sich auffälliges Systemverhalten identifizieren und analysieren?* und *iv. Wie lassen sich Einflussgrößen des Systemverhaltens identifizieren und analysieren?* wird betrachtet.

In Tabelle 4.3 wird in der ersten Spalte der den Betrachtungen zugrundeliegende **Simulationstyp** differenziert. Neben der erhöhten Komplexität dynamischer Systeme, unterliegt dieses Feld stärkerer Regulierung durch Gesetze und Verbraucherschutz, was höhere Entwicklungsaufwände begründet. Defizite können hinsichtlich der detaillierten Betrachtung von datengetriebenen Analysemethoden für gekoppelte Multiphysik-Simulationen aber auch die domänenübergreifende Betrachtung identifiziert werden. Ein Beispiel für die domänenübergreifende Betrachtung kann die intelligent vernetzte Anomaliedetektion in Crash- und Steifigkeitsanalyse eines Systems sein.

Hinsichtlich der eingesetzten **Lerntypen** sind ausschließlich Supervised und Unsupervised Learning vorzufinden. Hervorzuheben ist hier insbesondere [PGS23; PGS25], da dies der einzige Ansatz auf Basis der Graphentheorie ist. Dieser basiert auf einem Ähnlichkeitsmaß, das Knoten in einer Graphendatenbank vergleicht. Daher erfolgt die Zuordnung zum Unsupervised Learning. Der Hauptfokus relevanter Ansätze liegt im Bereich der Dimensionsreduktion, die ebenfalls dem Unsupervised Learning zuzuordnen ist. Lediglich zwei der Supervised-Learning-basierten Ansätze sind im Feld der Simulation dynamischer Systeme verortet. RL-basierte Ansätze, die beispielsweise zur interaktiven Analyse und Dokumentation des Systemverhaltens eingesetzt werden könnten, sind in diesem Kontext noch kein zentraler Gegenstand der Literatur. Dasselbe gilt für Verfahrenskombinationen wie Self-Supervised Learning.

Die meisten Methoden fokussieren sich dabei auf die Verarbeitung von **FE-Daten** (vgl. Abbildung 3.2 auf S. 50), da dies der grundlegendste Datentyp ist. Im Rahmen der FE-Daten können verschiedene Feldgrößen wie Knotenverschiebungen [IG19; HIG20], aber auch plastische Dehnungen [Kra+23] oder Spannungen [BGW23] verarbeitet werden. Der in Tabelle 4.3 dargestellte Erfüllungsgrad entspricht der Flexibilität des Ansatzes beliebige geometrische Formen zu analysieren. Dieser Aspekt wird nicht in jeder Methode abgedeckt,

ist jedoch für die praktische Anwendung relevant, da sich Formen, insbesondere in frühen Designphasen, kurzfristig und grundlegend ändern können. Selbiges gilt für die Verarbeitung verschiedener Elementtypen wie Oberflächen- (Shell) und Volumenelementformulierungen (Solid). Um diesen Aspekten gerecht zu werden, verwenden [KGS20; Kra+23] Voxelgitter und [BGW23] eine sphärische Projektionstechnik. Für die übrigen Methoden, in denen FE-Daten verarbeitet werden, aber keine genaueren Informationen zu diesem Aspekt enthalten, liegt ein halber Erfüllungsgrad vor. Hahner et al. [HIG20; Hah24] behandeln semireguläre Oberflächennetze und eine Grenzformbasis in späteren Arbeiten, jedoch nicht direkt im Bereich der FE-Simulationsanalyse. Bisher bezieht jedoch keine der Methoden mehrere Attribute wie Materialdaten und Knotenverschiebungen mit ein. Ebenso gibt es in diesem Kontext keine Verwendung von Graph Neural Networks, obwohl diese schon zur Prognose von Simulationsdaten eingesetzt wurden [Pfa+21; Kne+25].

Lediglich zwei als relevant identifizierte Ansätze diskutieren die Analyse von **1D-Zeitreihendaten**, wobei sich deren Betrachtungsebene deutlich unterscheidet. Während in [Iza+23] – analog zu eigenen vorangegangenen Arbeiten [MDV23] – die Analyse von Datensätzen durch eine dimensionsreduzierte Repräsentation lokal im Entwicklungsprozess im Mittelpunkt steht, fokussiert der graphenbasierte Ansatz von [PGS23; PGS25] auf die Exploration großer Datenmengen mittels Energiefeatures, beispielsweise auf Projektebene. Da in [PGS23; PGS25] hinsichtlich der Zeitserienanalyse ausschließlich Energiefeatures einbezogen werden, ist der in [IG19] vorgestellte Ansatz weitaus allgemeingültiger, weshalb für ersteren ein reduzierter Erfüllungsgrad in Tabelle 4.3 berücksichtigt wird. Eine Kombination von Dimensionsreduktion und Merkmalsextraktion, aber auch die Berücksichtigung und Umgang mit verschiedenen Einheiten, beispielsweise bei simultaner Verarbeitung von Kraft und Beschleunigungssensoren, sind bisher ebenfalls nicht hinreichend betrachtet worden. Ebenfalls kann die Betrachtung von Supervised-Learning-Modellen an dieser Stelle interessant sein. In [KWB18; KB18] werden diese Daten zur Prognose von Bauteilen und Aufprallpunkten verwendet – eine Übertragung auf die FE-Simulationsanalyse liegt jedoch nahe.

Die ausschließliche Verwendung von **skalaren Daten**, die geometrische Formen und resultierende Verformungen einer statischen Simulation beschreiben, wird in der Arbeit von [CK20] behandelt. Da diese Daten vergleichsweise einfach zu verarbeiten sind, ergeben sich hier wenig neue Aspekte die Gegenstand der Forschung sind.

Zusammenfassend lässt sich basierend auf Tabelle 4.3 feststellen, dass eine kombinierte Datenverarbeitung – wenn auch nur teilweise – in [Die+18] und [Ack+08] vorgeschlagen wird. Beide beziehen skalare Merkmale, die eine geometrische Form beschreiben, auf ein bestimmtes Verhalten der Stoßstange bzw. U-Struktur, jeweils in einer niedrigeren Dimension dargestellt. Die niedrigdimensionale Repräsentation der Knotenverschiebungen

wird dabei als Ausgabe verwendet. Entsprechend lassen sich Defizite auch im Bereich der kombinierten Datenverarbeitung feststellen.

Bezüglich des Analyseziels konzentrieren sich die meisten Ansätze vollständig auf die **Cluster- und Ausreißeranalyse**. Dies wird durch die Verwendung von Dimensionsreduktion wie in [IG19; Kra+23], aber auch durch die Prognose von Labels wie in [BGW23] erreicht. Ein kleiner Erfüllungsgrad wird angenommen, wenn ein Ansatz sich als Erweiterung zur Ausreißeranalyse verwenden lässt. Dies umfasst beispielsweise das Bereitstellen von Regeln, die zu Ausreißern führen [Die+18]. Die Methode fokussiert sich jedoch nicht explizit auf deren Auffindung. Gerade im klassischen Ingenieurskontext fehlt jedoch oft nötige Expertise im Bereich ML und KI [GGW23; Son+24b], was die Interpretation der Einbettungsräume und die darin abgebildeten Zusammenhänge schwer verständlich macht. Defizite herrschen daher auch in der intuitiven und ingenieurmäßigen Darstellung der Ergebnisse vor – beispielsweise unter Einbezug physikalischer Einheiten.

Deutlich weniger Methoden behandeln die (qualitative) Bestimmung von Hauptursachen in Simulationseingabe und -ausgabe für bestimmtes Verhalten im Rahmen einer **Ursachenanalyse**. Hierbei wird ein reduzierter Erfüllungsgrad angenommen, wenn diese durch die Bestimmung von Korrelationen zwischen skalaren Parametern, die die geometrische Form und Deformationsmodi beschreiben [Ack+08], Hauptkomponenten [Tho+10], Ausreißerkennwerten [Kra+23] oder einer intelligenten Definition von Klassenlabels [BGW23] gewonnen werden. Obwohl dies keine kausale Ursachenanalyse im mathematischen Sinne ist, werden tiefgreifendere Informationen als bei der klassischeren Ausreißeranalyse bereitgestellt. Dies erleichtert die Bestimmung möglicher Ursachen für auffälliges Systemverhalten in praktischen Anwendungen. Eine Zwischenlösung ist beispielsweise die Verwendung eines Entscheidungsbaums [Die+18], der komplexere Zusammenhänge modellieren kann. Oft sind diese Modelle jedoch eher ungeeignet, da die betrachteten Probleme in der Auslegung crashbelasteter Systeme nichtlinear sind. Kausale Analysen können durch die Verwendung kausaler Modelle wie Bayesian Networks durchgeführt werden, wie in [CK20] initial demonstriert. Eine umfassendere Betrachtung kausaler Modelle oder auch die Verwendung von interpretierbarem ML zur Analyse ist bisher noch in keinem der Ansätze ein zentraler Gegenstand.

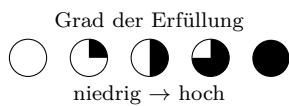
**Beitragsanalysen** werden als quantitative Bestimmung von Merkmalsbeiträgen verstanden, insbesondere in der Simulationseingabe, da dies eine intuitive Ableitung von Designänderungen ermöglicht. Diese Art von Information wird beispielsweise durch einen Entscheidungsbaum in [Die+18] bereitgestellt, der auch den globalen Beitrag von Merkmalen zu einem bestimmten Ergebnis aufzeigt. Eine tiefgreifendere Analyse unter Einbeziehung von Wahrscheinlichkeitsverteilungen wird durch das Bayesian Network von Conti und Kajima [CK20] ermöglicht. Obwohl dieser Aspekt in [CK20] nicht umfassend behandelt wird,

erlauben bedingte Wahrscheinlichkeiten die Nachverfolgung des Einflusses von Variablen auf die Wahrscheinlichkeit verschiedener Ergebnisse, aus denen Merkmalsbeiträge abgeleitet werden können. Im einfachsten Fall beziehen sich Beiträge von Eingangsmerkmalen zu potenziellen Ursachen auf Korrelationskoeffizienten (vgl. [Ack+08]). An dieser Stelle ergibt sich eine Schnittmenge mit den im Rahmen von Abschnitt 2.2.2 auf S. 27 diskutierten Methoden des interpretierbaren ML und der globalen Sensitivitätsanalyse, die bisher in diesem Kontext, abseits von inhärent interpretierbaren Modellen wie den Decision Trees [Die+18], in diesem Kontext ungeachtet bleiben.

**Tabelle 4.3:** Einordnung der Literatur zu datengetriebener Analyse

		Simulationstyp	Lernverfahren	$\mathcal{G}(t, \xi)$ – FE-Daten	1D – (Zeit-)Serien Daten	0D – Skalare	Cluster- und Ausreißeranalyse	Ursachenanalyse	Beitragsanalyse	Automatischer Prozess	
Skalare Abschnitt 3.4.1	Ackermann et al. [Ack+08]	DY	UL	◐	○	●	◐	◐	◐	○	
	Conti und Kajijima [CK20]	ST	SL	○	○	●	◐	◐	◐	○	
	Diez et al. [Die+16; Die+18]	DY	SL	◐	○	●	◐	◐	◐	◐	
1D-Zeitreihe Abschnitt 3.4.2	Iza Teran et al. [Iza+23]	DY	UL	○	◐	○	◐	○	○	○	
	Pakiman et al. [PGS23; PGS25]	DY	UL	○	◐	○	◐	◐	○	○	
Finite Elemente Abschnitt 3.4.3	Thole et al. [Tho+10]	DY	UL	◐	○	○	◐	◐	◐	○	
	Bohn et al. [Boh+13]	DY	UL	◐	○	○	◐	○	○	○	
	Iza Teran [Iza14]	DY	UL	◐	○	○	◐	○	○	○	
	Iza Teran und Garcke [IG19]	DY	UL	◐	○	○	◐	○	○	◐	
	Kracker et al. [KGS20; Kra+23]	DY	UL	◐	○	○	◐	○	○	●	
	Hahner et al. [HIG20]	DY	SL	◐	○	○	◐	○	○	○	
	Bickel et al. [BGW23]	ST	SL	◐	○	○	◐	◐	○	◐	

DY = Dynamisch, ST = Statisch, UL = Unsupervised Learning, SL = Supervised Learning



Ein **automatischer Prozess** bezieht sich auf den Automatisierungsgrad der vorgestellten Methode. In [BGW23; Kra+23] wird dieser Aspekt des Ansatzes hervorgehoben. Kracker [Kra24, S. 91–92] nimmt zudem eine detaillierte Betrachtung des Schwellwertes vor, um Ausreißer zuverlässig zu finden. In keiner der betrachteten Arbeiten wird näher darauf eingegangen, wie Ergebnisse möglichst nutzenorientiert dargelegt werden können. Dazu zählt beispielsweise die Kombination aus automatischer und interaktiver Betrachtungsweise sowie die KI-basierte Ergebnisdarstellung.

## 4.2 Konkretisierung der Forschungsfragen

Entsprechend des Aufbaus der Präskriptiven Studie nach Abbildung 1.1 auf S. 6 verteilen sich die Beiträge dieser Arbeit auf vier Teilbereiche, deren Richtung bereits die Forschungsfragen vorgeben.

Zur Beantwortung der Frage *i. Wie lässt sich die situative Auswahl und Integration von Methoden in Entwicklungsprozesse unterstützen?* werden die folgenden Beiträge erarbeitet:

- Einführung einer allgemeinen und abstrakten Betrachtungsweise zur Orientierung in komplexen Schnittfeldern.
- Entwicklung eines modularen Vorgehensmodells für die datengetriebene Produktentwicklung am Beispiel der Auslegung crashbelasteter Systeme in der passiven Fahrzeugsicherheit zur anwendungsbezogenen Darstellung von Zusammenhängen hinsichtlich Aktivitäten und Phasen sowie zur Integration von Methoden.
- Herausstellung von Merkmalen, anhand derer eine einfache, zielgerichtete Einordnung und Auswahl von Methoden entsprechend der Situation im Entwicklungsprozess ermöglicht wird.
- Sicherstellung der allgemeinen Anwendbarkeit durch eine Generalisierungsbetrachtung anhand eines weiteren Schnittfelds.

Zur Beantwortung der Frage *ii. Wie lässt sich die generations- und aufgabenstellungsübergreifende Systemauslegung optimieren?* werden die folgenden Beiträge erarbeitet:

- Entwicklung einer Methode zur datengetriebenen Systemauslegung für große FE-Simulationsmodelle, die den Übertrag einer Auslegungsstrategie auf ähnliche Entwicklungsaufgaben – auch in verschiedenen Produktgenerationen – ermöglicht.
- Prozessintegration der Methode durch die Effizienzsteigerung des Trainingsprozesses und den Schulterschluss zur Produktgenerationsentwicklung.
- Anwendung in industrienahen Evaluationsszenarien im Bereich der Rückhaltesystemauslegung in der passiven Fahrzeugsicherheit.

Zur Beantwortung der Frage *iii. Wie lässt sich auffälliges Systemverhalten identifizieren und analysieren?* werden die folgenden Beiträge erarbeitet:

- Entwicklung einer Methode zur datengetriebenen Analyse von Crashesimulationen basierend auf 1D-Zeitseriendaten (hier: Crashsensordaten) diverser Messpunkte.
- Detektion von Ausreißern über verschiedene Sensoren in allen betrachteten Crashesimulationen.
- Identifikation möglicher Ursachen für die Ausreißer über die Rekonstruktion einer Eventkette.
- Systematische und anwenderfreundliche Darstellung der Ergebnisse innerhalb eines voll automatischen Prozesses.
- Anwendung in industrienahen Evaluationsszenarien im Bereich der Auslegung von Insassenzelle und Rückhaltesystem.

Zur Beantwortung der Frage *iv. Wie lassen sich Einflussgrößen des Systemverhaltens identifizieren und analysieren?* werden die folgenden Beiträge erarbeitet:

- Entwicklung einer Methode zur modularen Verarbeitung von FE-, 1D-Zeitserien- und skalaren Daten – unabhängig vom Entwicklungsstand oder der Geometrie der betrachteten Simulationsmodelle.
- Übertrag von interpretierbarem ML in das Feld der Simulationsdatenanalyse zur automatischen Erklärung von Zusammenhängen in Crashesimulationen.
- Bereitstellung intuitiver Ergebnisdarstellungen, die Systemansichten sowie einen datengetriebenen A/B-Vergleich ermöglichen.
- Anwendung in industrienahen Evaluationsszenarien im Bereich der Auslegung von Insassenzelle, Rückhaltesystem und Fahrzeugstruktur.

## 4.3 Abgrenzung der Arbeit

Die Abgrenzung dieser Dissertation knüpft an die Eingrenzung in Abschnitt 1.3.4 auf S. 6 an. Diese ergibt sich vor allem durch die Entwicklung wissenschaftlich neuer Methoden mit klarem Fokus auf praktischen Mehrwert und Anwendungsbezug. Einerseits werden hierzu aktuelle ML-Algorithmen mit Anforderungen der industriellen Praxis zusammengeführt. Dies umfasst sowohl die Sicherstellung eines varianten- und generationsübergreifenden Einsatzes als auch die Berücksichtigung praxisrelevanter Fragestellungen, die bisher in der Forschung unzureichend adressiert wurden. Das ist beispielsweise die Identifikation möglicher Ursachen für das Systemverhalten einzelner Datenpunkte, da diese Gegenstand alltäglicher Arbeit sind. Andererseits wird durch das Vorgehensmodell eine methodische Grundlage zur Orientierung innerhalb des Schnittfelds vorgeschlagen. Das ist aus Sicht der Produktentwicklung der Ausgangspunkt für alle weiteren Betrachtungen. Wie später vertieft, erweisen sich insbesondere diese Aspekte häufig als Hemmnis für die Einführung neuer und verbesserter ML-Methoden und Software-Tools.

---

Im Gegensatz zu wertvollen wissenschaftlichen Beiträgen, insbesondere aus Mathematik und Informatik, auf denen die im Rahmen dieser Arbeit erarbeiteten Methoden aufbauen, liegt hier der Fokus auf einer generalistischeren Betrachtung. Dadurch können Ergebnisse aus dem gesamten Spektrum verschiedener Lerntypen des ML zusammengeführt und Potenziale für den Einsatz in der Auslegung crashbelasteter Systeme ganzheitlich abgeleitet werden. Die gewonnenen Erkenntnisse bilden anschließend einen idealen Ausgangspunkt für weiterführende und spezifischere Fragestellungen – auch im Bereich der Algorithmen selbst.



## 5 Einführung

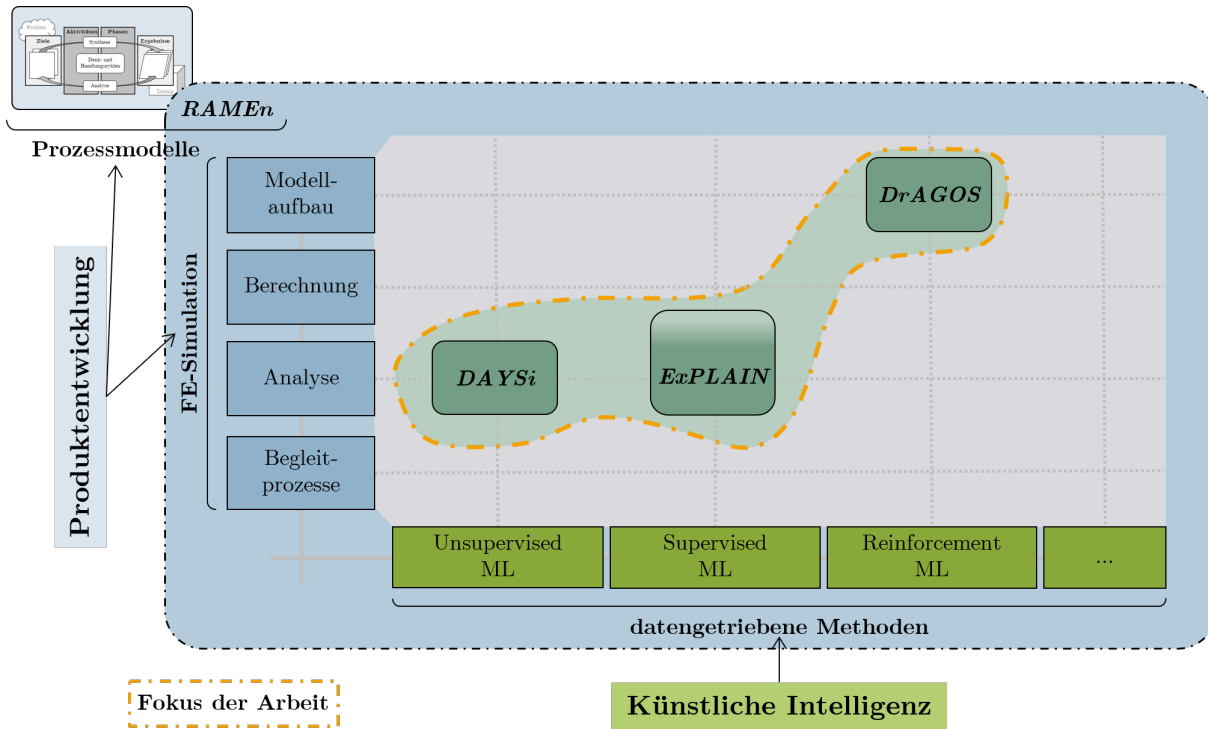
Die in diesem Kapitel abgehandelten Aspekte entsprechen der Präskriptiven Studie (Phase III) des Aufbaus nach der DRM in Abbildung 1.1 auf S. 6. Zur besseren Lesbarkeit werden die folgende Akronyme eingeführt:

- **VoR**gehensmodell zur situativen **A**uswahl und Prozessintegration datengetriebener **M**ethoden in **E**ntwicklungsprozesse zur Auslegung crashbelasteter Systeme (*RAMEn*)
- **D**atengetriebene **A**ufgaben- und **G**enerationsübergreifende **O**ptimierung von **S**ystemen (*DrAGOS*)
- **D**atengetriebene **A**nalyse von auffälligem **S**ystemverhalten in Crashsensor**S**ignalen (*DAYSi*)
- **D**atengetriebene **E**rklärung und **V**ergleich von Systemverhalten (*ExPLAIN*)

Das Vorgehensmodell wird in Abschnitt 5.1 ab S. 87 eingeführt. Die Methoden zur datengetriebenen Systemauslegung, der Analyse von Crashsensordaten sowie der Erklärung und dem Vergleich von Systemverhalten werden in den darauffolgenden Sektionen Abschnitt 5.2 ab S. 94, Abschnitt 5.3 ab S. 99 und Abschnitt 5.4 ab S. 107 vorgestellt. Während der Mehrwert der Ansätze in Kapitel 6 ab S. 117 ausführlich evaluiert wird, fokussiert sich dieses Kapitel vorwiegend auf die Konzeption und die theoretischen Aspekte.

*RAMEn*, *DrAGOS*, *DAYSi* und *ExPLAIN* sind im Kontext des Schnittfeldes, das bereits zum Beginn des Stands der Technik in Kapitel 3 ab S. 43 aufgespannt wurde, in Abbildung 5.1 zum besseren Verständnis eingeordnet.

Mit den drei Methoden, die sich im gesamten Schnittfeld verteilen, werden konkrete Möglichkeiten aufgezeigt, Entwicklungsprozesse in der FE-Simulation zu optimieren. Auf übergeordneter Ebene wird mit *RAMEn* zudem eine theoretische Grundlage bereitgestellt wie solche Methoden systematisch eingesetzt werden können. Durch den verbesserten Einsatz von Methoden und das zur Verfügung gestellte Wissen, lassen sich anschließend weitere Effizienzpotenziale heben. Aus Abbildung 5.1 geht ebenfalls der Fokus der vorliegenden Arbeit hervor. Dieser orientiert sich insbesondere an den Alltagstätigkeiten eines Berechnungsingenieurs, da Pre- und Postprocessing etwa 50 % bzw. 30 % der Arbeitszeit zur Bearbeitung von FE-Problemen in Anspruch nehmen [Kle15, S. 7]. Im Erstausrüster-Umfeld werden jedoch die klassischen FE-Modellierungstätigkeiten im Preprocess oft ausgelagert, weshalb hier verstärkt Analysemethoden betrachtet werden. Die Grundlage für die entwickelten Methoden bilden die im Entwicklungsprozess anfallenden Daten, mit denen auch die Ingenieure selbst alltäglich interagieren.



**Abbildung 5.1:** Eingeführte Methoden im Kontext des Stands der Technik

Des Weiteren ist davon auszugehen, dass perspektivisch auch vortrainierte Grundlagenmodelle für CAE im Kontext des Berechnungsprozesses eine Rolle spielen werden. Dieser beansprucht etwa 70% der Rechenressourcen [Kle15, S. 7]. Die Modellaufbau und Analysetätigkeiten fallen jedoch auch weiterhin an und könnten durch zunehmende Variantenvielfalt, die perspektivisch in Echtzeit berechnet werden könnten, sogar vermehrt Zeit in Anspruch nehmen. Das macht die entwickelten Methoden generalisiert einsetzbar. Die in Abbildung 5.1 gezeigten Begleitprozesse sind vor allem für IT-Infrastruktur oder für Simulationssoftwarehersteller relevant. Beispielsweise können mittels KI eine automatische Dokumentation oder KI-gestützte Anleitungen [Ans24] realisiert werden. Im Kontext dieser Arbeit ist insbesondere die Einbindung solcher Ansätze in die Entwicklungsprozesse gefragt und weniger die Entwicklung selbst.

## 5.1 *RAMEn* – Methodenauswahl

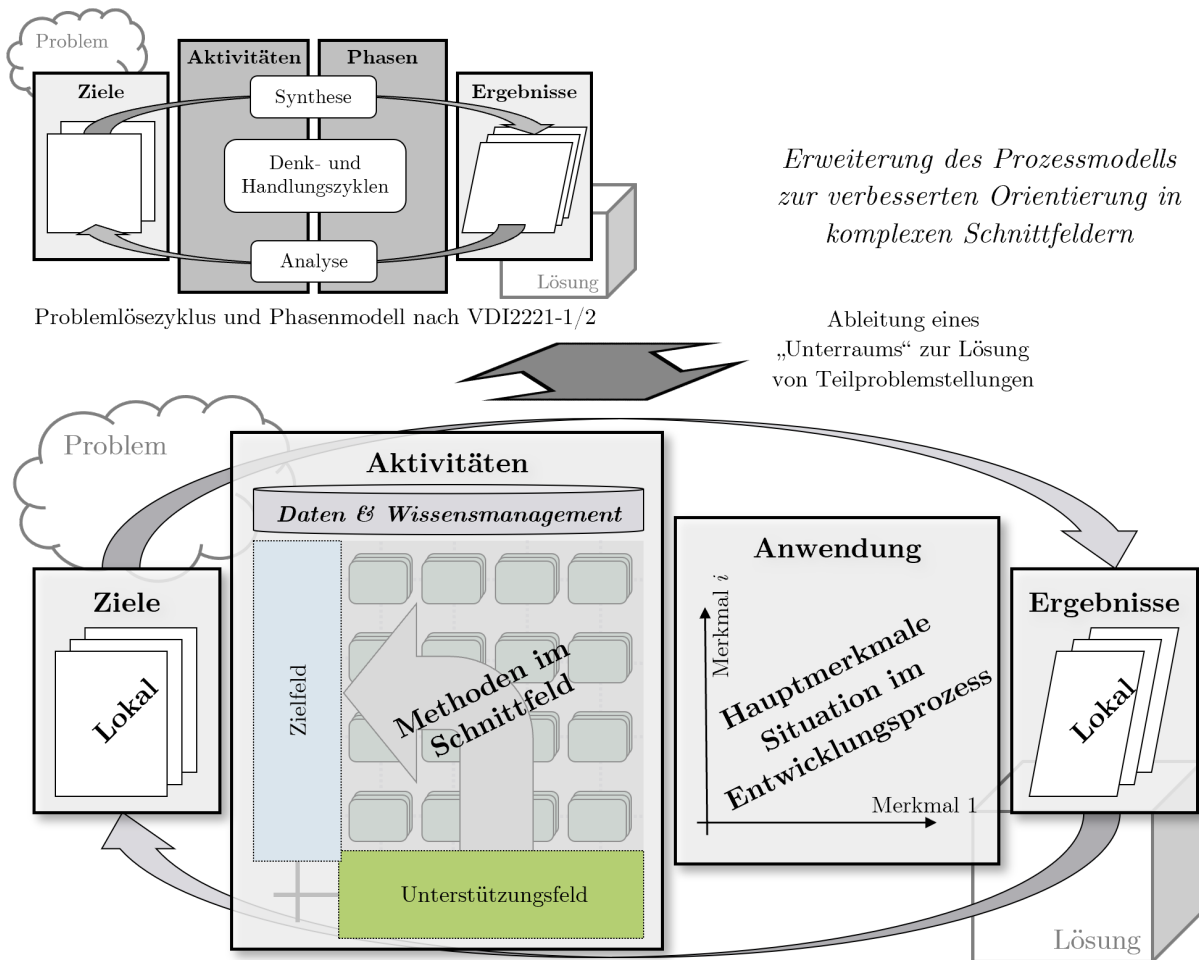
Ausgehend von Abbildung 5.1 soll durch *RAMEn* ein Vorgehensmodell bereitgestellt werden, das in komplexen Schnittfeldern die Orientierung verbessert. Beteiligte Ingenieure werden im Produktentwicklungsprozess situativ unterstützt, neue Methoden einzuordnen und passende auszuwählen. Das gilt insbesondere für *DrAGOS*, *DAYSi* und *ExPLAIN*, soll aber auch auf andere Methoden aus dem Stand der Technik gültig sein. Zudem wird die Entscheidungsfindung im Management – beispielsweise in Bezug auf die Anschaffung neuer Software-Tools oder die Festsetzung von neuen Forschungsprojekten – unterstützt. Für den größtmöglichen Mehrwert in der Praxis, wird *RAMEn* spezifisch für das betrachtete Schnittfeld nach Abbildung 5.1 formuliert. Ein Übertrag auf andere Szenarien erfolgt als Teil der Evaluation in Abschnitt 6.1 ab S. 117.

*RAMEn* basiert auf dem Problemlösungszyklus und dem Phasenmodell nach [VDI 2221-1; VDI 2221-2]. Der Problemlösungszyklus gilt dabei als grundlegendes Prinzip der Produktentwicklung und beschreibt diese abstrakt mittels iterativer Optimierungszyklen. Die Zyklen wechseln zwischen der Analyse einer Referenz im Sinne der Produktgenerationsentwicklung (vgl. Abschnitt 2.1.2 auf S. 14) zur Festlegung von Entwicklungszielen und der Synthese der neuen Generation zu einem gewünschten Zustand. Eine iterative Verfeinerung erfolgt durch die Anwendung verschiedener Methoden und wird so lange fortgesetzt, bis eine optimale Lösung erreicht ist.

### 5.1.1 Orientierung in komplexen Schnittfeldern

Zunächst wird der Problemlösungszyklus und das Phasenmodell so erweitert, dass die Orientierung innerhalb komplexer Schnittfelder verbessert wird. Diese neuen Prinzipien sind in Abbildung 5.2 dargestellt. *Ziele* und *Ergebnisse* sind lokal für das entsprechende Schnittfeld zu formulieren, wobei diese im Einklang mit globalen Zielen der Entwicklung sein müssen. Beispielsweise ist die Optimierung der Fahrzeugsicherheit im Einklang mit dem globalen Ziel die Produktperformance zu verbessern. Analog gilt das für das Systemmodell und die Architektur eines Sicherheitsgurts mit dem Systemmodell und der Architektur eines Gesamtfahrzeugs. Die *Aktivitäten* werden systematisch durch *Zielfeld* und *Unterstützungsfeld* aufgespannt, in deren Ebene Methoden sortiert und gruppiert werden können. Die Methoden werden im Kontext des Daten- und Wissensmanagements angeordnet, das die notwendigen Voraussetzungen zur Anwendung bereitstellt. Entsprechend werden Methoden als durch den Ingenieur ausführbare Aktivitäten betrachtet.

Die Anwendung dieser Aktivitäten erfolgt nicht zwingend nur in zeitlichen Phasen, wie das in [VDI 2221-2] vorgeschlagen wird, sondern anhand von *Hauptmerkmalen*, die die (lokal



**Abbildung 5.2:** Problemlösezyklus nach [VDI 2221-1] mit eingeführten Erweiterungen zur verbesserten Orientierung in komplexen Schnittfeldern

relevanten) Situationen innerhalb des Produktentwicklungsprozesses beschreiben. Für die passive Fahrzeugsicherheit werden diese im weiteren Verlauf beispielhaft bestimmt.

Um die grundlegenden Prinzipien beizubehalten und die lokale Formulierung im Einklang mit der globalen Formulierung zu ermöglichen, werden auch hier iterative Denk- und Handlungszyklen vorgeschrieben. Das iterative Vorgehen bildet die Grundlage auch für den kombinierten Einsatz von Methoden.

### 5.1.2 Vorgehensmodell

Das Vorgehensmodell *RAMEn* ist in Abbildung 5.3 dargestellt. Dieses basiert auf den im vorangegangenen Abschnitt vorgeschlagenen Erweiterungen und betrachtet spezifisch das Schnittfeld des FEM-Prozesses und den datengetriebenen Methoden im Kontext der passiven Fahrzeugsicherheit.

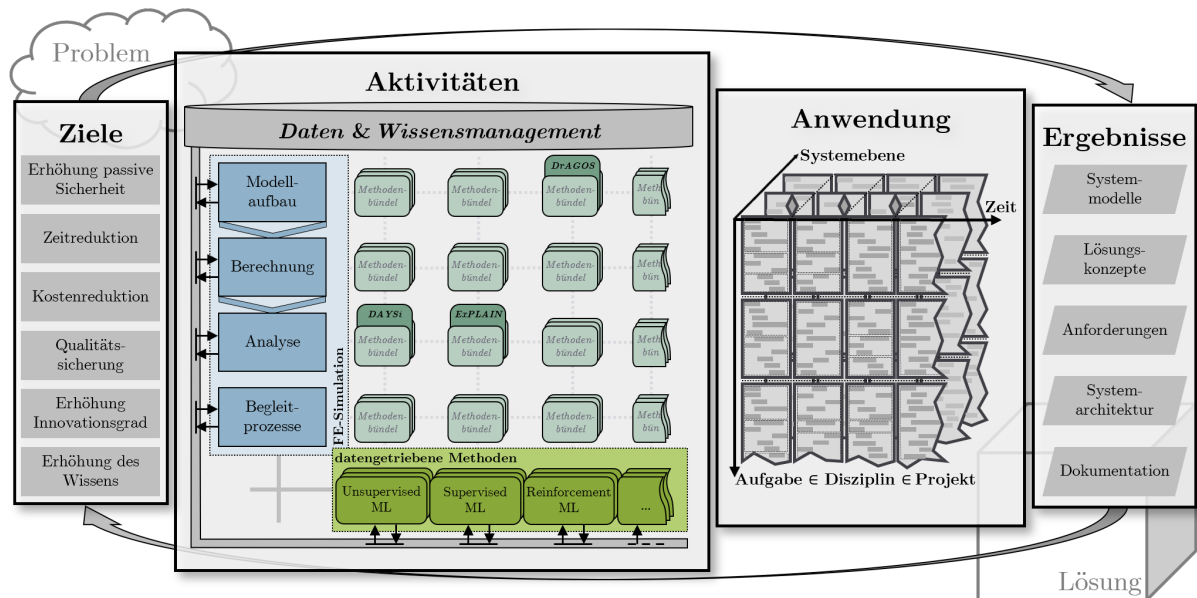


Abbildung 5.3: Schaubild RAMEn: Vorgehensmodell

### 5.1.2.1 Ziele und Ergebnisse

Zu den **Zielen** zählen die Erhöhung der passiven Fahrzeugsicherheit, die Einsparung von Entwicklungszeit und -kosten, die Sicherung der (Prozess-)Qualität, eine Erhöhung des Innovationsgrades sowie der Wissenszuwachs für beteiligter Personen und Organisationen. Die **Ergebnisse** beziehen sich auf die Beiträge der Organisationseinheit zum Fahrzeugentwicklungsprozess im Allgemeinen und enthält daher Systemmodelle, Lösungskonzepte und Anforderungen, Systemarchitektur sowie Dokumentation. An dieser Stelle wird auf Wechselwirkungen und den Austausch bezüglich der Ergebnisse und Ziele angrenzender Fachgebiete hingewiesen. Diese müssen lokal aus der Perspektive der passiven Fahrzeugsicherheit analysiert und berücksichtigt werden, wobei potenziell widersprüchliche Ziele während der Entwicklung iterativ aufzulösen sind, um das bestmögliche Produkt zu entwickeln.

### 5.1.2.2 Aktivitäten

Wie in Abbildung 5.3 dargestellt, finden die *Aktivitäten* im Schnittfeld statt. Dieses wird hier durch den FEM-Prozess und die datengetriebene Methoden, insbesondere die, die dem ML zugeordnet werden können, aufgespannt. Der FEM-Prozesses wurde bereits in Abbildung 2.4 auf S. 38 erläutert. Relevant, damit der Simulationsprozess selbst funktioniert, sind *Begleitprozesse*, wie die Bereitstellung von Informationen durch eine Anleitung oder das Schreiben von Skripten zum automatischen Kopieren von Daten. Auch diese können mit Hilfe von datengetriebenen Methoden optimiert werden. Die Gruppierung nach [ISO 23053] deutet an, dass zu jedem Lernverfahren verschiedene Algorithmen existieren,

wie beispielsweise Decision Trees oder neuronale Netzwerke für das Supervised Learning. Eine Erweiterung dieser Methoden durch klassische Methoden der Statistik, wie der Berechnung der Standardabweichung zur Ermittlung von Ausreißern oder auch die Berechnung linearer Korrelationen, ist möglich und daher in Abbildung 5.3 angedeutet. Werden mehrere Teilbereiche der beiden Felder abgedeckt, können Methoden auch mehrfach zugeordnet werden. Jedes Methodenbündel kann somit mehrere Methoden enthalten, die ähnliche Zwecke verfolgen, aber auf unterschiedlichen Algorithmen basieren. In Abbildung 5.3 sind beispielhaft die drei im Rahmen dieser Arbeit entwickelten Methoden eingeordnet. Das Wissen darüber, wie eine bestimmte Methode funktioniert, wie sie angewandt wird und welche Daten zu einem bestimmten Zeitpunkt verwendet werden, verbindet das Ziel mit dem Unterstützungsfeld. Diese Art von Wissen gilt als Grundlage für die Integration neuer Konzepte und Methoden in bestehende Prozesse und muss allen relevanten beteiligten Ingenieuren zugänglich gemacht werden. Dementsprechend ist dieses direkt bei den Aktivitäten verortet. Eine Umsetzung in der Praxis ist beispielsweise durch ein Wissensmanagementsystem wie Confluence [Atl25] möglich.

### 5.1.2.3 Anwendung

Die betrachteten datengetriebenen Methoden basieren entsprechend ihrer Namensgebung auf Daten und sind daher direkt von deren Eigenschaften (Anzahl, Typ, Dimensionalität, Verteilung, ...) abhängig. Das hat einen hohen Einfluss auf die *Anwendung* und damit auch auf die Auswahl von Methoden bzw. einzelner ML-Algorithmen. Um die Methoden nun besser in Einklang mit dem Entwicklungsprozess bringen zu können, wurden auf Basis des Stands der Technik in Kapitel 3 ab S. 43 drei Hauptmerkmale identifiziert, die für die Anwendung der Methoden im betrachteten Kontext relevant sind. Hierzu zählen die *Zeit*, die *Systemebene* sowie die (Ingenieurs-) *Aufgabe*. Darüber hinaus können Aufgaben einer *Disziplin* zugeordnet werden, die wiederum je nach Organisationsstruktur mindestens einem *Projekt* zugeordnet werden kann. Wird die Grundidee der Produktgenerationsentwicklung auf verfügbare Daten im Entwicklungsprozess übertragen, kann davon ausgegangen werden, dass sich auch diese nicht sprunghaft und grundlegend verändern werden. Das hat eine projektübergeordnete zeitliche Konstanz der Einordnungsprinzipien zur Folge. Einschränkungen abseits der Hauptmerkmale, die beispielsweise spezifische Datentypen für einzelne Methoden ein- oder ausschließen sind in der Methodendokumentationsvorlage auf Abbildung 5.4 vorgesehen. Eine Integration solch feingranularer Aspekte in den Überblick von *RAMEn* nach Abbildung 5.3 selbst, ist nicht vorgesehen, da hierfür in erster Linie übergeordnete Aspekte als relevant betrachtet werden.

Das erste Hauptmerkmal im Sinne der eingeführten Erweiterungen ist der *Zeitpunkt im Entwicklungsprozess*. Während in sehr frühen Konzeptphasen des Produktentwicklungsprozesses noch gar keine Simulationsdaten zur Verfügung stehen, da selbst die Computer Aided

Design (CAD)-Modelle noch im Aufbau sind, erhöht sich diese Zahl in späteren Phasen deutlich. Da typischerweise Insassensimulationsmodelle auf Struktursimulationsmodellen aufsetzen, sind diese ebenfalls nicht zur selben Zeit verfügbar. In frühen Phasen sind Daten eines Referenzprodukts verfügbar, wohingegen mit fortschreitender Entwicklungszeit zusätzlich auch physische Crashtests verfügbar sind. Dies ist auch im Einklang mit den klassischerweise im Automobilbau verwendeten Stage-Gate-Prozessen.

Die *Systemebene* gemäß dem V-Modell [VDI 2206] umfasst Komponentenebene (Airbag, Sicherheitsgurt, ...), Subsystemebene (Rückhaltesystem, Vorderwagen, ...) und Systemebene (Gesamtfahrzeug). Große und rechenintensive Simulationsmodelle, die das gesamte Fahrzeugsystem abbilden weisen hohe Freiheitsgrade und viel Gestaltungsraum auf. Daher ist die Menge der erzeugten Daten typischerweise deutlich geringer, als das bei Subsystemmodellen, wie beispielsweise Schlittensimulationen im Insassenschutz, der Fall ist. Zum Training für ML-Modelle werden jedoch aufgrund der Freiheitsgrade größere Datenmengen benötigt.

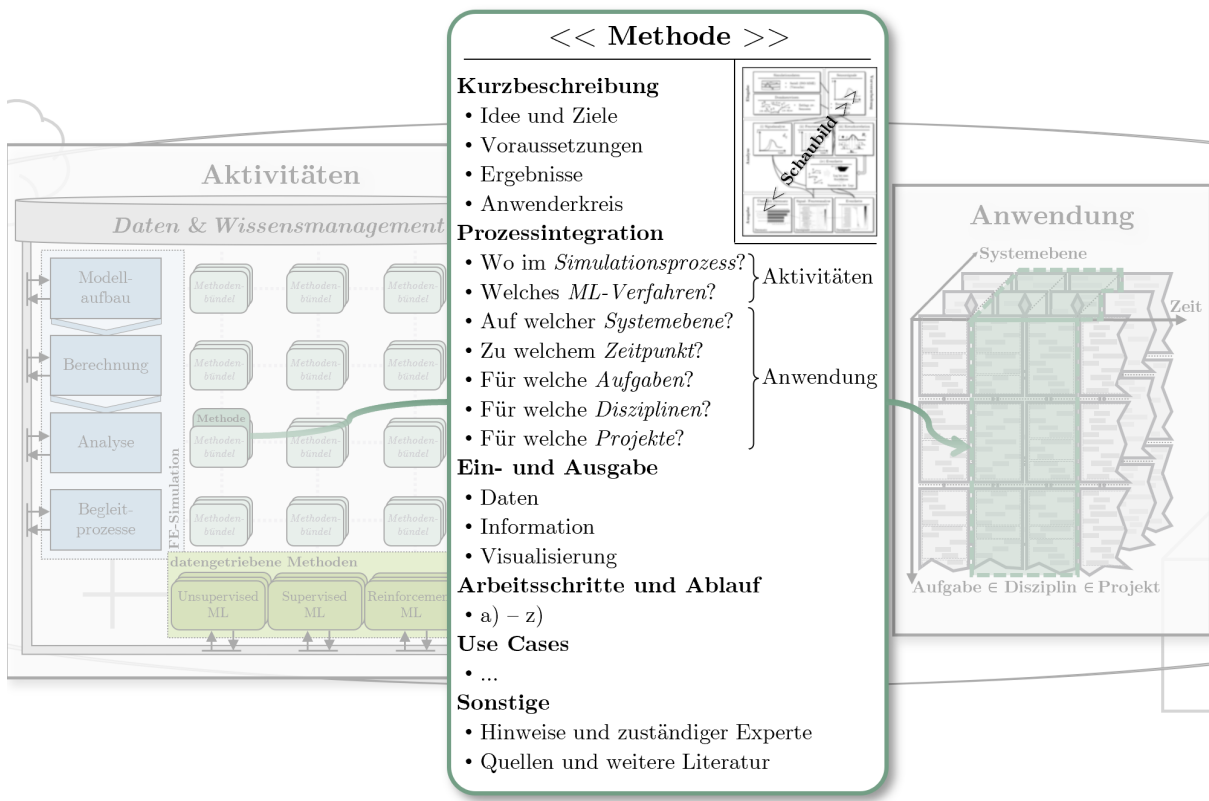
Zu den *Ingenieuraufgaben* in der Anwendung, die durch entsprechende Aktivitäten unterstützt und gelöst werden, zählen die Optimierung des Systemverhaltens, die Evaluation von Konzept-, Design- oder Materialänderungen oder auch Robustheitsbewertungen. Dabei werden unterschiedlich viele Simulationsdaten generiert, wobei besonders große Mengen in Robustheits- oder Optimierungskampagnen anfallen. *Disziplinen* im betrachteten Anwendungsfeld, in denen sich die Aufgaben nach Abbildung 5.3 einordnen lassen, sind die Auslegung der Rückhaltesysteme, Insassenzelle und Fahrzeugstruktur (vgl. Abschnitt 2.3.1 auf S. 33). Aus Sicht eines Automobilherstellers, der ein Portfolio mehrerer Fahrzeugbaureihen und Derivate besitzt, treten Aufgaben und oder Disziplinen mehrfach in verschiedenen Fahrzeugprojekten zu unterschiedlichen Zeitpunkten auf.

Das Anwendungsschema ermöglicht zudem die Darstellung von Wechselwirkungen und Mehrwerten der Methoden in anwendungsorientierter Weise. Ein Beispiel hierfür könnte eine Methode sein, um physikalisch gemessene Unsicherheiten, beispielsweise bei einer Airbagentfaltung (Komponentenebene), in eine Simulation auf Subsystem- oder Systemebene zu integrieren. Ebenfalls ist der Wissenstransfer zwischen Aufgaben, Disziplinen und Projekten relevant. Optimierungsstrategien von RL-Agenten oder Transfer-Learning-Methoden könnten hier eine Wiederverwendung von Daten realisieren und dabei helfen hohe Freiheitsgrade und Unsicherheiten gerade in den frühen Anfangsphasen der Entwicklung zu reduzieren.

#### 5.1.2.4 Dokumentation von Methoden

Die Dokumentation einzelner Methoden in *RAMEn* auf Basis von Abbildung 5.3 ist in Abbildung 5.4 dargestellt. Die Methoden, als Teil eines Methodenbündels, erhalten jeweils eine

Beschreibung. Der Inhalt dieser orientiert sich dabei an in [Pon07, S. 100][Alb+15][Fle+24, S. 40] aufgeführten Beschreibungen.



**Abbildung 5.4:** Integration und Auswahl von Methoden im Vorgehensmodell *RAMEn*

Dem Überblick dient eine Kurzbeschreibung, die neben der grundlegenden Idee auch die Ziele und Voraussetzungen sowie den Anwenderkreis beinhaltet. Dies wird durch ein Schaubild ergänzt, das Schritte und den Ablauf visualisiert, bevor diese im Detail erläutert werden. Voraussetzungen meint im Kontext datengetriebener Methoden insbesondere Rechenleistung und Speicherkapazitäten, auch wenn diese – aufgrund von Abhängigkeiten zur konkreten Problemstellung – schwer allgemein quantifizierbar sind. Die in Abbildung 5.4 aufgeführten Fragen dienen dabei als Leitfaden und erlauben eine Anordnung im Schnittfeld sowie eine bedarfsgerechte Integration in die Anwendung. Die Ein- und Ausgabe informiert den Nutzer über bereitzustellende und ausgegebene Information, Daten und Visualisierung. Dies dient in der Praxis insbesondere dem Aufbau von Schnittstellen. Die Arbeitsschritte und der Ablauf sorgen für Transparenz und sollen dem Nutzer ermöglichen ein tiefgreifendes Verständnis der Methode aufzubauen. Ebenfalls lassen sich basierend darauf einzelne Schritte hinterfragen. Die Verbindung von Schnittfeld, Methodenbeschreibung und Bereiche der Anwendung verschafft darüber hinaus einen schnellen grafischen Überblick. Die vorgeschlagene Struktur wird zur Einführung der Methoden in den drei Folgeabschnitten verwendet.

### 5.1.2.5 Einordnung und Auswahl von Methoden

Gemäß der Architektur von *RAMEn* nach Abbildung 5.3 auf S. 89 erfolgt die Einordnung der Methoden anhand der folgenden Schritte.

- i. Identifikation einer (übergeordneten) **Problemstellung**.
- ii. Festlegung von **Zielen** und **Ergebnissen**, die durch das Durchlaufen der Methode erreicht werden sollen.
- iii. Einordnung der **Aktivität** im Schnittfeld durch Festlegung relevanter Schritte im FEM-Prozess, die durch die Verwendung der Methode unterstützt werden, und Spezifikation des Typs der verwendeten datengetriebenen Methode.
- iv. Festlegung von mindestens einem Szenario in der **Anwendung** – hier die Einordnung der Verwendbarkeit einer Methode hinsichtlich Aufgabe, Disziplin, Projekt, Entwicklungszeitpunkt und Systemebene.
- v. Hervorheben des Transfers und der Speicherung von **Wissen** in der Anwendung.
- vi. Definition der **Übergabe** von Artefakten wie Anforderungen oder Systemmodellen an weitere Teile der globalen Produktentwicklung.
- vii. Befüllung der **Methodendokumentation** nach Abbildung 5.4 auf S. 92 zur Speicherung relevanter Information, wie Eingabe- und Ausgabedatentypen, Ansprechpartner oder Einschränkungen.

Die Auswahl erfolgt durch sukzessives Eingrenzen basierend auf den in den Schritten aufgeführten Merkmalen. Beispielsweise können alle Methoden aufgelistet werden, die nur zur frühen Zeitpunkten und zur Kostenreduktion eingesetzt werden können.

## 5.2 *DrAGOS* – Datengetriebene Systemauslegung

Der Optimierungsmethode *DrAGOS* liegt – gemäß Abschnitt 2.2.1.3 auf S. 26 – der ML-Typ RL zugrunde. Während die metamodellbasierte Optimierung, wie in Abbildung 2.6 auf S. 40 visualisiert, auf der statischen Approximation eines physikalischen Modells aufbaut, erlernt der RL-Agent eine Entscheidungsstrategie, die direkt auf optimale Designanpassungen abzielt. Da der RL-Agent ein exploratives Verhalten mitlernt und sich Nebenbedingungen direkt über die Belohnung abbilden lassen, ist er insbesondere bei einer schwachen Abdeckung des Designraums robuster als ein Metamodell [Cim+25; HLM25]. Zudem entspricht die gelernte Strategie einer inhärent interpretierbaren Abbildung von Zuständen auf Designänderungen. Dies macht Designentscheidungen nachvollziehbar und kann dadurch zur Akzeptanz solcher Methoden in der Praxis beitragen.

Diese Potenziale sollen zusammen mit einem Anwendungskonzept für besonders rechenintensive Simulationsmodelle durch *DrAGOS* gehoben werden. Die Einführung von *DrAGOS* erfolgt anhand der in Abbildung 5.4 auf S. 92 vorgeschlagenen Dokumentation für Methoden.

### 5.2.1 Kurzbeschreibung und Schaubild

Die **Idee** von *DrAGOS* ist die Übertragung von Wissen innerhalb wiederkehrender Entwicklungsaufgaben und verschiedener Produktgenerationen in den Produktentwicklungsprozessen einer Organisation. Mit Wissen ist hier die Entscheidungsfindung und Strategie zur Optimierung von Systemen unter Crashbelastung gemeint. Das dabei verfolgte **Ziel** ist die zeit- und oder kosteneffizientere Optimierung solcher Systeme in Bezug auf festgelegte Zielgrößen und Grenzwerte. Einen Überblick über den Ablauf von *DrAGOS*, der in Abschnitt 5.2.4 ab S. 96 noch detaillierter erläutert wird, ist Abbildung 5.5 zu entnehmen.

**Voraussetzungen** für den Einsatz ergeben sich aus der Verfügbarkeit von mindestens einem Referenzprodukt und den zugehörige Systemmodellen, die im Idealfall über mehrere Produktgenerationen oder Problemstellungen hinweg reichen. Eine umfangreiche Datenbasis befähigt die bestmögliche Abbildung funktionaler Zusammenhänge im Trainingsprozess des zugrundeliegenden RL-Agenten. Systemmodelle schließen hier jegliche Modelle oder Daten ein, anhand derer eine Systemantwort bestimmt werden kann. Hierzu können FE-Simulationsmodelle, theoretisch auch physische Tests verwendet werden. Zur Vergleichbarkeit der Systeme oder Aufgaben müssen alle einbezogenen Daten gemeinsame Parameter (Metaparameter) oder eine gemeinsame Basis, auf die die Parameter transformiert werden können, aufweisen. Ein Beispiel für solche Metaparameter sind Eigenschaften des Rückhaltesystems, da in jedem Fahrzeug Kraftniveaus und Zeitpunkte des Kraftniveauschaltens für den Sicherheitsgurt oder auch die Sitzposition spezifiziert werden [Rab+22].

Zum Training der Metamodelle muss eine hinreichend performante Recheninfrastruktur bereitgestellt werden, die von der Problemgröße und -komplexität abhängig ist.

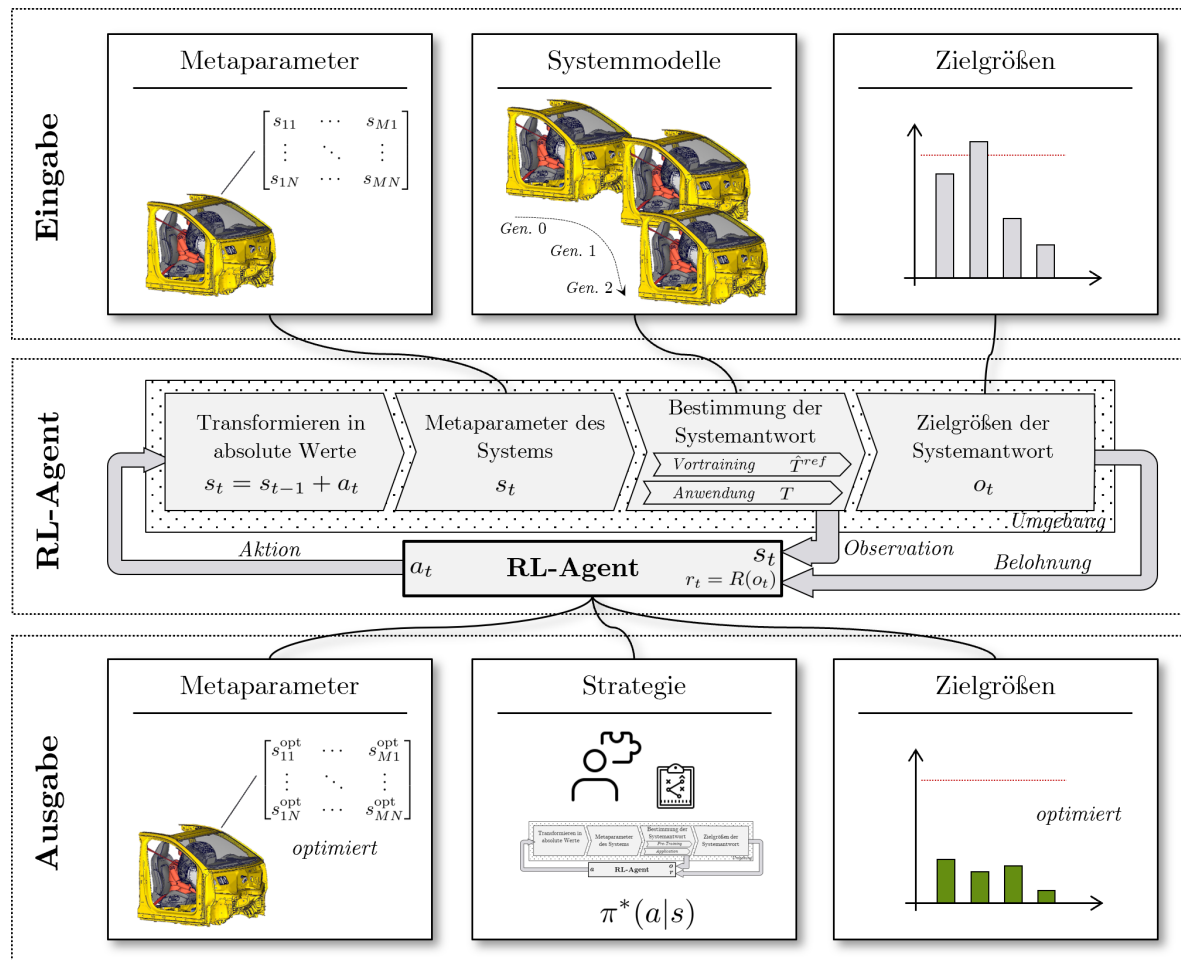


Abbildung 5.5: Schaubild *DrAGOS*: Arbeitsschritte und Ablauf

Die **Ergebnisse**, die basierend auf den später spezifizierten Ausgaben von *DrAGOS*, wieder in den Entwicklungsprozess überführt werden können, bestehen aus den optimierten Systemmodellen sowie dem Lösungskonzept mit dem ein technisch besseres Produkt erzielt werden kann. Dieses kann auch Änderungen in der Systemarchitektur enthalten.

Zum **Anwenderkreis** zählen Ingenieure, die in der Auslegung crashbelasteter Systeme mitwirken und dabei Änderungsvorschläge am System erarbeiten müssen.

### 5.2.2 Prozessintegration

In Bezug auf die in *RAMEn* eingeführten Ingenieursaktivitäten, die im Schnittfeld zwischen ML und dem Simulationsprozess liegen, ist *DrAGOS* als *RL-basierte Methode* zu betrachten, die ein verbessertes Modell automatisch spezifiziert und damit die Zeit im *Modellaufbau* verkürzt. *Alle Systemebenen* eignen sich zum Einsatz von *DrAGOS*, wohingegen

der Zeitpunkt des Einsatzes sich auf *Phasen nach der Konzeptentwicklung und vor der Serienentwicklung* beschränkt. In den frühen Phasen sind noch keine vollständigen Systemmodelle vorhanden, die der Methode bereitgestellt werden können. Ebenfalls können keine Abschätzungen auf Basis bestehender Referenzprodukte getroffen werden. In zu späten Phasen der Entwicklung sind die Freiheitsgrade zu gering, um große Änderungen durch die Methode in das System einzubringen. Zu den Ingenieursaufgaben, die durch die Methode unterstützt werden können, zählen *Optimierungsaufgaben in der Systemauslegung*, in denen die Metaparameter anzupassen sind. Das kann die Optimierung von Rückhaltesystemen oder auch die Dimensionsoptimierung einer Crashbox sein. Entsprechend ist ein *disziplinsübergreifender Einsatz* für Optimierungsaufgaben in der Fahrzeugstruktur, Insassenzelle, aber auch den Rückhaltesystemen möglich. Analog ist damit auch der *projektübergreifende Einsatz* von DrAGOS möglich und erlaubt es sogar Wissen über  $\pi^*$  zwischen Projekten und Aufgaben zu übertragen.

### 5.2.3 Ein- und Ausgaben

**Eingaben**, die DrAGOS zur Verfügung gestellt werden müssen, sind einerseits *Metaparameter* und *Zielgrößen* sowie *Systemmodelle* aus verwandten Entwicklungsaufgaben und Produktgenerationen oder Produktderivaten und andererseits das zu entwickelnde Produkt. Zielgrößen sind Werte anhand derer eine Bewertung des Systems vorgenommen werden kann – beispielsweise Belastungswerte für Insassen oder auch die Energieabsorptionseigenschaften eines Strukturbauteils. Relevante Verfahrensparameter, die angegeben werden müssen und von denen eine Abhängigkeit der Ergebnisse ausgeht, sind im Folgenden aufgelistet.

- Auswahl und Anzahl der Metaparameter, Zielgrößen und Systemmodelle
- Definition Belohnungsfunktion  $R$  und Diskretisierung des Aktionsraums  $\mathcal{A}$
- RL-Algorithmus (Algorithmus und Hyperparameter) (Standard: PPO)
- Abbildung funktionaler Zusammenhänge (Standard: ML-basiertes Metamodell  $\hat{T}^{\text{ref}}$ )

Die **Ausgaben** sind *angepasste Metaparameter*, die zugehörigen *optimierten Zielgrößen* und eine *angepasste Strategie* des RL-Agenten. Da die Metaparameter sukzessive angepasst werden, kann auch der Verlauf der Optimierung relevante Einblicke in das System liefern, die für beteiligte Ingenieure einen Mehrwert bieten.

### 5.2.4 Arbeitsschritte und Ablauf

Das generelle Vorgehen ist dem Schaubild in Abbildung 5.5 zu entnehmen. Auf Basis von  $M$  Metaparametern  $s_t \in \mathcal{S} \subset \mathbb{R}^M$ ,  $D$  Zielgrößen  $o_t \in \mathcal{O} \subset \mathbb{R}^D$  und den Referenzsystemmodellen  $T^{\text{ref}}$  wird einem RL-Agenten eine Optimierungsstrategie  $\pi^{\text{pre}}(a|s)$  beigebracht, die dieser dann nach dem Vortraining auf das zu entwickelnde Produkt anwendet. Die Grund-

lagen für RL, was die Basis für die Methode darstellt, wurden bereits in Abschnitt 2.2.1.3 auf S. 26 abgehandelt. Da die Wertebereiche von Metaparametern in der Produktgenerationsentwicklung variieren können, besteht der diskrete Aktionsraum  $a_t \in \mathcal{A} \subset \mathbb{R}^M$  aus relativen Parameteränderungen  $\Delta s$ , wobei

$$a_t = \Delta s_t = s_t - s_{t-1} \quad \text{mit} \quad a_t \in \{-\delta, 0, +\delta\}^M \quad (5.1)$$

gilt. Dabei entspricht  $M$  der Anzahl der Metaparameter,  $\delta$  der Schrittweite und  $t$  dem aktuellen Zustand der Umgebung. Der RL-Agent kann somit beispielsweise vorschlagen die Rückhaltekraft des Sicherheitsgurts in einem bestimmten Szenario um 0.25 kN zu reduzieren. Basierend auf den Änderungen an der Umgebung mittels der Aktion werden durch

$$s_t = s_{t-1} + a_t \quad (5.2)$$

absolute Werte spezifiziert. Die Systemantwort  $o_t$  kann durch FE-Simulation, physische Tests oder dem Ziehen aus einer Datenbank bestimmt werden und wird mittels einer Übergangsfunktion  $T$  durch

$$o_t = T(s_t) \quad (5.3)$$

beschrieben, welche die Systemantwort bestimmt. Da die Bestimmung dieser, insbesondere bei großen FE-Simulationen, aufwändig sein und mehrere Tage in Anspruch nehmen kann, wird hier die Verwendung eines approximierenden ML-basierten Metamodells  $\hat{T}^{\text{ref}}$  vorgeschlagen, das funktionale Zusammenhänge des Referenzsystemmodells  $T^{\text{ref}}$  approximiert, sodass

$$o_t \approx \hat{T}^{\text{ref}}(s_t) \quad (5.4)$$

gilt. Dies ermöglicht ein zeiteffizientes Vortrainieren des RL-Agenten auf den funktionalen Zusammenhängen, die sonst ein Ingenieur im Laufe der Entwicklung verstehen muss. Das Anwenden im darauffolgenden Schritt, beispielsweise direkt auf FE-Simulationen, ermöglicht das Optimieren des betrachteten Produkts. Die Strategie wird dabei weiter angepasst. Das Training des RL-Agenten erfolgt also iterativ und in zwei Phasen. Dabei kann die Strategie sukzessive an neue Produkte oder Systemänderungen angepasst werden. Formal lässt sich dies nach Gleichung (5.5) formal durch zwei Update-Schritte beschreiben und ist zur Verdeutlichung auch in Abbildung 5.5 visualisiert.

$$\begin{cases} \pi_t^{\text{pre}} = \text{Update}(\pi_0, \hat{T}^{\text{ref}}, n^{\text{pre}}) & \text{Vortraining Zusammenhänge Referenzsystems,} \\ \pi_{t+1} = \text{Update}(\pi_t^{\text{pre}}, T, n) & \text{Weitere Anpassung auf realem System} \end{cases} \quad (5.5)$$

Ein Update-Schritt bezieht sich auf das Durchlaufen einer festgesetzten Zahl  $n$  Epochen, nach denen die Strategie aktualisiert wird. Hierbei ist  $\pi_0$  die initiale Strategie und  $\pi^{\text{pre}}$  die vortrainierte Strategie. Nach Abbildung 5.5 auf S. 95 erhält der RL-Agent für jeden Schritt eine Belohnung  $r_t = R(o_t)$ , wobei  $R$  die Belohnungsfunktion ist, mit der die Qualität des Zustands der Umgebung in Bezug auf die Zielgrößen gemessen wird. Ziel des Agenten ist es, eine Strategie  $\pi^*(a|s)$  zu finden, die die erwartete kumulierte Belohnung maximiert (vgl. Gleichung (2.31) auf S. 26).

Das beschriebene Vorgehen von *DrAGOS* imitiert die Entscheidungen eines Ingenieurs, der relative Änderungen an Parametern vorschlägt und deren Auswirkungen bewertet. Über mehrere Produktgenerationen und Aufgaben hinweg wird der Agent sukzessive verbessert und an neue Systeme angepasst.

## 5.3 *DAYSi* – Analyse von Crashsensordaten

Der Analysemethode *DAYSi* liegt – gemäß Abschnitt 2.2.1.2 auf S. 22 – der ML-Typ Unsupervised Learning zugrunde. Während im Handlungsbedarf für die Analyse von FE-Simulationsdaten in Abschnitt 4.1.3 auf S. 77 der großen Mehrheit der Methoden detaillierte FE-Daten zugrunde liegen, soll mit *DAYSi* nun ein ressourceneffiziente Alternative geschaffen werden. Zudem soll *DAYSi* Ingenieure auch dabei unterstützen mögliche Ursachen für gefundene Auffälligkeiten zu identifizieren.

Die Einführung von *DAYSi* erfolgt anhand der in Abbildung 5.4 auf S. 92 vorgeschlagenen Dokumentation für Methoden.

### 5.3.1 Kurzbeschreibung und Schaubild

Die **Idee** von *DAYSi* besteht darin dem Nutzer einen schnellen ersten Eindruck über diverse 1D-Zeitseriendaten aus Crashsensoren zu verschaffen. Dabei werden neben Auffälligkeiten auf verschiedenen Betrachtungsebenen auch mögliche Ursachen identifiziert, denen dann weiter nachgegangen werden kann. Das **Ziel** besteht darin, die Qualität durch das Finden von Auffälligkeiten zu sichern und durch das Vorgeben der Richtung für weitere Analysen Entwicklungszeit einzusparen. Einen Überblick über den Ablauf von *DAYSi*, der in Abschnitt 5.3.4 ab S. 102 noch detaillierter erläutert wird, ist Abbildung 5.6 zu entnehmen.

**Voraussetzungen** für den Einsatz ergeben sich insbesondere aus der Verfügbarkeit von Simulations- und oder Versuchsdaten sowie einer hinreichend performanten Recheninfrastruktur, gerade bei der parallelen Verarbeitung vieler Sensoren und Datenpunkte. Während in Fahrzeugversuchen hunderte dieser Sensoren gefunden werden können, übersteigt die Anzahl in großen Crashsimulationen mehrere Tausend. Neben dem Zieldatenpunkt, der analysiert werden soll, beispielsweise eine neu durchgeführte Simulation, müssen mindestens zwei weitere Datenpunkte für den Kontext bereitgestellt werden.

Das **Ergebnis**, das basierend auf den später spezifizierten Ausgaben von *DAYSi*, wieder in den Entwicklungsprozess überführt werden kann, ist in erster Linie Dokumentation, da es sich um eine reine Analysemethode handelt. Basierend auf den Ergebnissen können Änderungen an den Systemmodellen oder den Lösungskonzepten durch Ingenieure hervorgebracht werden.

Zum **Anwenderkreis** zählen daher vor allem Ingenieure, die sich mit Simulations- bzw. Versuchsauswertung und Abgleich in Bezug auf Sensordaten beschäftigen.

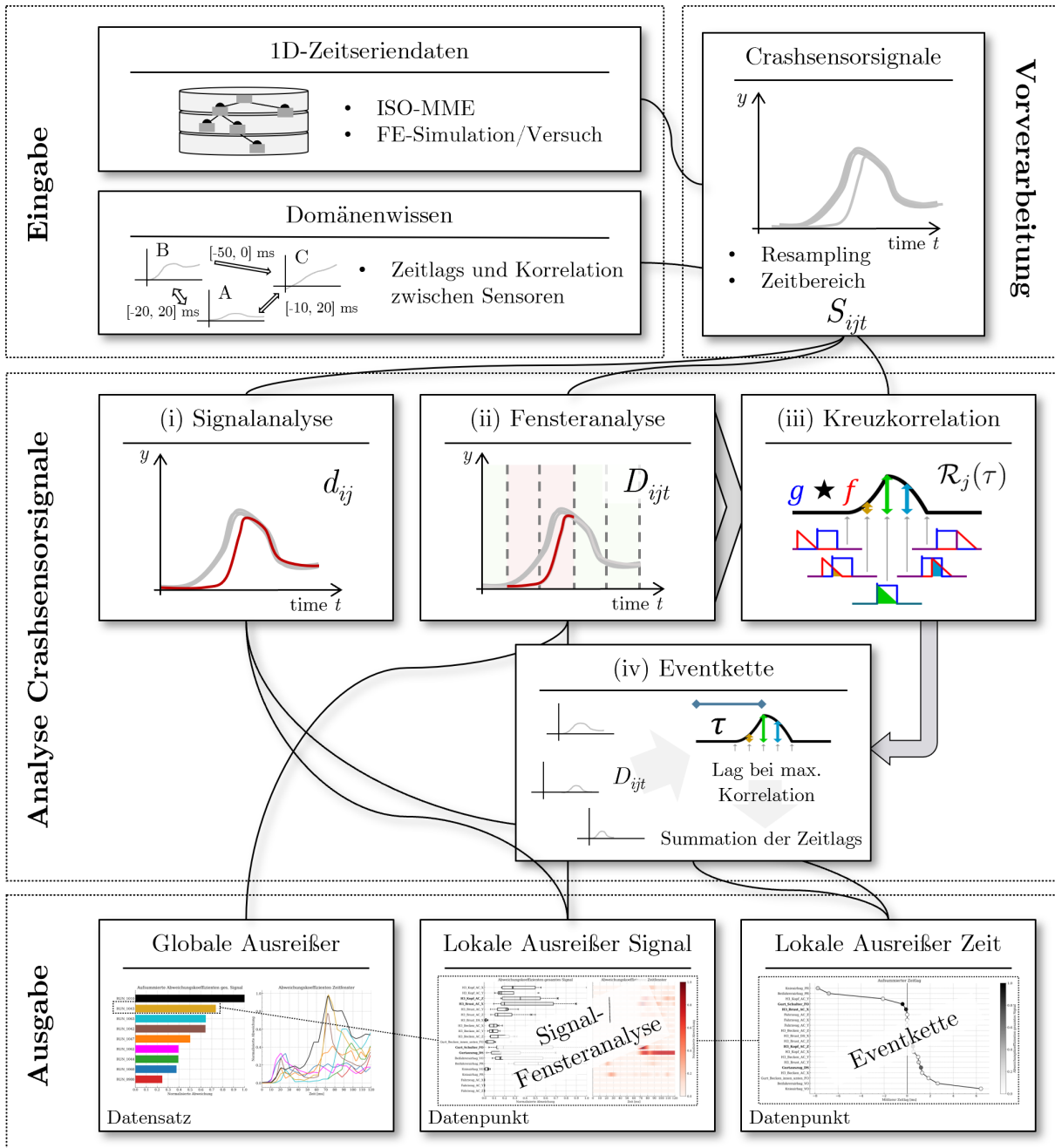


Abbildung 5.6: Schaubild *DAYSi*: Arbeitsschritte und Ablauf

### 5.3.2 Prozessintegration

In Bezug auf die in dem *RAMEn* eingeführten Ingenieursaktivitäten, die im Schnittfeld zwischen ML und dem FEM-Prozess liegen, lässt sich *DAYSi* als *Unsupervised-Learning-basierte Methode*, die die *Analyse von Simulationsdaten* und zugehörigen Versuchen unterstützt, einordnen. *Alle Systemebenen* eignen sich zum Einsatz von *DAYSi*, wohingegen der Zeitpunkt des Einsatzes sich auf *Phasen nach der frühen Konzeptentwicklung* beschränkt. Abschätzungen auf Basis bestehender Daten sind mit *DAYSi* nicht möglich, da

die Unsupervised-Learning-Methode in erster Linie nach Strukturen in den bereitgestellten Daten sucht. In frühen Phasen der Konzeption existieren jedoch noch keine Simulationsmodelle, die im großen Stil analysiert werden müssen. Sobald erste rechenfähige (Teil-)Modelle verfügbar sind, können diese mit Referenzen aus anderen Baureihen abgeglichen werden. Das ist durch die einheitliche Benennung der ISO-MME-Kanäle einfach möglich. Zu den Ingenieursaufgaben, die durch die Methode unterstützt werden können, zählen *Analyseaufgaben in Bezug auf 1D-Zeitreihendaten* sowohl im Versuch als auch der Simulation. Entsprechend ist ein *disziplinsübergreifender Einsatz* für die Auslegungsaufgaben in der Fahrzeugstruktur, Insassenzelle, aber auch den Rückhaltesystemen möglich. Große Einsparpotenziale sind dort gegeben, wo viele Simulationen bzw. Sensoren ausgewertet werden müssen. Beispielsweise in der Plausibilisierung von Daten einer Robustheitskampagne. Dadurch, dass keine FE-Daten verarbeitet werden müssen, ist der Rechenaufwand entsprechend deutlich geringer und vereinfacht den alltäglichen Einsatz. Hinzu kommt, dass *DAYSi* unabhängig von FE-Daten (Netzkonfiguration und Form) und dem verwendeten FEM-Löser ist. Damit ist auch ein *projektübergreifender Einsatz* von *DAYSi* möglich. Ein Transfer von Wissen, wie das mit *DrAGOS* realisiert wird, ist mit *DAYSi* hingegen nicht möglich.

### 5.3.3 Ein- und Ausgaben

Die **Eingabe** für *DAYSi* besteht aus *1D-Zeitreihendaten* (im Datenformat ISO-MME [ISO 13499]) und *Domänenwissen* (optional). Letzteres wird durch erlaubte Zeitlags und Korrelationen zwischen den Auffälligkeiten verschiedener Sensoren abgebildet. Das Datenformat ISO-MME wurde bereits in Abschnitt 3.2 auf S. 50 vorgestellt und ermöglicht die gemeinsame Verarbeitung von Versuchs- und Simulationsdaten unabhängig von Baureihe oder Entwicklungsstand. Relevante Verfahrensparameter von denen eine mögliche Abhängigkeit der Ergebnisse ausgeht, sind im Folgenden aufgelistet.

- Auswahl und Anzahl Datenpunkte (Simulationen/Versuche) und Sensoren
- Zeitlags und untere Grenze Korrelation nach Pearson zwischen Abweichungen verschiedener Sensoren (Standard:  $\tau \leq \pm 20$  ms und  $\rho \geq 0.2$  für alle Sensoren)
- Vorgehensweise und Abstandsmaß zur Berechnung der Abweichungen (Standard: Abstand zum Hauptclustermittelpunkt (MCD))
- Skalierung der Daten (Standard: Gruppenweise auf Basis physikalischer Einheiten mit dem absoluten Maximum)
- Zeitfenstergröße (Standard: 5 ms)
- Filterung (Standard: cfc60)

Zur **Ausgabe** zählen die berechneten *Abweichungskoeffizienten* und die *Plots*, die diese für den Nutzer grafisch zum besseren Verständnis aufbereiten. Die generierten Plots sind

dabei hierarchisch aufgebaut, sodass ein einfaches Verfolgen von Auffälligkeiten möglich ist. Beispielsweise ausgehend vom auffälligsten Datenpunkt über den dort auffälligsten Abweichungskoeffizient eines Sensors in Bezug auf Signal- und Zeitbereich zu dem Signal selbst. Darüber hinaus, werden die Abweichungskoeffizienten in Form von Textdateien abgespeichert.

### 5.3.4 Arbeitsschritte und Ablauf

Die Arbeitsschritte sind in Abbildung 5.6 dargestellt. Die eingegebenen ISO-MME-Daten werden im ersten Schritt der Vorverarbeitung auf ein gemeinsames Zeitintervall zwischen einzelnen Messwerten gebracht (Resampling mit  $\Delta t = 0.1$  ms) und auf einen identischen Zeitbereich (beispielsweise  $[0, 120]$  ms) beschnitten. Die extrahierten Zeitseriendaten lassen sich als Tensor dritter Ordnung  $S_{ijt}$  darstellen, wobei  $i$  die Sensoren,  $j$  die Datenpunkte (Simulationen/Versuche) und  $t$  die Signalwerte über die Zeit (Merkmale) repräsentieren. Die Analyse der Zeitseriendaten ist in Abbildung 5.6 verdeutlicht und besteht aus der Betrachtung (i) gesamter Signale und (ii) einzelner Zeitbereiche. Hierbei werden jeweils einzelne Sensoren für alle Datenpunkte betrachtet. Die Vorgehensweise zur Berechnung von Abweichungen basiert auf dem k-Means-Clusteringalgorithmus nach Gleichung (5.6). Ziel hierbei ist die Minimierung der Cluster-Varianz  $J_i^1$ .

$$\min\{J_i^1\} = \min_{u, \mu} \left\{ \sum_{k=1}^K \sum_{j=1}^N u_{kj} \|S_{ijt} - \mu_{kt}\|^2 \right\} \quad \text{s.t.} \quad \begin{cases} \text{(i)} & u_{kj} \in \{0, 1\} \quad \forall k, j \\ \text{(ii)} & \sum_{k=1}^K u_{kj} = 1 \quad \forall j \end{cases} \quad (5.6)$$

Hierbei werden Clustermittelpunkte  $\mu_{kt}$  ermittelt, die im weiteren Verlauf auch zur Berechnung der Abweichungskoeffizienten verwendet werden. Die Anzahl der Cluster  $K$  ist datenabhängig und muss individuell festgelegt werden. Damit dies nicht händisch erfolgen muss, wird die Silhouette Score [Rou87] durch

$$K^{\text{opt}} = \arg \max \left\{ \frac{1}{N} \sum_{j=1}^N q_j(K) \right\} \quad (5.7)$$

maximiert. Die Silhouette Score ist ein Maß dafür, wie gut die Datenpunkte innerhalb eines Clusters gruppiert sind und wie klar die Cluster voneinander getrennt sind. Diese entspricht daher einem Qualitätsmaß unabhängig von der Anzahl Clusterzentren. So soll für alle Sensoren eine vergleichbare und nachvollziehbare Basis geschaffen werden. Basierend auf der optimalen Clusteranzahl  $K^{\text{opt}}$  wird nun das Hauptcluster durch

$$\mu_t^{\text{main}} = \mu_{kt} \quad \text{mit} \quad k = \arg \max_k \sum_{j=1}^N u_{kj} \quad (5.8)$$

bestimmt. Das Hauptcluster repräsentiert das dominierende Systemverhalten, da dort die meisten Datenpunkte zugeordnet sind. Dieses Vorgehen wird für jeden Sensor durchgeführt und liefert die assemblierte Hauptclustermatrix  $\mu_{it}^{\text{main}}$ .

Die Abweichungskoeffizienten für alle Sensoren und Datenpunkte  $d_{ij}$  werden im nächsten Schritt mit

$$d_{ij} = \|S_{ijt} - \mu_{it}^{\text{main}}\| \quad (5.9)$$

bestimmt und entsprechen damit dem euklidischen Abstand eines Datenpunkts zum Hauptclusters des jeweiligen Sensors. Dieses Abstandsmaß wird im Folgenden als MCD bezeichnet. Das beschriebene Vorgehen wird anschließend analog für die Bestimmung der Abweichungskoeffizienten für die Zeitfensteranalyse angewendet. Dazu wird jedes Signal in gleich lange Zeitfenster unterteilt, über die dann zur Berechnung der zeitlichen Abweichungskoeffizienten iteriert wird. Diese resultierende Dimension ist geringer als die der Zeit in den Signalen  $S_{ijt}$ . Um jedem Merkmal in  $S_{ijt}$  einen Abweichungskoeffizienten zuzuordnen, wird ein Upsampling in der Zeitdomäne durchgeführt, sodass die zeitlichen Abweichungskoeffizienten als  $D_{ijt}$  ausgedrückt werden können. Auf Basis von  $D_{ijt}$  werden dann, wie in Abbildung 5.6 dargestellt, Kreuzkorrelationen (iii) für jeden Datenpunkt berechnet. Die Kreuzkorrelation  $\mathcal{R}_j(\tau)$ , in Abhängigkeit des Zeitlags  $\tau$ , wird durch

$$\mathcal{R}_j(\tau) = (D_{ij\bullet} \star D_{i'j\bullet})(\tau) \quad \text{mit} \quad \mathcal{R}_j \in \mathbb{R}^{M \times M \times T} \quad (5.10)$$

ermittelt. Der Index  $i'$  bezeichnet dabei einen weiteren Kanal auf derselben Dimension  $i$ , und weist entsprechend auf den Vergleich der zeitlich abhängigen Abweichungskoeffizienten zwischen zwei Sensoren hin. Die maximale Korrelation und der zugehörige Zeitlags ergeben sich durch

$$l_{ii'j} = \arg \max_{0 \leq \tau < T} \mathcal{R}_j(\tau) \quad \wedge \quad r_{ii'j} = \max_{0 \leq \tau < T} \mathcal{R}_j(\tau). \quad (5.11)$$

Da erlaubte Zeitlags im physikalischen System mit Domänenwissen grob abgeschätzt werden können, erlaubt *DAYS<sub>i</sub>* nach Abbildung 5.6 auf S. 100 eine manuelle Spezifikation dieser nach  $[\mathcal{T}^{\min}, \mathcal{T}^{\max}]$ . Beispielsweise ist eine Reaktion des ATDs in Bezug auf den Gurt, der direkt auf diesem aufliegt innerhalb weniger Millisekunden zu erwarten, während für eine veränderte Bewegung des ATDs aufgrund einer anderen Fahrzeugverzögerung mehrere zehn Millisekunden vergehen. Zur kompakteren Darstellung wird abschließend noch der mittlere kumulierte Zeitlag

$$\bar{l}_{ij} = \frac{1}{M} \sum_{i'=1}^M l_{ii'j} \quad \text{s.t.} \quad \begin{cases} \text{i)} & l_{ii'j} > \mathcal{T}^{\min} \quad \forall j \\ \text{ii)} & l_{ii'j} < \mathcal{T}^{\max} \quad \forall j \end{cases} \quad (5.12)$$

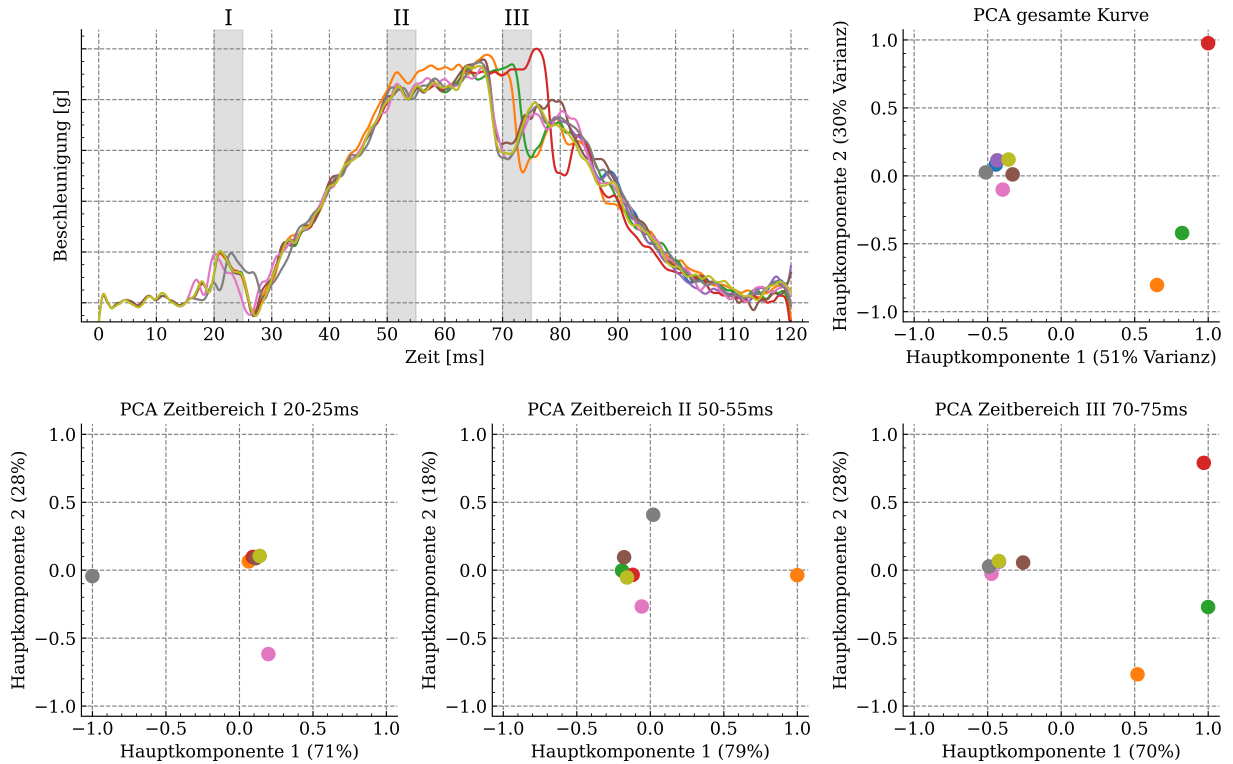
eingeführt. Dieser wird neben dem globalen Abweichungskoeffizient für das gesamte Signal  $d_{ij}$  dazu verwendet um in Abbildung 5.6 auf S. 100 die Eventkette (iv) zu bestimmen. Abschließend werden alle ermittelten Daten nach einem hierarchischen Schema geplottet, um dem Ingenieur eine möglichst einfache Analyse zu ermöglichen. Wie in Abbildung 5.6 gezeigt, wird der Überblick über den Datensatz (alle Simulationen) mittels der Summation über  $j$  und  $jt$  der zeitlichen Abweichung  $D_{ijt}$  generiert. Das ermöglicht grundsätzlich interessante Simulationen, aber auch Zeitpunkte darin zu ermitteln. Anschließend erfolgt die Darstellung der Abweichungskoeffizienten  $d_{ij}$  in einem Boxplot, um eine auf Basis des globalen Eindrucks ermittelte Simulation im Kontext der übrigen zu repräsentieren. Rechts im Plot dient eine Heatmap zur Ermittlung auffälliger Zeitbereiche des Sensors in der ausgewählten Simulation. Tiefrote Bereiche weisen dabei auf Auffälligkeiten hin. Die Eventkette wird als Heatmap-Barplot dargestellt. Die Balken selbst repräsentieren den summierten Zeitlag  $\bar{l}_{ij}$  für die  $j$ -te Simulation (links negativ, rechts positiv). Die Einfärbung anhand von  $d_{ij}$  (hell – geringe Auffälligkeit, dunkel – hohe Auffälligkeit) soll ein globales Bild darüber liefern, welche Sensoren früh – also bei negativem Zeitlag – auch als auffällig identifiziert werden.

#### 5.3.4.1 Analyse von Zeitbereichen

Um zu verdeutlichen, warum eine Berechnung der Abweichungskoeffizienten nur für die gesamte Kurve nicht hinreichend ist, zeigt Abbildung 5.7. Dort sind neun verschiedene Signale der Brustbeschleunigung eines ATDs dargestellt. Das entspricht einem typischen Vergleichsszenario in der Praxis. Während sechs dieser Signale global betrachtet sehr ähnlich sind, weisen drei davon – insbesondere im Bereich von 50 ms bis 85 ms – Unterschiede auf. Dies lässt sich auch durch die Anwendung einer PCA auf das gesamte Signal und die Visualisierung der zweidimensionalen Einbettung gut darstellen. Durch die beiden ersten Hauptkomponenten werden 81 % der Varianz in den initialen Daten erklärt, was einem hohen Informationsgehalt entspricht. In der Einbettung zeichnen sich zudem deutliche Cluster ab. Innerhalb einzelner Zeitbereiche können jedoch auch andere Kurven als auffällig identifiziert werden. Dies geht ebenfalls aus den Einbettungen hervor. Während die Kurven in Grau und Pink in Bezug auf die gesamte Kurve unauffällig sind, unterscheiden sich diese im Zeitbereich I hingegen deutlich. Diese Information ist gerade zur Identifikation möglicher Ursachen in frühen Zeitbereichen essenziell. Dementsprechend ist die Berücksichtigung globaler  $d_{ij}$  und lokaler  $D_{ijt}$  Information hier essenziell.

#### 5.3.4.2 Abstandsmaß zur Ausreißerererkennung

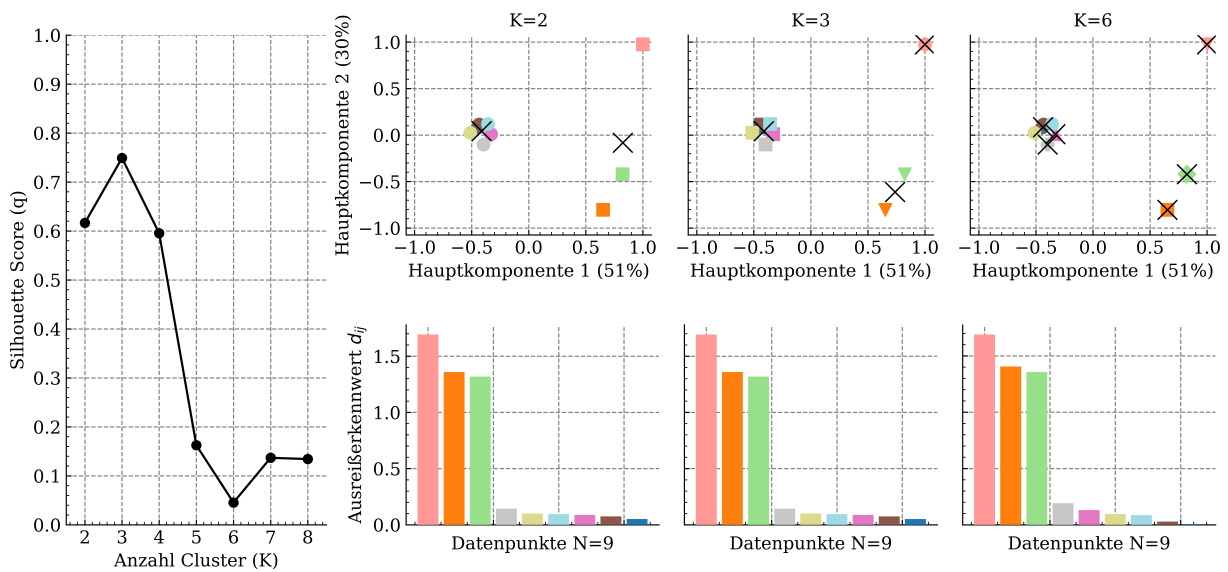
Zur Bestimmung der Abweichungskoeffizienten wird der MCD verwendet, der den Abstand einzelner Datenpunkte zum Mittelpunkt des Hauptclusters beschreibt. Da sich dort die Mehrzahl der ähnlichen Datenpunkte konzentriert, wird angenommen, dass dieses Cluster



**Abbildung 5.7:** Analyse von Zeitbereichen am Beispiel von Brustbeschleunigungskurven

das maßgebliche Systemverhalten eines Sensors innerhalb des Datensatzes repräsentiert. Für das verwendete k-Means-Clustering ist jedoch die Anzahl der Clusterzentren manuell vorzugeben. Abbildung 5.8 zeigt exemplarisch die Ergebnisse des k-Means-Clustering in der PCA-Einbettung der gesamten Signale aus Abbildung 5.7. Die Clusterzentren sind durch schwarze „X“-Symbole gekennzeichnet. Unterschiedliche Markierungen visualisieren die jeweilige Clusterzugehörigkeit des Datenpunkts. Zur Bewertung der Clusterqualität wird im Rahmen von *DAVS* die Silhouette-Score eingesetzt, die ebenfalls in Abbildung 5.8 über die möglichen Clusteranzahlen für den betrachteten Datensatz dargestellt ist. Festzustellen ist, dass die resultierenden Abstände  $d_{ij}$  zum Hauptcluster für verschiedene Clusteranzahlen robust bleiben – und damit weitgehend unabhängig von der Anzahl Clusterzentren sowie der Silhouette-Score sind. Die Maximierung der Silhouette-Score stellt jedoch einen nachvollziehbaren Ausgangspunkt dar, da ein hoher Wert auf eine klare Separation der Cluster sowie auf eine geringe Streuung innerhalb der Cluster hinweist. Ebenso gibt die Anzahl der Cluster Aufschluss darüber wie viele unterschiedliche Verhaltensweisen des jeweiligen Sensors zu finden sind. In Abbildung 5.8 ist beispielhaft dargestellt, dass für  $K = 3$  die höchste Silhouette Score erreicht wird und auch visuell ein stabiler und interpretierbarer Clusterverlauf erkennbar ist. Da die Silhouette Score jedoch nicht zwangsläufig die inhaltlich oder physikalisch sinnvollste Clusteranzahl liefert und insbesondere bei kleinen Datensätzen kleinere Cluster überproportional begünstigen kann,

werden alternative Abstandsmaße im Rahmen der Evaluation in Abschnitt 6.3.4.4 ab S. 158 genauer untersucht.



**Abbildung 5.8:** Silhouette Score und Abstandsmaß MCD zur Ausreißererkenennung

## 5.4 *ExPLAIN* – Erklärung von Systemverhalten

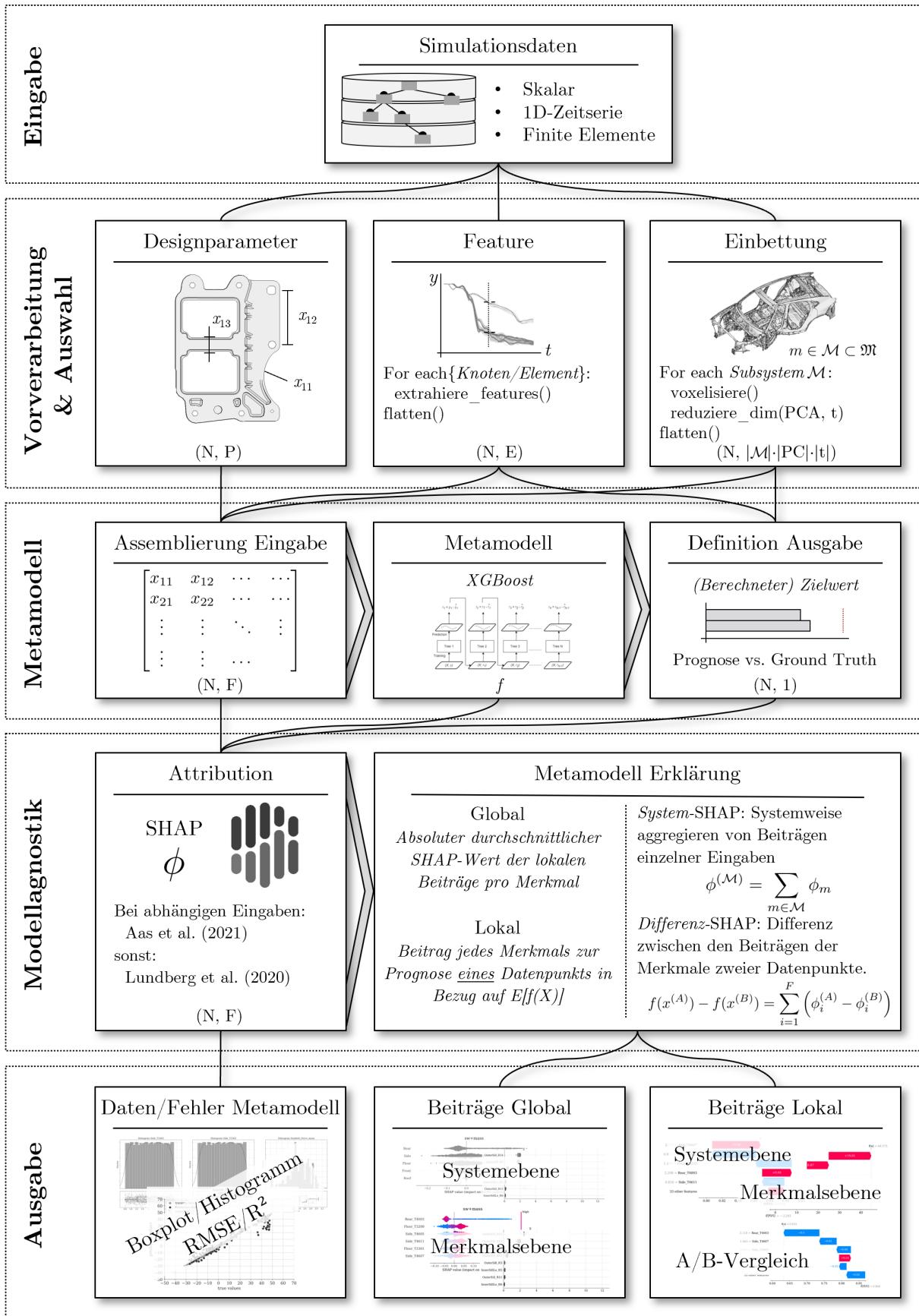
Der Analysemethode *ExPLAIN* liegt – gemäß Abschnitt 2.2.1.1 auf S. 18 – insbesondere der ML-Typ Supervised Learning zugrunde. Während sich im Handlungsbedarf für die Analyse von FE-Simulationsdaten in Abschnitt 4.1.3 auf S. 77 die große Mehrheit der Methoden auf einen der drei Datentypen stützt, sollen nun eine gemeinsame modulare Verarbeitung sowie detaillierte Erklärungen der Ergebnisse durch den Einsatz von interpretierbarem ML realisiert werden.

Die Einführung von *ExPLAIN* erfolgt anhand der in Abbildung 5.4 auf S. 92 vorgeschlagenen Dokumentation für Methoden.

### 5.4.1 Kurzbeschreibung und Schaubild

Die **Idee** von *ExPLAIN* besteht darin, die Simulationsdatenanalyse durch eine automatisierte Erklärung von Zielgrößen, wie Deformationsverhalten oder Insassenbelastungen, zu unterstützen. Dabei wird die Erklärung global – über den gesamten Designraum – und lokal – für einzelne Datenpunkte (Simulationen) – realisiert. Basierend hierauf werden eine in Echtzeit anpassbare Systemansicht sowie datengetriebene A/B-Vergleiche realisiert und ermöglichen tiefe Einblicke in das Systemverhalten. Mit *ExPLAIN* wird ein modulares Konzept vorgeschlagen, in dem alle relevanten Datentypen der FEM (vgl. Abbildung 3.2 auf S. 50) verarbeitet und situativ beliebig, also unabhängig von Geometrievariante, Entwicklungsstand oder Baureihe, kombiniert werden können.

Durch die Zeitabhängigkeit der Crashsimulationen kann somit neben einem Eingabe-Ausgabe ((i-o))-Zusammenhang auch ein Ausgabe-Ausgabe ((o-o))-Zusammenhang betrachtet werden. Im (i-o)-Ansatz werden Eingaben, wie Designparameter oder Geometrie-einbettungen, zum Nullzeitpunkt  $t_{\text{Eingabe}}^{(i-o)} = t_0 = 0$  ms in Relation zu einer Zielgröße bei  $t_{\text{Ausgabe}}^{(i-o)} > 0$  ms betrachtet. Dies entspricht den Beziehungen, die beispielsweise auch Gegenstand einer metamodellbasierten Optimierung sind (vgl. Abbildung 2.6 auf S. 40). Es besteht jedoch auch die Möglichkeit, ein (o-o)-Zusammenhang zu betrachten. Diese sind besonders interessant, wenn ein heterogener Simulationsdatensatz analysiert werden soll, der nicht parametrisierbare heuristische Designänderungen enthält. Als Eingabe werden dann Informationen wie Features zu einem späteren Zeitpunkt berücksichtigt,  $0 \text{ ms} < t_{\text{Eingabe}}^{(o-o)} < t_{\text{Ausgabe}}^{(o-o)} \leq t_{\text{Ende}}$ . Dies ermöglicht beispielsweise, Zusammenhänge zwischen Bauteilverformungen herzustellen. Das **Ziel** dabei ist beteiligte Ingenieure bei der Wissensgenerierung und im Systemverständnis flexibel zu unterstützen. Diese verbringen im Alltag viel Zeit damit, Abhängigkeiten zu verstehen, um geeignete Designs zu finden, die alle relevanten Zielgrößen erfüllen.



$N$  = Anzahl Simulationen,  $F$  = Anzahl Eingaben Metamodell,  $P$  = Anzahl Designparameter,  $E$  = Anzahl extrahierter Features,  $|\mathcal{M}|$  = Anzahl voxelisierter Subsysteme,  $|PC|$  = Anzahl Hauptkomponenten,  $|t|$  = Anzahl Zeitschritte

Abbildung 5.9: Schaubild ExPLAIN: Arbeitsschritte und Ablauf

Die Zielgrößen werden dabei typischerweise durch Grenzwerte aus gesetzlichen Vorschriften oder auch internen Anforderung einer Organisation vorgegeben und sind damit ständiger Gegenstand in der Entwicklung. Ein Überblick über den Ablauf von *ExPLAIN*, der in Abschnitt 5.4.4 ab S. 111 noch detaillierter erläutert wird, ist Abbildung 5.9 zu entnehmen.

**Voraussetzungen** für den Einsatz ergeben sich vor allem in der Verfügbarkeit von hinreichend Simulationsdaten. Typischerweise sollten – abhängig von der Problemstellung – mindestens 50 Datenpunkte vorhanden sein. Für das Training der Metamodelle sowie deren modellagnostische Analyse wird eine hinreichend performante Recheninfrastruktur, je nach Metamodell auch mit genügend grafischer Speicherkapazität, benötigt.

Das **Ergebnis**, das basierend auf den später spezifizierten Ausgaben von *ExPLAIN*, wieder in den Entwicklungsprozess überführt werden kann, ist in erster Linie Dokumentation, da es sich bei *ExPLAIN* um eine reine Analysemethode handelt. Basierend auf den Ergebnissen können Änderungen an den Systemmodellen oder den Lösungskonzepten durch Ingenieure hervorgebracht werden.

Zum **Anwenderkreis** zählen daher insbesondere Ingenieure, die sich mit Robustheits- und Optimierungskampagnen beschäftigen, da gerade dort hohe, oft auch systematisch parametrisierte, Datenmengen anfallen.

## 5.4.2 Prozessintegration

In Bezug auf die in *RAMEn* eingeführten Ingenieursaktivitäten, die im Schnittfeld zwischen ML und dem FEM-Prozess liegen, lässt sich *ExPLAIN* als *Supervised-Learning-Methode*, die die *Analyse von Simulationsdaten* unterstützt, einordnen. *ExPLAIN* ist vorwiegend für den *Einsatz auf höheren Systemebenen* geeignet, da dort die Komplexität und damit auch das Potenzial der Wissensgenerierung tendenziell höher ist. Bestandteil der Betrachtung auf den unteren Ebenen ist beispielsweise die simulative Analyse von Ersatzversuchen. Eine vergleichsweise einfache Fragestellung sowie wenige Parameter und Auswertegrößen machen diese Daten auch manuell gut überschaubar. Idealerweise wird *ExPLAIN* dort eingesetzt, wo viele Eingabegrößen berücksichtigt werden müssen. Ein Beispiel hierfür ist die Optimierung eines ganzen Rohbaus. Ebenfalls interessant sind Fragestellungen auf Systemebenen, wo Erfahrungen und Expertise fehlen. Zeitlich ist die Verwendung von *ExPLAIN* in erster Linie für die *Phasen nach der frühen Konzeptentwicklung* vorgesehen. Analog zu *DAYSi* fehlen auch für *ExPLAIN* in den frühen Phasen Daten, die Gegenstand einer Analyse sein können. Bei der Verwendung von Metaparametern – analog dem Konzept von *DrAGOS* – oder dem Encodieren von Zeitserien- oder FE-Daten aus verschiedenen Produkten oder Aufgaben ist mittels dem Supervised-Learning-Ansatz auch ein übergreifender Einsatz denkbar. Entsprechend der verwendeten Daten lässt sich also ein Wissenstransfer indirekt – durch die Dokumentation von extrahiertem Wissen bei bestehenden Analysen –

aber auch direkt – durch die Inferenz entsprechender Metamodelle – realisieren. Zu den Ingenieursaufgaben, die durch ExPLAIN unterstützt werden können, zählen insbesondere *rechenintensive Aufgaben wie Robustheits- und Optimierungskampagnen*. Entsprechend ist ein *disziplinsübergreifender Einsatz* für die *Auslegungsaufgaben in der Fahrzeugstruktur, Insassenzelle, aber auch den Rückhaltesystemen* möglich, da all diese Systeme optimiert und auf Robustheit überprüft werden müssen.

Durch die Möglichkeit flexibel beliebige Zusammenhänge zu analysieren, die für Ingenieure relevant sind, wird eine flexible Prozessintegration ermöglicht. Bei parametrisierten Problemstellungen in Optimierungs- oder Robustheitsuntersuchungen eignet sich die Analyse des (i-o)-Zusammenhangs. Für die Verifikation von Designänderungen, in denen keine systematische Parametrisierung vorhanden ist, kann stattdessen ein (o-o)-Zusammenhang betrachtet werden. Ein Beispiel dafür ist die Verwendung unterschiedlicher Airbagvarianten deren Verarbeitung durch die Voxelisierung mit anschließender Dimensionsreduktion oder auch der Extraktion von Merkmalen aus den Zeitserien ermöglicht wird. Zudem trägt die einfache und intuitive Darstellungsweise zum Vertrauen und zur Akzeptanz datengetriebener Methoden generell bei, da diese – wie in Abschnitt 3.4 ab S. 60 erläutert – bisher abstrakt sind.

### 5.4.3 Ein- und Ausgaben

Die **Eingabe** für ExPLAIN besteht aus *Skalaren-, 1D-Zeitserien- und FE-Daten*, die in den Datenformaten Excel-CSV, ISO-MME [ISO 13499] und dyna3d vorliegen (vgl. Datentypen FEM: Abbildung 3.2 auf S. 50). Durch die Modularität von ExPLAIN, ergeben sich Verfahrensparameter, die spezifiziert werden müssen. Diese sind im Folgenden aufgelistet.

- Auswahl und Anzahl FE-Simulationen
- Auswahl und Anzahl Eingabedaten aus Parameter-, 1D-Zeitserien- und FE-Daten unter Berücksichtigung möglicher Abhängigkeiten
- Diskretisierung und Dimensionsreduktionsmethoden für das Voxelgrid der FE-Daten (Standard: 20 mm, PCA [Jol02])
- Metamodell (Standard: XGBoost [CG16])
- Berechnungsmethode von SHAP (Standard: TreeSHAP [Lun+20])

Die **Ausgabe** enthält *Datenüberblick und Fehlermetriken des Metamodells* sowie die Plots der *SHAP-Werte*, um die Ergebnisse der Analyse visuell für die Ingenieure aufzubereiten. Die Fehlermetriken des Metamodells dienen dabei auch zur initialen Bewertung der Plausibilität berechneter SHAP-Werte. Wenn das Metamodell nicht in der Lage ist die Zusammenhänge in den Daten abzubilden, ist ebenso bei den SHAP-Werten davon auszugehen.

#### 5.4.4 Arbeitsschritte und Ablauf

Arbeitsschritte und Ablauf für *ExPLAIN* sind in Abbildung 5.9 auf S. 108 detailliert dargestellt. Die Eingabe für *ExPLAIN* besteht aus manuell ausgewählten Simulationsdaten (vgl. Skalar-, 1D-Zeitreihen- und oder FE-Daten), aus denen im Rahmen der Analyse drei relevante Typen – *Designparameter*, *Feature* und *Einbettung* – abgeleitet und automatisch vorverarbeitet werden.

**Designparameter** beziehen sich dabei direkt auf Designaspekte, Auslegungsgrößen oder auch Randbedingungen des analysierten Systems in der Simulationseingabe. Beispiele hierfür sind Wandstärken oder Vorspannungskräfte. Ein spezieller Vorverarbeitungsschritt, abgesehen von einer Skalierung, abhängig vom verwendeten Metamodell, ist nicht erforderlich.

**Features** bezeichnen hier extrahierte Merkmale aus der Simulationsergebnisdatei, wie beispielsweise Maximal-, Minimal- oder Mittelwerte von Schnittkräften, Drücken, Verschiebungen oder FE-Elementspannungen sein. Daten können aus einer Menge von Knoten oder Elementen extrahiert werden. Auch die kombinierte Verarbeitung ist denkbar. Beispielsweise die Anzahl der Elemente oberhalb einer plastischen Dehnungsgrenze im Motorlager im letzten Simulationszeitpunkt. Die zugrundeliegenden Daten müssen zunächst aber auf eine identische Abstrakte gebracht werden, um die Vergleichbarkeit im Zeitbereich sicherzustellen. Nach der Extraktion ist hier ebenfalls, abhängig vom verwendeten Metamodell, eine Skalierung vonnöten.

**Einbettungen** beschreiben latent geometrisches Verhalten des Simulationsmodells in einem niedrigdimensionalen Einbettungsraum. Neben Knotenverschiebungen, die die Geometrie beschreiben, sind auch weitere FE-bezogene Attribute wie plastische Dehnungen oder Versagen denkbar. Zunächst ist auch hier erforderlich die zugrundeliegenden zeitabhängigen Daten auf eine identische Abstrakte zu bringen, um die Vergleichbarkeit im Zeitbereich sicherzustellen. Um die FE-Daten unabhängig von den geometrischen Eigenschaften und der FE-Netzkonfiguration zu verarbeiten, wird eine Voxelisierung – in Gleichung (5.13) allgemein ausgedrückt – für jedes der Subsysteme aus  $\mathfrak{M}$  und für jede der  $N$  Simulationen im Datensatz durchgeführt.

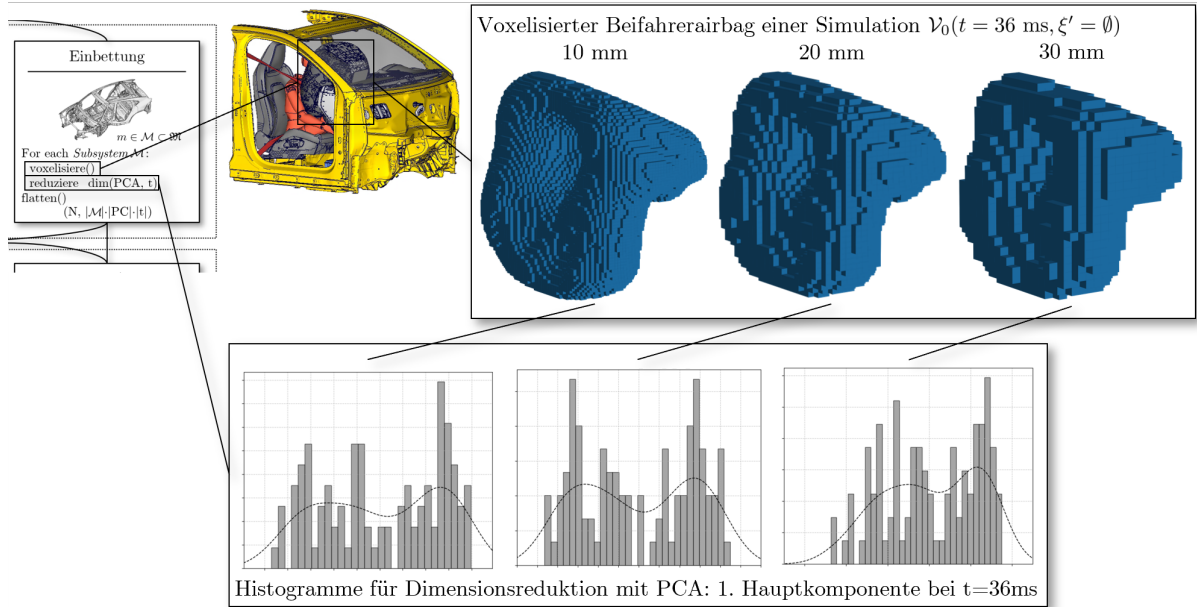
$$\Phi : \mathcal{G}(t, \xi) \mapsto \mathcal{V}(t, \xi') \quad \text{mit} \quad \mathcal{V}(t, \xi') \in \mathbb{R}^{N \times X \times Y \times Z \times C}, \quad \xi' \subseteq \xi \quad (5.13)$$

Allgemein können die FE-Daten als zeitabhängiger und attributierter Graph  $\mathcal{G}(t, \xi)$  gesehen werden.  $\mathcal{V}(t, \xi')$  entspricht dem resultierenden Voxelgrid mit räumlicher Auflösung  $X, Y, Z$  und  $C$  Kanälen (aggregierten Attributen) für ein Subsystem bei gegebenem Datensatz. Durch die nicht injektive Transformation  $\Phi$  wird jedoch nur ein Teil der ursprünglichen Attribute ( $\xi' \subseteq \xi$ ) abgebildet. Des Weiteren treten Diskretisierungsfehler auf, die es zu

berücksichtigen gilt. Die Voxelisierung wird innerhalb einer Bounding-Box durchgeführt, die alle einbezogenen Komponenten des Subsystems aller Datenpunkte zu jedem Zeitpunkt umschließt. Neben einer aus Ingenieursicht einfach interpretierbaren räumlichen Repräsentation ermöglicht der Ansatz auch die Integration weiterer Information  $\xi'$  wie Teile der Materialeigenschaften oder Randbedingungen in einzelnen Voxeln.

Zur Erkennung von Mustern und Strukturen in den Daten wird eine Transformation der hochdimensionalen Daten in einen niedrigdimensionalen Einbettungsraum durchgeführt (vgl. Abschnitt 2.2.1.2 ab S. 22). Als Grundlage wird hierzu eine PCA verwendet, da diese Methode linear und deterministisch ist. Einzelne Komponenten repräsentieren dabei Richtungen der größten Varianz. Weitere, insbesondere nichtlineare, Methoden wie DiffMaps oder UMAP sind denkbar und daher später Teil der Evaluation. Für diese Methoden sind potenzielle Sensitivitäten gegenüber Hyperparametern zu berücksichtigen. Im Reduktionsprozess kann die Zeitinformation der Voxelrepräsentation  $\mathcal{V}(t, \xi')$  erhalten werden, sodass spezifische Zeitpunkte betrachtet werden können. Ein Beispiel hierfür ist die niedrigdimensionale Repräsentation im Endzustand der Simulation. Die Zeitinformation kann jedoch zusammen mit den räumlichen Informationen reduziert werden, um globale Einblicke in das Subsystem über die gesamte Simulationsdauer zu ermöglichen. Analog zur Voxelisierung wird die Einbettung separat für Subsysteme  $\mathcal{M}$  durchgeführt, um die Rückverfolgbarkeit der dimensionsreduzierten Repräsentation auf einzelne Systeme zu ermöglichen. Nach der Einbettung ist hier ebenfalls, abhängig vom verwendeten Metamodell, eine Skalierung vonnöten. Sensitivitäten gegenüber potenziellen Ausreißern bei unüberwachten Dimensionsreduktionsmethoden sind für *ExPLAIN* nur bedingt relevant, da keine Generalisierung für andere Problemstellungen angestrebt wird. Der Fokus liegt vor allem auf der Analyse eines gegebenen Simulationsdatensatzes.

Ein Beispiel zur Verdeutlichung der Voxelisierung und der Dimensionsreduktion ist Abbildung 5.10 zu entnehmen. In jeder der beispielhaft betrachteten 98 FE-Simulationen, die zur Analyse eines Frontalaufpralls verwendet werden, erfolgt die Extraktion und anschließende Voxelisierung des Beifahrerairbags zu einem Zeitpunkt von 36 ms. Zudem sind in Abbildung 5.10 verschiedene Diskretisierungsgrößen von 10 mm bis 30 mm aufgezeigt. Das Voxelgrid mit der Größe 20 mm weist eine Größe von  $\dim\{\mathcal{V}(t = 36 \text{ ms}, \xi' = \emptyset)\} = (98, 1, 39, 35, 38)$  auf und verbraucht dabei etwa 5 Megabyte Speicherplatz. Anschließend wird exemplarisch eine Dimensionsreduktion durchgeführt, wobei die erste ermittelte Hauptkomponente, die in allen drei Fällen etwa 20% der Varianz in den unkomprimierten Daten abbildet, in einem Histogramm gezeigt ist. Ersichtlich ist, dass diese zwei Cluster bilden. Abseits der Position im Raum enthält das Voxelgrid keine weiteren Informationen, weshalb  $\xi' = \emptyset$  gilt.



**Abbildung 5.10:** Voxelisierung von Subsystemen am Beispiel eines Airbags

Wie im Ablauf von *ExPLAIN* nach Abbildung 5.9 ersichtlich, ist die Auswahl, welche Aspekte des mechanischen Systems wie beschrieben werden, den Ingenieuren selbst überlassen. Eingaben und Ausgaben können entsprechend dem Ziel der Untersuchung situativ festgelegt werden. Der verfolgte Ansatz bietet Flexibilität, es wird jedoch Domänenwissen in Form von einem Grundsystemverständnis benötigt.

Bevor die ausgewählten Daten dem Metamodell als Eingabe und Ausgabe zugeordnet und übergeben werden ist eine Prüfung von inhärenten Abhängigkeiten der aggregierten Daten notwendig. Stark korrelierende Eingabemerkmale oder auch fehlende Abhängigkeiten zur Ausgabe können den Lernprozess erschweren und erfordern auch eine gesonderte Berücksichtigung in der modellagnostischen Betrachtung. Dies kann manuell durch Berechnung der Pearson-Korrelationskoeffizienten und Datenvisualisierung in Histogrammen oder automatisch durch das Festsetzen von Grenzwerten erfolgen.

Als Metamodell wird entsprechend Abbildung 5.9 auf S. 108 XGBoost [CG16] vorgeschlagen, da dieses nichtlineare Zusammenhänge lernen kann und auch bei moderat großen Datensätzen performant ist. Der letzte Aspekt ist entscheidend, da die typische Anzahl an Crashsimulationen für spezifische Problemstellungen selten mehrere Hundert überschreitet. Das Metamodell wird mit dem entsprechenden Zielwert trainiert und bildet anschließend funktionale Abhängigkeiten der Simulation nach.

Auf das trainierte Metamodell erfolgt die Anwendung von SHAP [LL17; Lun+20]. Die SHAP-Werte stellen lokale Effekte einzelner Eingaben für jeden Datenpunkt bereit. Diese lassen sich anschaulich darstellen und sind gerade aufgrund ihrer Additivität intuitiv inter-

pretierbar. Da die Beiträge im Ausgaberaum vorliegen, weisen diese entsprechend – sofern eine vorhanden – auch dieselbe physikalische Einheit auf. Da XGBoost auf Entscheidungsbäumen basiert, lassen sich mittels TreeSHAP [Lun+20] SHAP-Werte in polynomieller Zeit ermitteln (vgl. Abschnitt 2.2.2 ab S. 27). KernelSHAP hingegen skaliert theoretisch exponentiell mit der Anzahl Eingaben. Folglich ist der kombinierte Einsatz von XGBoost und TreeSHAP effizient und daher vielversprechend für den praktischen Einsatz.

Die abschließende Interpretation der SHAP-Werte gemäß Abbildung 5.9 auf S. 108 im Kontext domänenspezifischen Wissens (vgl. [Ros+20]) erlaubt die Formulierung einer ingenieurwissenschaftlichen Erklärung der analysierten Zielgröße. Um nun diesen Ansatz weiter auf den Ingenieurbereich mit dem Fokus Simulationsdatenanalyse zu übertragen, werden im Folgenden *System-* und *Differenz-SHAP-Werte* eingeführt. Hierbei werden mathematische Eigenschaften der SHAP-Werte ausgenutzt.

#### 5.4.4.1 System-SHAP

Gemäß dem Paradigma des Systems Engineering, bei dem Produkte in Systemen gedacht werden [Kos+11], lässt sich der Beitrag von mehreren Merkmalen  $m$ , die ein Subsystem als Teil eines Systems  $m \in \mathcal{M} \subset \mathfrak{M}$  beschreiben, zusammenfassen. Formal ist das in Gleichung (5.14) dargelegt. Auf Basis einmal berechneter SHAP-Werte  $\phi$  können System-SHAP-Werte  $\phi^{(\mathcal{M})}$  für verschiedene Subsysteme entsprechend dem Nutzerwunsch in Echtzeit dargestellt werden. Die rechenintensive Modellagnostik muss dafür nur einmal durchgeführt werden. Perspektivisch kann das in der Praxis dabei helfen Bauteilverantwortliche oder Zulieferer bestimmter Subsysteme zurückzuverfolgen, die Änderungen zur Verbesserung des Gesamtverhaltens umsetzen können. Damit trägt System-SHAP auch zu dem Grundprinzip des Systems Engineering der ständigen Nachverfolgbarkeit im Entwicklungsprozess bei.

$$\phi^{(\mathcal{M})} = \sum_{m \in \mathcal{M}} \phi_m \quad (5.14)$$

#### 5.4.4.2 Differenz-SHAP

Ingenieure vergleichen im Rahmen ihres Alltagsgeschäfts häufig einzelne Simulationsergebnisse. Fachlich wird das typischerweise als A/B-Vergleich bezeichnet. Um nun zu verstehen, was zum Unterschied von *Simulation A* zu *Simulation B* beiträgt, können die mit SHAP erhaltenen Beiträge merkmalsweise subtrahiert werden. Formal ist das in Gleichung (5.15) gezeigt, wobei  $F$  die Anzahl der Merkmale der assemblierten ML-Modelleingaben bezeichnet. Damit die Bildung der Differenz erlaubt ist, müssen beide Datenpunkte in den berechneten SHAP-Werten  $\phi$  vorhanden sein. Damit beziehen sich ursprünglich beide auch auf die selbe Referenz  $\mathbb{E}[f(X)]$  und die Beiträge sind konsistent. Der Vergleich zweier Systemkonfigurationen ist damit datengetrieben und direkt im Ausgaberaum möglich,

wobei Unterschiede in der Konfiguration als Beitragsdifferenz hinsichtlich der analysierten Zielvariable ausgedrückt werden. Auch wenn Differenzen in den Eingaben große sind, müssen diese nicht maßgeblich für eine große Änderung in der Zielvariable verantwortlich sein. Dieser Aspekt, der konventionell erst manuell herausgearbeitet werden muss, lässt sich nun automatisch abbilden. Ein relevanter Vergleich in der Praxis ist beispielsweise die Betrachtung der optimierten bezüglich der initialen (Basis-)Variante in einer Optimierungskampagne. Ebenfalls ist im Rahmen einer Robustheitsanalyse auch der Vergleich der Basis mit einem Ausreißer denkbar.

$$f(x^{(A)}) - f(x^{(B)}) = \sum_{i=1}^F (\phi_i^{(A)} - \phi_i^{(B)}) \quad (5.15)$$

## 5.5 Zusammenfassung

Im Rahmen von Kapitel 5, das der Präskriptiven Studie (Phase III) des Aufbaus nach der DRM in Abbildung 1.1 auf S. 6 entspricht, wurden das Vorgehensmodell *RAMEn* sowie drei eigenständige Methoden, *DrAGOS*, *DAYSi* und *ExPLAIN*, eingeführt.

Die Erkenntnisse dieses Kapitels können folgendermaßen zusammengefasst werden:

- Auf Basis der Defizite in Kapitel 4 ab S. 69, wurde der Fokus der Arbeit insbesondere auf den Bereich der Analysemethoden in der Datenauswertung sowie Optimierungsmethoden im Kontext des Modellaufbaus geschärft.
- Das Vorgehensmodell *RAMEn* ermöglicht die Auswahl und Prozessintegration von datengetriebenen Methoden für Entwicklungsprozesse rund um die FE-Simulation in der passiven Fahrzeugsicherheit und stellt das dafür benötigte Wissen strukturiert bereit.
- Die Methode *DrAGOS* hebt das Potenzial von RL, um eine vortrainierte Entscheidungsstrategie in der Produktgenerationsentwicklung wiederzuverwenden. Zudem wird zum Vortraining dieser Entscheidungsstrategie ein Metamodell eingesetzt, um lange Rechenzeiten zu umgehen.
- Die Methode *DAYSi* ist ein ressourceneffizienter Ansatz Signale von Crashesensoren automatisch zu analysieren und Ingenieure bei der Identifikation von Auffälligkeiten zu unterstützen. Zudem wird eine Eventkette rekonstruiert, die bei der Ursachenfindung unterstützt.
- Die Methode *ExPLAIN* ermöglicht die modulare Verarbeitung von skalaren, 1D-Zeitreihen- und FE-Daten aus FE-Simulationen und stellt Erklärungen für vorherrschendes Systemverhalten bereit. Erklärungen werden dabei auf verschiedenen Ebenen und im A/B-Vergleich bereitgestellt.
- Die drei eingeführten Methoden verteilen sich im gesamten Schnittfeld, das durch *RAMEn* abgebildet wird. Die Methoden sind dabei konkrete Maßnahmen die Entwicklungsprozesse im Kontext der FE-Simulation zu optimieren. Das Vorgehensmodell betrachtet hingegen übergeordnete Aspekte und stellt das theoretische Fundament für den bestmöglichen Einsatz bereit. Das zielt auch darauf ab, Ingenieure zum kombinierten Einsatz datengetriebener Methoden für verschiedenste Problemstellungen zu befähigen.

Der Fokus lag bisher vorwiegend auf der Konzeption und den theoretischen Aspekten der eingeführten Ansätze. Das liefert die Grundlage für Kapitel 6 ab S. 117 und Kapitel 7 ab S. 209 in, denen die Evaluierung und Diskussion stattfindet.

## 6 Evaluierung

Die in diesem Kapitel abgehandelten Aspekte entsprechen der Deskriptiven Studie II (Phase IV) des Aufbaus nach der DRM in Abbildung 1.1 auf S. 6. Die neu eingeführten entwicklungsunterstützenden Maßnahmen – *RAMEn*, *DrAGOS*, *DAYSi* und *ExPLAIN* – werden an repräsentativen Use Cases aus der Entwicklung passiver Sicherheitssysteme von Abschnitt 6.1 bis Abschnitt 6.4 evaluiert. Dies liefert die Grundlage für die jeweilige Diskussion der Ergebnisse anhand bestehender Methoden aus dem Stand der Technik, identifizierter Defizite dieser Methoden sowie den eingeführten Kriterien zur Bewertung und Einordnung.

### 6.1 *RAMEn* – Methodenauswahl

Die Evaluierung von *RAMEn* erfolgt mittels der gezielten Einordnung und Auswahl repräsentativer Methoden. Ebenfalls wird das Potenzial von strukturiert bereitgestelltem Wissen anhand einer Umfrage diskutiert. Zudem erfolgt die Generalisierung anhand einer weiteren Schnittfeldebetrachtung.

#### 6.1.1 Use Case I: Einordnung von Methoden

*RAMEn* kann zur systematischen Einordnung bestehender Methoden sowohl in den Aktivitäten als auch in der Anwendung genutzt werden, um den zielgerichteten Einsatz aktiv zu unterstützen (vgl. Abschnitt 5.1.2.5 auf S. 93). Ebenfalls unterstützt werden kann die Entscheidung über die Anschaffung neuer Softwaretools oder auch die interne Entwicklung neuer Methoden. Darüber hinaus lassen sich blinde Flecken identifizieren, in denen auch aus wissenschaftlicher Sicht neue Methoden entwickelt werden können. Dieses Vorgehen wird beispielhaft an der Methode CrashNet [BRK21] gezeigt. CrashNet prognostiziert mittels eines Convolutional Neural Networks auf Basis multimodaler Eingaben den zeitlichen Verlauf der Brustbeschleunigung des Fahrers (1D-Zeitserie). Die Eingaben setzen sich aus der Fahrzeugverzögerung (Crashpuls, 1D-Zeitserie) und Parameter des Rückhaltesystems (skalare Daten) zusammen. Als Datengrundlage wird auf eine öffentlich zugängliche Datenbank der NHTSA zurückgegriffen. Dies ermöglicht gute Generalisierungseigenschaften über verschiedene Fahrzeugtypen. In der praktischen Anwendung kann CrashNet zur Bewertung des Crashpulses in der frühen Entwicklungsphase verwendet werden.

- i. Die übergeordnete **Problemstellung** ist die Absicherung der passiven Fahrzeugsicherheit in der Entwicklung einer neuen Fahrzeuggeneration.

- ii. Das **Ziel** ist die Zeit- und Kostenreduktion durch die Abschätzung von Insassenbelastungen in der frühen Entwicklungsphase. Das **Ergebnis** ist dementsprechend die Ableitung von Anforderungen an die Fahrzeugstruktur und das Rückhaltesystem, die aus der Bewertung abgeleitet werden können, um spätere Phasen der Entwicklung in die bestmögliche Richtung zu lenken. Eine Anforderung kann beispielsweise das Kraftniveau eines Sicherheitsgurtes sein.
- iii. Die **Aktivität** im Schnittfeld kann folgendermaßen spezifiziert werden: Im Simulationsprozess wird die *Berechnung* im FEM-Löser unterstützt bzw. vollständig ersetzt. Die Eingaben werden entsprechend nicht mehr in einen FEM-Löser sondern direkt in das vorgeschlagene Metamodell eingegeben. Dieses ist – im trainierten Stadium – signifikant weniger rechenaufwändig und benötigt daher auch kein Hochleistungsrechencluster. Zusätzlich können durch die Einsparung möglicher Konzeptmodelle zur Crashpulsbewertung auch Aufwände im Modellaufbau reduziert werden. Die verwendete Architektur basiert auf einem Convolutional Neural Network ist daher in Bezug auf die *ML-Lernverfahren* dem Supervised Learning zuzuordnen.
- iv. Eine **Anwendung** der Methode ist vor allem in der *frühen Phase* der Entwicklung zum Treffen von ersten Abschätzungen sinnvoll. In späteren Phasen können Änderungen im Design genauer mittels Crashsimulationen und auch physischen Crashtests bestimmt werden. Da die Methode nur für den Insassen auf der Fahrerposition genutzt werden kann, und diese Betrachtungen in der Entwicklung zum großen Teil mittels Schlittensimulationen und Schlittenversuchen betrachtet werden, erfolgt eine Zuordnung der Betrachtungen zur *Subsystemebene*. Die *Aufgabe*, für die die Methode eingesetzt werden kann, ist die Bestimmung von Insassenbelastungen im Frontalaufprall. Die relevante *Disziplin* ist sowohl die Fahrzeugstruktur-, als auch die Rückhaltesystementwicklung, da die Bewertung des Fahrzeugpulses auch im Kontext der Rückhaltesystementwicklung stattfindet. In erster Linie sind damit die Bewertung des Fahrzeugpulses sowie die Konzeption und Anforderungserstellung des Rückhaltesystems als relevante Entwicklungsaufgaben aufzuführen. Durch die breite Generalisierung aufgrund der verwendeten Daten ist ein *projektübergreifender Einsatz* möglich.
- v. Durch die Generalisierung des Metamodells ist der Transfer von **Wissen** zwischen Projekten und Disziplinen möglich. Das Feintuning des Metamodells auf einen speziellen Fahrzeugtyp oder ein sukzessives Anpassen durch neu generierte Daten ist außerhalb des Betrachtungsrahmens.
- vi. Zur **Übergabe** zählen die abgeleiteten Anforderungen an die Fahrzeugstruktur und das Rückhaltesystem.

- vii.** In Bezug auf die **Methodendokumentation** lassen sich beispielsweise Eingabedaten – Fahrzeugpuls (1D-Zeitreihe) sowie Parameter des Rückhaltesystems (skalare Daten) – und Ausgabedaten – Brustbeschleunigung (1D-Zeitreihe) – festlegen. Der Anwenderkreis sind Ingenieure in der Fahrzeugsicherheit, die sich in der Konzeptentwicklung mit der Bewertung von Fahrzeugpulsen beschäftigen.

Das gerade angewendete Schema nach Abschnitt 5.1.2.5 auf S. 93 ist ebenfalls zur Einführung der in dieser Arbeit vorgestellten Methoden *DrAGOS*, *DAYSi* und *ExPLAIN* in Abschnitt 5.2 bis Abschnitt 5.4 als Teil der Präskriptiven Studie (Phase III) gezeigt worden. Diese drei Methoden, aber auch das gerade diskutierte *CrashNet*, adressieren alle unterschiedliche Problemstellungen in der Entwicklung crashbelasteter Systeme, was eine allgemeine Gültigkeit von *RAMEn* in diesem Anwendungsfeld attestiert.

Zur weiteren Untermauerung sind in dem in Abbildung 6.1 auf S. 120 aufgezeigten Schnittfeld weitere im Stand der Technik vorgestellte Methoden angeordnet. Zu sehen ist, dass das Schnittfeld den Betrachtungsrahmen vollständig abdeckt. Zusätzlich ist der Fokus der vorliegenden Arbeit in Orange hervorgehoben. Außerhalb des Fokus ist beispielsweise die generative Erstellung von FE-Netzen in [Sid+24], die Kompression von Simulationsdaten [Mül+23] oder auch große Sprachmodelle wie ChatGPT [Ope23] zu finden. Letztere zählen zu den Supervised-ML-Methoden, können jedoch, je nach Art und Umfang des Feintunings, auch bei den RL-Methoden eingeordnet werden. Gerade in Bezug auf Begleitprozesse können diese in diversen Alltagsaufgaben wie der Skripterstellung oder auch allgemeinen Fragen eine erste Hilfe darstellen. Spezifische Versionen in Bezug auf CAE-Workflows werden mittlerweile kommerziell angeboten [Ans24].

Aktivitätsfelder in Abbildung 6.1, die dünn besetzt oder leer bleiben, deuten auf bisher weniger stark betrachtete Aspekte in der Forschung hin. Der Hauptfokus dieser Arbeit, der ebenfalls in Abbildung 6.1 hervorgehoben ist, wurde entsprechend den zeitintensivsten Ingenieursaktivitäten – dem Modellaufbau und dem Postprocessing – festgesetzt. Ansätze außerhalb – wie ein RL-Agent zur Auswertung von FE-Simulationen – können als Teil zukünftiger Recherche- und Forschungsaktivitäten genauer zu untersucht werden. Wichtig zu erwähnen ist, dass wissenschaftlich neue Methoden auch innerhalb der einzelnen Aktivitätsfelder zu finden sind. Beispiele hierfür sind die Verarbeitung wenig fokussierter Datentypen, wie den 1D-Zeitreihendaten, oder die Entwicklung neuer physikbasierter ML-Algorithmen für CAE. In *RAMEn* wird daher die weitere Differenzierung anhand der Anwendung sowie den Detailinformationen in der Methodenbeschreibung vorgenommen.

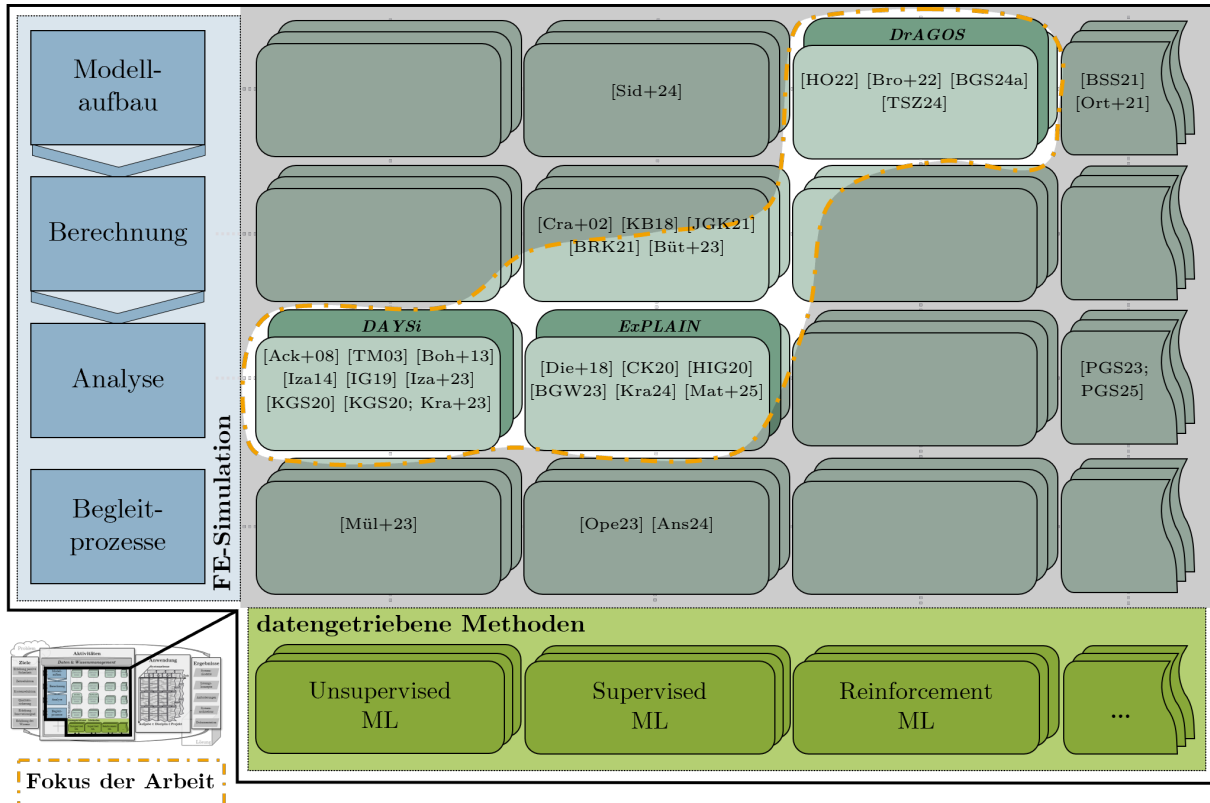
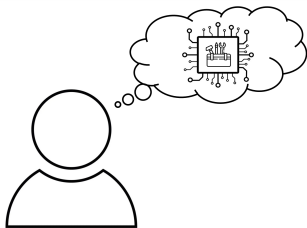


Abbildung 6.1: Einordnung von Methoden aus dem Stand der Technik

### 6.1.2 Use Case II: Auswahl von Methoden

Die zuvor erläuterte Einordnung bildet die Grundlage für die nun diskutierte systematische Methodenauswahl, die entsprechend rückwärts abläuft. Basierend auf den zugewiesenen Kategorien können bestimmte Methoden je nach Zugehörigkeit ein- oder ausgeschlossen werden. Anhand der beispielhaften Fragestellung wird dies im Folgenden erläutert.



*Inwiefern kann mich eine intelligente Methode bei dem Abgleich von Versuch und Simulation eines Pfahl-Seitencrashes unterstützen, um Zeit einzusparen? Da ich im letzten Projekt einen fehlerhaften Sensor übersehen habe, möchte ich im Idealfall auch auf Fehler automatisch hingewiesen werden!*

Zu Zwecken der Anschaulichkeit sind die einzelnen Schritte in Abbildung 6.2 verdeutlicht. Eine Vorauswahl kann anhand der Aktivitäten im Schnittfeld erreicht werden, da es sich um Aktivitäten im Bereich der *Analyse* handelt. Die Sortierung erfolgt unabhängig vom ML-Lernverfahren, da aus der Frage keine Anforderungen dahingehend hervorgehen. Eine Verfeinerung erfolgt durch die Betrachtung der Anwendung im Prozess, wobei die *mittlere bis späte Entwicklungsphase* aufgrund bereits vorhandener Versuche, sowie die *Systemebene*, da der Seitencrash meist auf Gesamtfahrzeugebene betrachtet wird, im Fokus stehen. Eine

weitere Differenzierung ist an dieser Stelle nicht vorzunehmen, aufgrund fehlendem Bezug zu einem bestimmten *Projekt* oder einer *Disziplin*. Die *Aufgabe*, für die der Ingenieur Unterstützung sucht, kann allgemein als Sensordatenanalyse beschrieben werden. Die Methodenbeschreibungen bilden nun die Grundlage für weitere Entscheidungen und den detaillierten Vergleich. Relevant ist hier insbesondere der Datentyp, da Sensorsignale (1D-Zeitreihendaten) verarbeitet werden müssen. Dieser ist in der Methodendokumentation aufgeführt.

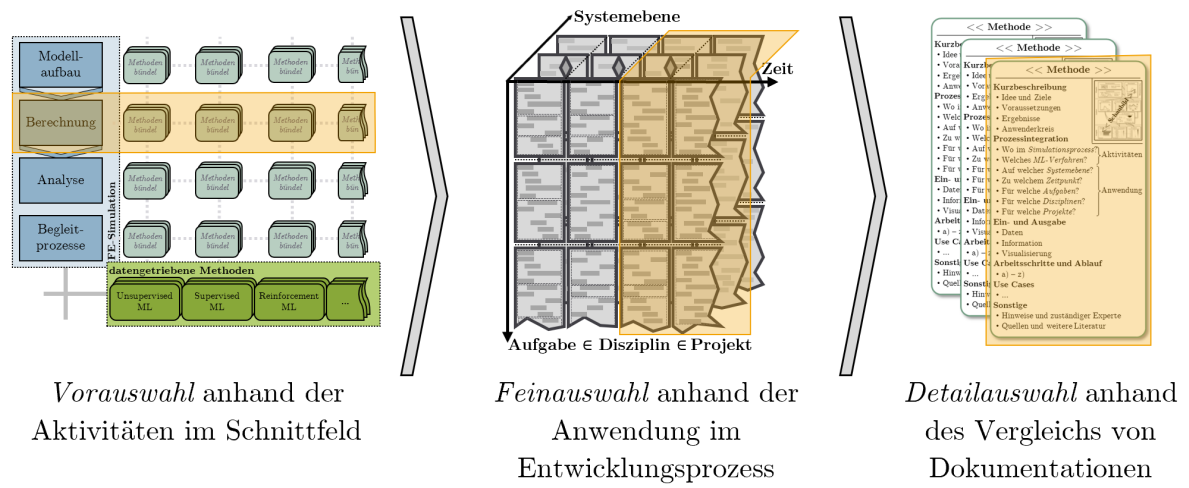


Abbildung 6.2: RAMEn: Auswahl von Methoden im Vorgehensmodell

Methoden, die darauf zutreffen, sind [Iza+23] oder *DAYSi*. Zusätzlich kann noch die eigene Arbeit [MDV23], auf der *DAYSi* aufbaut, genannt werden. Da nach einem automatischen Fehlerhinweis gefragt wird, ist *DAYSi* an dieser Stelle zutreffender. Würde hingegen nach einer interaktiven Möglichkeit der Datenanalyse gefragt, ist [Iza+23] vorzuziehen.

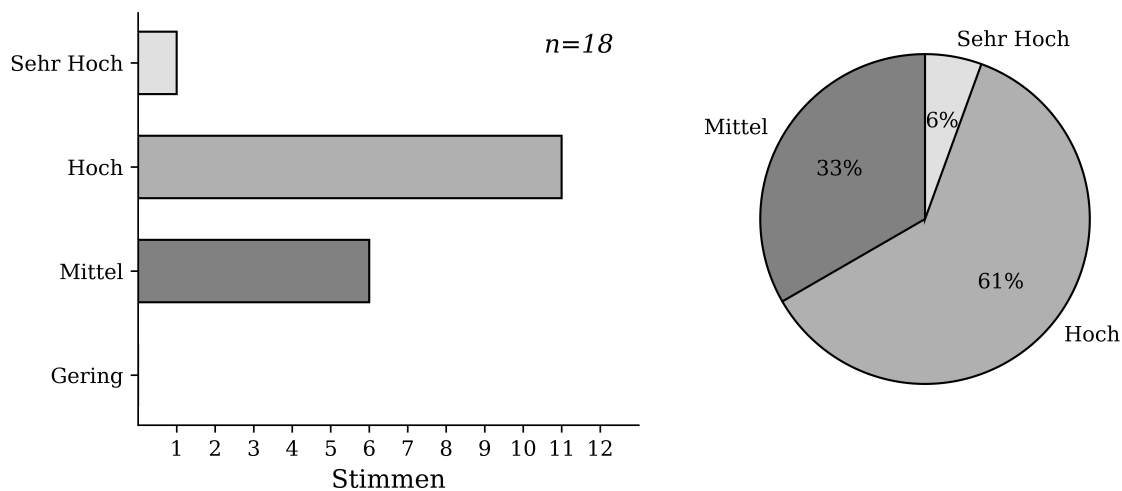
### 6.1.3 Strukturierte Verfügbarkeit von Wissen

Um zu evaluieren wie hoch das Potenzial von strukturiert verfügbarem Wissen ist, wurden im September 2024 18 Personen, die im Bereich CAE und Fahrzeugsicherheit tätig sind, anonym befragt. Alle Beteiligten sind Entwicklungsingenieure und sind entweder im Bereich CAE mit dem Fokus Crashsimulation oder dem Bereich Versuch tätig. Die überwiegende Mehrheit der Befragten hat mehr als fünf Jahre Berufserfahrung in diesem Tätigkeitsfeld. Die Fragen, Antwortmöglichkeiten und deren Häufigkeit sind in Abbildung 6.3, Abbildung 6.4 und Abbildung 6.5 gezeigt. Während bei den ersten beiden Fragen nur jeweils eine Antwortmöglichkeit abgegeben werden konnte, waren bei der dritten Frage zwei verschiedene Antworten zulässig.

Die erste Frage in Abbildung 6.3 beschäftigt sich mit dem grundlegenden Potenzial von KI in der passiven Fahrzeugsicherheit. Es konnte zwischen vier Stufen der Zustimmung

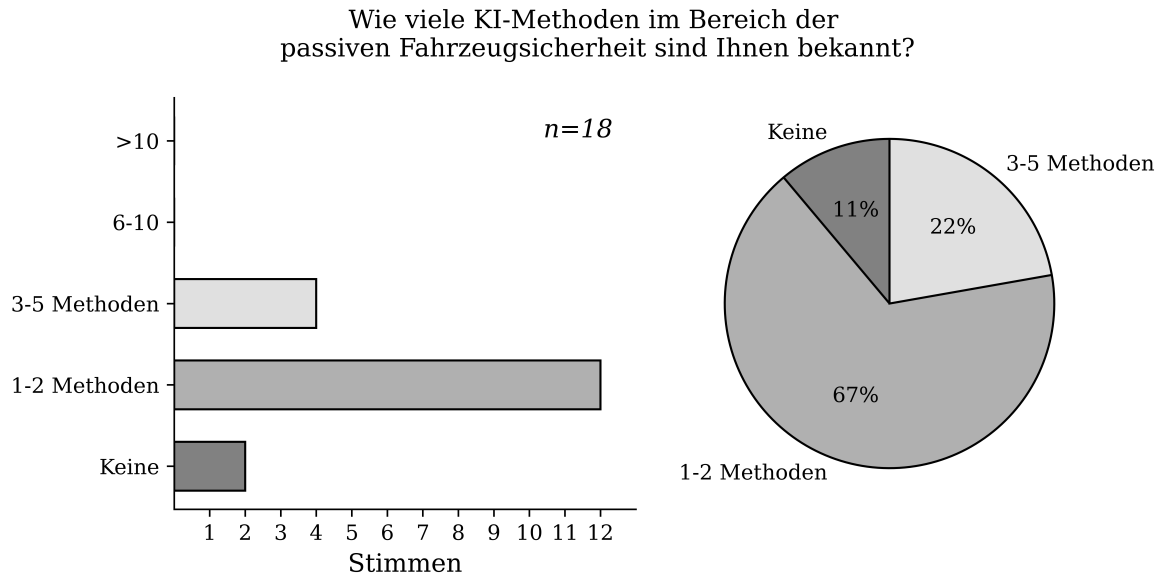
gewählt werden. Alle Befragten sehen das grundsätzliche Potenzial des Einsatzes von KI mindestens als mittelgroß an. Die Mehrheit von 67 % schätzt das Potenzial sogar hoch bis sehr hoch ein. Auch wenn davon auszugehen ist, dass die Kernexpertise der Befragten eher im Bereich des Ingenieurwesens und weniger im Bereich der KI selbst ist, weisen alle Befragten ein hohes Maß an generellem technischen Verständnis auf. Folglich lässt sich trotzdem eine klare Tendenz und Erwartung ablesen.

Wie hoch schätzen Sie das Potenzial von künstlicher Intelligenz (KI) in Bezug auf die Optimierung der Entwicklungsprozesse im Bereich der passiven Fahrzeugsicherheit ein?

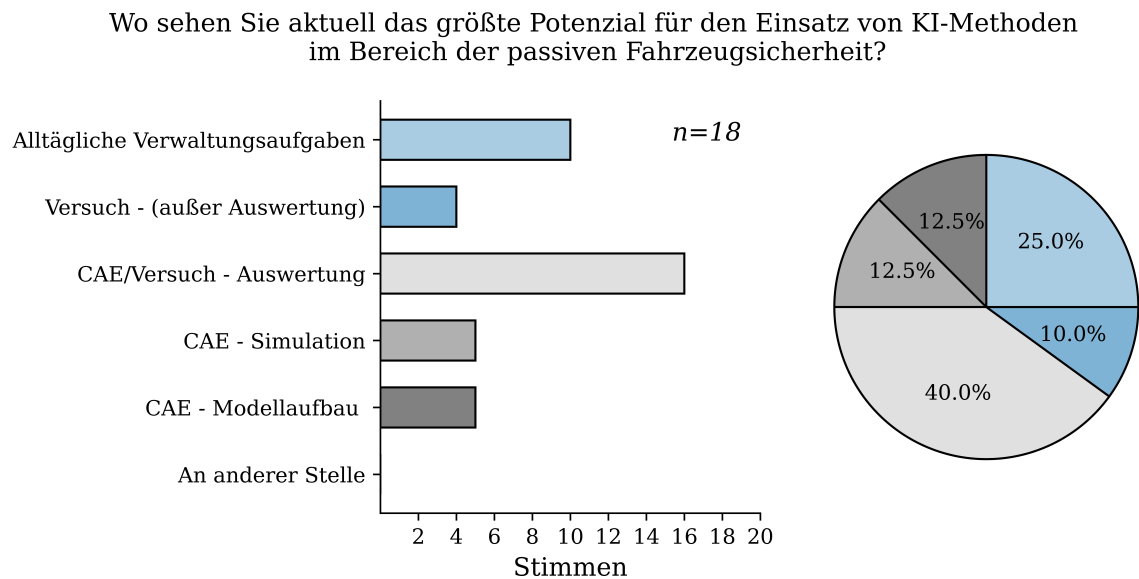


**Abbildung 6.3:** Umfrage zu KI in der passiven Fahrzeugsicherheit – Frage I

Ebenfalls wurde gefragt, wie viele KI-Methoden den Befragten jeweils bekannt sind. Die Ergebnisse sind Abbildung 6.4 zu entnehmen. Der Begriff KI-Methoden ist hier als allgemeiner Überbegriff für intelligente Methoden zu verstehen. Während einer Minderheit von 11 % keine Methoden bekannt sind, geben 89 % der Befragten an mindestens eine Methode zu kennen. Eine Mehrheit von 67 % kennen eine bis zwei Methoden. Im Stand der Technik dieser Arbeit wurden allerdings weitaus mehr Methoden für unterschiedliche Anwendungsfälle identifiziert. Dieses Ergebnis steht konträr zu der ersten Frage in Abbildung 6.3, da dem hohen Potenzial eher wenig spezifisches Wissen gegenübersteht. Folglich ist davon auszugehen, dass die mittels *RAMEn* strukturiert bereitgestellte Information dazu beiträgt neue Methoden bekannt zu machen und Ingenieure aktiv dabei unterstützt diese zu adaptieren. Basierend auf dem hohen Potenzial ist davon auszugehen, dass beteiligte Ingenieure Methoden – sofern diese zur Verfügung stehen – auch einsetzen und das grundlegende Interesse da ist, sich mit dem Thema zu beschäftigen.



**Abbildung 6.4:** Umfrage zu KI in der passiven Fahrzeugsicherheit – Frage II



**Abbildung 6.5:** Umfrage zu KI in der passiven Fahrzeugsicherheit – Frage III

Abschließend wurde abgefragt, welchen Aufgaben das höchste Potenzial zugewiesen wird. Unter der Rubrik „An anderer Stelle“ konnten eigene Antwortmöglichkeiten hinzugefügt werden, was dafür spricht, dass die Hauptaufgaben durch die gegebenen Möglichkeiten abgedeckt sind. Insbesondere in der Auswertung von Versuch und CAE, die Analysemethoden unterstützt werden können, sehen 40% der Befragten das höchste Potenzial. Interessanterweise entfallen jedoch mit 25% die zweitmeisten Stimmen auf alltägliche Verwaltungsaufgaben. Die Ergebnisse sind in guter Übereinstimmung mit dem Fokus

dieser Arbeit, da die Analysemethoden im Vordergrund stehen. Dies ist schlüssig, da ein großer Teil der Arbeitszeit aller Beteiligten auf die Analyse und Interpretation von Crashsimulationen und zugehörigen Versuchen, insbesondere auf Subsystem- und Systemebene, entfällt.

#### 6.1.4 Generalisierbarkeit

Die Grundidee – die verbesserte Orientierung in komplexen Schnittfeldern im Kontext der Produktentwicklung – wurde bereits in Abschnitt 5.1.1 auf S. 87 eingeführt. Bisher wurde gezeigt, dass *RAMEn* für die Entwicklung crashbelasteter Systeme funktioniert und dort zur Bereitstellung von Informationen zur Wissensgenerierung und -bewahrung beiträgt. Um nun weiter von der zugrundeliegenden Idee zu profitieren, wird nun beispielhaft das Schnittfeld aus *Leichtbau* und *datengetriebenen Methoden* betrachtet, um das Generalisierungspotenzial aufzuzeigen.

Eine Gesamtgewichtsreduktion oder ein verringerter Ressourcenverbrauch können in diesem Fall als *Ziele* formuliert werden. Zu den Ergebnissen zählen eine funktionale Systemarchitektur und ein dreidimensionales Systemmodell. Die *Aktivitäten* im Schnittfeld sind in Abbildung 6.6 abgebildet. Analog zum bisherigen Beispiel und in Anlehnung an [ISO 23053] sind die datengetriebenen Methoden in die drei Lerntypen des ML und sonstige Verfahren untergliedert. Das Feld Leichtbau wird hier in Form der fünf Leichtbaustrategien [KG19, S. 37–40] dargelegt. Der *Materialleichtbau* beschreibt den gezielten Austausch konventioneller Werkstoffe durch leichtere Alternativen, um das Bauteilgewicht zu reduzieren. Der *Formleichtbau* konzentriert sich auf die optimale Verteilung des Materials innerhalb des Bauteils, indem es gezielt dort eingesetzt wird, wo es aufgrund der wirkenden Belastungen erforderlich ist. Der *Bedingungsleichtbau* berücksichtigt äußere Einflussfaktoren und Nutzungsbedingungen, um das Gewicht durch Anpassungen von Belastungen und Lebensdauer zu verringern. Der *Konzeptleichtbau* beschäftigt sich mit der Auswahl bestmöglicher Bauweisen, wie der Integral- oder Differentialbauweise. Schließlich zielt der *Fertigungsleichtbau* darauf ab, durch verbesserte Fertigungsverfahren und Fügeverfahren Gewichtseinsparungen zu realisieren. Auch hier lassen sich bereits beispielhaft einige datengetriebenen Methoden identifizieren. Hinsichtlich dem Formleichtbau lassen sich hier ebenfalls die bereits besprochenen Methoden von Trilling, Schumacher und Zhou [TSZ24] und Büttner et al. [Büt+23] einordnen, die die Topologie bzw. die Dimensionierung optimieren und dadurch Gewicht einsparen können.

Darüber hinaus lassen sich Methoden im Bedingungsleichtbau finden, die unter Verwendung einer PCA einen Gesundheitsindikator für rotierende Maschinenteile ableiten [Son+24a]. In Bezug auf Fertigungsleichtbau wird in [Joh+24] der Zusammenhang zwischen Stellgrößen im Schweißprozess und Zielgrößen wie der Festigkeit betrachtet. Diese Methoden tragen

alle dazu bei, Bauteile besser auszulegen oder im Betrieb zu überwachen, was eine bessere Ausnutzung der Materialien ermöglicht und so einen Beitrag zur Ressourceneffizienz leistet. Weitere Überlegungen hinsichtlich der Generalisierung in dem Schnittfeld Leichtbau und Nachhaltigkeit sind der eigenen vorangegangenen Arbeit [MV24] zu entnehmen.

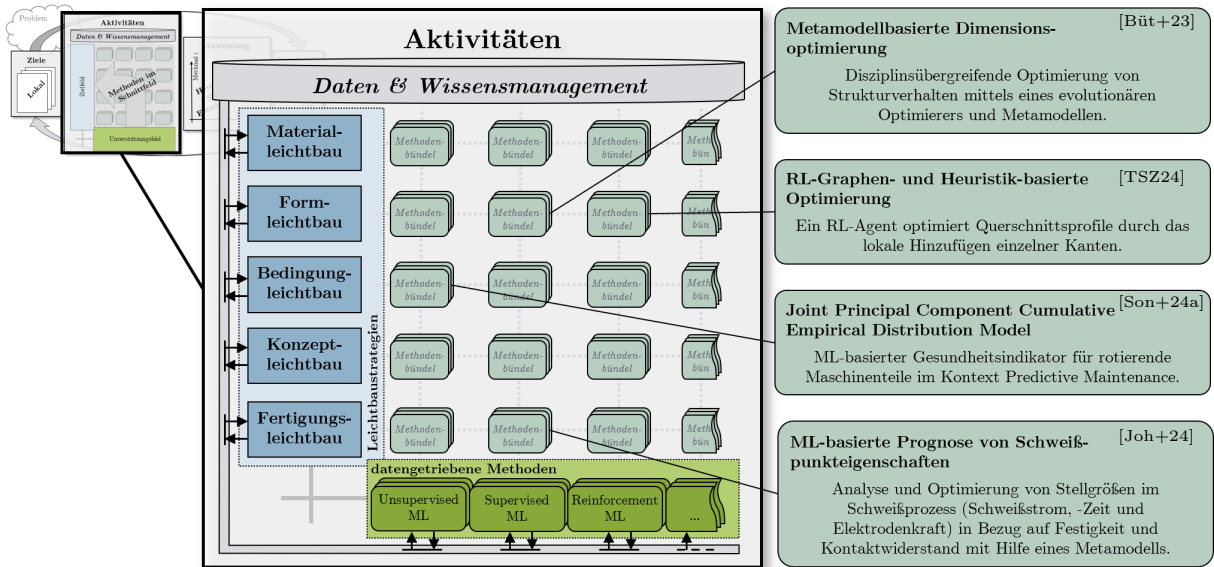


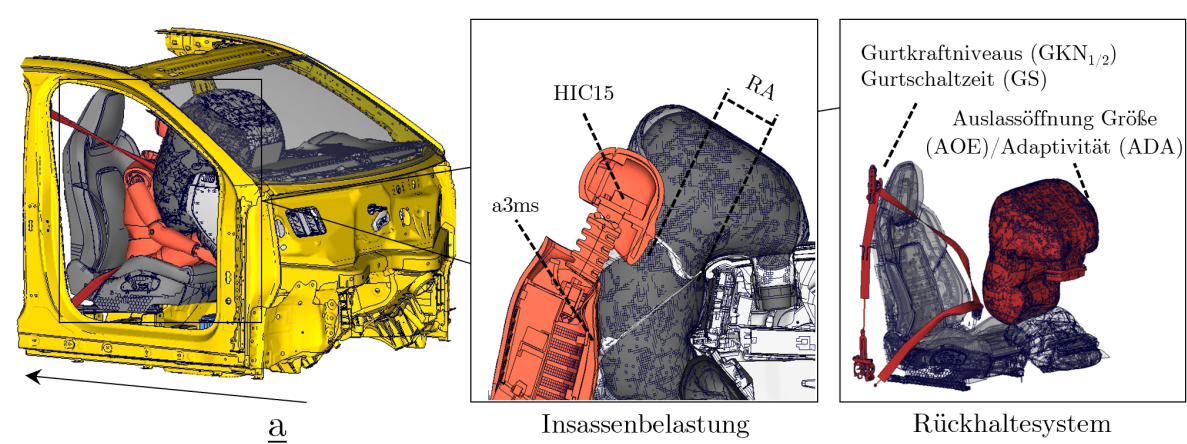
Abbildung 6.6: *RAMEn* Generalisierung: Leichtbau und datengetriebene Methoden

## 6.2 *DrAGOS* – Datengetriebene Systemauslegung

Die Evaluierung von *DrAGOS* erfolgt anhand der Auslegung von Rückhaltesystemen. Zur Bewertung der Ergebnisse wird zusätzlich ein Benchmark mit einem reinen Metamodellansatz betrachtet. Ebenfalls werden die in Abschnitt 5.2 auf S. 94 erwähnten Verfahrensparameter diskutiert. Für die Betrachtung der Use Cases selbst, werden die Verfahrensparameter auf den bereits definierten Standardwerten belassen.

### 6.2.1 Rückhaltesystemauslegung

Ein repräsentatives Beispiel für eine produktgenerationsübergreifende Aufgabenstellung ist die Auslegung von Rückhaltesystemen, da diese in jedem Fahrzeug verbaut sein müssen, um Insassen in einem Crashszenario bestmöglich zu schützen. In der Praxis ist die Problemstellung komplex, da das Rückhaltesystem für jede Sitzposition so ausgelegt werden muss, dass Belastungswerte für mehrere ATD-Typen in mehreren Lastfällen erfüllt werden. Da die Kopplung wieder eine separate Problemstellung darstellt, wird hier – vereinfacht – nur ein ATD Typ in einem Lastfall betrachtet. Gegenstand der Evaluation ist die Beifahrerposition im Lastfall Frontalaufprall  $0^\circ$  gegen eine starre Wand mit 56 km/h (FN-56) gemäß FMVSS 208 [FMVSS208].



**Abbildung 6.7:** *DrAGOS* Use Case: Rückhaltesystemauslegung

Um den numerischen Aufwand in der Auslegung zu verringern, wird das FE-Modell reduziert. Betrachtet werden dabei typischerweise der Rohbau mit Interieur, das Rückhaltesystem und der ATD. Der Rohbau wird mit einer vordefinierten Beschleunigung aus der Crashsimulation eines Gesamtfahrzeugs oder einem physischen Crashtest beschleunigt und das Rückhaltesystem gezündet. Anschließend können Kinematik und Belastungen des Insassen ausgewertet werden. Ein solches FE-Modell ist in Abbildung 6.7 gezeigt und findet sich auch in mehreren Literaturquellen wieder [Thi+06; JGK21; HCK24]. Diese sog.

Schlittensimulationen sind trotz Reduktion rechenintensiv und weisen Laufzeiten von bis zu 20 Stunden auf einem Hochleistungsrechnercluster auf. Ebenfalls sind in Abbildung 6.7 die Metaparameter des Rückhaltesystems aufgeführt, die für die betrachtete Problemstellung relevant sind. Hierbei handelt es sich in Bezug auf das Rückhaltesystem um zwei Gurtkraftniveaus  $GKN_{1/2}$ , die Schaltzeit GS zwischen diesen Kraftniveaus sowie die Adaptivität ADA und Größe AOE der Auslassöffnung am Beifahrerairbag. Für die Bewertung der Insassenbelastung wird das HIC15 und die maximale Brustbeschleunigung über einem Intervall von 3 ms (a3ms) verwendet. Der Restabstand ATD-Kopf zu Instrumententafel (RA) stellt eine Randbedingung dar, die erfüllt sein muss.

All diese Größen beschreiben typische Auslegungsgrößen und Bewertungskriterien, die in der Entwicklung einer neuen Fahrzeuggeneration relevant sind. Von weiteren Größen, wie Zündzeiten oder weitere Straffer, wie am Endbeschlag des Gurts, soll der Einfachheit halber abgesehen werden. Die beschriebene Problemstellung ist auf Basis der in Abschnitt 2.3.2.3 auf S. 38 eingeführten Optimierungstheorie in Gleichung (6.1) dargestellt.

$$\min_{s_t \in \mathbb{R}^5} [\text{HIC15}(s_t), \text{a3ms}(s_t)] \quad \text{s.t.} \quad \text{RA}(s_t) > \text{RA}_{\min}, s_t = \begin{bmatrix} \text{GKN}_1 \\ \text{GKN}_2 \\ \text{GS} \\ \text{ADA} \\ \text{AOE} \end{bmatrix} \quad (6.1)$$

## 6.2.2 Anpassung der Architektur

Der grundlegende Aufbau von *DrAGOS* wurde in Abbildung 5.5 auf S. 95 bereits erläutert. Die Anpassung für die Rückhaltesystemauslegung ist in Abbildung 6.8 auf S. 128 abgebildet. Aufgrund der Anforderung, dass relative Änderungen in das System eingebracht werden, muss der RL-Agent in der Lage sein diskrete Aktionsräume zu verarbeiten. Für die vorliegende Problemstellung wird ein PPO-RL-Agent verwendet. Dieser ermöglicht die Findung einer Balance zwischen Exploration und Exploitation, was für die Entdeckung von neuen Strategien begünstigt. In zahlreichen Anwendungen hat sich PPO als sampleeffizient erwiesen [Bey19, S. 36–38].

Der RL-Agent spezifiziert die Aktionen innerhalb eines definierten Aktionsraums. Dieser Aktionsraum hat nach Abbildung 6.8 fünf Dimensionen, von denen jede drei diskrete Aktionen anbietet, um die Parameter zur Spezifikation der Rückhaltesystemkonfiguration zu verändern. Die Parameter können erhöht, verringert oder konstant gehalten werden. Die zulässigen Schritte sowie der sich daraus ergebende Aktionsraum  $\mathcal{A}$  sind dabei Gleichung (6.2) zu entnehmen. Die Belohnungsfunktion ist in Gleichung (6.3) dargestellt und beschreibt die gewichtete Summe des quadrierten Verhältnisses des gesetzlichen Grenzwerts gemäß FMVSS 208 zu den relativen Belastungen des HIC15 und der a3ms.

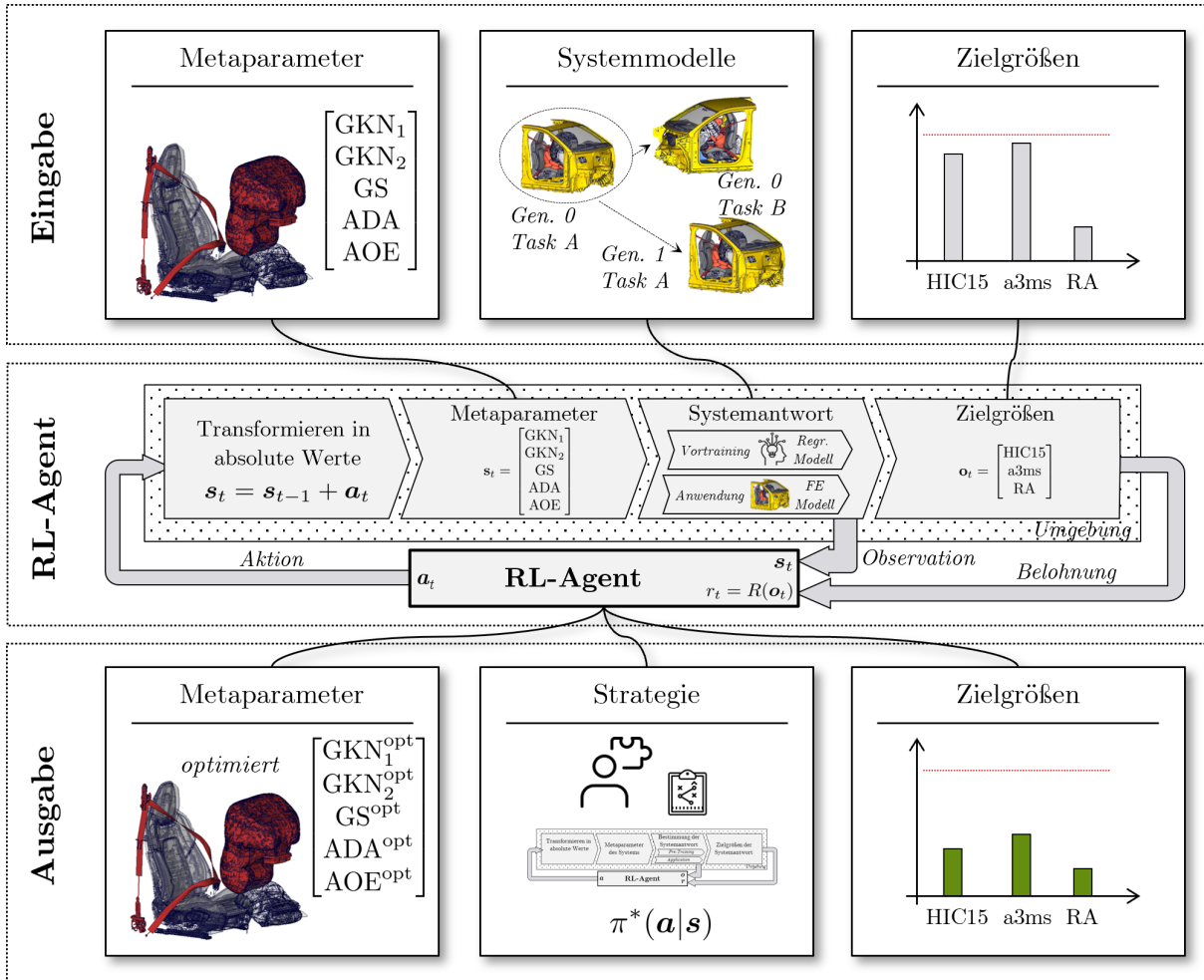


Abbildung 6.8: DrAGOS Formulierung für die Rückhaltesystemauslegung

Eine Belohnung nach Gleichung (6.3) erhält der Agent nur, wenn erlaubte Grenzen nach Gleichung (6.1) erfüllt sind. Andernfalls wird dieser mit  $-1$  bestraft. Diese Bestrafung zwingt den Agenten, insbesondere im Trainingsvorgang, den Lösungsraum zu explorieren und dabei alle Zielvariablen in dem gewünschten Bereich zu halten. Der Observationsraum enthält die Parameter des Rückhaltesystems nach Abbildung 6.7 und spiegelt somit den Zustand der Umgebung  $s_t$  wider.

$$\mathcal{A} = \left\{ a_t \in \mathbb{R}^5 \mid a_i \in \begin{bmatrix} -0.25 & -0.25 & -4 & -4 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0.25 & 4 & 4 & 1 \end{bmatrix}, i = 1, \dots, 5 \right\} \quad (6.2)$$

mit Einheiten  $\begin{cases} \text{kN} & \text{für } i = 1, 2 \\ \text{ms} & \text{für } i = 3, 4 \\ \text{mm} & \text{für } i = 5 \end{cases}$

$$r_t = R(o_t) = -0.05 + \begin{cases} \left(\frac{700}{\text{HIC15}}\right)^k + \left(\frac{60}{a_{3\text{ms}}}\right)^k & \text{für } \begin{cases} \text{RA} > \text{RA}_{\min} \wedge \\ \text{HIC15} < 700 \wedge \\ a_{3\text{ms}} < 60 \text{ g} \end{cases} \\ -1, & \text{sonst} \end{cases}, \text{ mit } k = 2 \quad (6.3)$$

### 6.2.3 Vortraining mittels Metamodell

Wie in Abbildung 6.8 gezeigt, unterscheidet *DrAGOS* zwischen einem Trainings- und einem Anwendungsmodus. Im Trainingsmodus erfolgt das Training des RL-Agenten auf einem Metamodell, um die hohen Rechenzeiten der Schlittenmodelle zur Bestimmung der Insassenbelastungen zu umgehen. Dafür muss das Metamodell erst auf Basis vorhandener Daten trainiert werden. Hierfür werden nach Abbildung 5.5 auf S. 95 Daten eines Fahrzeugs der  $i$ -ten Generation für eine Problemstellung  $A$  herangezogen.

Problemstellung  $A$  entspricht nach Abbildung 6.7 der Auslegung des Rückhaltesystems auf der Beifahrerposition. Der repräsentative Datensatz für diese Problemstellung enthält Simulationen aus verschiedenen Lastfällen im Frontalaufprall mit  $0^\circ$  (32 km/h und 40 km/h ungesurtert sowie 50 km/h und 56 km/h gesurtert) aus mehreren Entwicklungsphasen und verschiedene Fahrzeugkonfigurationen für drei verschiedene ATD-Typen und zwei Fahrzeuggenerationen. Insgesamt werden 271 Schlittenmodelle verwendet. Die Daten sind repräsentativ für die Aufgaben, die im Laufe einer typischen Rückhaltesystementwicklung entstehen.

Um die Vielfalt der Daten abzubilden, werden jedoch mehr als nur die Metaparameter benötigt. Weitere Informationen zu den Daten, der Architektur und verwendeten Hyperparametern sind dem Anhang (vgl. Abschnitt A.1.1 auf S. 257) zu entnehmen. Dort sind ebenfalls Plots zur visuellen Interpretation der Prognosegüte gezeigt.

#### 6.2.3.1 Auswahl des Metamodells

Um das bestmögliche Modell für den Anwendungsfall zu ermitteln, werden verschiedene Metamodelle unter selben Bedingungen trainiert und Hyperparameter optimiert. Die Bewertung des Modells wird mittels einer 5-fold Cross Validation vorgenommen, wobei die Auswertegrößen über diese gemittelt werden. Berechnet werden  $R^2$  und RMSE jeweils einzeln für die drei Ausgaben HIC15,  $a_{3\text{ms}}$  und RA. Diese sind in Tabelle 6.1 gelistet. Verwendet werden TabPFN [Hol+23; Hol+25], XGBoost [CG16], Support Vector Regression [Dru+96], Multilayer Perceptron<sup>1</sup> [RHW86], Decision Trees [Bre+84] und Lasso Regression [Tib96]. In Bezug auf die Auswertegrößen für alle betrachteten Ausgaben weist das TabPFN

<sup>1</sup>Die Architektur wird im Anhang in Tabelle A.4 auf S. 260 genau erläutert.

mit bis zu einem  $R^2$  von 0.93 eindeutig die höchste Prognosegüte auf. Während  $R^2$  auch ausgabenübergreifend zum Vergleich verwendet werden kann, kann der RMSE nur zum Vergleich der Güte innerhalb einer Größe dienen. Die höchste Güte ergibt sich in allen Modellen jeweils für den HIC15. Der geringste  $R^2$  von 0.15 wird für den RA und Decision Trees gefunden.

**Tabelle 6.1:** Metamodelle zur Prognose von ATD Belastungen im Frontalaufprall

		TabPFN	XGBoost	Support Vector Regr.	Multilayer Perceptron	Decision Trees	Lasso Regr.
$R^2$	HIC15	<b>0.93</b>	0.91	0.83	0.83	0.84	0.73
	a3ms	<b>0.77</b>	0.74	0.55	0.58	0.54	0.44
	RA	<b>0.89</b>	0.87	0.84	0.82	0.15	0.53
RMSE	HIC15	<b>47.0</b>	52.0	75.4	74.4	71.7	93.8
	a3ms	<b>2.78</b>	2.97	4.19	3.99	4.15	4.67
	RA	<b>8.75</b>	9.61	10.6	11.3	24.6	18.3

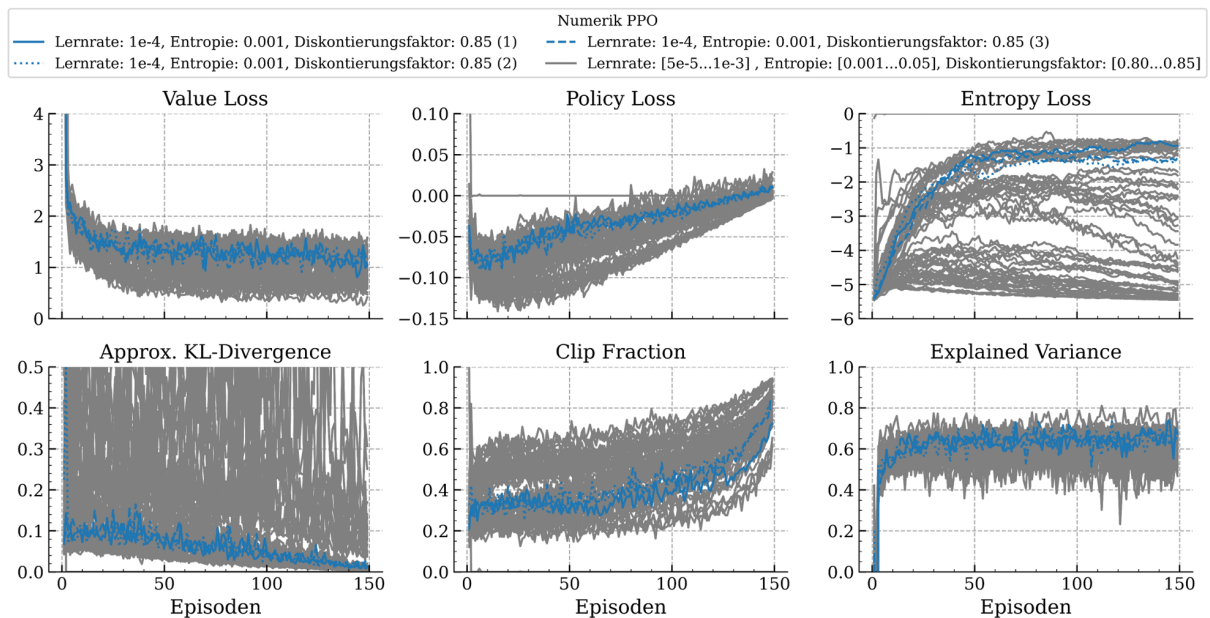
Die Brustbeschleunigung weist tendenziell das geringste Bestimmtheitsmaß auf. Von den klassischen Metamodellen ist XGBoost die eindeutig beste Wahl. In Bezug auf das Bestimmtheitsmaß ist die Prognosegüte maximal 3 % schlechter als für TabPFN. Ebenfalls im Training ist TabPFN mit Zeiten unter 3s äußerst effizient, da dieses aufgrund der Architektur keine Hyperparameter Optimierung benötigt. Das Training von XGBoost dauert im Schnitt 15s ohne und 55 min mit Hyperparameteroptimierung. Im weiteren Vergleich von XGBoost und TabPFN fällt jedoch insbesondere die deutlich schnellere Inferenzzeit von XGBoost positiv ins Gewicht, da gerade diese relevant für das Vortraining des RL-Agenten ist. TabPFN weist eine um 82 % höhere Inferenzzeit auf. Dies entspricht auch den Erkenntnissen von [Hol+25], da das Modell nicht auf Inferenz optimiert wurde. Folglich würde das Vortraining des RL-Agenten nahezu doppelt so viel Zeit in Anspruch nehmen, weshalb für die akademischen Untersuchungen aufgrund der Effizienz XGBoost verwendet wird. Die übrigen Metamodelle in Tabelle 6.1 dienen lediglich als Benchmark, werden aber nicht weiter betrachtet.

### 6.2.3.2 Trainingsprozess des Reinforcement-Learning-Agenten

Im nächsten Schritt wird dann entsprechend Abbildung 6.8 der RL-Agent auf dem XGBoost-Metamodell vortrainiert. Benötigte Eingaben außerhalb des Rückhaltesystems werden entsprechend Abschnitt A.1.1 einfach fest gesetzt. Der PPO RL-Agent wird in zwei Schritten angepasst: (1) Setzung der PPO-spezifischen Parameter, zur Gewährleistung numerischer Stabilität und (2) Analyse der Netzwerkgrößen, Schrittweite und Belohnungsfunktion. Das Training umfasst 150 Episoden mit jeweils maximal 300 Schritten. Zu Beginn jeder Episode wird die Umgebung auf einen beliebigen Startpunkt zurückgesetzt, was bedeutet,

dass der RL-Agent das Systemverhalten von mehreren Startpunkten aus optimieren muss. Der gesamte Trainingsprozess zur Ermittlung der optimalen Strategie  $\pi^*$  dauert 45 min auf einer CAE-Workstation.

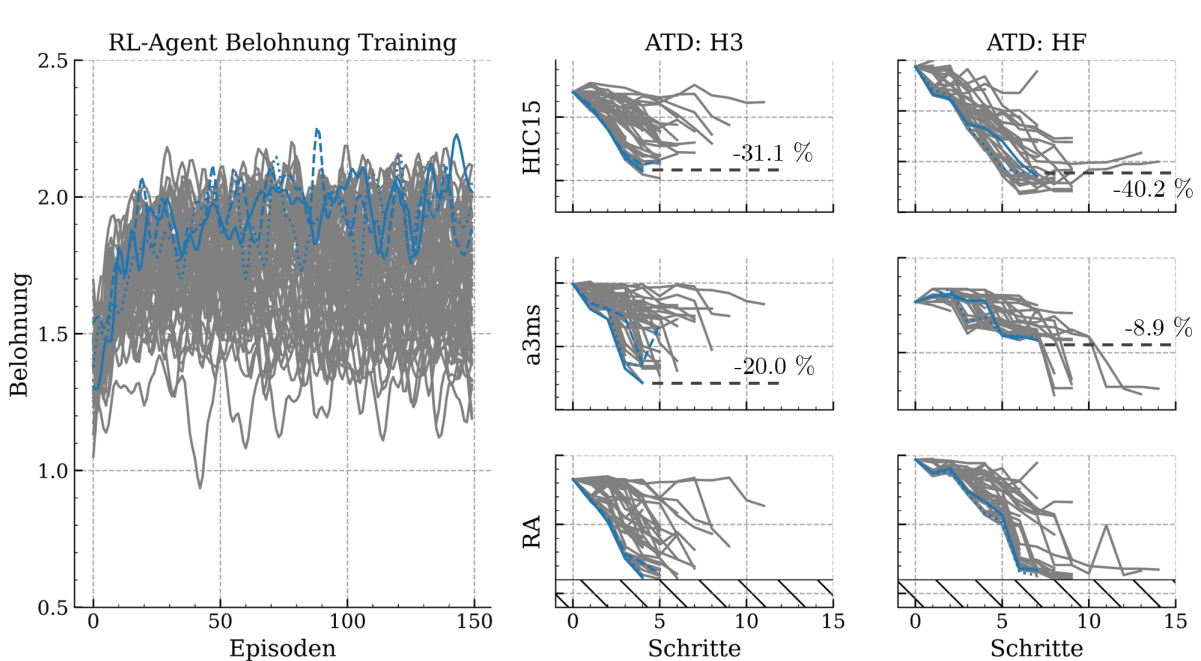
Die während dem Trainingsvorgang ausgewerteten PPO-spezifischen Größen sind in Abbildung 6.9 gezeigt. Eine genaue Erklärung dieser ist dem Anhang in Abschnitt A.1.3 auf S. 260 zu entnehmen. Wichtige Hyperparameter sind die Lernrate  $\eta$ , der Entropiekoeffizient  $\beta_{\mathcal{H}}$  zur Steuerung der Exploration und Exploitation, sowie der Diskontierungsfaktor  $\gamma$  zur Steuerung des Gewichts zukünftiger Belohnungen [Sch+17]. Diese werden im Rahmen einer Robustheitsstudie in den Grenzen  $\eta \in [5e-5, 3e-4, 1e-4, 1e-3]$ ,  $\beta_{\mathcal{H}} \in [0.8, 0.85, 0.9, 0.95]$  und  $\gamma \in [0.001, 0.01, 0.05]$  variiert. Die vollständigen Plots sind ebenfalls Abschnitt A.1.3 auf S. 260 zu entnehmen. Zwecks Platzgründen wird hier nur die für am besten befundene Konfiguration im Kontext der übrigen diskutiert.



**Abbildung 6.9:** *DrAGOS* Robustheit: Trainingsprozess mit variiertem Lernrate, Entropiekoeffizient und Diskontierungsfaktor

Abbildung 6.9 ist zu entnehmen, dass das Verfahren nicht für beliebige Parameter stabil ist. Insbesondere die approximierten Kullback-Leibler-Divergenz (KL-Divergenz) weist für zu hohe Lernraten große Sprünge auf. Die Verluste für Value und Policy sowie die Explained Variance konvergieren hingegen für alle Konfigurationen. Der Entropieverlust konvergiert jedoch nur für niedrigere Entropiekoeffizienten zu geringeren Werten. Mit fortlaufendem Training wird der RL-Agent also deterministischer. Für hohe Entropiekoeffizienten 0.1 oder 0.05 divergiert der Entropieverlust leicht. Der Agent exploriert auch im späteren Training

mehr. Ein leichter Einfluss ist auch auf den mittlere Belohnung pro Episode festzustellen, die für niedrigere Entropiekoeffizienten etwas höher ist.



**Abbildung 6.10:** *DrAGOS* Robustheit: Trainings- und Anwendungsprozess

In Abbildung 6.9 und Abbildung 6.10 in blau hervorgehoben sind drei identisch Durchläufe für die Lernrate  $1e-4$ , Entropiekoeffizient 0.001 und Diskontierungsfaktor 0.85. Die genannten Verfahrensparameter wurden gewählt, da hiermit in Abbildung 6.10 die beste mittlere Belohnung ermittelt wurde und ausgehend von den Auswertegrößen in Abbildung 6.9 ein stabiles Training möglich ist. Eine Steigerung der Belohnung findet zwar für alle Verfahrensparameterkombinationen statt, allerdings fällt diese unterschiedlich hoch aus. Für den hervorgehobenen Verfahrensparameter liegt eine Steigerung von initial 1.31 bis 2.26 vor, was 73% entspricht. Die Ergebnisse sind zudem reproduzierbar, weshalb anzunehmen ist, dass die Initialisierung eine untergeordnete Rolle spielt. Dementsprechend wird im Folgenden mit Lernrate  $1e-4$ , Entropiekoeffizient 0.001 und Diskontierungsfaktor 0.85 fortgefahren.

Um die Optimierung greifbarer darzustellen, werden im rechten Teil von Abbildung 6.10 die ersten Schritte der Optimierung ausgehend von einem festgesetzten Startpunkt gezeigt. Das Vorgehen entspricht in Abbildung 6.8 der Anwendung – wird hier jedoch beispielhaft aufgrund von Rechenkosten vorerst erneut an dem Metamodell durchgeführt. Die Anwendung wird für den im Training betrachteten H3 aber auch für die HF für den Frontalaufprall mit 56 km/h und Beifahrersitzpositionen (vgl. Abbildung 6.7 auf S. 126) betrachtet. Der HF-ATD ist folglich ungesehen und ermöglicht einen ersten Eindruck in Bezug auf die Generalisierung der Strategie. Dieser weist aufgrund von weniger Gewicht und anderer

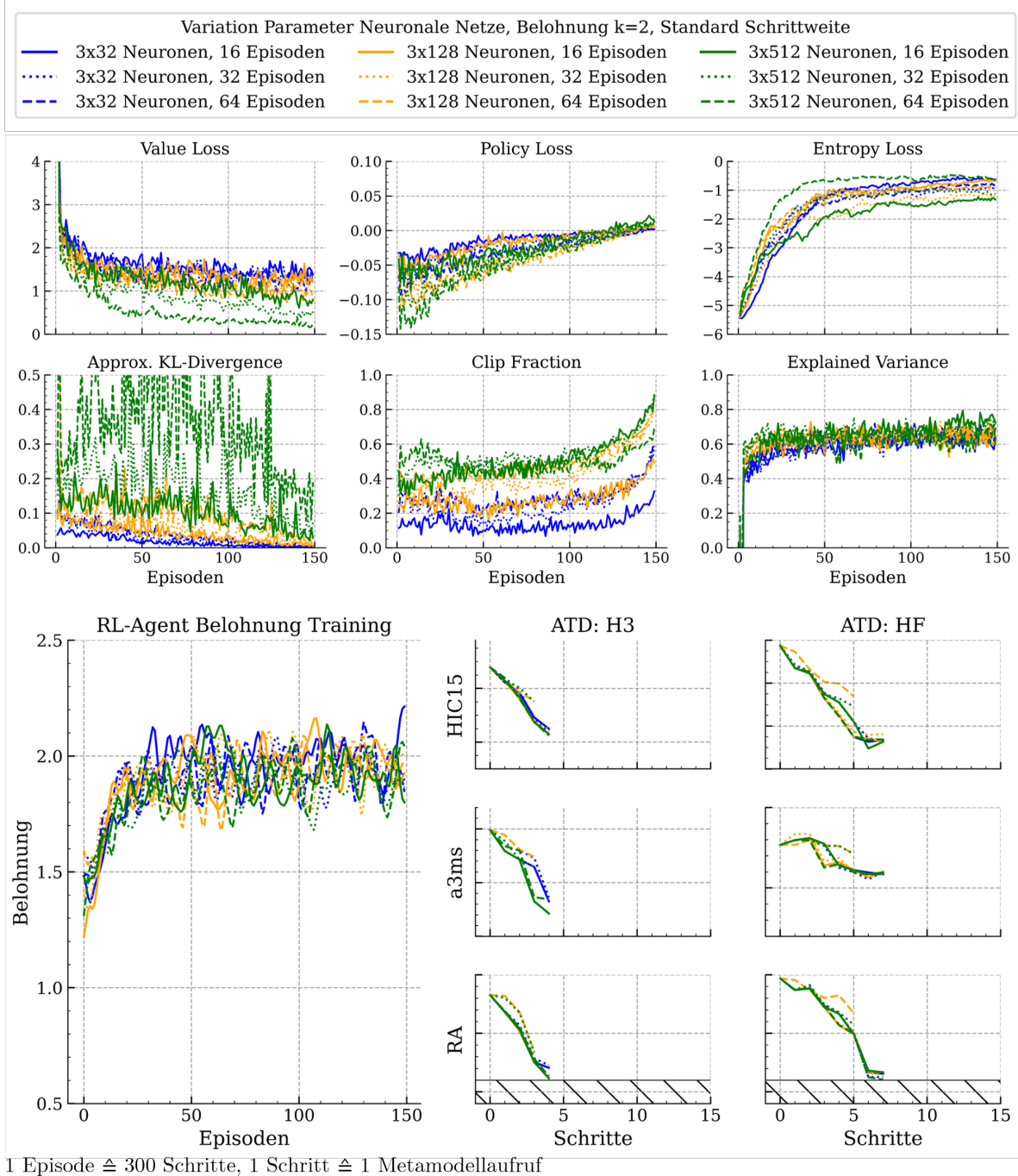
Sitzposition eine stark abweichende Systemantwort auf. Die Systemantwort für den HF wird ermittelt, indem entsprechend Abschnitt A.1.1 auf S. 257 ein anderer ATD-Typ in das Prognosemodell eingegeben wird. Die y-Achsenbeschriftung in Abbildung 6.10 wurde aufgrund der Vertraulichkeit der verwendeten Daten entfernt. Die wichtigsten Informationen, wie stark die Belastungen durch die Verwendung der Strategie reduziert wurden und welchen Optimierungspfad der Agent verfolgt hat, sind jedoch weiterhin klar erkennbar. Ausgehend von der initialen Rückhaltesystemkonfiguration werden HIC15 und a3ms um 31.1% bzw. 20.0% für den H3 und um 40.2% bzw. 8.9% für den HF reduziert. Der RL-Agent wird in der Anwendung terminiert, sobald sich die Brustbeschleunigung a3ms zwei Schritte in Folge verschlechtert oder der Restabstand RA unter den festgelegten Grenzwert  $RA_{\min}$  fällt. Im Training ist das nicht der Fall, dadurch, dass der RL-Agent verschiedene Lösungsstrategien explorieren soll. In Abbildung 6.10 ist ersichtlich, dass die Anwendung weitgehend robust ist. Lediglich für HF gibt es drei Fälle, in denen der HIC15 erhöht wird. Alle übrigen Strategien erlaubten eine Belastungsreduktion. Das spricht für die Robustheit des Verfahrens, auch wenn Sprünge in der KL-Divergence oder fehlende Konvergenz des Entropy Losses (vgl. Abbildung 6.9) festgestellt werden können.

### 6.2.3.3 Robustheit des Reinforcement-Learning-Agenten

Nach Festlegung der wichtigsten Verfahrensparameter für den PPO-Agent im Kontext des Trainingsprozesses, wird die Robustheit nun weitergehend untersucht. Betrachtet werden dabei die Größe der Policy- und Value-Netzwerke (vgl. Abbildung 6.11), die Linearisierung der Belohnung (vgl. Abbildung 6.12) und die Verringerung der Schrittweite (vgl. Abbildung 6.13). Die übrigen Verfahrensparameter verbleiben auf den festgelegten Werten für Lernrate  $1e-4$ , Entropiekoeffizient 0.001 und Diskontierungsfaktor 0.85.

Abbildung 6.11 zeigt Trainingsdurchläufe des PPO-RL-Agenten mit Variationen in der Größe der neuronalen Netzwerke. Dabei schätzt das Policy-Netzwerk die Wahrscheinlichkeitsverteilung über mögliche Aktionen im aktuellen Zustand und das Value-Netzwerk den erwarteten kumulierten Belohnungswert dieses Zustands. In der Betrachtung werden verschiedene Anzahl Neuronen [32, 128, 512] auf drei verdeckten Lagen untersucht. Zusätzlich wird die Anzahl Epochen [16, 32, 64] untersucht, wie oft pro Update-Zyklus die Policy- und Value-Netzwerke mit den gesammelten Übergängen (Observation, Aktionen, Belohnungen) aus mehreren Schritten (dem sog. Rollout-Datenbatch) angepasst werden.

Bei Betrachtung der Auswertegrößen in Abbildung 6.11 zeigt lediglich die KL-Divergence Unstetigkeiten für zu große Netzwerke. Zusätzlich steigt das Clipping ab etwa 120 Episoden deutlich an. Bei 64 Epochen zum Training von Policy- und Value-Netzwerk wird dies zusätzlich begünstigt.



**Abbildung 6.11:** *DrAGOS* Robustheit: Trainingsprozess – Netzwerkgröße

Ein starkes Clipping  $> 0.5$  deutet darauf hin, dass der RL-Agent versucht häufig starke Änderungen an der Strategie vorzunehmen. Auf die mittlere Belohnung der Trainingsepisoden hat dies jedoch einen untergeordneten Einfluss – diese sind entsprechend Abbildung 6.11 vergleichbar. Ebenso die Optimierungsstrategie für H3 und HF ausgehend von einem festen Startpunkt.

Ebenfalls wird eine Abänderung der Belohnungsfunktion (vgl. Gleichung (6.3) auf S. 129) untersucht. Um die Sensitivität der Belohnungsfunktion  $R(o_t)$  zu erhöhen wurde  $k = 2$  gesetzt. Die Ergebnisse bei einer Abänderung auf  $k = 1$  ist Abbildung 6.12 auf S. 136 zu entnehmen. Die Belohnung ist durch den Wegfall der quadratischen Zusammenhänge um im Schnitt 0.5 Punkte geringer. Ein Einfluss – auch in Bezug auf abgeänderte Netzwerkparameter – ist nicht zu erkennen. Trotz abgeänderter Belohnungsfunktion ist auch hier ersichtlich, dass Unstetigkeiten in der KL-Divergence sowie die Clipping Fraction ab 120 Episoden für große Netzwerke bzw. mehr Trainingsepochen der Netzwerke ansteigen. Auch die Optimierungsstrategie für H3 und HF ausgehend von einem festen Startpunkt ist für die Belohnungsfunktion mit  $k = 1$  unabhängig der Netzwerkparameter und vergleichbar mit den Ergebnissen für  $k = 2$  in Abbildung 6.11.

Abschließend soll noch die Verringerung der Schrittweite in Kombination mit dem linearen Belohnung ( $k = 1$  in Gleichung (6.3) auf S. 129) untersucht werden. Die initial definierte Schrittweite ist Gleichung (6.2) auf S. 128 zu entnehmen. Die Ergebnisse sind in Abbildung 6.13 auf S. 137 gezeigt, wobei auch hier eine Variation der Parameter für Policy- und Value-Netzwerke vorgenommen wird. Die absoluten Schrittweiten werden auf  $[0.15, 0.15, 1.0, 1.0, 1.0]$  abgesenkt. Die bisherigen Ergebnisse der Robustheitsstudie werden auch an dieser Stelle bestätigt. Trotz abgeänderter Belohnungsfunktion ist auch hier ersichtlich, dass Unstetigkeiten in der KL-Divergence sowie die Clipping Fraction ab 120 Episoden für große Netzwerke bzw. mehr Trainingsepochen der Netzwerke ansteigen. Der Anstieg der mittleren Belohnung in den Trainingsepisoden fällt schwächer aus als für die beiden vorangegangenen Fälle, trotzdem ist eine Konvergenz zu erkennen. An der beispielhaften Anwendung für H3 und HF ist die Absenkung der Lasten deutlich kleiner. Die Grenze  $RA_{\min}$  wird nicht erreicht, da das Konvergenzkriterium für a3ms vorher greift. Dieses müsste also entsprechend angepasst werden, um auch das System in der Anwendung effektiv zu optimieren. Da der RL-Agent jedoch auch in diesem Szenario stabil trainiert, wäre eine Verwendung von kleineren Schritten entsprechend der Situation im Entwicklungsprozess denkbar. Untersuchungen zur Erhöhung der Schrittweite des RL-Agenten sind nur bedingt interessant, da die Aktionen zu grob werden um feinfühligere Abänderungen vorzunehmen.

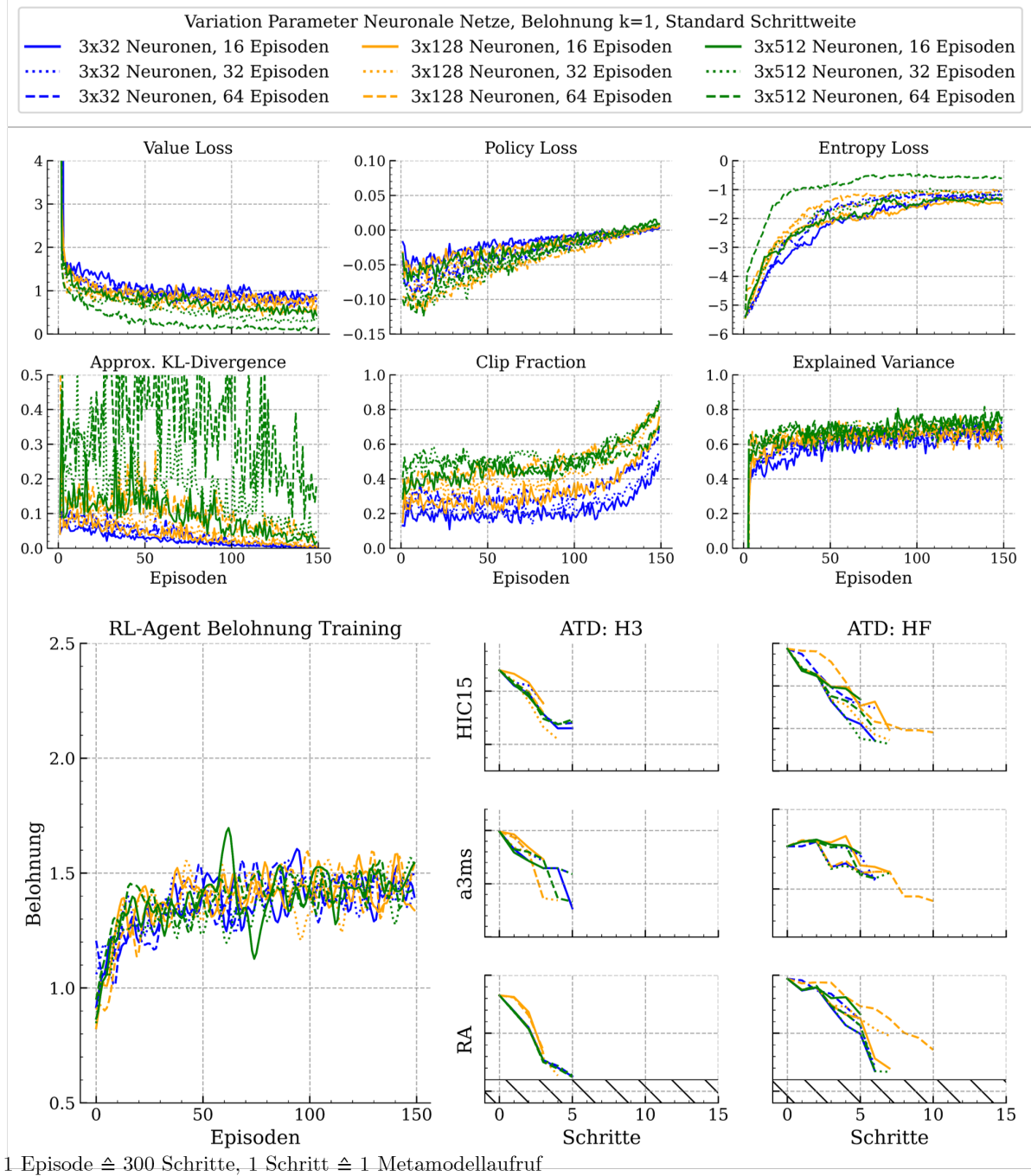
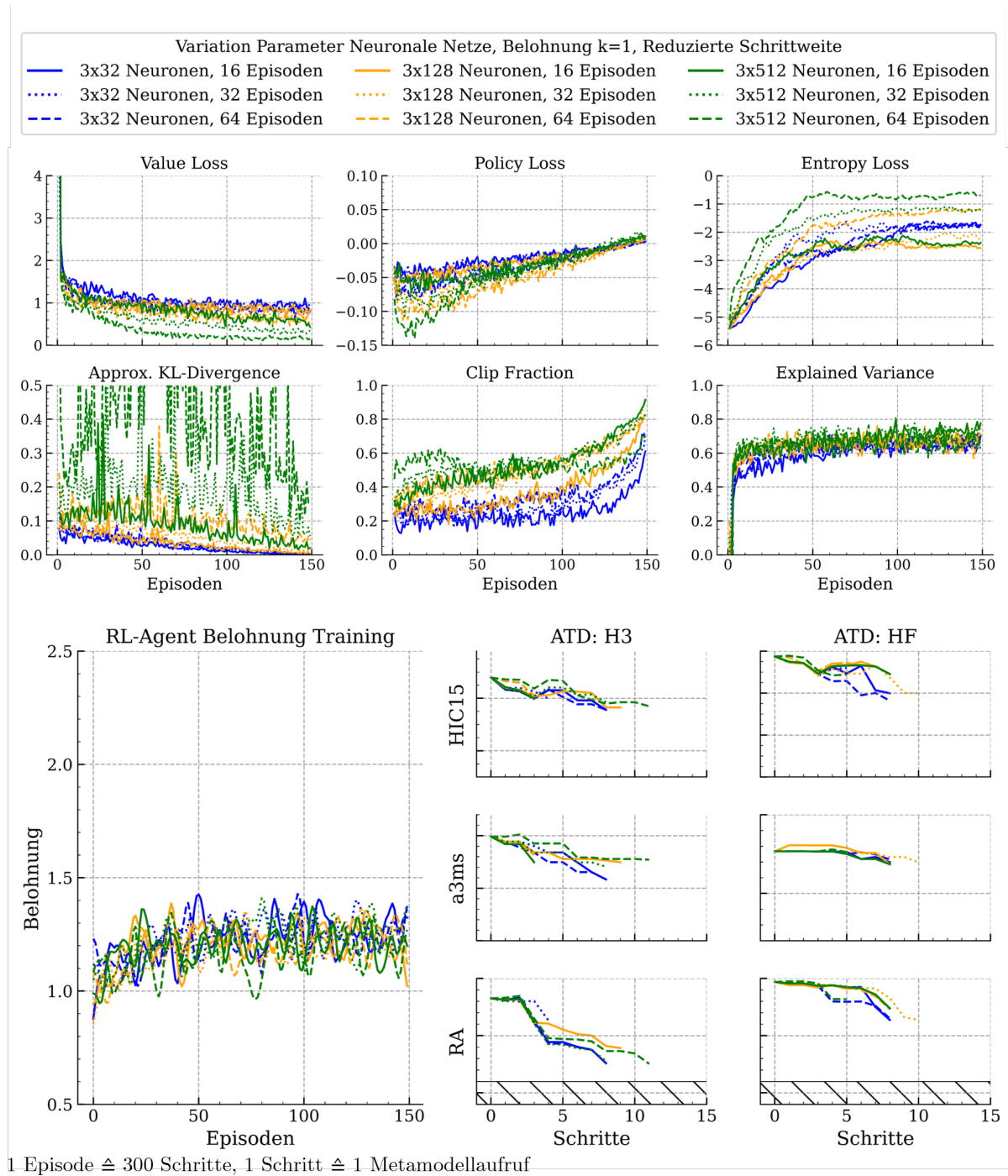


Abbildung 6.12: *DrAGOS* Robustheit: Trainingsprozess – Belohnung

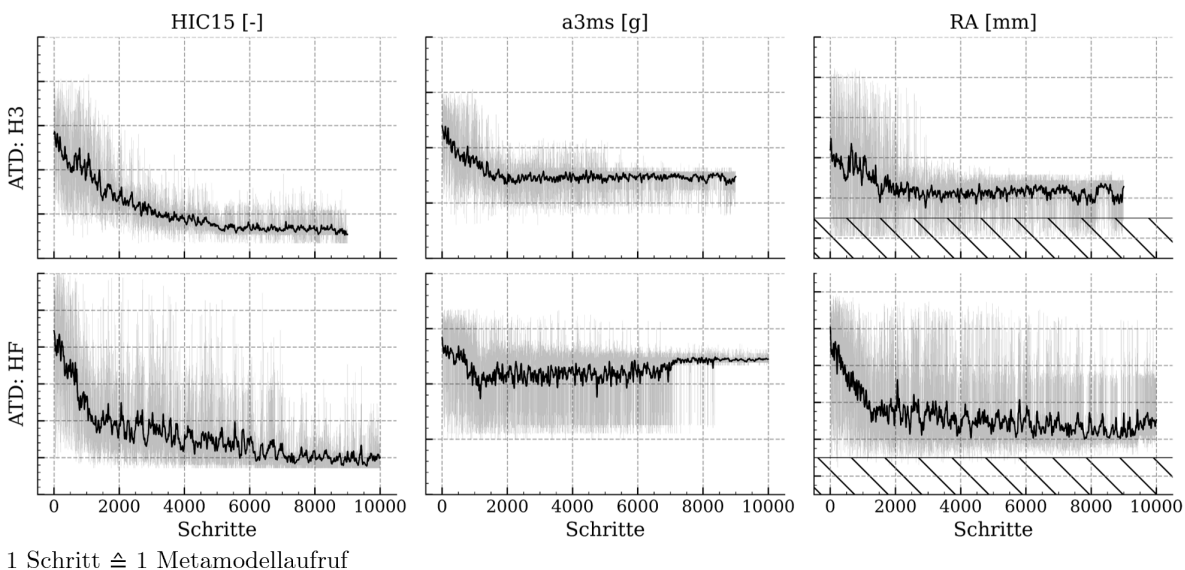


**Abbildung 6.13:** *DrAGOS* Robustheit: Trainingsprozess – Schrittweite

### 6.2.4 Benchmark mit globalem Optimierungsalgorithmus

Um die mittels dem RL-Agenten erzielten Ergebnisse einzuordnen, wird nun analog der bisherigen Anwendung auf H3 und HF basierend auf dem XGBoost-Metamodell eine Optimierung mit einem Differential Evolution Algorithmus [SP97] durchgeführt. Das grundlegende Vorgehen entspricht der metamodellbasierten Optimierung, wie bereits in Abbildung 2.6 auf S. 40 eingeführt und von [Xue+12; Büt+23] zur Optimierung crashbelasteter Systeme betrachtet wurde. Identisch bleibt das Optimierungsproblem nach Gleichung (6.1) auf S. 127. Um die bestmögliche Vergleichbarkeit herzustellen, wird als Optimierungsgegenstand die Belohnungsfunktion nach Gleichung (6.3) auf S. 129 sowie identische Schrittweiten des Aktionsraums nach Gleichung (6.2) auf S. 128 verwendet.

Die Ergebnisse der Belastungsminimierung für die drei betrachteten Ausgaben HIC15, a3ms und RA sind in Abbildung 6.14 für beide ATD-Typen (oben H3, unten HF) gezeigt. Um den Verlauf der drei Ausgaben besser erkennen zu können wurde zusätzlich ein Gauss-Filter ( $\sigma = 9.6$ ) angewandt. Beide Optima erfüllen die vorgeschriebene Randbedingung, wobei eine sukzessive Minimierung der Belastung ersichtlich ist. Interessant ist vor allem der Vergleich mit den Lösungen nach Abbildung 6.10 auf S. 132, die mit Hilfe des RL-Agenten (Lernrate:  $1e-4$ , Entropiekoeffizient: 0.001, Diskontierungsfaktor: 0.85) ermittelt wurden.



**Abbildung 6.14:** *DrAGOS* Benchmark: Metamodellbasierte Optimierung mittels Differential Evolution Algorithmus für H3 (oben) und HF (unten)

In Bezug auf den H3 ist das mittels Differential Evolution ermittelte Optimum technisch besser, da für die Lösung eine um 27% höhere Belohnung vorliegt. Während ein um 18% geringerer HIC15 ermittelt werden kann, ist jedoch die a3ms um 1.4% höher. In Bezug auf den HF ist das mittels Differential Evolution ermittelte Optimum hingegen schlechter,

da für die Lösung eine um 7.3 % geringere Belohnung vorliegt. Der HIC15 und a3ms sind um 4.2 % bzw. 2.9 % höher. Alle Lösungen erfüllen die Randbedingung.

Im Vergleich auffällig ist die hohe Anzahl Schritte von bis zu 10,000, die der Differential Evolution Algorithmus benötigt. Ähnliche Größenordnungen werden auch in der Arbeit von [Xue+12] ermittelt. Eine direkte Interaktion des Optimierers mit dem FE-Modell ist hier nicht denkbar.

### 6.2.5 Anwendung *DrAGOS* auf Crashsimulationen

Im nächsten Schritt wird nun die Anwendung der trainierten Strategie  $\pi^*$  für die beiden Use Cases (1) Minimierung der Insassenbelastung des HF auf der Fahrersitzposition in der  $i$ -ten Produktgeneration, und (2) Minimierung der Insassenbelastung des H3 auf der Beifahrersitzposition in der  $(i+1)$ -ten Produktgeneration untersucht. Für die beiden Use Cases wird jetzt der Modus gemäß Abbildung 6.8 auf Anwendung gesetzt. Der RL-Agent interagiert nun direkt mit einer LS-DYNA-Simulationsumgebung, die auf einem Hochleistungsrechnercluster läuft.

Im Kontext der Produktgenerationsentwicklung dient als Startpunkt der Optimierung die Standardkonfiguration des zum Training verwendeten Referenzmodells. Nachvollziehbar definierte Startpunkte können zudem das Vertrauen beteiligter Ingenieure in die Methoden erhöhen. Im Training konnte jedoch gezeigt werden, dass der RL-Agent auch von beliebigen Startpunkten aus optimieren kann.

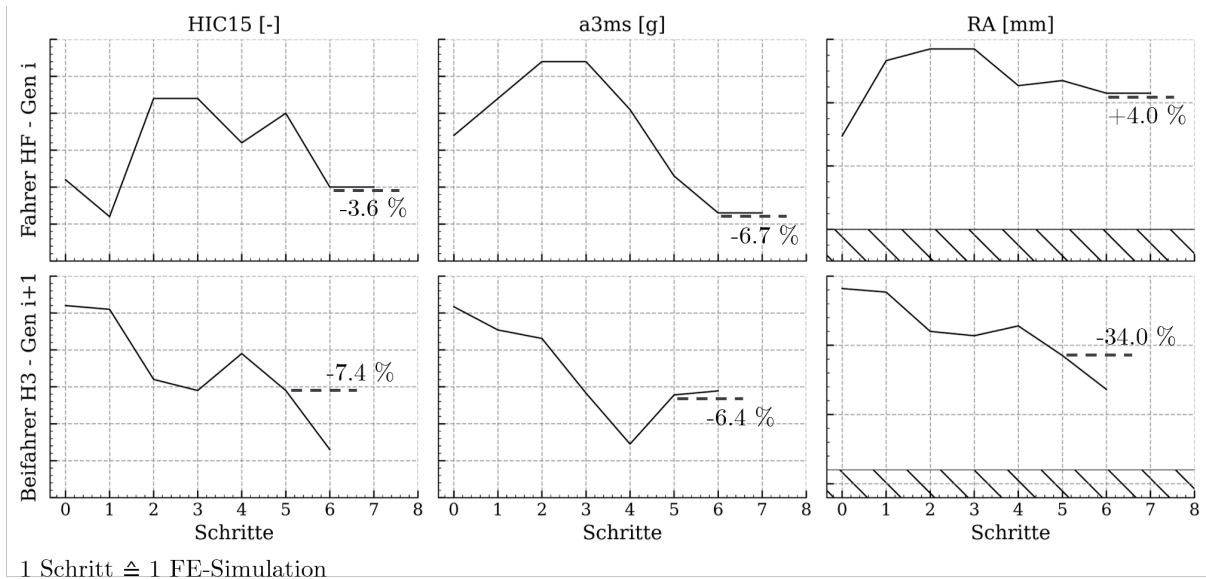
#### 6.2.5.1 Use Case I: Aufgabenübergreifender Einsatz

Die Ergebnisse für den aufgabenübergreifenden Einsatz der vortrainierten Strategie sind im oberen Teil von Abbildung 6.15 auf S. 140 dargestellt. Normalerweise kann zur Minimierung der Belastungen auf der Fahrersitzposition auch die Retraktionskraft der Lenksäule angepasst werden [JGK21]. Da dieser Parameter vom Agenten bisher nicht berücksichtigt wurde, muss dieser unverändert bleiben. In den ersten Schritten vergrößern die Änderungen die Belastungen auf den ATD signifikant, bis zu 8.1 % für das HIC15. Dies ist wahrscheinlich auf die empfindlichere Systemreaktion des HF zurückzuführen, die auf etwa 40 % weniger Gewicht und eine andere Sitzposition aufgrund der Größe zurückzuführen ist [Mer+89; FMVSS208]. Ab dem dritten Schritt werden HIC15 und a3ms jedoch sukzessive reduziert. Im Vergleich zum Ausgangswert wird eine Reduktion von 3.6 % bzw. 6.7 % für HIC15 und a3ms erreicht. Der RA in diesem Szenario wird als minimaler Abstand zwischen dem Kopf des Fahrers und dem Lenkrad betrachtet. Dieser wird um 4 % erhöht. Für den RA darf der festgelegte Grenzwert nicht unterschritten werden. Obwohl das Szenario deutlich vom Trainingsszenario des RL-Agenten abweicht, ist dieser dennoch in der Lage, die Belastungen basierend auf dem gegebenen Startpunkt zu reduzieren. Nach 10 Schritten schlägt der RL-Agent keine Änderungen mehr vor, weshalb der Optimierungsvorgang

terminiert wird. Das spezifizierte Rückhaltesystem ist aus ingenieurtechnischer Sicht auch für den betrachteten Use Case sinnvoll.

### 6.2.5.2 Use Case II: Produktgenerationsübergreifender Einsatz

Die Ergebnisse für den produktgenerationsübergreifenden Einsatz der vortrainierten Strategie sind im unteren Teil von Abbildung 6.15 dargestellt. Unterschiede zwischen den Produktgenerationen sind in der Sitzposition, der Instrumententafel sowie im Crashpuls gegeben. Obwohl im ersten Use Case der Restabstand zuvor zugenommen hatte, wurde er im zweiten Use Case um 34 % reduziert, verbleibt aber auch hier überhalb des Grenzwertes. Die Werte von HIC15 und a3ms, werden um jeweils 7.4 % bzw. 6.4 % reduziert. Das Optimierungsverfahren wird nach 8 Schritten aufgrund der Konvergenz von a3ms gestoppt. Eine Minimierung der Belastungen ist also unter Einhaltung der Randbedingung auch für diesen Use Case möglich.



**Abbildung 6.15:** *DrAGOS* Anwendung vortrainierter Strategie auf FE-Simulationen: Aufgabenübergreifender Einsatz (oben) und Produktgenerationsübergreifender Einsatz (unten)

## 6.3 *DAYSi* – Analyse von Crashsensordaten

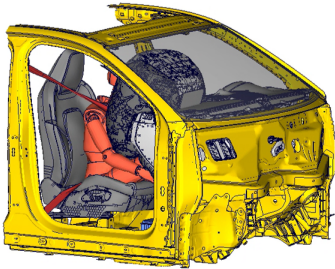
Die Evaluierung von *DAYSi* erfolgt anhand von drei repräsentativen Use Cases aus der Entwicklung passiver Sicherheitssysteme. Ebenfalls werden die bei der Einführung der Methode in Abschnitt 5.3 ab S. 99 gelisteten Verfahrensparameter abgeprüft um die Robustheit zu bestimmen.

Für die Betrachtung der Use Cases selbst, werden die Verfahrensparameter auf den bereits definierten Standardwerten belassen.

### 6.3.1 Use Case I: Bewertung einer Entwicklungsmaßnahme

Ein Überblick über den ersten Use Case ist in Abbildung 6.16 gegeben. Betrachtet wird der Vergleich von neun Schlittensimulationen, die den FN-56 abbilden. Die FE-Modelle entsprechen im Aufbau denen, die bereits schon für die Evaluation von *DrAGOS* verwendet wurden. Alle Simulationen wurden im Kontext der Betrachtung verschiedener Rückhaltesystemkonfigurationen, insbesondere von Gurt und Beifahrerairbag, durchgeführt. Entsprechend werden 20 Sensorsignale, die das Verhalten von ATD, dem Rückhaltesystem und dem Fahrzeug beschreiben, verwendet. Energien, weitere Schnitt- und Kontaktkräfte sowie numerische Größen, wie der Zeitschritt, werden daher nicht einbezogen. Die Ingenieuraufgabenstellung entspricht der Plausibilisierung des Systemverhaltens der neusten Simulation im Kontext vorangegangener Simulationen.

**Use Case I – Bewertung einer Entwicklungsmaßnahme**



Simulationsmodell  $t > 0$  ms

- **9 Schlittenmodelle**
  - Frontalaufprall  $0^\circ$  56 km/h auf starre Wand
  - Beifahrer, 50% Mann Hybrid III (H3)
- Simulationen aus einer Entwicklungsphase, Unterschiede im Rückhaltesystem
- **Auswertung von 20 Kanälen:** Hauptbewegung ATD (Kopf, Brust, Becken), Kräfte Gurt und Druck/Volumen Airbag, Fahrzeugpuls

»»

Analyse der Entwicklungsmaßnahme „Änderung Schaltzeit Gurtkraftniveau“.

**Abbildung 6.16:** *DAYSi* Use Case I: Problemstellung

Auf Basis der Information, dass die Schaltzeit der Kraftniveaus des Sicherheitsgurts abgeändert wurde, soll nun sichergestellt werden, dass dies erfolgreich stattgefunden hat, und keine Auffälligkeiten an unerwünschten Stellen auftreten. Die Ergebnisse bei der Anwendung von *DAYSi* sind in Abbildung 6.17 und Abbildung 6.18 gezeigt. Die Ergebnisdarstellungen entsprechen denen, die auch im Schaubild von *DAYSi* in Abbildung 5.6 auf S. 100 bereits

allgemein eingeführt wurden. Eine globale Übersicht der Abweichungskoeffizienten auf der Datensatzebene ist im oberen Teil von Abbildung 6.17 dargestellt, wobei der linke Plot die über Sensoren und Zeit kumulierten Abweichungskoeffizienten  $\sum_{i,k} D_{ijk}$  und der rechte Plot die Sensoren kumulierten Abweichungskoeffizienten  $\sum_i D_{ijk}$  für jedes Zeitfenster zeigt. Da diese Werte nicht physikalisch interpretierbar sind, werden diese jeweils in den Wertebereich  $[0, 1]$  skaliert, wobei 0 einer geringen und 1 einer hohen Abweichung entspricht. Die Simulation *RUN\_1043* entspricht der Zielsimulation, die näher betrachtet werden soll, und ist daher hervorgehoben. Die vergebene Identifikationsnummer entspricht nicht der zeitlichen Reihenfolge der Durchführung sondern verweisen auf hier nicht näher relevante Konfigurationspools. Im Vergleich zu den anderen Simulationen im Datensatz tritt für *RUN\_1043* die zweithöchste Abweichung auf. In Bezug auf die Zeit sind diese hauptsächlich zwischen 65 ms und 100 ms aufzufinden. In *RUN\_1062* hingegen liegen diese zwischen 15 ms und 30 ms vor. Diese Information hilft dabei neben einzelnen Simulationen auch interessante Zeitbereiche aufzufinden.

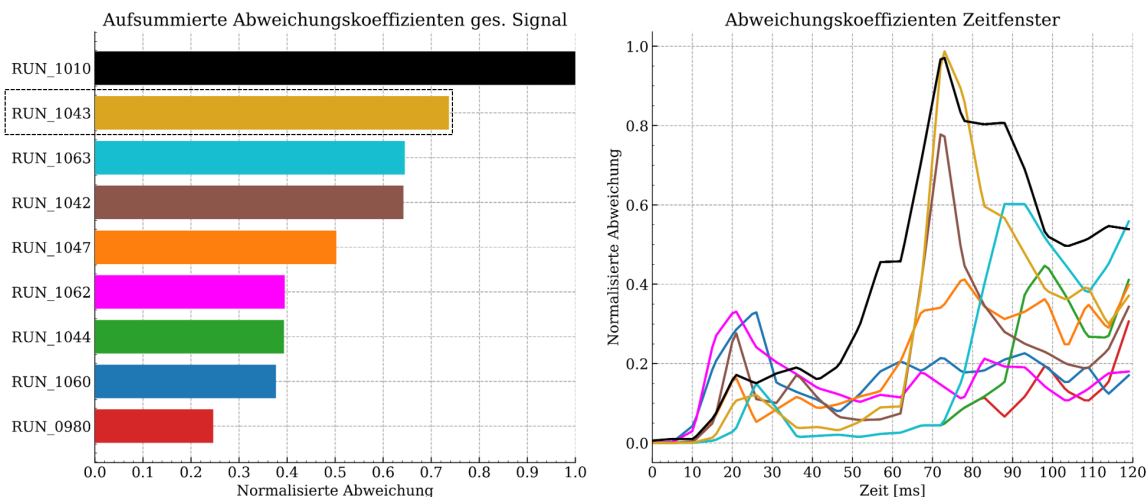


Abbildung 6.17: *DAYSi* Use Case I: Überblick Datensatz

Die in Abbildung 6.18 dargestellten Plots zeigen Informationen spezifisch für die Zielsimulation *RUN\_1043*. Der erste zeigt die Abweichungskoeffizienten in Bezug auf den Signalverlauf und besteht aus einem horizontalen Boxplot, der die Abweichungskoeffizienten für die vollständigen Signale darstellt. Rechts daneben eine Heatmap abgebildet, die die Abweichungskoeffizienten in den Zeitfenstern zeigt. Beide Teile des Plots teilen sich die y-Achse, auf der die Sensoren gelistet sind. Der Boxplot links repräsentiert die Abweichungswerte für die vollständigen Signale aller Simulationen am jeweiligen Sensor. Die analysierte Simulation, hier *RUN\_1043*, wird mit einem „X“ hervorgehoben. Dies liefert Informationen darüber, wie sich die Signale der einzelnen Sensoren voneinander unterscheiden und wo dort der analysierte Datenpunkt einzuordnen ist. Während *RUN\_1043* einen

unauffälligen Knieairbagdruck aufweist, der mittelstark streut, ist bei ähnlicher Streubreite eine auffällige Beckenbeschleunigung in Z-Richtung zu verzeichnen. Eine große Streubreite ist in der Brustbeschleunigung in X-Richtung zu sehen. Dort entspricht RUN\_1043 dem auffälligsten Datenpunkt.

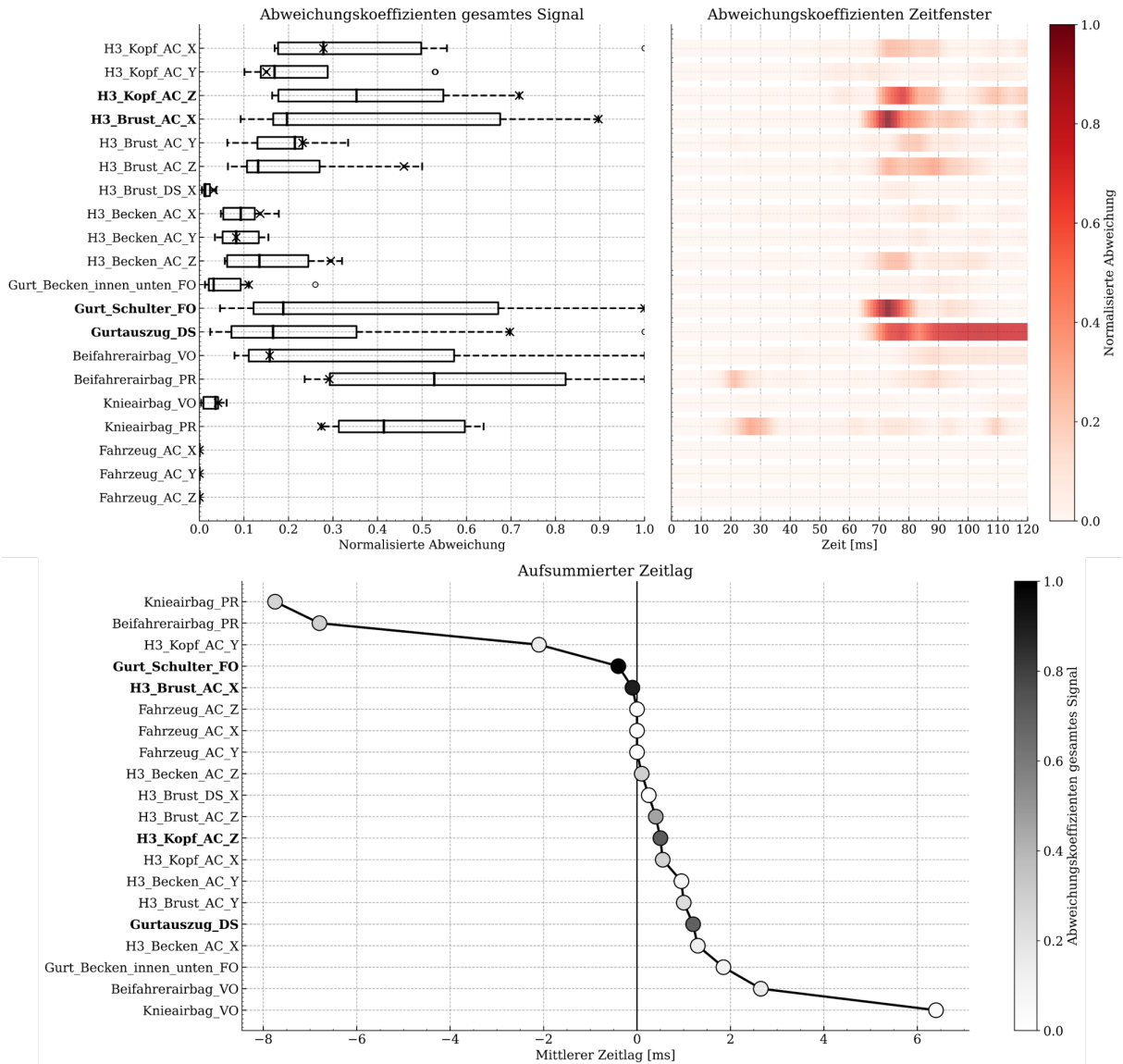


Abbildung 6.18: DAYSi Use Case I: Lokale Ausreißer und Eventkette RUN\_1043

Die daneben verortete Heatmap liefert weitere Informationen zu den Signalen der Ziel-simulation und visualisiert auf Sensorebene zu welchen Zeitpunkten die Abweichungen auftreten. Die Farbgebung orientiert sich, entsprechend der Legende, an der Amplitude des Abweichungswertes. In den ATD-Sensoren zeigt sich in RUN\_1043 die Becken- und Kopfbeschleunigung in X-Richtung vergleichsweise geringe Abweichungen. Die übrigen ATD-Signale weisen jedoch hohe Abweichungen in dem bereits als interessant identifizierten

Zeitintervall von 65–100 ms auf. Die längsten Abweichungen treten in der Kopfbeschleunigung in Z-Richtung auf, die frühesten in der Brustbeschleunigung in X-Richtung. In den Fahrzeugsignalen treten marginale Abweichungen im Volumen und Druck des Beifahrer- und Knieairbags zwischen 20–30 ms auf. Hohe Abweichungen treten kurz nach 65 ms in der Gurtkraft an der Schulter auf. Das ist früher als die nachfolgenden Änderungen in der ATD-Belastung und die Abweichungen in den übrigen Gurtschnittkräften. Die Gurtkraft am Becken zeigt hingegen wenig Varianz und auch in RUN\_1043 nur marginale Abweichungen. Dieses Verhalten ist plausibel, da die Gurtkraft an der Schulter aufgrund der Nähe zum Gurtretractor Änderungen in Kraftniveau und Schaltzeit deutlicher widerspiegelt. Das einzige Fahrzeugsignal, das nach 100 ms noch deutliche Abweichungen aufweist, ist der Gurtauszug. Im Fahrzeugpuls sind keine Abweichungen zu finden, da dieser überall identisch ist.

Mittels der Kreuzkorrelationsanalyse wird nun auch eine zeitliche Reihenfolge der Auffälligkeiten ermittelt. Frühe und große Abweichungen sind besonders interessant, da diese Auswirkungen auf das spätere Geschehen des dynamischen Systems haben. Entsprechend sind die Kreise, deren Position in X sich aus dem aufsummierten Zeitlag ergibt, anhand der Abweichungskoeffizienten des gesamten Signals der betrachteten Datenpunkte eingefärbt. Mögliche Hauptursachen und Sekundäreffekte können Abbildung 6.18 direkt entnommen werden, da die Abweichungen im Signal mit der höchsten negativen Zeitverzögerung im Durchschnitt den Abweichungen in den anderen Signalen vorauslaufen. Für die betrachtete Simulation RUN\_1043 kann die Gurtkraft an der Schulter als Hauptursache für Abweichungen sowohl in den ATD- als auch in den Fahrzeugsignalen identifiziert werden. Als relevante Sekundäreffekte werden eine auffällige Kopf- und Brustbeschleunigung in Z-Richtung sowie die Änderung im Gurtauszug hervorgehoben. Diese Ergebnisse sind – auch unter Einbezug des Ingenieurverständnisses basierend auf den eingebrachten Änderungen – plausibel.

### 6.3.2 Use Case II: Erkennung von Ausreißern

Ein Überblick über den zweiten Use Case ist in Abbildung 6.19 gegeben. Betrachtet wird der Vergleich von 14 Schlittensimulationen, die ebenfalls den FN-56 abbilden. Alle Crashsimulationen wurden im Kontext der Betrachtung verschiedener Rückhaltesystemkonfigurationen, insbesondere von Gurt und Beifahrerairbag, durchgeführt. In diesem Fall enthält der Datensatz jedoch Simulationen aus mehreren Entwicklungsphasen. Die Auswahl der Sensoren ist analog zum ersten Use Case. Die Ingenieursaufgabenstellung entspricht hier dem Vergleich von Simulationen, ausgehend von der Annahme, dass der Ingenieur kein vorhergehendes Wissen hat.

Die Ergebnisse bei der Anwendung von *DAYSi*, wie bereits im Schaubild der Methode selbst in Abbildung 5.6 auf S. 100 abgebildet, sind in Abbildung 6.20 bis Abbildung 6.22 gezeigt.

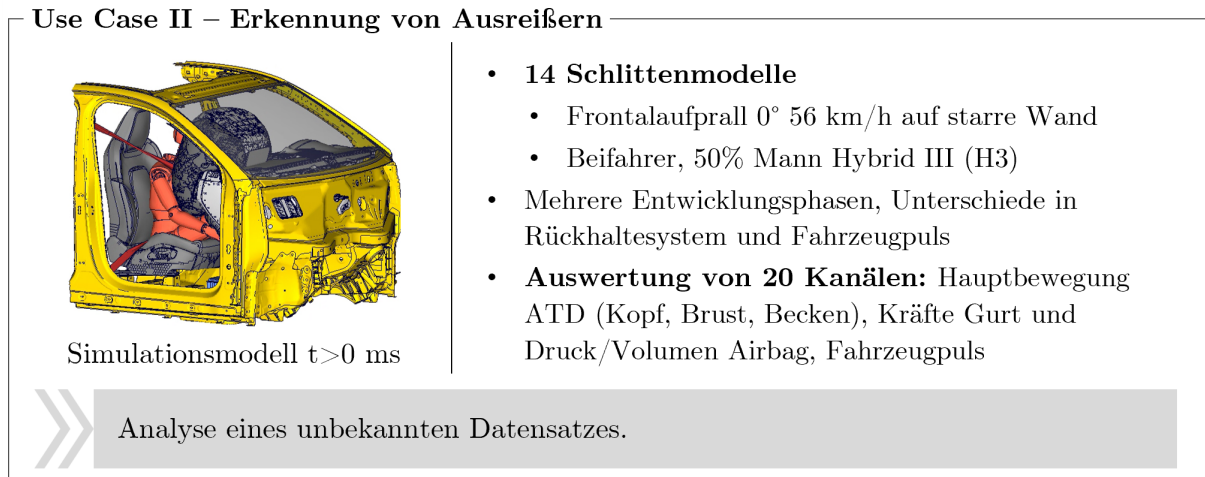


Abbildung 6.19: *DAYSi* Use Case II: Problemstellung

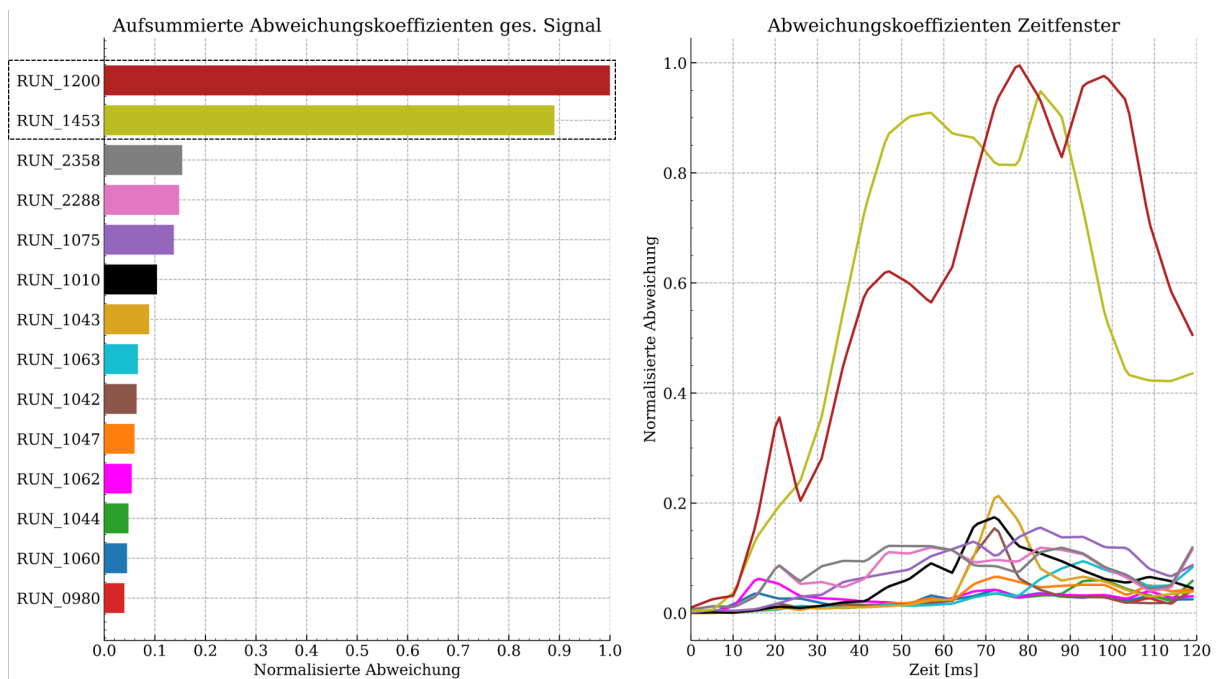


Abbildung 6.20: *DAYSi* Use Case II: Überblick Datensatz

Eine globale Übersicht der Abweichungskoeffizienten auf der Datensatzebene ist in Abbildung 6.20 dargestellt. Eine genauere Beschreibung, wie die Plots zu interpretieren sind, kann dem ersten Use Case entnommen werden. Ausgehend davon, dass interessante Datenpunkte in einem unbekanntes Datensatz ermittelt werden sollen, stehen RUN\_1453

und RUN\_1200 besonders hervor. Im Vergleich zu den übrigen Datenpunkten liegen hier sehr deutliche Abweichungen vor, was – insbesondere im Kontext der vergleichsweise geringen Unterschiede in den globalen Abweichungskoeffizienten im ersten Use Case – auf größere Fehler hinweist.

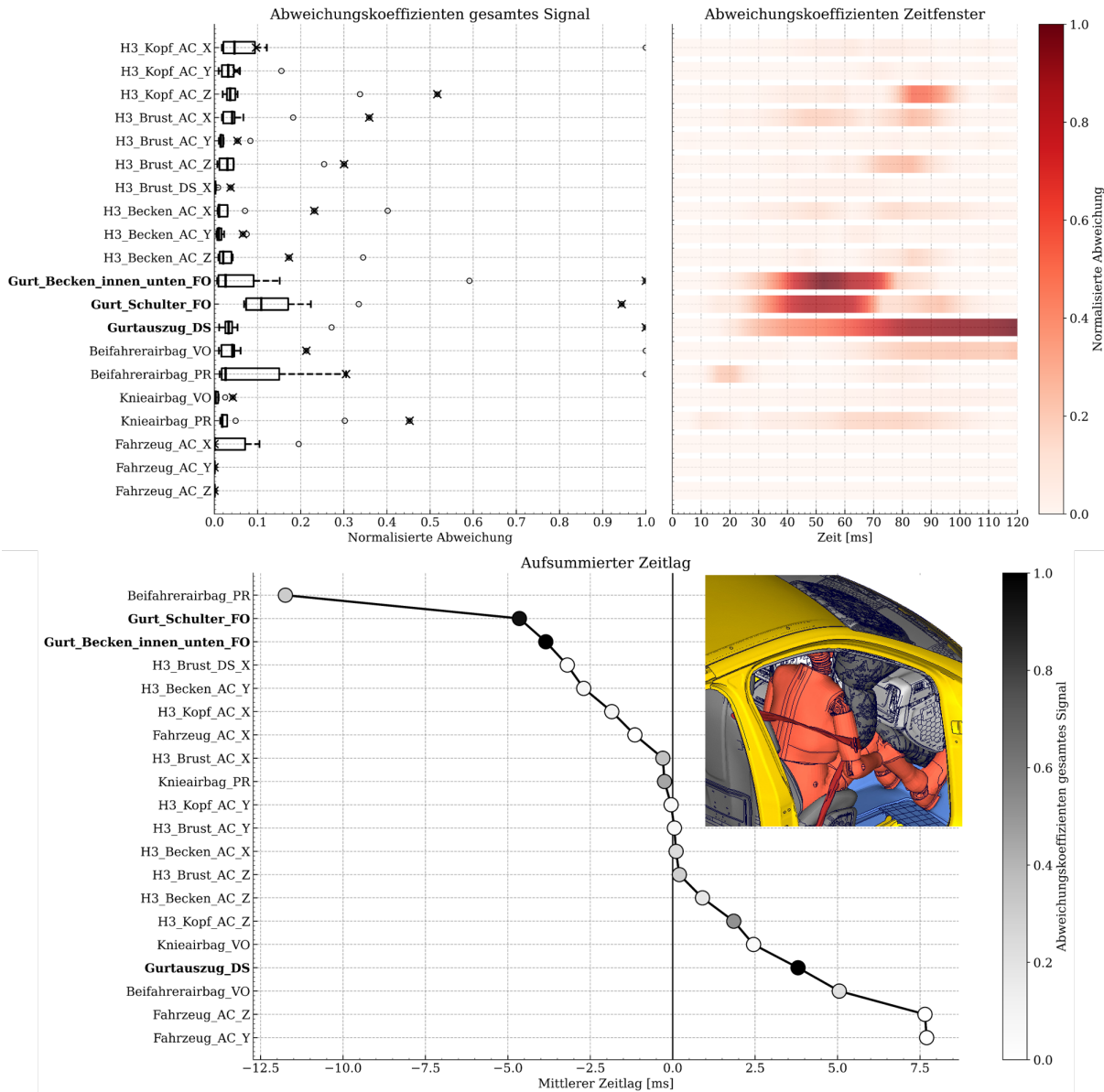


Abbildung 6.21: *DAYSi* Use Case II: Lokale Ausreißer und Eventkette RUN\_1453

Die nach Abbildung 5.6 auf S. 100 abgeleitete datenpunktspezifische Information – für RUN\_1453 und RUN\_1200 – sind in Abbildung 6.21 und Abbildung 6.22 dargestellt. Eine Erklärung der Plots ist auch hier dem ersten Use Case zu entnehmen. In Abbildung 6.21 für RUN\_1453, der der auffälligsten Simulation entspricht, sind insbesondere die Signale des Gurts – Auszug, Schulter und Becken – hervorgehoben. Während an der Messstelle

nahe der Schulter bereits ab 20 ms erste Auffälligkeiten zu verzeichnen sind, treten diese in der Messstelle an Becken und dem Auszug nachgelagert auf.

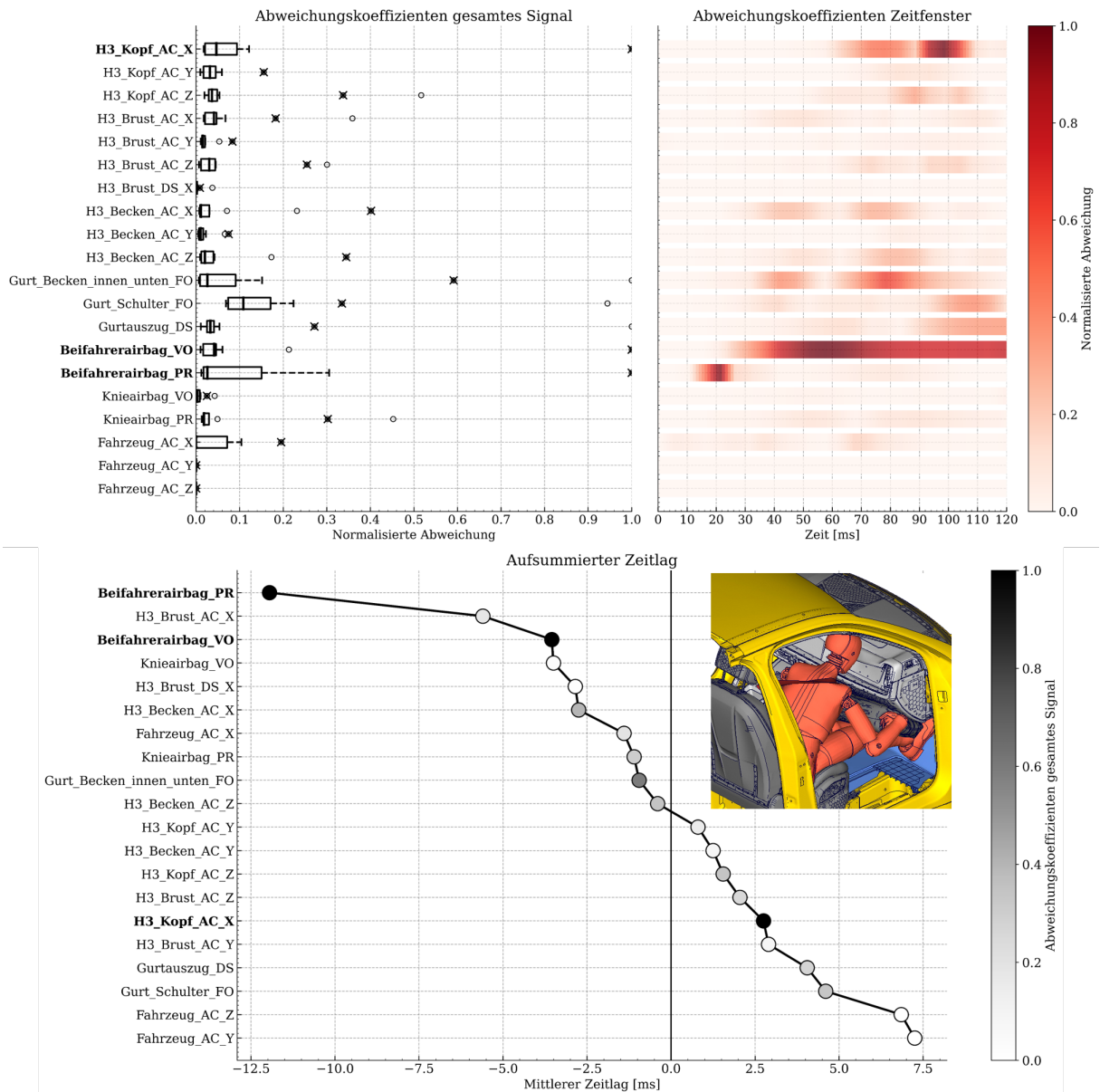


Abbildung 6.22: DAYSi Use Case II: Lokale Ausreißer und Eventkette RUN\_1200

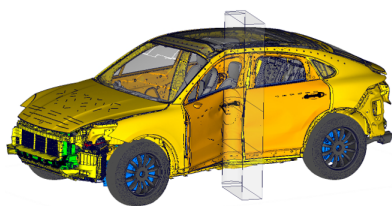
In den Boxplots, die die Abweichung des gesamten Signals beschreiben, liegt RUN\_1453 außerhalb des Bereichs, der durch das eineinhalbfache des Interquartilsabstands oberhalb des dritten Quartils und unterhalb des ersten Quartils definiert ist. Solche Punkte werden auch als „Flier“ bezeichnet. Starke Abweichungen liegen auch in den Drücken der beiden Airbags vor. Am ATD werden Brust- und Kopfbeschleunigung als besonders starke Ausreißer identifiziert. Die zeitlichen Abweichungen sind hier weniger stark ausgeprägt. Die

zeitliche Einsortierung der Signale im unteren Teil von Abbildung 6.21 mittels mittlerer Zeitlags zeigen deutlich, dass gerade im frühen Zeitbereich der Gurt auffällig ist.

Die Abweichung in den Airbags sowie dem Auszug des Gurts werden als sekundäre Effekte hervorgehoben. Bei näherer Betrachtung der Simulation im Bereich des Beckens ist ersichtlich, dass der Kontakt zwischen Gurtschloss und Gurtzunge nicht korrekt definiert ist und der Gurt sich daher öffnet. Dies sorgt dafür, dass der ATD ungegurtet in den Airbag fällt. Als Konsequenz lassen sich, wie bereits durch *DAYSi* extrahiert Auffälligkeiten in den Airbags als Sekundäreffekt feststellen. Ebenfalls vorauseilend aber mit weniger starkem Einfluss ist die Beckenbeschleunigung in Y- sowie der Fahrzeugpuls in X-Richtung.

In RUN\_1200, für den die Ergebnisse in Abbildung 6.22 dargestellt sind, werden Auffälligkeiten insbesondere im Beifahrerairbag identifiziert. Das Verhalten des Knieairbags entspricht hingegen den Erwartungen. Zusätzlich ist auch der Fahrzeugpuls in X-Richtung als auffällig zu betrachten. Hinsichtlich dem ATD liegen in den Beschleunigungen Brust-Y sowie Kopf-X und -Y Auffälligkeiten vor. In Bezug auf die zeitliche Einordnung der Abweichungen im unteren Teil von Abbildung 6.22 sind einige früh abweichende Signale festzustellen. Hierzu zählt grundsätzlich das Becken sowie der angekoppelte Gurt und der Knieairbag Druck. Die stärksten Auffälligkeiten in der frühen Phase der Simulation hingegen sind dem Beifahrerairbag zuzuschreiben. Bei einer genaueren Betrachtung der Simulation lässt sich feststellen, dass die Zündung des Beifahrerairbags fehlgeschlagen ist und dieser nicht entfaltet.

### 6.3.3 Use Case III: Bewertung einer Modellierungsmaßnahme



Simulationsmodell  $t > 0$  ms

- **10 Gesamtfahrzeugmodelle**
  - Seitenaufprall Pfahl 75° 32 km/h
  - Beifahrer, 50% Mann WorldSID
- Eine Entwicklungsphase, Reibung, Fügetechnik, Initialgeschwindigkeiten und Zündzeiten Seitenairbag.
- **Auswertung von 24 Kanälen:** ATD, B-Säule, Tür und Seiten- und Curtainairbag.



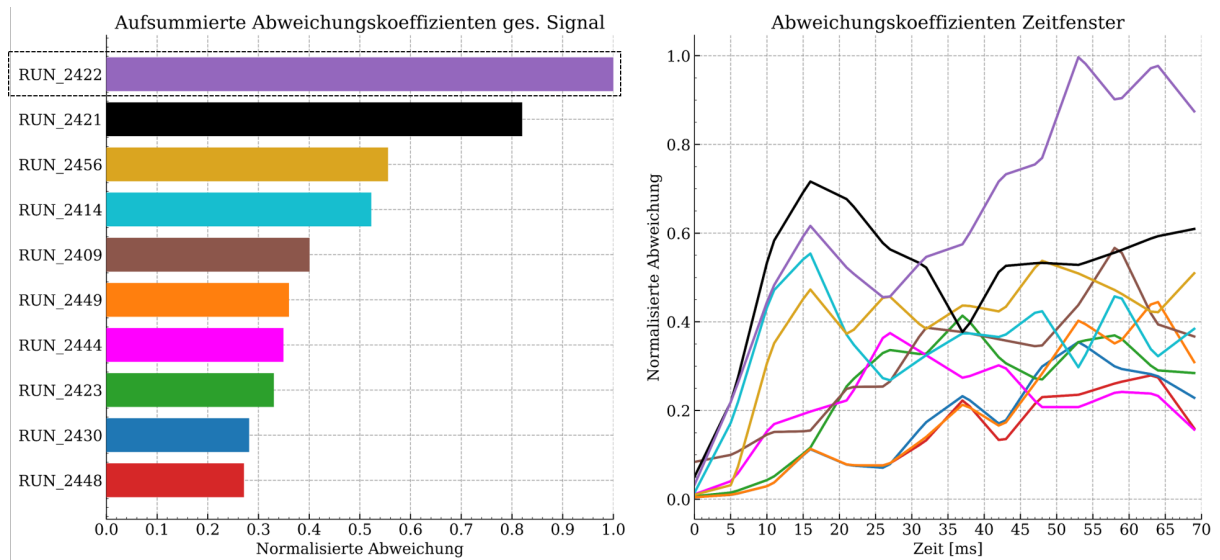
Analyse der Modellierungsmaßnahme „Reduktion Reibung Pfahl“.

**Abbildung 6.23:** *DAYSi* Use Case III: Problemstellung

Ein Überblick über den dritten Use Case ist in Abbildung 6.23 dargestellt. Betrachtet wird der Vergleich von 10 Gesamtfahrzeugsimulationen, die einen Pfahl-Seitencrash 75°

und 32 km/h abbilden. Die Ingenieursaufgabenstellung entspricht – angelehnt an Use Case I – der Analyse einer Zielsimulation im Kontext vorangegangener Simulationen. Entgegen Use Case I wurde hier eine Anpassung in der Modellierung der FE eingebracht. Die Simulationen weisen Unterschiede in Modellierungsparametern wie der Reibung auf, aber auch abgeänderte Lage von Schweißpunkten. Auch hier wird eine Auswahl der wichtigsten Sensoren verwendet, die für den ATD relevantes Fahrzeugverhalten abbilden. Hierzu gehören Tür und B-Säule, Curtain- und Seitenairbag sowie die Belastung des ATDs.

Die Ergebnisse bei der Anwendung von *DAYSi* sind in Abbildung 6.24 und Abbildung 6.25 gezeigt. Eine globale Übersicht der Abweichungskoeffizienten auf der Datensatzebene ist in Abbildung 6.24 dargestellt. Eine genauere Beschreibung, wie die Plots zu interpretieren sind, kann dem ersten Use Case entnommen werden.



**Abbildung 6.24:** *DAYSi* Use Case III: Überblick Datensatz

Die betrachtete Simulation RUN\_2422 weist im Kontext der übrigen Simulationen die höchsten Abweichungskoeffiziente auf. Im Zeitbereich in Abbildung 6.24 ist dies zwischen 10 ms und 25 ms sowie 45 ms und 70 ms ersichtlich. Bei Betrachtung der globalen Abweichungskoeffiziente können gröbere Fehler, wie sie im zweiten Use Case aufgetreten sind, ausgeschlossen werden. Dies lässt sich durch eine nachgelagerte manuelle Analyse bestätigen. Die Auswertung für RUN\_2422 ist Abbildung 6.25 zu entnehmen. Auffälligkeiten an der Fahrzeugstruktur sind insbesondere an der B-Säule (Sensor mitte mitte) und der Tür (Sensor hinten oben) festzustellen. Aber auch im Curtainairbagvolumen und Seitenairbagdruck. Entsprechend werden einerseits Sensoren gefunden, von denen auszugehen ist, dass diese durch die eingebrachte Änderung abweichendes Verhalten zeigen. Andererseits werden aber auch zwei Airbagsignale hervorgehoben.

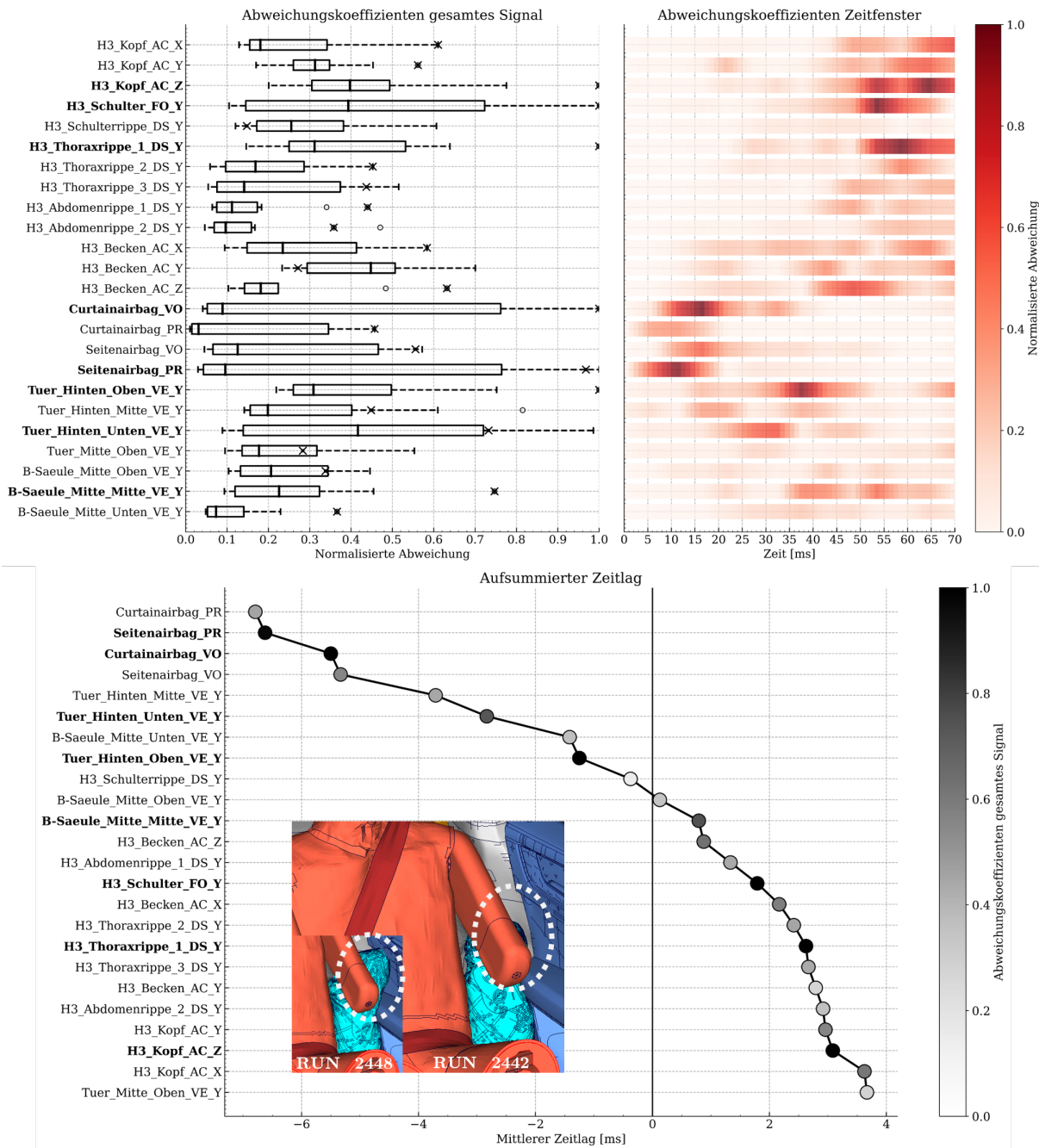


Abbildung 6.25: DAYSi Use Case III: Lokale Ausreißer und Eventkette RUN\_2422

Diese Abweichungen können jedoch auf Basis der initial verfügbaren Information nicht direkt erklärt werden und benötigen daher eine weitergehende Untersuchung. Auf der Heatmap in Abbildung 6.25 ist dies im Zeitbereich zwischen 10 ms und 25 ms auch deutlich zu sehen. In der Eventkette sind beiden Sensoren auch im oben Bereich verortet. Am Seitenairbag ist in Abbildung 6.25 beispielhaft gezeigt, dass im Vergleich zu dem als unauffällig identifizierten RUN\_2448 eine schwächere und spätere Entfaltung zu sehen ist.

Folglich lassen sich identifizierte Auffälligkeiten auch visuell nachweisen. In der Eventkette sind Tür und B-Säule den Abweichungen im Rückhaltesystem vorgelagert, auch wenn diese in der Heatmap weniger stark sind. Dies lässt sich zum Teil auf die veränderte Reibung zurückzuführen, die den Verlauf der Simulation von Beginn an beeinflusst. Entsprechend ist die vorliegende Problemstellung komplex, da Auffälligkeiten von zwei möglichen Ursachen herrühren und nicht vollständig durch die eingebrachten Änderungen am System erklärt werden können. Beide wurden jedoch durch die Verwendung von *DAYSi* in der Heatmap und der Eventkette hervorgehoben und helfen dabei eine Richtung für folgende Untersuchungen vorzugeben.

### 6.3.4 Analyse Verfahrensparameter

Die in Abschnitt 5.3 auf S. 99 eingeführten Verfahrensparameter sollen nun noch weiter untersucht werden. Im Vorangegangenen wurde bereits der Mehrwert von *DAYSi* mit demselben Standardverfahrensparametersatz anhand von drei Use Cases diskutiert.

#### 6.3.4.1 Auswahl und Anzahl Datenpunkte

Die Auswahl und Anzahl der Datenpunkte, die in die Analyse einbezogen werden, geben den Kontext vor. Um diesen Aspekt näher zu beleuchten, werden im Datensatz des zweiten Use Cases die auffälligen Simulationen RUN\_1200 und RUN\_1453 entfernt. Die zugehörigen Visualisierungen sind in Abbildung 6.26 und Abbildung 6.27 zu finden.

Unabhängig davon, ob die beiden Datenpunkte mit berücksichtigt werden oder nicht, bleibt die Reihenfolge für die fünf folgenden Simulationen bis einschließlich RUN\_1043 identisch. Die abnehmende Signifikanz und Ähnlichkeit der noch weniger auffälligen Simulationen sorgt dafür, dass die Reihenfolge hier nicht mehr robust bestimmt werden kann. Allerdings zeigt RUN\_980 in allen vier Datensätzen die geringsten Auffälligkeiten. Diese Positionierung ist dahingehend schlüssig, dass alle übrigen Simulation auf dieser Konfiguration aufbauen. Um abseits der globalen Information auch die lokal extrahierten Information einzelner Datenpunkten zu bewerten, ist die Eventkette für RUN\_2358 neben dem Datensatzüberblick in Abbildung 6.26 und Abbildung 6.27 dargestellt. RUN\_2358 folgt in der Reihenfolge auf die beiden entfernten Datenpunkte RUN\_1200 und RUN\_1453 hinsichtlich der globalen Auffälligkeit.

Dies soll verdeutlichen, dass mit *DAYSi* auch für weniger auffällige Datenpunkte nützliche und konsistente Informationen extrahiert werden können. Als deutlich auffällig, mit einem Abweichungskoeffizienten größer als 0.65, werden im reduzierten Datensatz ohne RUN\_1200 und RUN\_1453 auf Abbildung 6.27 vier Sensoren identifiziert, die entsprechend fett hervorgehoben werden. In den übrigen Datensätzen weist kein Sensor in RUN\_2358 hinreichend hohe Abweichungskoeffizienten auf, um als Anomalie fett hervorgehoben zu werden. Das ist schlüssig, da im Kontext der gegebenen Simulationen kein auffälliges

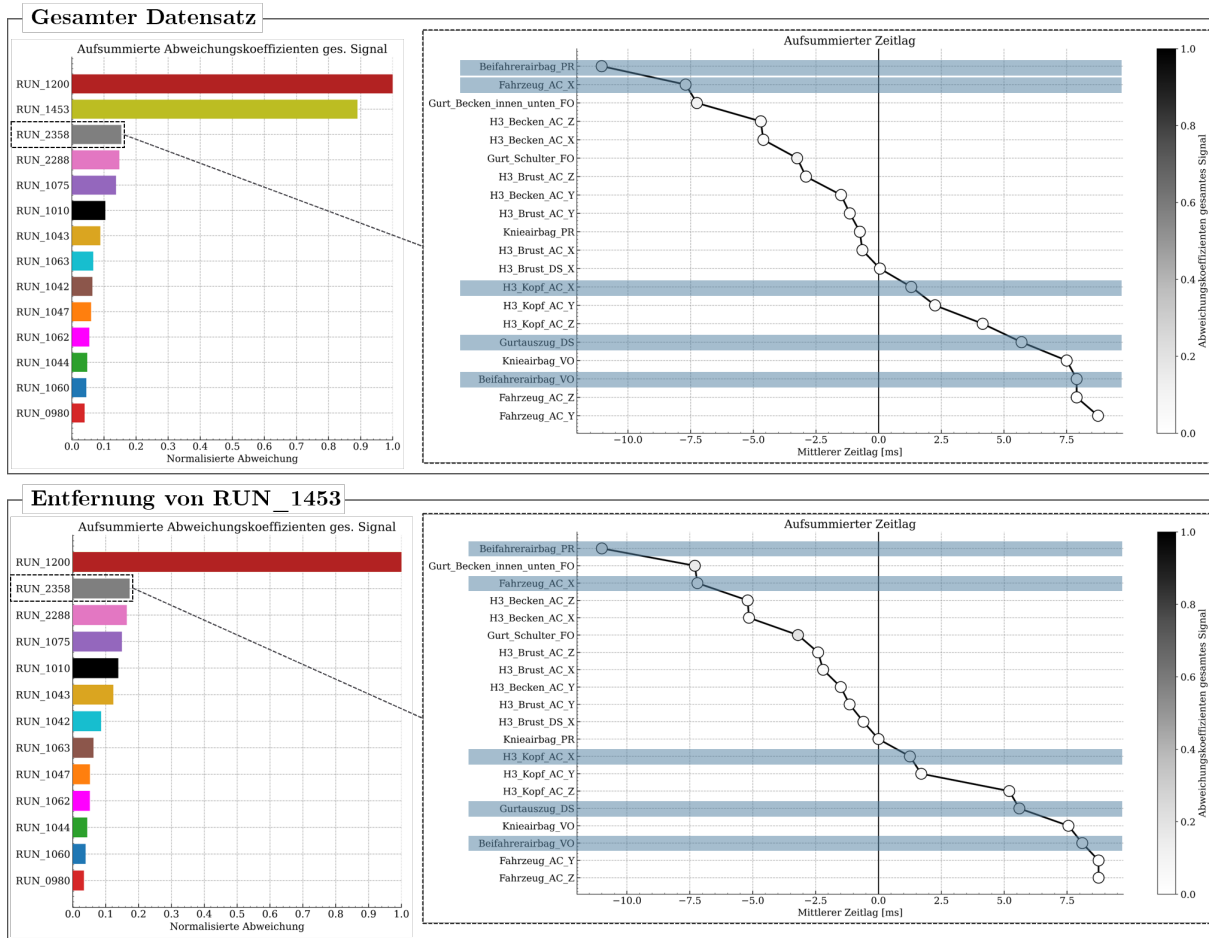


Abbildung 6.26: *DAYSi* Robustheit: Use Case II mit weniger Datenpunkten – Überblick Datensatz und Eventkette I

Verhalten vorliegt. Dies hebt aber auch die Relevanz des gegebenen Kontextes hervor. Um die fünf Sensoren – Beifahrerairbag Druck, Fahrzeugpuls in X, Beifahrerairbag Volumen und Gurtauszug sowie Kopfbeschleunigung in X – einfacher nachzuverfolgen, sind diese in den restlichen Teilen der beiden Abbildungen manuell blau hinterlegt worden. Evident ist, dass die berechnete Eventkette trotz der unterschiedlichen Abweichungskoeffizienten nahezu vollständig konsistent ist. Das betrifft insbesondere die Reihenfolge der blau hervorgehobenen Sensoren, aber auch die Differenzierung zwischen vor und nachgelagerten Effekten. Vor- und nachgelagert meint hier, dass der mittlere Zeitlag kleiner bzw. größer als 0 ms ist. In allen drei reduzierten Datensätzen wird der Mittelpunkt bei durchschnittlich etwa 0 ms mittlerem Zeitlag durch die Knieairbagdruck besetzt. Lediglich im gesamten Datensatz in Abbildung 6.26 rutscht dieser um 1 ms in die vorgelagerten Effekte. Ebenfalls vergleichbar ist das relevante Zeitintervall der mittleren Zeitlags von  $[-11, 8]$  ms.

Folglich konnte gezeigt werden, dass die Verwendung der Kreuzkorrelationen zur Ermittlung einer sinnvollen Reihenfolge der Auffälligkeiten beiträgt. Das hilft dabei mögliche Ursachen

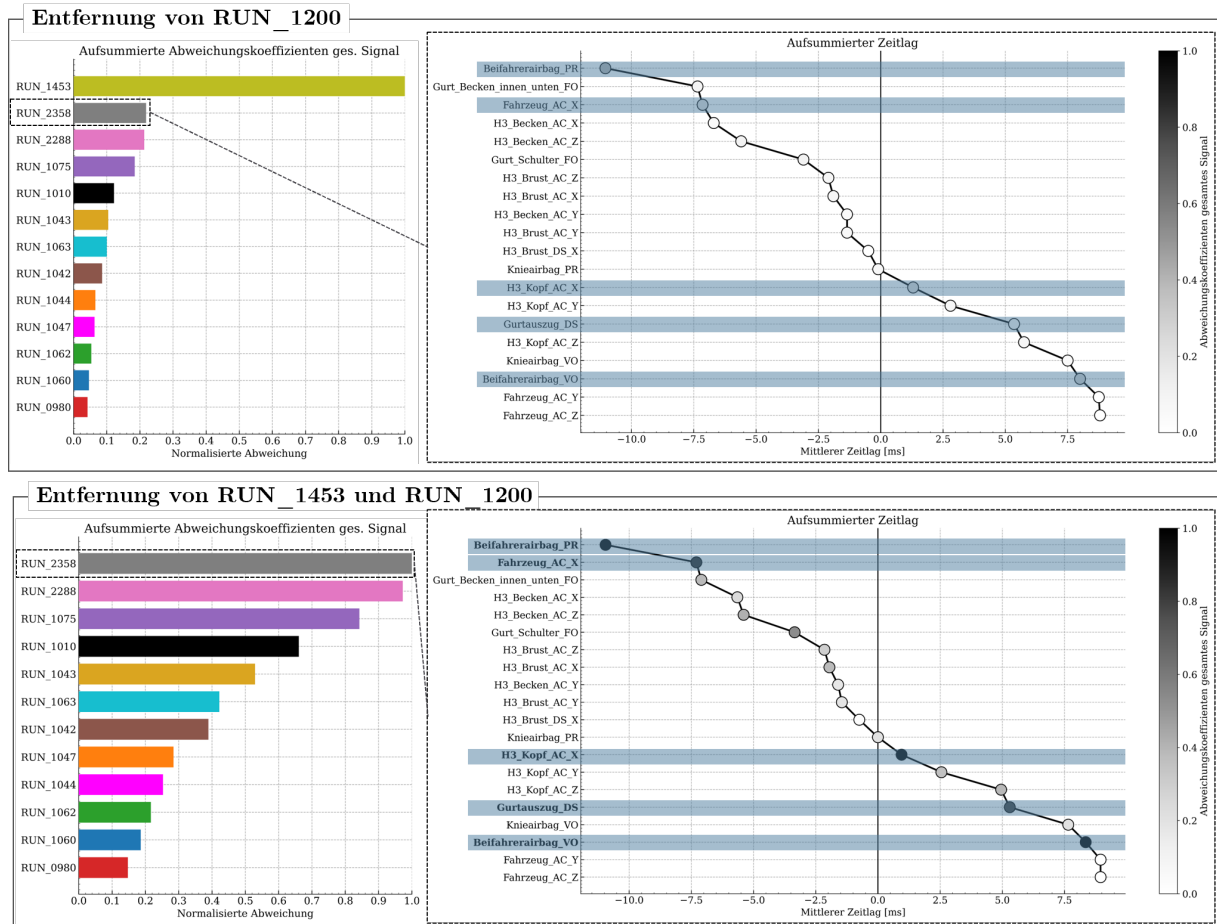


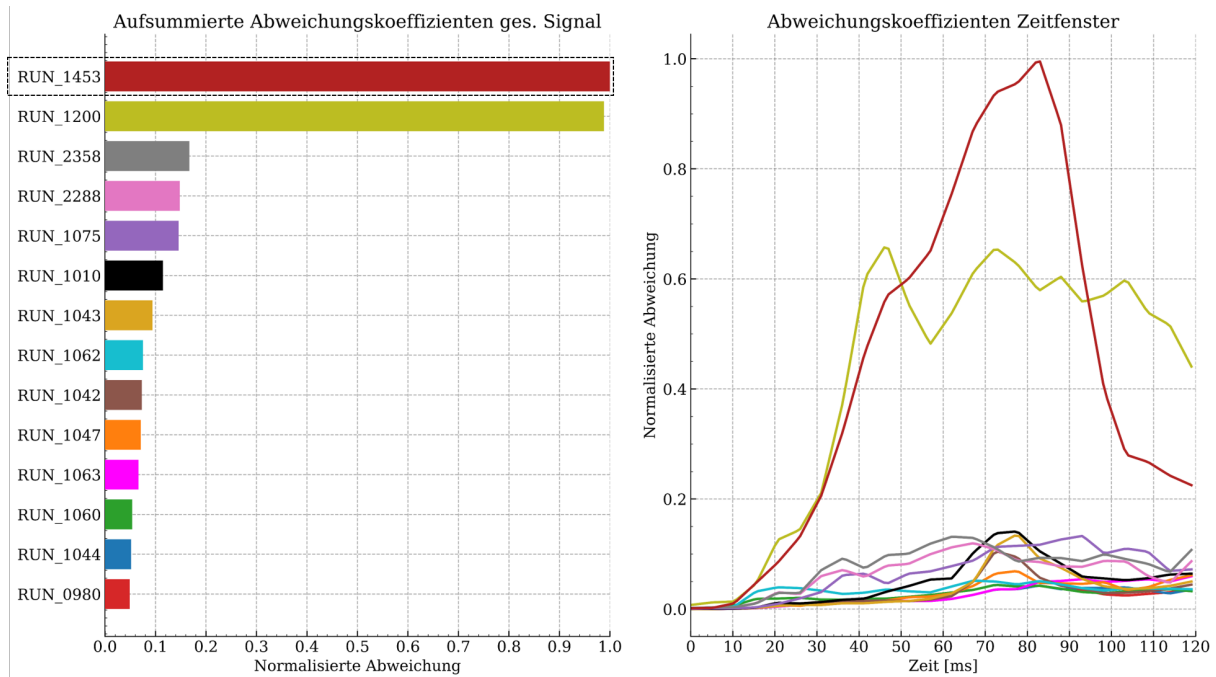
Abbildung 6.27: *DAYSi* Robustheit: Use Case II mit weniger Datenpunkten – Überblick Datensatz und Eventkette II

für auffälliges Verhalten in den frühen Zeitbereichen der Simulation zu bestimmen, ohne dass die absolute Höhe der Abweichungskoeffizienten relevant ist. Die Eventkette kann also auch außerhalb der auffälligsten Datenpunkte spannende Einblicke liefern.

#### 6.3.4.2 Auswahl und Anzahl Sensoren

In den betrachteten Use Cases wurden bisher diejenigen Sensoren verwendet, die die wichtigsten Teile der betrachteten Systeme abbilden. Entsprechend übersichtlich sind die Darstellungen, die das Ziel verfolgen einen ersten Eindruck über das Systemverhalten bereitzustellen. Grundsätzlich kann jedoch auch die Analyse beliebig vieler Sensoren erfolgen. Dies wird anhand Use Case II in Abbildung 6.28 bis Abbildung 6.30 verdeutlicht. Die Analyse ist analog zum zweiten Use Case durchgeführt worden, schließt aber alle verfügbaren Sensoren mit ein.

In dem Datensatzüberblick in Abbildung 6.28 ist ersichtlich, dass die beiden auffälligsten Simulationen RUN\_1200 und RUN\_1453 die Reihenfolge getauscht haben. Dies ist insbesondere im zeitlichen Verlauf im Zeitbereich 60 ms bis 90 ms durch erhöhte Abweichung



**Abbildung 6.28:** *DAYSi* Robustheit: Use Case II mit zusätzlichen Sensoren – Überblick Datensatz

von RUN\_1453 begründbar. Folglich sind die vorher außer Acht gelassenen Sensoren in RUN\_1453 stärker ausreißerbehafet als die in RUN\_1200. Die übrige Reihenfolge im linken Plot in Abbildung 6.28 ist konsistent zu den Erkenntnissen aus der vorangegangenen Robustheitsanalyse.

In Abbildung 6.28 und Abbildung 6.29 sind die Analyseergebnisse für RUN\_1453 gezeigt. Auffällig sind insbesondere alle Gurtkräfte, der Gurtauszug sowie Waden- und Oberschenkelkräfte des ATD. Diese wurden in den bisherigen Analysen nicht berücksichtigt, da die Bewegung des ATD im Vordergrund stand. Die in Abbildung 6.29 visualisierte Eventkette identifiziert jedoch analog zu Abbildung 6.21 identische vor- und nachgelagerte Effekte in der Simulation. Eine gewisse Abhängigkeit zu den ausgewählten Sensoren ist entsprechend gegeben, wobei grundlegende Erkenntnisse aber unverändert bleiben.

Folglich konnte gezeigt werden, dass die Auswahl der Sensoren eine Rolle spielt, globale Tendenzen bei einer sinnvollen Auswahl jedoch dadurch nicht verändert werden. In der Praxis kann beispielsweise abhängig vom Lastfall und ATD eine Liste mit relevanten Sensoren und Richtungen erstellt werden. Grundsätzlich können beliebig viele analysiert werden, sofern diese verfügbar sind. Dies gilt insbesondere für Problemstellungen auf Gesamtfahrzeugebene. Die statischen Darstellungen kommen dann jedoch langsam an ihre Grenzen. Aus Gründen der Ergonomie sollte für diesen Fall eine interaktive Oberfläche verwendet werden. Die Filterung der Ergebnisse ist dann einfacher möglich.

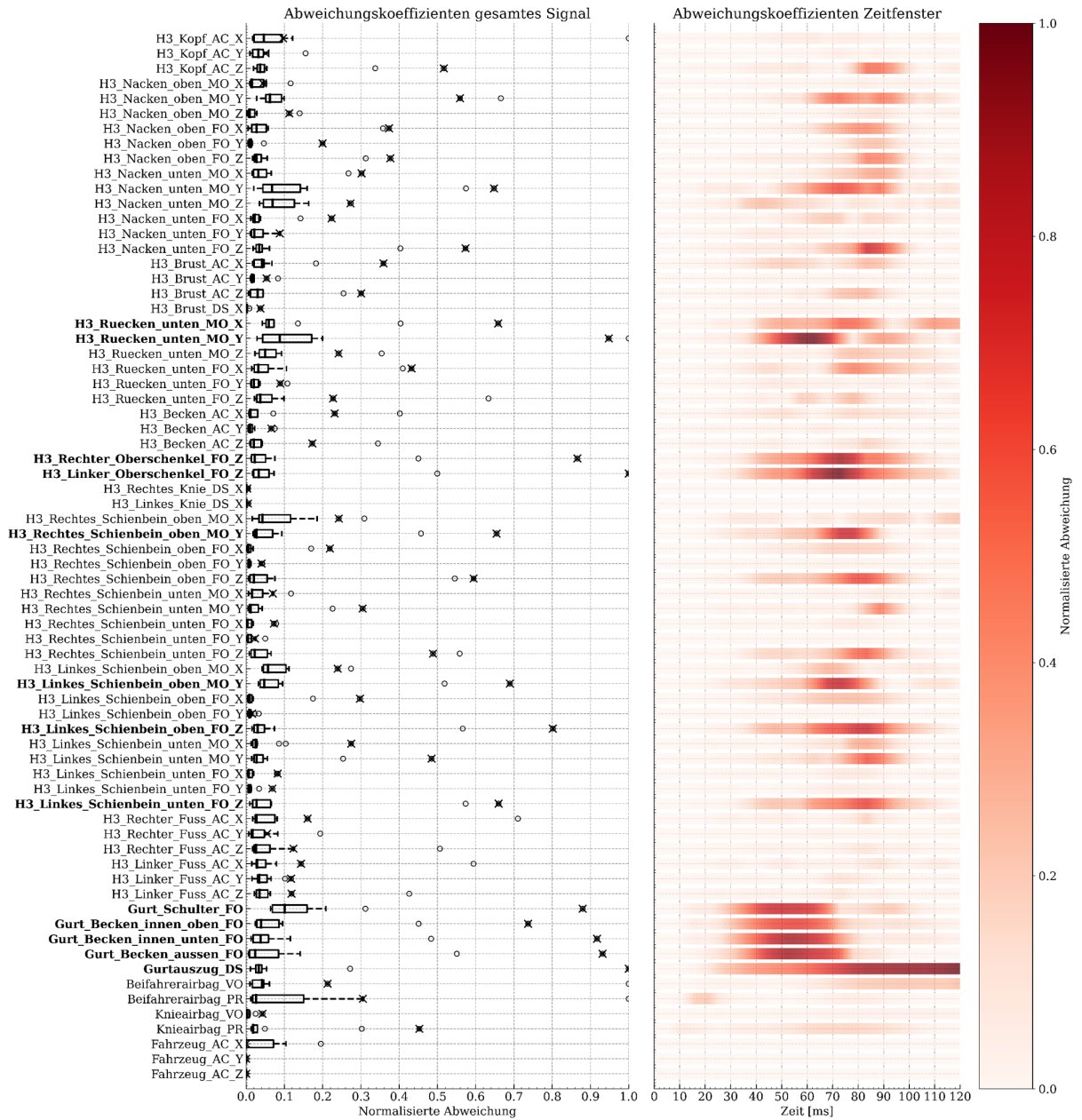


Abbildung 6.29: *DAYSi* Robustheit: Use Case II mit zusätzlichen Sensoren – Lokale Ausreißer RUN\_1453

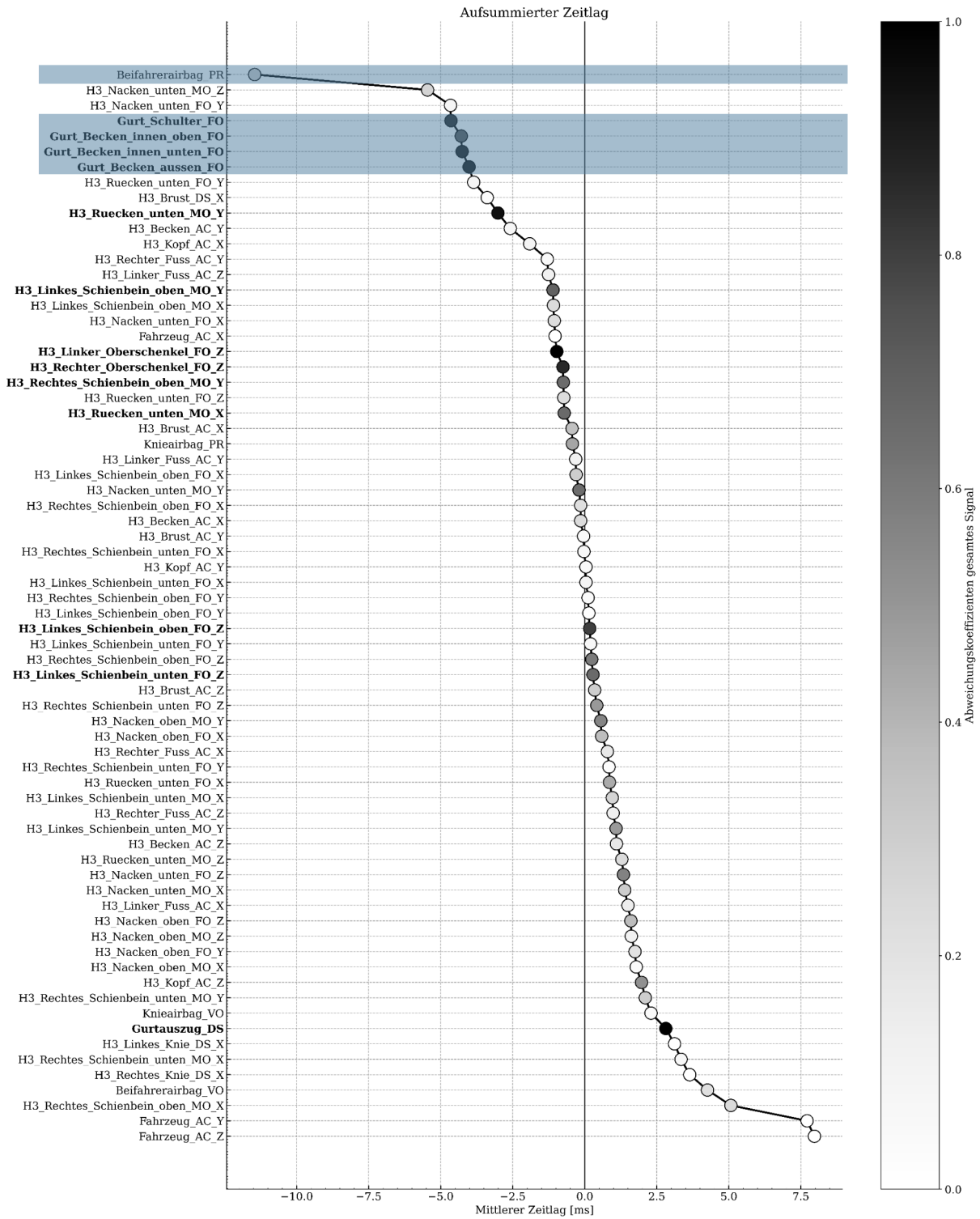


Abbildung 6.30: *DAYSi* Robustheit: Use Case II mit zusätzlichen Sensoren – Eventkette RUN\_1453

### 6.3.4.3 Zeitlags und Korrelation zwischen den Abweichungen

Bisher wurde als erlaubter Zeitlag zwischen allen Sensoren,  $-20\text{ ms} < \tau < 20\text{ ms}$  sowie eine korrespondierende Pearson Korrelation von  $\rho > 0.2$  angenommen. Dies soll in erster Linie physikalisch unrealistische Zusammenhänge aus den Daten filtern und Rauschen durch zu geringe Korrelationen ausschließen. Entsprechend liegt die Annahme zugrunde, dass sich einzelne Teile des Systems nicht beliebig lange beeinflussen. Anhand des ersten Use Cases mit RUN\_1043 werden Eventketten für Zeitintervalle von  $\pm 10\text{ ms}$ ,  $\pm 20\text{ ms}$  und  $\pm 50\text{ ms}$  sowie minimalen Pearson Korrelationen von 0.2 und 0.6 ermittelt. Diese sind in Abbildung 6.31 gezeigt.

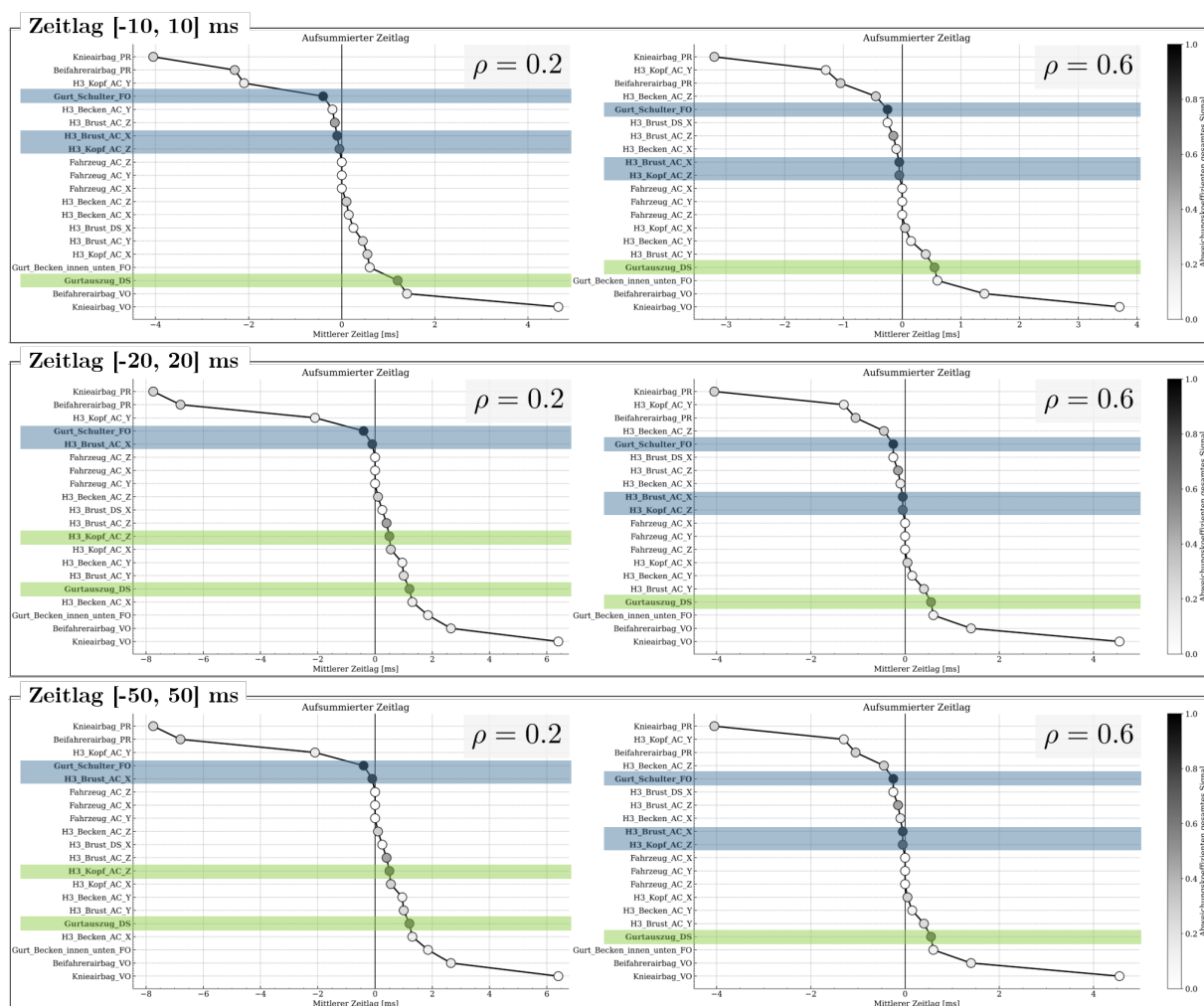


Abbildung 6.31: *DAYSi* Robustheit: Use Case I mit anderen Zeitlags und Korrelationen – Vergleich Eventkette RUN\_1043

Wie bereits im ersten Use Case im Detail betrachtet, ändert sich die Schaltzeit des Gurtkraftniveaus in RUN\_1043. Zusätzlich sind auf der Heatmap (vgl. Abbildung 6.18 auf S. 143) leichte Auffälligkeiten im Bereich 20 ms im Verhalten der Airbags ersichtlich. Auf diese leichten Abweichungen folgen starke Abweichungen in der Gurtkraft an der Schulter,

die sich wiederum direkt auf die Brustbeschleunigung auswirkt. Die anschließend nachgelagerten Effekte treten als Folge in der Kopfbeschleunigung in Z sowie dem Gurtauszug auf. Diese plausible Eventkette wird nur durch die Zeitlagintervalle  $-20 \text{ ms} < \tau < 20 \text{ ms}$  in Kombination mit einer Pearson Korrelation von  $\rho > 0.2$  erreicht. Dieser Verlauf entspricht auch der Erwartung ausgehend von der Kinematik des Systems. Bei einem zu hohen Korrelationsgrenzwert oder einem zu geringen Zeitintervall, wird die Kopfbeschleunigung in Z den vorgelagerten Effekten zugeordnet. Folglich ist gerade bei einem zu hohen Grenzwert für die Pearson Korrelation davon auszugehen, dass weniger starke, aber dennoch relevante Zusammenhänge ausgeschlossen werden. Selbiges gilt auch für zu kurze Zeitintervalle. Trotz der starken Unterschiede in den festgelegten Grenzwerten für Zeitlag und Korrelation ist die Reihenfolge der vier auffälligen Sensoren jedoch in allen Fällen identisch. Dies gilt auch für diverse weitere Größen wie den Knieairbagdruck oder Beifahrer- und Knieairbagvolumen.

Folglich ist die Spezifikation sinnvoller Grenzwerte relevant und kann einen Einfluss auf die Ergebnisse nehmen. Trotzdem lassen sich dadurch Ergebnisse nicht grundlegend verfälschen, da die Tendenzen auch weiterhin übereinstimmen. Theoretisch ist auch eine lokale Spezifikation für einzelne Sensorpaare möglich. So können genauere Annahmen, wie beispielsweise das Ausschließen von einem vorgelagerten Einfluss von ATD auf den Crashpuls, berücksichtigt werden. Dies wird hier jedoch nicht näher beleuchtet und kann Gegenstand zukünftiger Untersuchungen sein. Mit der globalen Definition der Grenzen wurden hier hinreichend genaue Ergebnisse erzeugt.

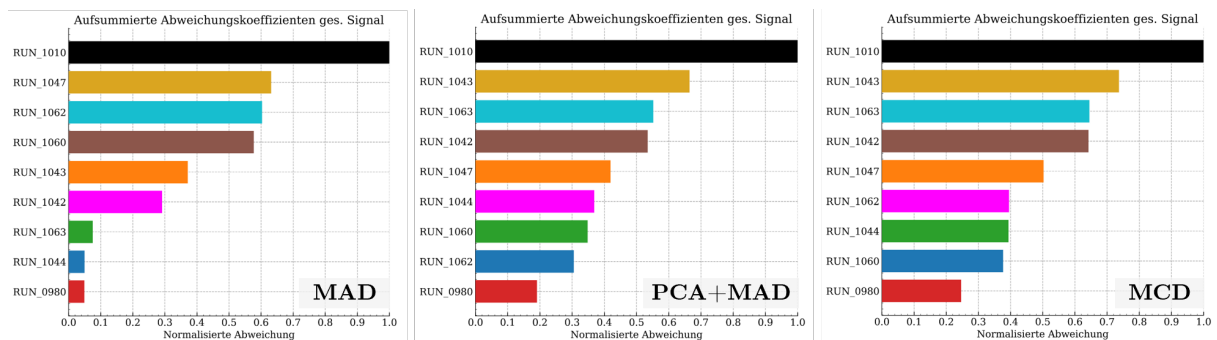
#### 6.3.4.4 Abstandsmaß zur Berechnung von Ausreißern

Die Berechnung der Abweichungskoeffizienten wurde bislang mit dem in Abschnitt 5.3 ab S. 99 vorgeschlagenen MCD durchgeführt. Um dieses Vorgehen einzuordnen, werden Abstände zur Bestimmung der Ausreißer mit dem Abstandsmaß Absolute Distanz zum Median (MAD) herangezogen. Dieses Verfahren ist robust und wird auch in [Kra+23] für Ausreißer in FE-Daten eingesetzt und berechnet sich nach Gleichung (6.4).

$$\text{MAD} = \text{median} (\|x_i - \text{median}(x)\|) \quad (6.4)$$

Da die verwendete euklidische Norm (L2-Norm)  $\|\cdot\|$  in hohen Dimensionen an Aussagekraft verliert, wird in einer dritten Variante zunächst eine PCA in der Zeitdomäne jeden Sensors durchgeführt, um die Distanzberechnung in einem niedrigdimensionalen Raum vorzunehmen zu können. Es werden so viele Hauptkomponenten verwendet, wie zum Erreichen von 90 % erklärter Varianz vonnöten sind. Im weiteren Verlauf wird dies als PCA+MAD bezeichnet. Der Vergleich der globalen Datensatzübersicht der drei Methoden ist Abbildung 6.32 zu entnehmen. Wird der MAD verwendet, stimmen in der Übersicht in

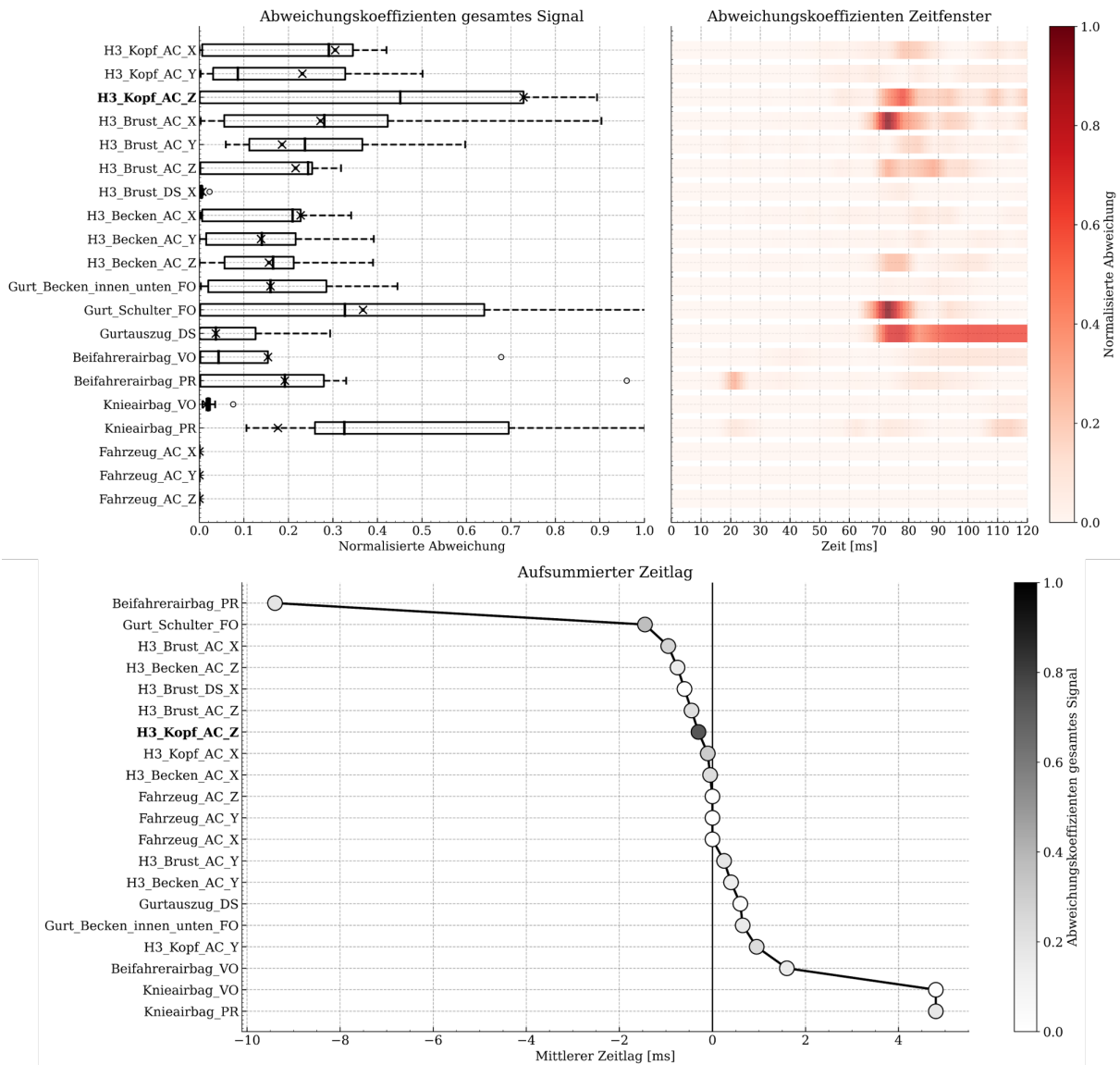
Abbildung 6.32 lediglich die auffälligste Simulation RUN\_1010, in der eine Erhöhung des Gurtkraftniveaus in Kombination mit einer Änderungen des Beifahrerairbags eingebaut sind, und die unauffälligste Simulation RUN\_0980, auf der alle weiteren Varianten aufbauen, mit den Ergebnissen der anderen beiden Abstandsmaße überein. Die übrige Reihenfolge bei der Verwendung von MAD ist bei manueller Betrachtung der Simulationen nicht schlüssig. Die an zweiter Stelle angeordnete Simulation RUN\_1047 weist lediglich eine leichte Verkleinerung der Auslassöffnung des Beifahrerairbags auf. In logischer Konsequenz ist also nicht zu erwarten, dass hier eine grobe Auffälligkeit vorliegt. Eine große Änderungen der Schaltzeit des Grutkraftniveaus (+10 ms), wie in RUN\_1043, müsste aus Ingenieurssicht deutlich stärker ins Gewicht fallen. In Abbildung 6.32 ist ersichtlich, dass für MCD und PCA+MAD die auffälligsten fünf sowie die unauffälligste Simulation in identischer Reihenfolge angeordnet werden. Lediglich drei Simulationen, in denen die Signifikanz und Varianz der Abweichungskoeffizienten gering ist, unterscheiden sich in der Reihenfolge. Entsprechend können mit beiden Verfahren auf globaler Ebene gleichwertige Ergebnisse erzielt werden.



**Abbildung 6.32:** DAYSi Robustheit: Use Case I mit verschiedenen Abstandsmaßen – Vergleich Überblick Datensatz

Auch lokal ergeben sich Unterschiede in den Ergebnissen für Ausreißer in den Sensoren und der Eventkette. Diese sind für MAD in Abbildung 6.32 und für PCA+MAD in Abbildung 6.33 beispielhaft für RUN\_1043 für den Use Case I abgebildet. Für MCD ist diese Darstellung bereits in Abbildung 6.18 auf S. 143 zu finden.

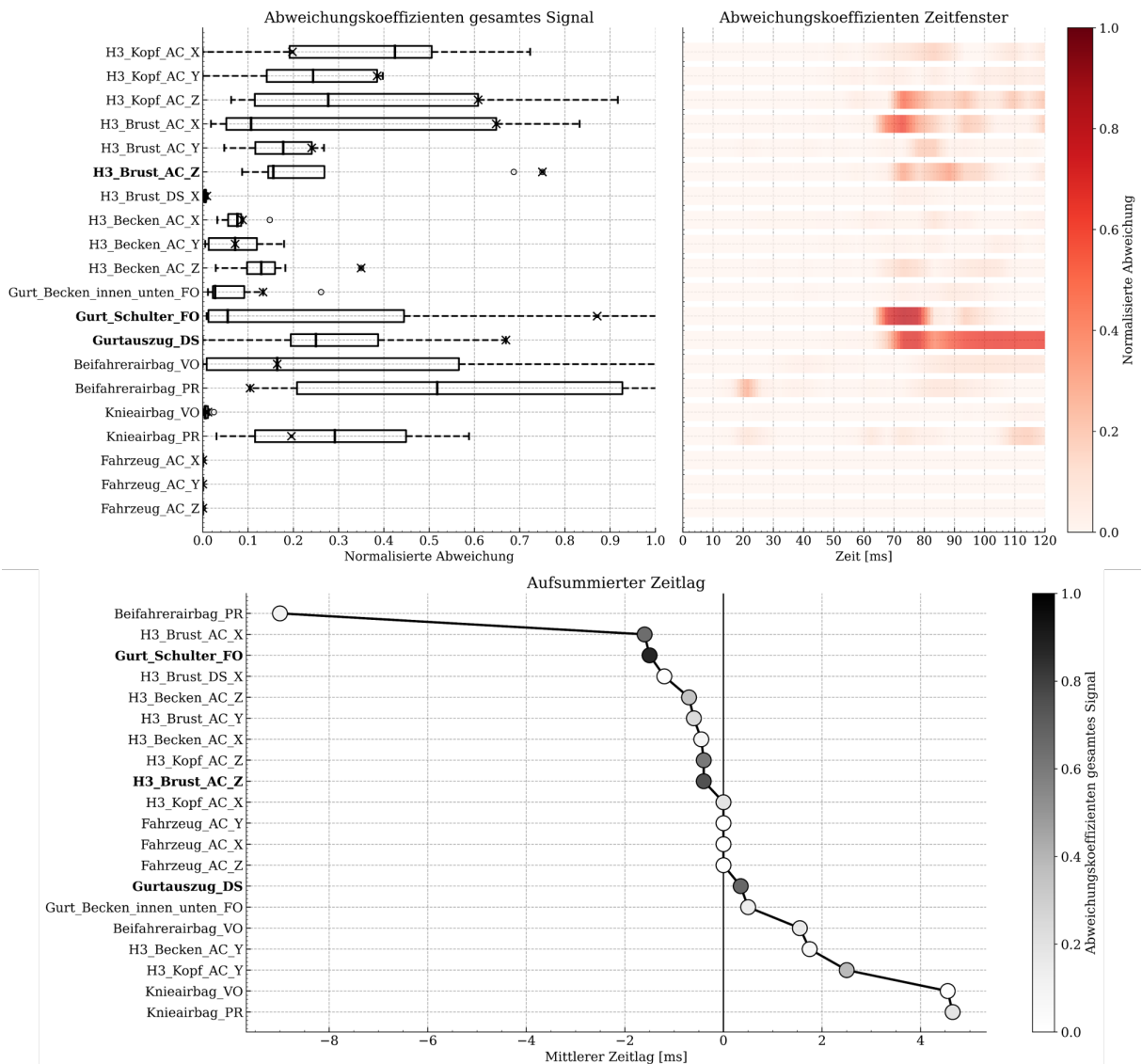
Bei der Verwendung von MAD (vgl. Abbildung 6.33) besteht das Problem, dass Ausreißer in Bezug auf das gesamte Signal nicht sinnvoll differenziert werden können. Oberhalb des Grenzwerts von 0.65 befindet sich lediglich die Kopfbeschleunigung in Z. Diese ist auch nach manueller Prüfung auffällig und wird ebenfalls von MCD und PCA+MAD gefunden. Andere Signale, die visuell weitaus auffälliger sind, werden aber nicht hervorgehoben. Wie erwartet, ist die Dimensionalität  $\mathbb{R}^{1200}$  zu hoch, um eine sinnvolle Differenzierung durch Abstände vorzunehmen. Lokal hingegen können mit MAD sinnvolle Ergebnisse erzielt werden.



**Abbildung 6.33:** *DAYSi* Robustheit: Use Case I mit MAD – Lokale Ausreißer und Eventkette RUN\_1043

Die leichten Auffälligkeiten bei etwa 20 ms in den Airbags sowie die deutlichen Abweichungen in Gurt und der Brustbeschleunigung werden in der Heatmap – analog zu PCA+MAD und MCD – hervorgehoben. Da die einzelnen Stücke des Signals bei einer Zeitfenstergröße von 5 ms in  $\mathbb{R}^{50}$  verortet sind, ist hier die L2-Norm weitaus aussagekräftiger. Dies wird vor allem am Beispiel des Gurtauszugs in Abbildung 6.33 klar. Etwa die Hälfte des Signals wird lokal als auffällig hervorgehoben, der Koeffizient des gesamten Signal beschreibt, ist jedoch lediglich der Median in der Verteilung. In Bezug auf die Eventkette, wird die Gurtkraft an der Schulter sowie die Brustbeschleunigung in X korrekt als vorgelagerte Effekte identifiziert. Trotz der Einordnung als unauffällig wird der Gurtauszug als sekundärer Effekt spezifiziert, da die Kreuzkorrelation auf den lokalen Abweichungskoeffizienten über die Zeit

aufsetzt. Entgegen einer physikalisch schlüssigen Kinematik wird die Kopfbeschleunigung in X und Z als vorgelagerte Effekte eingeordnet.



**Abbildung 6.34:** DAYSi Robustheit: Use Case I mit PCA+MAD – Lokale Ausreißer und Eventkette RUN\_1043

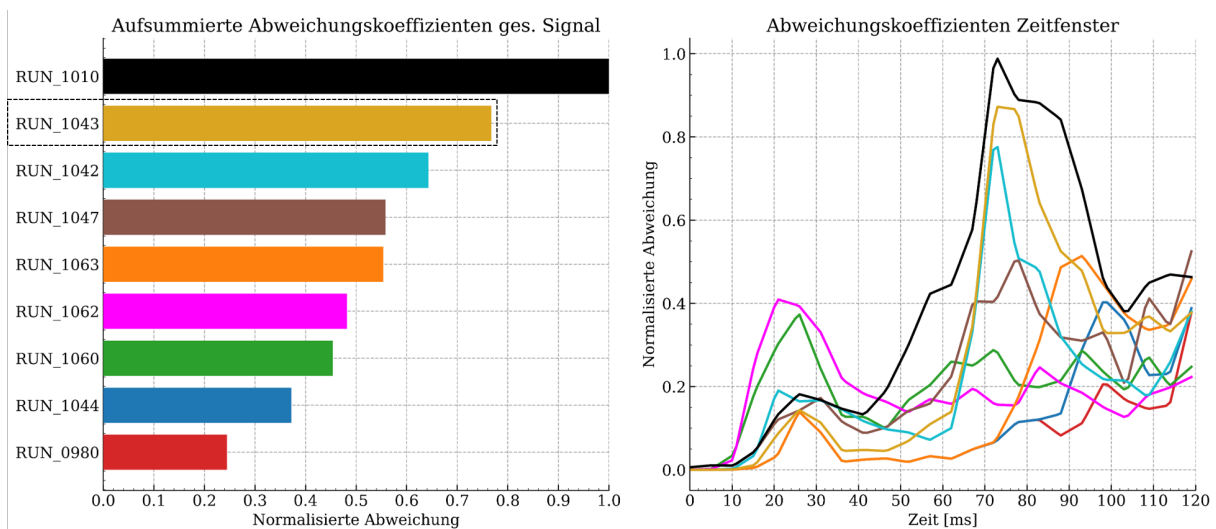
Wird PCA+MCD verwendet (vgl. Abbildung 6.34), werden drei Sensoren mit Abweichungskoeffizienten  $> 0.65$  spezifiziert: Die Gurtkraft an der Schulter, die Brustbeschleunigung in Z sowie der Gurtauszug. Die Identifikation der Brustbeschleunigung in Z ist hier nicht ganz schlüssig, da der Einfluss der Gurtkraft vor allem auf die X-Richtung wirkt und diese – entsprechend der Erwartung – als Auffälligkeit hervorgehoben wird. Bei visueller Betrachtung der lokalen Ausreißer in Abbildung 6.34 sind kaum Unterschiede zur den anderen beiden Verfahren erkennbar. Im Vergleich mit MAD wird die Beckenbeschleunigung in Z zwischen 70 ms und 80 ms etwas weniger stark hervorgehoben. Für

MCD ist der Knieairbagdruck zwischen 20 ms und 30 ms sowie die Kopfbeschleunigung in X um 80 ms stärker eingefärbt. In Bezug auf die Reihenfolge der Eventkette ergibt sich ein ähnliches Bild wie in Abbildung 6.33. Schlüssiger sind nun nur die Auffälligkeiten der gesamten Signale, was die Identifikation der Hauptursache in RUN\_1043 – der Gurtkraft an der Schulter – und einem schlüssigen nachgelagerten Effekt – dem Gurtauszug – nun möglich macht. Jedoch wird auch hier entgegen einer physikalisch schlüssigen Kinematik die Kopfbeschleunigung in X und Z als vorgelagerte Effekte eingeordnet.

Folglich sind deutliche Ausreißer global – RUN\_1010 – und lokal – Gurtauszug RUN\_1043 – mit allen Abstandsmaßen grundsätzlich bestimmbar und können hervorgehoben werden. Bei einer feingranularen Differenzierung auf beiden Ebenen ergeben sich jedoch deutliche Unterschiede. Einerseits bei der Aussagekraft der Abweichungskoeffizient für das gesamte Signal und andererseits beim Aufsummieren der Abweichungskoeffizienten einzelner Signale zur Ermittlung der globalen Übersicht. Zusätzlich ist ein Einfluss auf die Reihenfolge der Eventkette ersichtlich, deren Schlüssigkeit durch Abweichungskoeffizienten beeinflusst wird. Im Vergleich liefert MCD die sinnvollsten Ergebnisse für dieses Beispiel. Jedoch sind auch hier Schwachstellen, wie der Vorgelagerte Effekt der Kopfbeschleunigung in Y ersichtlich.

#### 6.3.4.5 Skalierung der Daten

In den bisherigen Analysen wurde die Skalierung der einzelnen Zeitserien auf Basis der physikalischen Einheiten durchgeführt. Die Grundidee hierbei ist, dass die Größenordnung stärker ins Gewicht fällt.

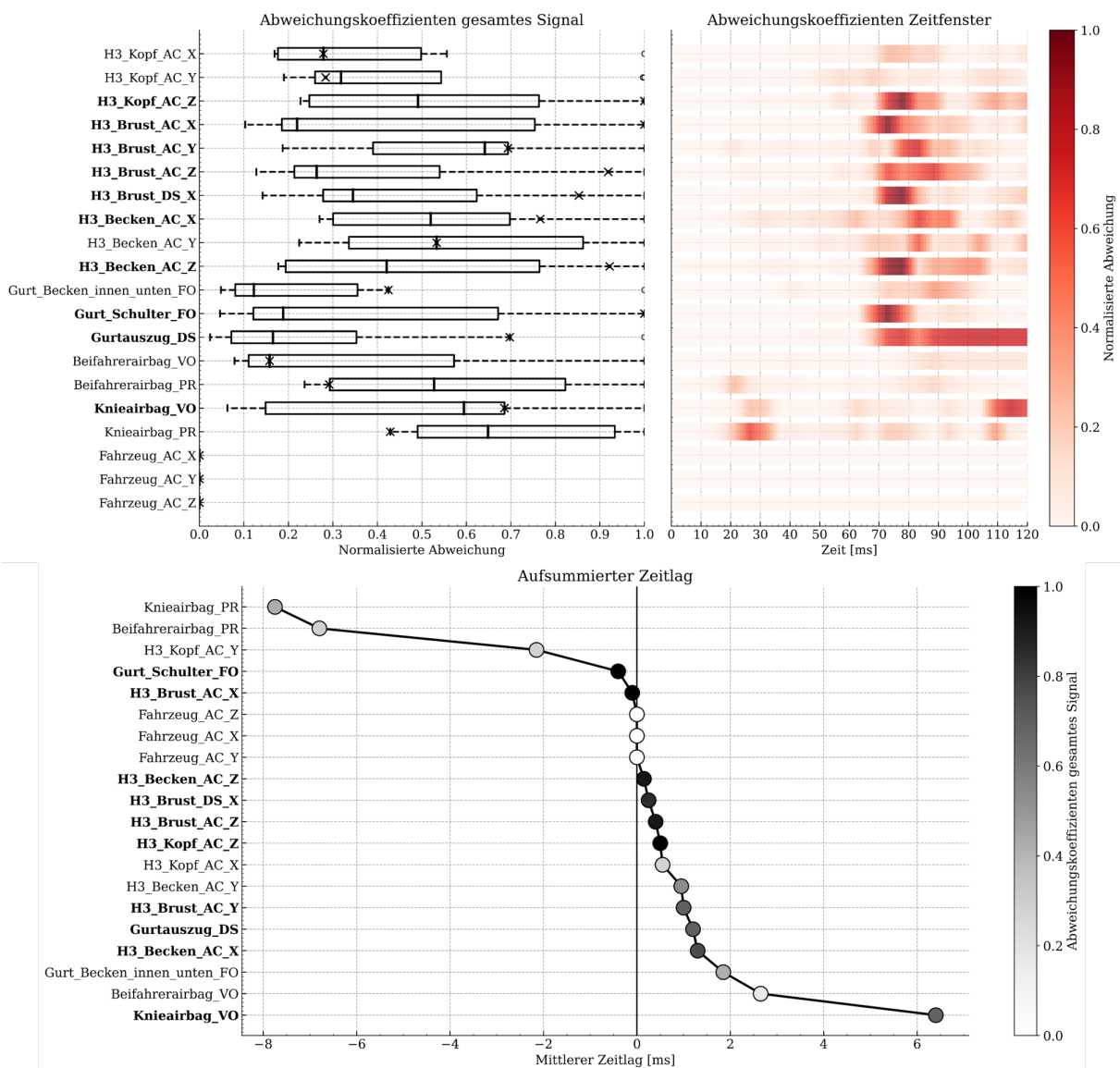


**Abbildung 6.35:** *DAYSi* Robustheit: Use Case I mit signalweise Skalierung – Überblick Datensatz

Dies kann beispielsweise durch eine Hauptbewegungsrichtung – X im Frontalaufprall – gegeben sein. Einige Signale in Z können Rauschen enthalten, was starke Abweichungen

suggeriert, obwohl keine Relevanz für das Verhalten des Systems vorliegt. Im klassischen Prozess – der manuellen Analyse – wird ebenfalls auf eine richtige Achsenskalierung geachtet, um keine falschen Schlüsse zu ziehen. Eine Alternative zur gruppenweise Skalierung der Daten ist die Skalierung jedes einzelnen Sensors. Dieses Beispiel wird ebenfalls anhand von RUN\_1043 im Use Case I diskutiert.

Der Überblick über den Datensatz ist in Abbildung 6.36 gezeigt. Die beiden auffälligsten Simulationen RUN\_1010 und RUN\_1043 sowie die unauffälligste Simulation RUN\_0980 stimmen mit den Ergebnissen der gruppenweise Skalierung in Abbildung 6.17 auf S. 142 überein.



**Abbildung 6.36:** DAYSi Robustheit: Use Case I mit signalweise Skalierung – Lokale Ausreißer und Eventkette RUN\_1043

Deutlich unübersichtlicher wird es in Abbildung 6.36 am Beispiel von RUN\_1043. Hinsichtlich den Abweichungskoeffizienten für das gesamte Signal ist erkennbar, dass viele Sensoren hervorgehoben werden, da sie den festgelegten Grenzwert von 0.65 übersteigen. Durch die sensorweise Skalierung wird die Brustbeschleunigung in allen Raumrichtungen als Ausreißer identifiziert. In der gruppenweise Skalierung ist lediglich die Raumrichtung X ein Ausreißer, da die übrigen Boxplots gesamtheitlich unter die Grenze fallen.

Folglich sind in allen ATD-Sensoren und dem Gurt tiefrote Bereiche erkennbar, was eine Differenzierung, wo die relevanten Ausreißer sind, deutlich erschwert. Da die Eventkette rein auf Korrelationen aufsetzt, spielt die Größenordnung und Skalierung für diese keine Rolle. Entsprechend ist die identifizierte zeitliche Abfolge identisch. Lediglich die Einfärbung entsprechend der Abweichungskoeffizienten für das gesamte Signal ist weniger gut differenzierbar. Eine Aussage hinsichtlich der Hauptursache – Gurtkraft Schulter – ist jedoch analog möglich.

Durch die verbesserte Differenzierung wird die gruppenweise Skalierung als grundsätzlich passender für die meisten Situationen betrachtet. Eine Auswahlmöglichkeit kann für bestimmte Fälle jedoch Flexibilität schaffen.

#### **6.3.4.6 Zeitfenstergröße und Filterung**

Ebenfalls untersucht wurden Änderungen in der Zeitfenstergröße und der Filterung der Signale. Diese sind jedoch aus Platzgründen dem Anhang in Abschnitt A.2 ab S. 262 zu entnehmen. Hier wurde gezeigt, dass zu kleine Fenstergröße für die lokale Analyse Rauschen verursacht, was Einfluss auf die Eventkette nimmt. Selbiges gilt für eine zu schwache Filterung, da die Kurven ebenfalls Rauschen aufweisen können. Bei den verwendeten Standardwerten – 5 ms Zeitfenster und cfc60 Filter – ist *DAYSi* robust.

## 6.4 *ExPLAIN* – Erklärung von Systemverhalten

Die Evaluierung von *ExPLAIN* erfolgt anhand von drei repräsentativen Use Cases aus der Entwicklung passiver Sicherheitssysteme. Ebenfalls werden die bei der Einführung der Methode in Abschnitt 5.4 ab S. 107 gelisteten Verfahrensparameter abgeprüft, um die Robustheit zu bestimmen. Für die Betrachtung der Use Cases selbst, werden die Verfahrensparameter auf den bereits definierten Standardwerten belassen.

### 6.4.1 Use Case I: Lösungsvergleich in der Strukturoptimierung

Ein Überblick über den ersten Use Case aus der Strukturoptimierung ist in Abbildung 6.37 gegeben. Dabei wird ein (i-o)-Zusammenhang zwischen 38 Wandstärken der Karosseriestruktur und dem Optimierungszielwert Sitzbreitenminderung und Masse (SwM) für einen Pfahl-Seitenaufprall 90° gemäß EuroNCAP analysiert.

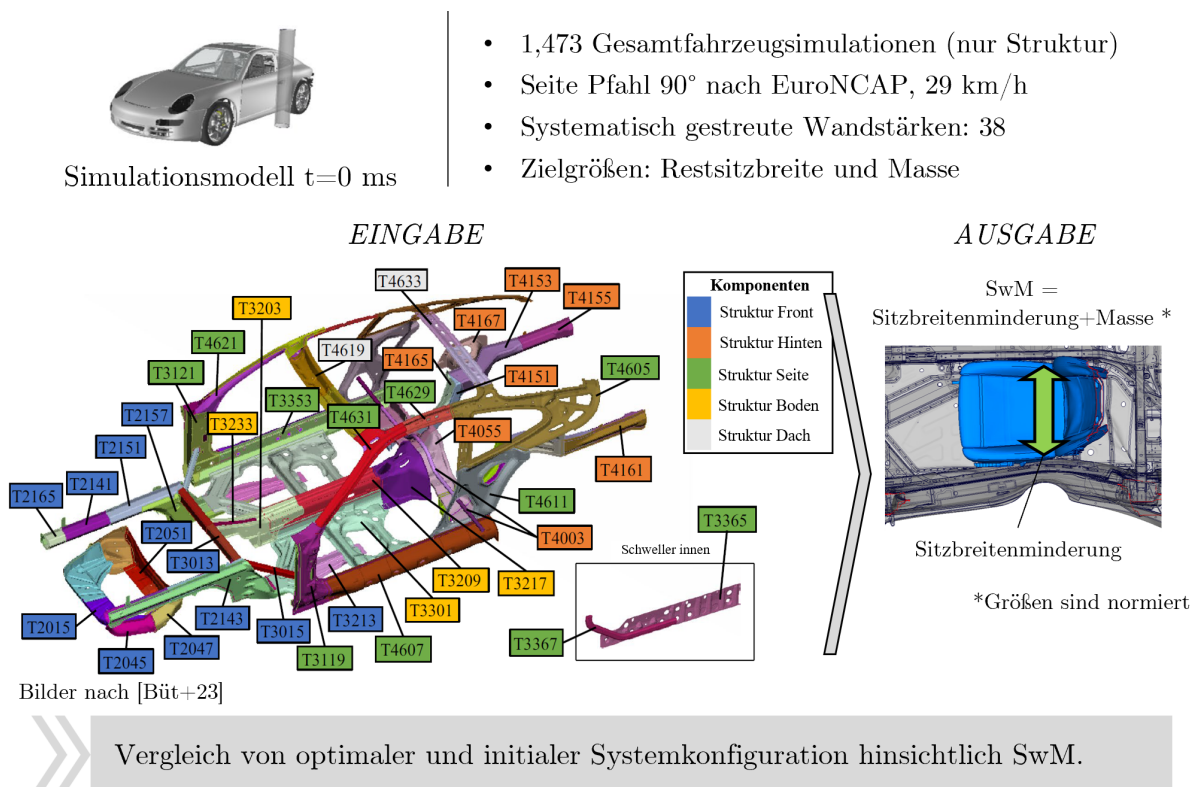
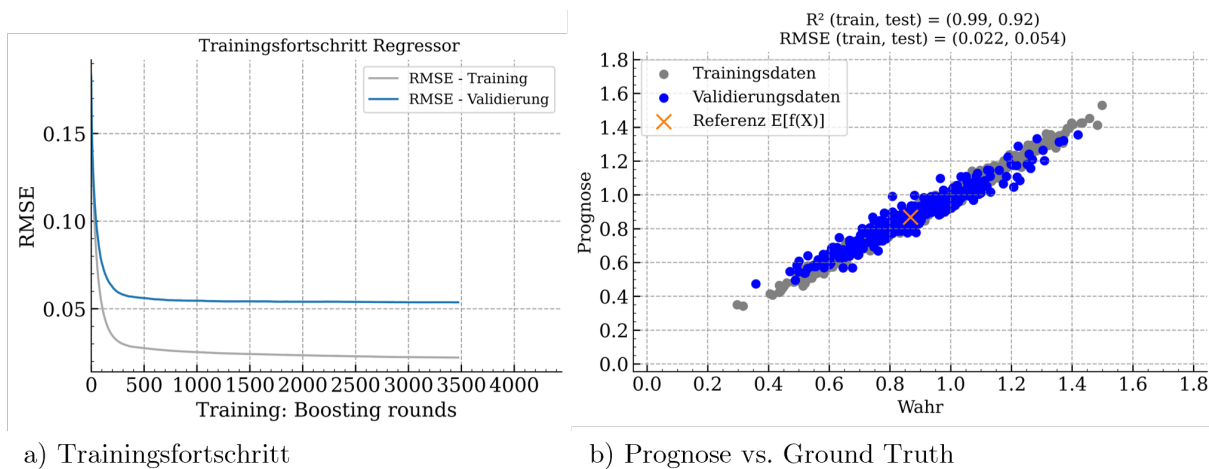


Abbildung 6.37: *ExPLAIN* Use Case I: Problemstellung

SwM entspricht gemäß Abbildung 6.37 der Summe der normierten Sitzbreitenminderung und Masse des Rohbaus. Die Normierung sorgt für Vergleichbarkeit der Einheiten, da sich die Ausgaben in unterschiedlichen Größenordnungen befinden. Die Daten stammen aus den Arbeiten von Büttner [Büt22; Büt+23] und enthalten 1,473 Gesamtfahrzeugsimulationen, in denen die Wandstärken der Rohbaukomponenten systematisch mittels Latin Hypercube

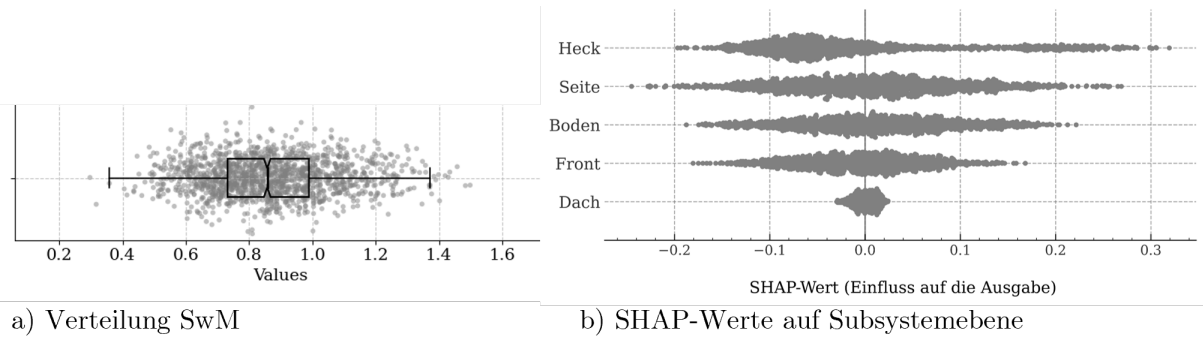
Sampling variiert wurden. Ziel dieser Betrachtung ist es nun die initiale und optimale Lösung durch Differenz-SHAP miteinander zu vergleichen.

Gemäß Abbildung 6.37 werden die Wandstärkenparameter als Eingabe und SwM als Ausgabe für das Metamodell verwendet. Bei den Eingaben handelt es sich damit um *Designparameter* und bei den Ausgaben um *Features*, die aus dem Berechnungsmodell  $t > 0$  ms extrahiert wurden. Die Fehlerkonvergenz des RMSE im Training des XGBoost-Modells sowie ein Fehlerplot für Trainings- und Validierungsdaten ist Abbildung 6.38 zu entnehmen. Das trainierte Metamodell erzielte ein Bestimmtheitsmaß  $R^2$  von 0.92 auf dem Testdatensatz im ersten Fold einer 5-fold Cross Validation. Es ist zu erwarten, dass die Metamodelle aufgrund von Nichtlinearitäten und dem zeitabhängigem Systemverhalten, das durch das Metamodell nicht erfasst wird, zum leichten Overfitting neigen. Dies wird auch im Konvergenzdiagramm in Abbildung 6.38 verdeutlicht – bei längerer Trainingsdauer hätte sich das Overfitting weiter verstärkt, obwohl die Anzahl der verfügbaren Simulationen (1,473) größer ist als das Quadrat der Eingaben (Wandstärken) mit ( $38^2 = 1,444$ ). Das gilt meist als grobe Faustregel für eine hinreichende Datenmenge. Die 5-fold Cross Validation zeigt, dass diese erreichte Güte über die fünf Folds hinweg reproduzierbar ist – diese streuen zwischen 0.87 und 0.92.



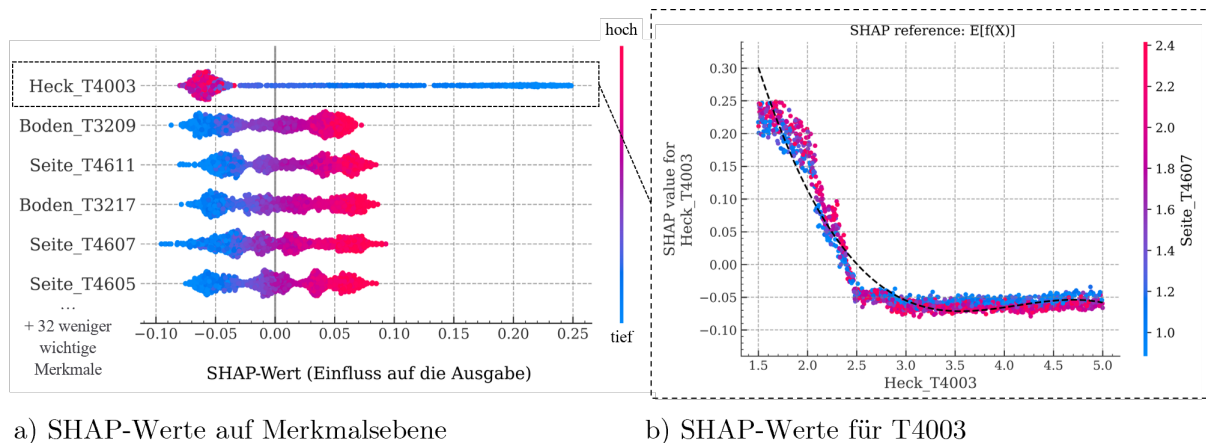
**Abbildung 6.38:** *ExPLAIN* Use Case I: Trainingsfortschritt und Güte des Metamodells in Fold 1

Ausgehend von dem trainierten Metamodell werden gemäß dem Ablauf nach Abbildung 5.9 auf S. 108 die SHAP-Werte berechnet. Auf globaler Ebene können diese Abbildung 6.39 entnommen werden. Zur Referenz ist hier auch die Verteilung des Zielwerts SwM im linken Teil der Abbildung 6.39 gezeigt. Die System-SHAP Repräsentation (vgl. Abbildung 6.39 b)) ermöglicht hier die Identifikation wichtiger Fahrzeugbereiche und spezifischer Komponenten. Hauptsächlich sind Komponenten aus dem Heck und der Seite sowie der Fahrzeugboden für den Zielwert aus Sitzbreitenminderung und Masse relevant.



**Abbildung 6.39:** *ExPLAIN* Use Case I: Verteilung SwM und SHAP-Werte auf Systemebene

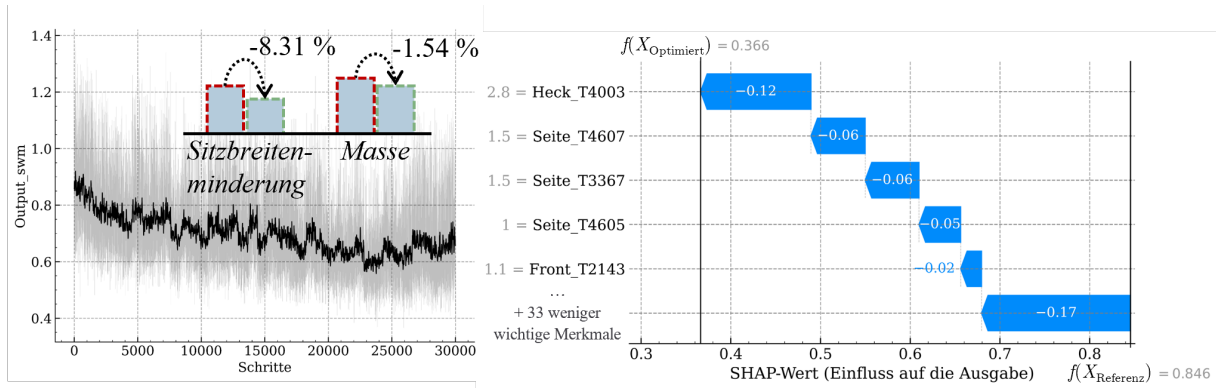
Von besonderem Interesse auf der Merkmalsebene, wie in Abbildung 6.40 a) dargestellt, ist T4003 (Absorberrohr bei den hinteren Sitzquerträgern), das bei geringer Dicke signifikant hohe Beiträge zum Zielwert leisten kann, wobei der positive Beitrag jedoch begrenzt ist. Die übrigen Werte weisen ähnliche Verteilungen auf und sind nach absolutem mittleren Einfluss sortiert. Der Übersichtlichkeit halber sind lediglich die wichtigsten sechs Merkmale gezeigt. Aus Sicht der Systemauslegung sind die generierten Informationen, insbesondere für T4003 von hoher Bedeutung, da sich sowohl die Richtung des Einflusses als auch die Änderung des Einflusses mit dem Wert der Eingabe identifizieren lässt. Das ist mit klassischen Verfahren wie den Sobol Indizes nicht möglich. Abbildung 6.40 b) zeigt die SHAP-Werte über den gesamten Parameterbereich für T4003 von 1,5 mm bis 5,0 mm. Deutlich ersichtlich ist, dass ab einer Wandstärke von etwa 2,5 mm eine weitere Erhöhung keinen zusätzlichen Nutzen für den Zielwert SwM mehr hat.



**Abbildung 6.40:** *ExPLAIN* Use Case I: SHAP-Werte auf Merkmalsebene

Der betrachtete Zielwert SwM, bestehend aus Sitzbreitenminderung und Masse, eignet sich zur gleichzeitigen Optimierung beider Größen. Zur Bestimmung einer optimalen

Konfiguration der Wandstärken im Rohbau wird das trainierte Prognosemodell an einen Differential Evolution Optimierungsalgorithmus [SP97] übergeben – analog zum bereits durchgeführten Benchmark für *DrAGOS* in Abschnitt 6.2.4 ab S. 138. Das Vorgehen entspricht der Metamodelloptimierung, die bereits in Abbildung 2.6 auf S. 40 eingeführt wurde.



a) Fortschritt Optimierung SwM

b) Difference SHAP: Optimierte vs. Initiale Konfiguration

**Abbildung 6.41:** ExPLAIN Use Case I: Optimierung und Differenz-SHAP-Werte

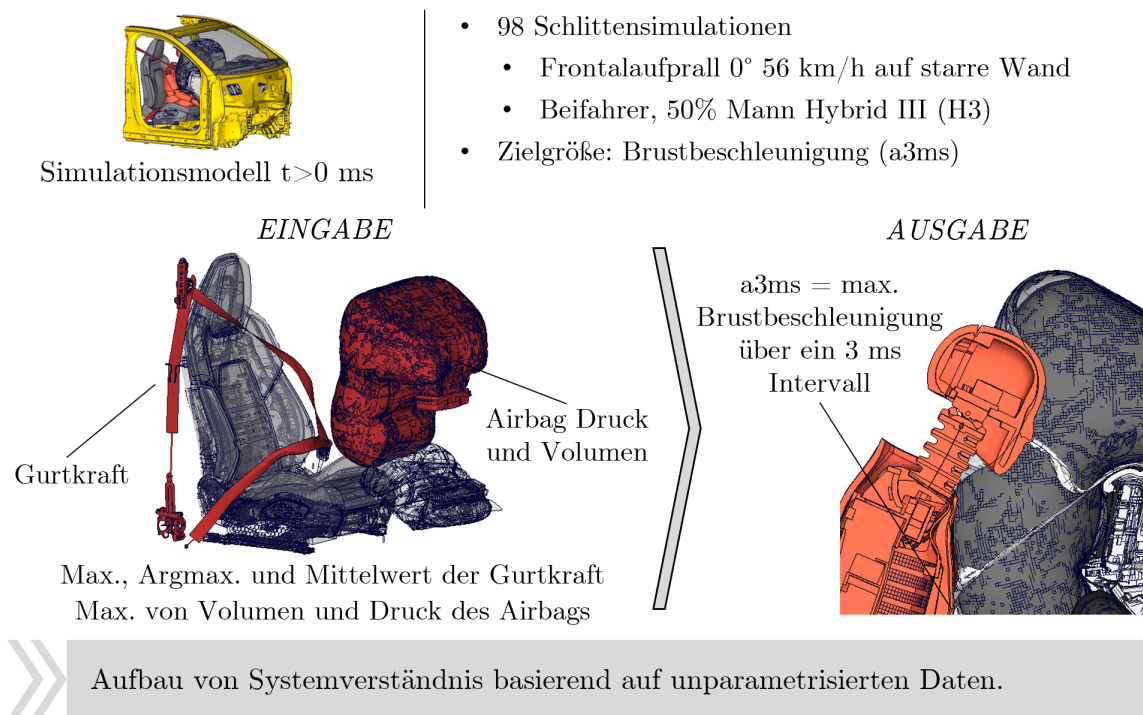
In Abbildung 6.41 a) ist der Fortschritt der Optimierung von SwM gezeigt. Die Optimierung läuft 30,000 Schritte und das finale Ergebnis weist eine Reduktion von Sitzbreitenminderung und Masse um 8.3 % bzw. 1.5 % auf. Ein Schritt entspricht dabei einem Aufruf des Metamodells. Welche Rohbauwandstärken Unterschiede zwischen der initialen und optimalen Lösung verursachen, kann nun mit dem in Gleichung (5.15) auf S. 115 eingeführten Differenz-SHAP beantwortet werden. Die Differenz-SHAP-Werte sind in Abbildung 6.41 b) als Wasserfalldiagramm abgebildet. Die Beiträge werden auf die Differenz der beiden Konfigurationen  $f(X_{\text{Optimiert}})$  und  $f(X_{\text{Referenz}})$  im Lösungsraum aufgeteilt. Hauptsächlich sind Parameteränderungen bei T4003, T4607 und T4605 entscheidend für die Optimalität der Lösung.

Wie anhand des Beispiels verdeutlicht, können Unterschiede in relevanten Lösungen mittels ExPLAIN einfach und intuitiv erklärt werden. Eine analoge Darstellung kann mit anderen lokalen Sensitivitätsanalysen oder modellagnostischen ML-Techniken aufgrund der erforderlichen Additivität im gesamten Designraum nicht erreicht werden.

### 6.4.2 Use Case II: Einflussgrößen auf die Insassenbelastungen

Für den zweiten Use Case werden – wie bereits in der Evaluierung von *DrAGOS* und *DAYSi* – Insassensimulationen eingesetzt, um die Anwendbarkeit im Bereich der Rückhalte-systemauslegung zu verdeutlichen. Verwendet wird das FE-Modell eines eines FN-56 gemäß FMVSS 208 mit dem H3 auf der Beifahrerposition. Der Datensatz umfasst 98 Crashsimu-

lationen, bestehend aus Varianten, die sowohl in einer Optimierungsstudie systematisch generiert als auch manuell spezifiziert wurden. Damit entsprechen sie typisch verfügbaren Daten im Entwicklungsprozess von Rückhaltesystemen. Gemäß Abbildung 6.42, der ein Überblick entnommen werden kann, sind Änderungen vor allem an zwei Komponenten vorhanden. Diese haben einen großen Einfluss auf die ATD-Belastung.

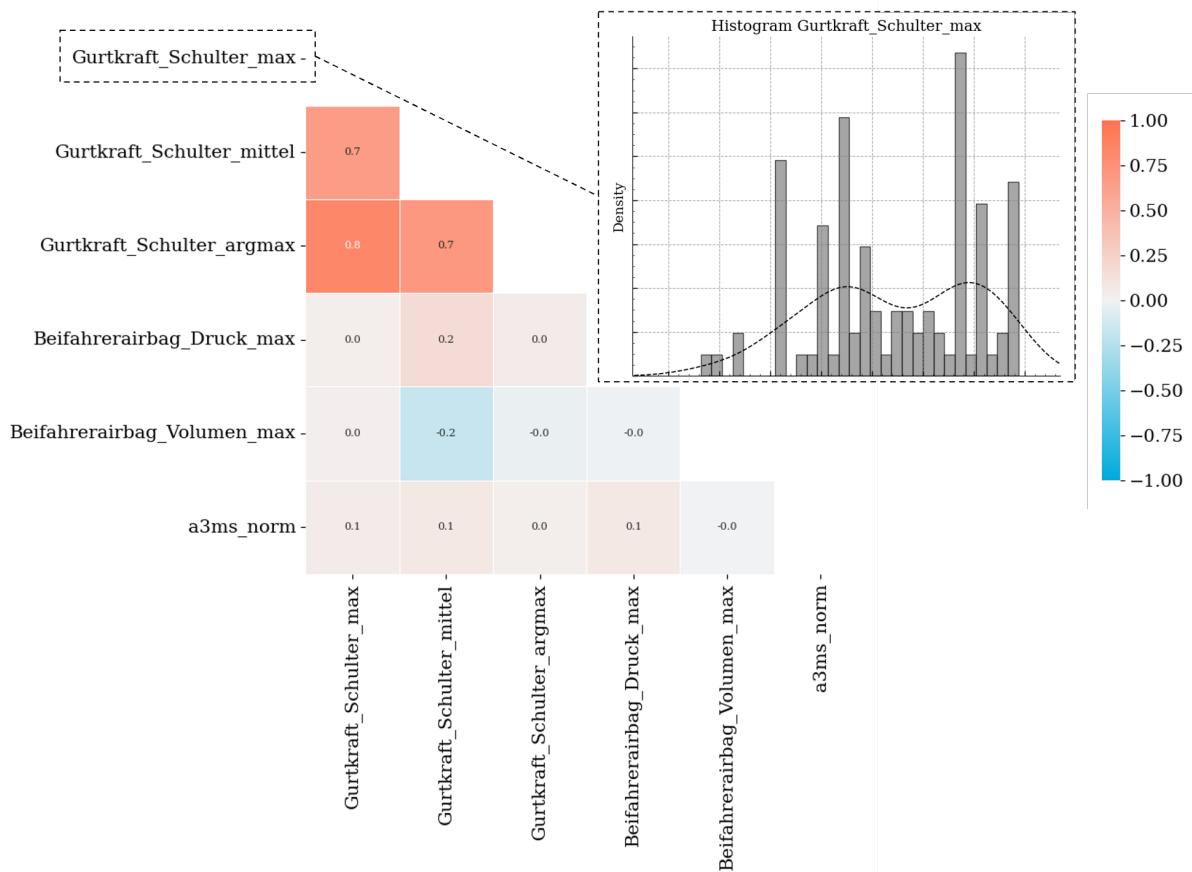


**Abbildung 6.42:** ExPLAIN Use Case II: Problemstellung

Die Einflussgrößen auf Insassenbelastungen – hier auf die  $a_{3ms}$  – werden nun mittels extrahierten *Features* von Sensorsignalen, die das Verhalten des Rückhaltesystems charakterisieren, beschrieben. Hierbei beschreiben Maximalkraft und Mittelwert der Schnittkraft des Gurts oberhalb der Schulter das allgemeine Rückhalteniveau des ATDs. Das Argument des Maximalwerts (argmax) der Gurtkraft lässt Rückschlüsse auf die Umschaltzeit zwischen den beiden Kraftniveaus des Gurts zu. Für den Beifahrerairbag werden das maximale Volumen und der maximale Druck extrahiert, die damit Größe und „Prallheit“ abbilden. Folglich wird hier ein (o-o)-Zusammenhang betrachtet. Die  $a_{3ms}$  wird hier als Zielgröße gewählt, da sie eine gesetztesrelevante Größe darstellt und die globale Rückhaltung des ATDs gut beschreibt. Aus Gründen der Vertraulichkeit ist die  $a_{3ms}$  daher auf ein Intervall von 0 bis 100 skaliert. Ziel der Betrachtung ist es, die grundlegenden Einflüsse auf den Zielwert darzulegen.

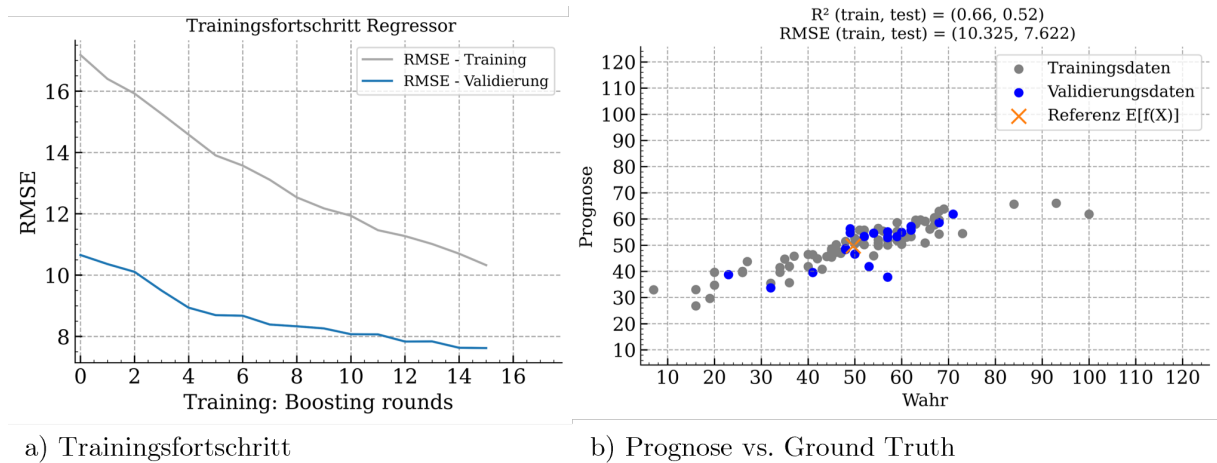
Durch die Extraktion der Features zu einem Zeitpunkt  $t > 0$  ms sind die Eingaben nicht mehr gleichverteilt. Zudem finden sich Korrelationen zwischen den Eingaben, die in

Abbildung 6.43 anhand einer Pearson Korrelationsmatrix aufgezeigt sind. Die Korrelation zwischen Maximum und dessen Argument sind mit 0.8 deutlich höher als Korrelationen, die zwischen den Eingaben und Ausgaben generell auftreten. Die Verteilung der Gurtkraft an der Schulter zeigt beispielhaft die ungleichmäßige Verteilung, die hier zwei Cluster ausbildet. Entsprechend ist zu erwarten, dass die Zusammenhänge schwerer mit einem ML-Algorithmus zu lernen sind als im vorangegangenen Use Case.



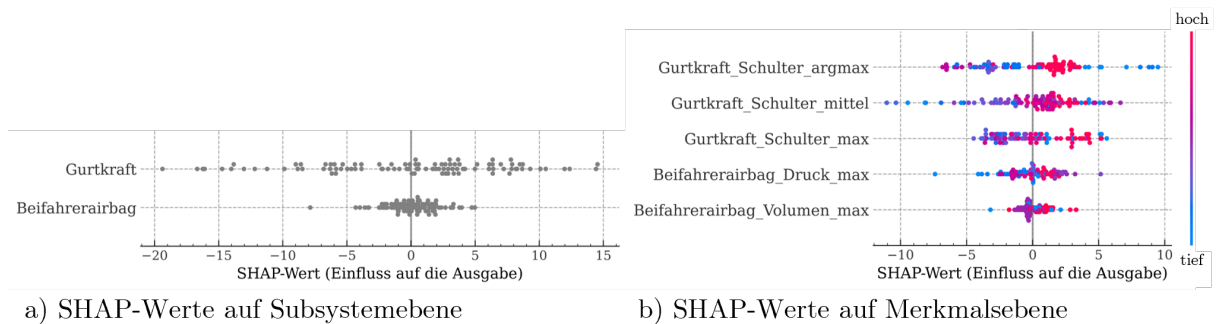
**Abbildung 6.43:** *ExPLAIN* Use Case II: Abhängigkeiten und Verteilung der Daten

Der Trainingsfortschritt und die Güte im ersten Fold für das XGBoost-Modell sind in Abbildung 6.44 gezeigt. Der  $R^2$  ist mit 0.52 in Bezug auf die ungesehenen Validierungsdaten deutlich niedriger als im vorangegangenen Use Case, obwohl eine größere Anzahl Datenpunkte (98) gemessen an den Eingaben (5) vorhanden ist. Deutliche Abweichung ist insbesondere bei hoher normierter Brustbeschleunigung von  $> 80\%$  zu sehen. Die Fehlerwerte in der 5-fold Cross Validation schwanken zudem auch deutlicher als bisher. Während für den  $R^2$  der Median bei 0.40 liegt, existiert zudem ein Ausreißer bei 0.15.



**Abbildung 6.44:** *ExPLAIN* Use Case II: Trainingsfortschritt und Güte des Metamodells in Fold 1

Diese Abhängigkeiten sind entsprechend dem Schaubild von *ExPLAIN* nach Abbildung 5.9 auf S. 108 auch bei der Berechnung der SHAP-Werte zu berücksichtigen. Daher wird hier CorrSHAP [AJL21] mit der empirischen Samplingstrategie verwendet, da die Eingabedaten nach Abbildung 6.43 weder normalverteilt sind, noch durch eine gaußsche Kopula modelliert werden können. Die Samplingstrategien legen die Teilmengen fest, die für die Bestimmung der lokalen Merkmal-Einflüsse herangezogen werden, da die Annahme einer Gleichverteilung bei abhängigen Merkmalen nicht mehr gültig ist. Die Robustheit dieser SHAP-Berechnung sowie Unterschiede zwischen den Samplingstrategien werden im Rahmen der Analyse der Verfahrensparameter später noch vertieft.

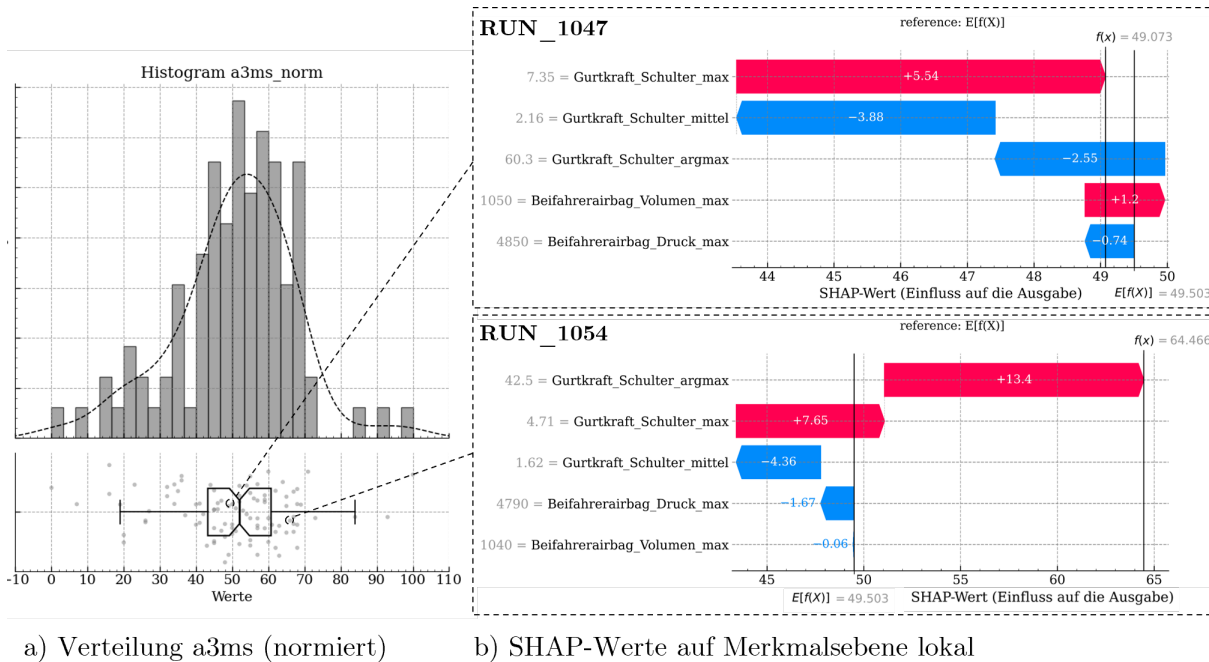


**Abbildung 6.45:** *ExPLAIN* Use Case II: SHAP-Werte auf Subsystem- und Merkmalsebene global

Ein allgemeines Verständnis der betrachteten Problemstellung kann mittels Abbildung 6.45 gewonnen werden. Erwartungsgemäß hat der Gurt – beschrieben über die Gurtkräfte – einen höheren Einfluss auf die Brustbeschleunigung, als der Airbag. Der Airbag kann jedoch ebenfalls einen maßgeblichen Einfluss haben. Dies kann damit begründet werden,

dass ein weicherer Airbag zu einer stärkeren Vorverlagerung des ATDs führt, was mit einer erhöhten Belastung der Brust einhergehen kann. Auf Merkmalsebene zeigen die SHAP-Werte ähnliche Streuungsbänder für die Gurtschnittkräfte, wobei hier das Argument des Maximalwerts den höchsten mittleren absoluten Beitrag leistet. Dies deutet darauf hin, dass der Umschaltzeitpunkt einen etwas größeren Einfluss hat als die Rückhaltekraftniveaus selbst. Trotz vergleichsweise geringer Prognosegüte versteht das Modell grundlegend den (o-o)-Zusammenhang in den Daten.

Aus Ingenieursicht sind neben den globalen auch die lokalen Einflüsse interessant, da sie bei der Analyse einzelner Simulationen helfen können. Zudem kann der Datensatz im Verlauf des Entwicklungsprozesses stetig erweitert werden. Durch wiederholte Ausführung von *ExPLAIN* können dadurch Erklärungen für neue Varianten generiert werden. Beispiele für zwei lokale Erklärungen sind in Abbildung 6.46 für Simulationen mit einem niedrigeren und einem höheren a3ms-Wert gezeigt.



**Abbildung 6.46:** *ExPLAIN* Use Case II: SHAP-Werte auf Merkmalsebene lokal

Der Referenzwert ist entsprechend [LL17] als erwartete Ausgabe des Metamodells  $\mathbb{E}[f(x)]$  definiert. Die Variante mit dem niedrigeren a3ms-Wert RUN\_1047 weist hohe konträrgerichtete Beiträge von mittlerer Gurtkraft und maximaler Gurtkraft auf, die sich gegenseitig kompensieren. Das Argument des Maximalwertes der Gurtkraft wird hier lokal nicht als wichtig erachtet, obwohl er allgemein in Abbildung 6.45 auf S. 171 als relevant identifiziert wird. Der Airbag hat einen erhöhenden Einfluss auf die Brustbeschleunigung. In der zweiten Variante mit dem höheren a3ms-Wert RUN\_1054 verursacht ein niedriger

Wert des Arguments des Maximalwertes der Gurtkraft eine hohe Brustbeschleunigung. Der Maximalwert der Gurtkraft wirkt ebenso in die positive Richtung. Mittelwert der Gurtkraft sowie Druck und Volumen des Airbags wirken hingegen in die negative Richtung. In beiden Simulationen spielen die Features des Airbags eine untergeordnete Rolle. Das ist im Einklang mit der globalen Sensitivität in Abbildung 6.45 auf S. 171.

Dieser Use Case verdeutlicht, dass durch die modulare Datenverarbeitung von *ExPLAIN* Simulationsdatensätze auf spezifische Zusammenhänge hin untersucht werden können, die bisher nicht realisierte und intuitive Einblicke ermöglichen können. Auch wenn keine direkte Anpassung der Parameter möglich ist, da es sich um extrahierte Features handelt, können interessante Komponenten und Systeme identifiziert werden. Inwiefern Schlüsse aus den Daten gezogen werden dürfen, ist stets durch die Einbindung von Experten zu hinterfragen. Perspektivisch können mit diesem Ansatz Airbags unterschiedlicher Form und verschiedene Gurtaufroller verarbeitet und verglichen werden, da keine Parametrisierung erforderlich ist.

### 6.4.3 Use Case III: Erklärung von Einbettungsräumen

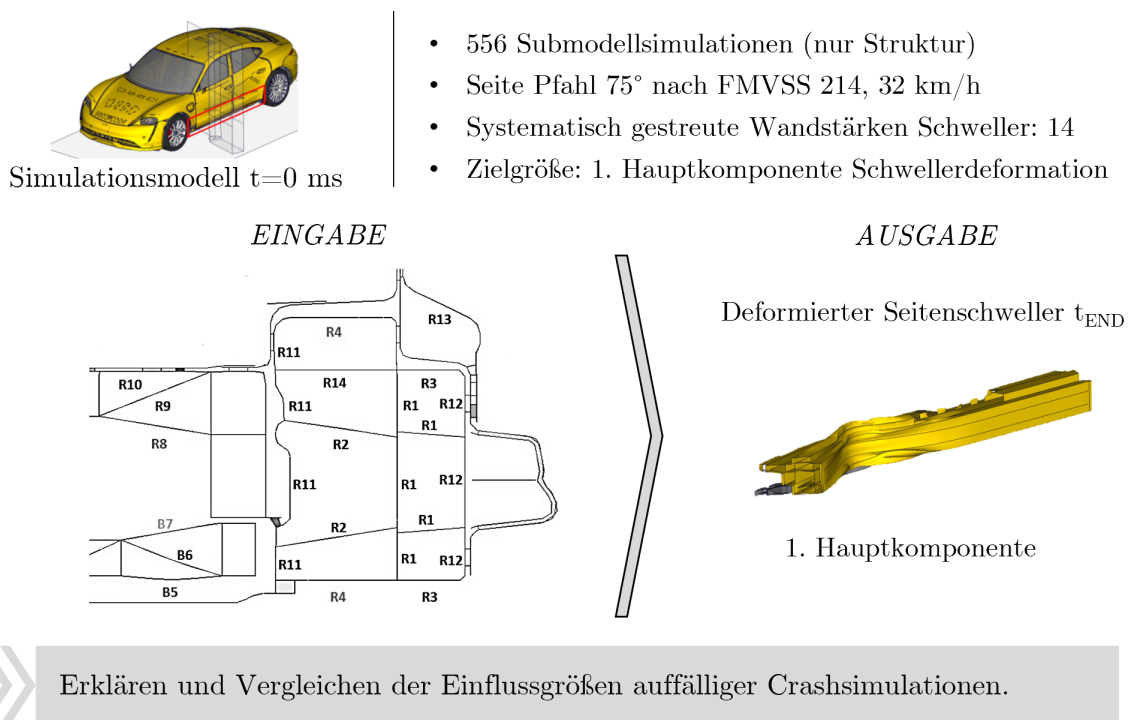


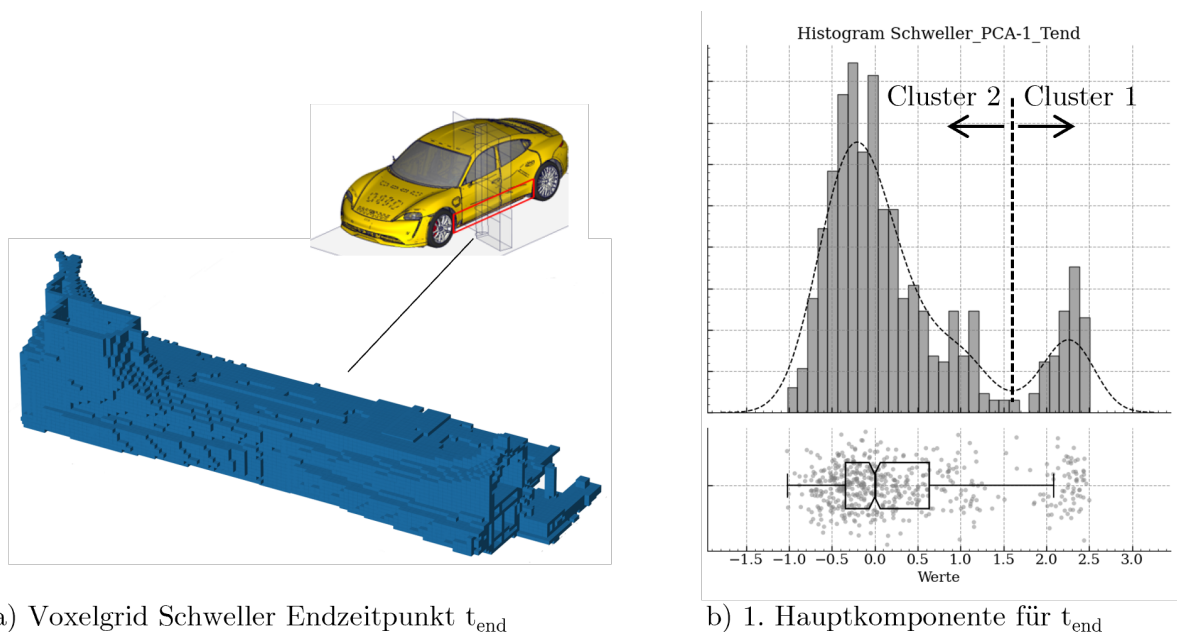
Abbildung 6.47: *ExPLAIN* Use Case III: Problemstellung

Im dritten Use Case wird ein Schweller-Submodell für einen Pfahl-Seitenaufprall nach FMVSS 214 im Kontext der Strukturauslegung analysiert. Ein Überblick ist in Abbildung 6.47 gegeben. Der Pfahl ist im Vergleich zu den  $90^\circ$  im ersten Szenario um  $75^\circ$

gedreht. Auch hier ist der Pfahl ortsfest und das Fahrzeug prallt auf diesen mit einer Geschwindigkeit von 32 km/h auf.

Der Seitenschweller nimmt in diesem Lastfall einen Großteil der kinetischen Energie des Fahrzeugs durch plastische Verformung auf. Aufgrund des hohen Rechenaufwands für die Berechnung einer vollständigen Fahrzeug-Crashsimulation wird hier ein Submodell verwendet. Dieses FE-Modell enthält, wie in Abbildung 6.47 gezeigt, nur den Schweller. Dieser wird über Punktmassen, Trägheiten und Fahrzeugbewegung an den Pfahl gedrückt. Die Rechenzeit des FE-Submodells ist 400-mal schneller als die der Gesamtfahrzeugsimulation. Letztere dauert auf Hochleistungsrechnerclustern zwischen einem und zwei Tagen. FE-Submodelle werden folglich vor allem in der Optimierung eingesetzt, da hier viele Varianten benötigt werden. Wichtig ist die Sicherstellung einer hinreichenden Übereinstimmung mit den Ergebnissen des ursprünglichen FE-Modells. Auch diese Daten wurden im Rahmen einer Optimierungskampagne generiert und unterscheiden sich in 14 Wandstärken im Seitenschweller. Der Querschnitt des Schwellers ist in Abbildung 6.47 gezeigt. Die Wandstärken wurden systematisch mit dem Latin Hypercube Sampling in 556 Varianten gestreut.

### 6.4.3.1 Analyse der PCA-Einbettung

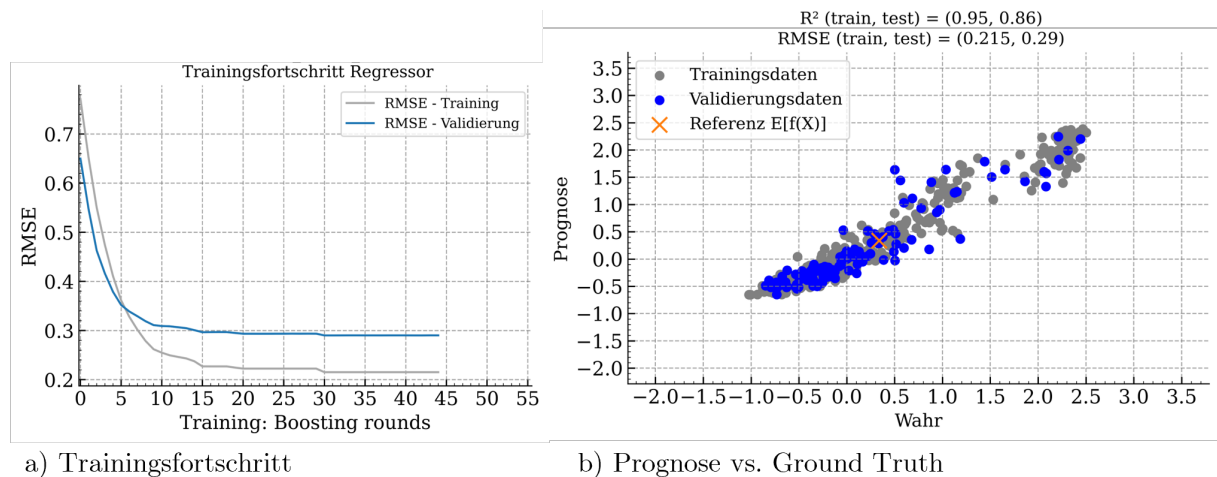


**Abbildung 6.48:** *ExPLAIN* Use Case III: Voxelgrid Seitenschweller und die erste Hauptkomponente der PCA

Analog zum ersten Use Case wird hier ebenfalls ein (i-o)-Zusammenhang betrachtet. Eingaben sind dabei die 14 Wandstärken des Seitenschweller und Ausgabe ist die erste Hauptkomponente der verformten Schwellergeometrie im letzten Zeitschritt der Simulation.

Abbildung 6.48 zeigt das Voxelgrid einer Simulation sowie das Histogramm der ersten Hauptkomponente über alle 556 Simulationen. Durch diese werden 16.5% der Varianz in den ursprünglichen Daten erklärt. Auffällig ist, dass sich zwei Cluster bilden. Entsprechend ist davon auszugehen, dass die Deformation des Schwellers in zwei deutlich unterschiedliche Gruppen eingeteilt werden kann. Allein durch die Dimensionsreduktion der FE-Daten können folglich interessante Einblicke in das Systemverhalten ermöglicht werden. Diese Erkenntnisse bestätigen damit vorhandene Literatur [IG19; Kra+23]. Ein bereits festgestelltes Defizit ist dort jedoch die Intransparenz, da Einflussgrößen auf die Einbettung nicht direkt ersichtlich sind. Mittels *ExPLAIN* kann nun ein (i-o)-Zusammenhang zwischen Wandstärken und der ersten Hauptkomponente mit einem Metamodell abgebildet und analysiert werden.

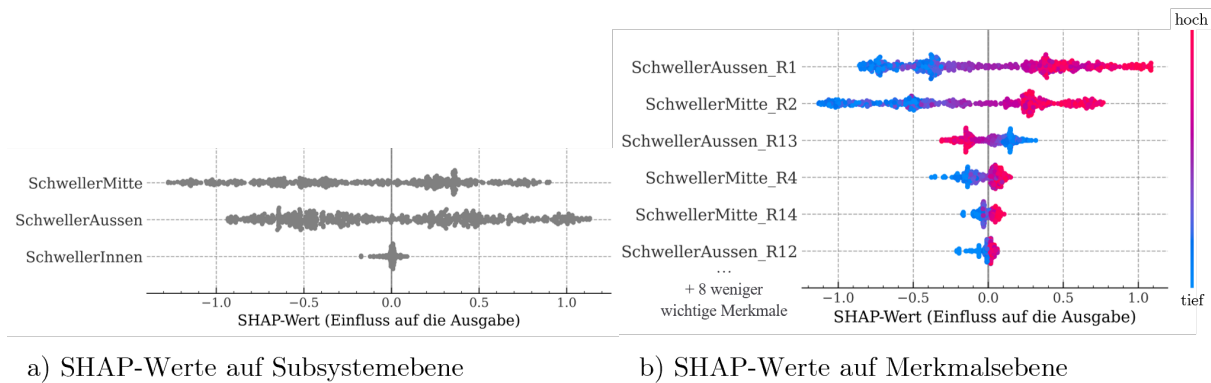
Abbildung 6.49 zeigt die Fehlerkonvergenz während dem Training und visualisiert die Güte des trainierten Metamodells. Erreicht wird im ersten Fold ein Bestimmtheitsmaß  $R^2$  von 0.95 im Training und 0.86 in der Validierung. Das zeigt, dass die Zusammenhänge erfolgreich gelernt werden können. Die 5-fold Cross Validation zeigt, dass Ergebnisse reproduzierbar sind. Die Güte in Bezug auf die ungesehenen Validierungsdaten verteilt sich zwischen 0.75 und 0.89, wobei der Median bei 0.86 liegt.



**Abbildung 6.49:** *ExPLAIN* Use Case III: Trainingsfortschritt und Güte des Metamodells in Fold 1

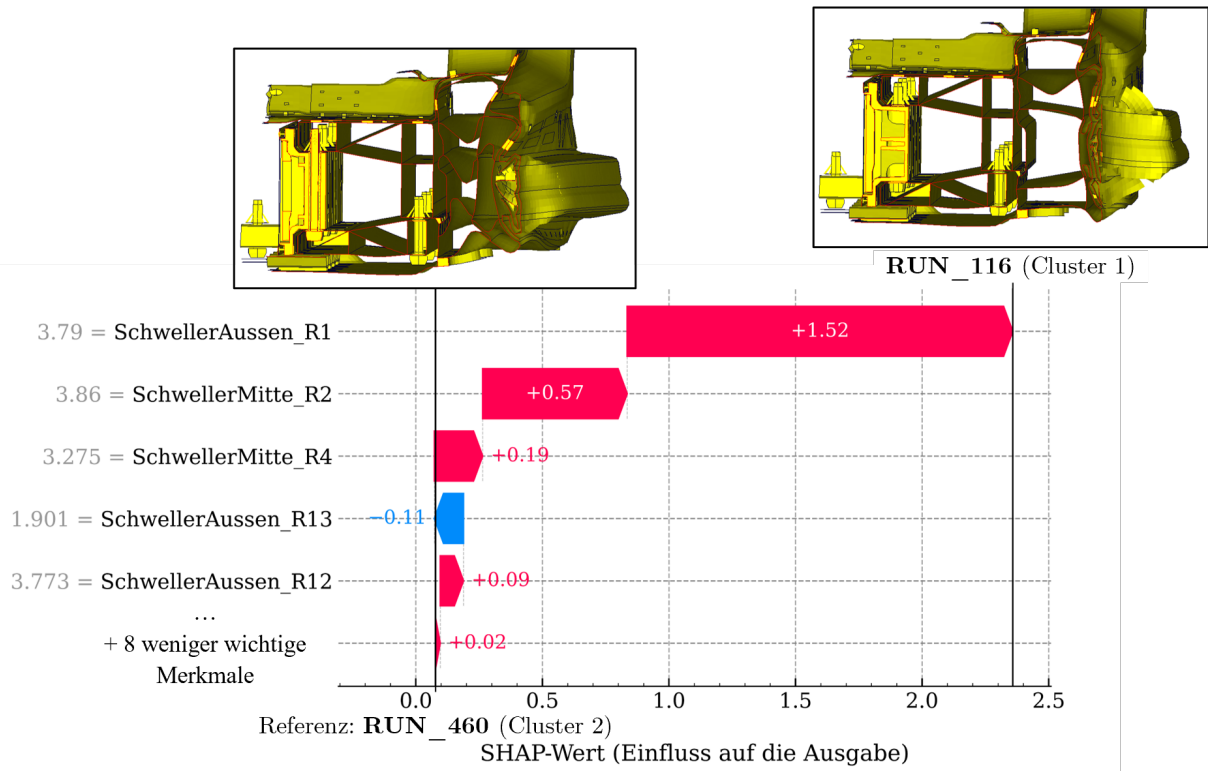
Einblicke in das Systemverhalten, das durch das Metamodell abgebildet wird, kann nun mittels SHAP hergestellt werden. Ein Überblick auf Subsystem- und Merkmalsebene ist in Abbildung 6.50 gezeigt. Auf der Subsystemebene weist vor allem der mittlere Bereich des Schwellers eine hohe Relevanz auf. Hierzu zählen R2, R4, R11 und R14. Der innere Bereich des Schwellers wird aufgrund der Simulationszeit lediglich von 10 ms nicht getroffen und hat daher einen vernachlässigbaren Einfluss. Auf der Merkmalsebene überlappen sich die

Einflüsse deutlich weniger als in den beiden vorangegangenen Use Cases. Maßgeblich wird das Verhalten, das durch die erste Hauptkomponente beschrieben wird, durch R1 und R2 beeinflusst. Diese beiden Wandstärken bilden die horizontalen Querstreben im Querschnitt und damit maßgeblich für die Deformationseigenschaften verantwortlich. Die Identifikation dieser Bestandteile ist aus Ingenieurssicht schlüssig und bestärkt die Sinnhaftigkeit der dimensionsreduzierten Repräsentation.



**Abbildung 6.50:** *ExPLAIN* Use Case III: SHAP-Werte auf Subsystem und Merkmalsebene global

Weiter vertieft werden kann dies durch die Verwendung von Differenz-SHAP. Zwei repräsentative Simulationen – RUN\_116 von Cluster 1 und RUN\_460 von Cluster 2 – können entsprechend einfach miteinander verglichen werden. Dies ist in Abbildung 6.51 gezeigt und kann dafür genutzt werden Unterscheide zwischen den Verhaltensweisen, die das jeweilige Cluster gruppiert, herauszustellen und zu begründen. Bei Betrachtung der deformierten Schwellerquerschnitte von RUN\_460 und RUN\_116 lässt sich feststellen, dass für Simulationen aus Cluster 2 die horizontalen Wände falten, was eine stärkere Intrusion des Schwellers zur Folge hat. Für Simulationen aus Cluster 1 hingegen ist das nicht der Fall und die horizontalen Wände werden kaum deformiert. Dies ist auch in den lokalen Beiträgen, die die Unterschiede in den beiden Simulationen repräsentieren, ersichtlich. Den auch global besonders relevanten Wandstärken R1 und R2 wird auch hier ein hoher Einfluss zugeordnet. Der Einfluss von R1 ist hier jedoch mehr als doppelt so groß wie der von R2. Alle übrigen Wandstärken leisten nur einen marginalen Beitrag. Dementsprechend hebt Abbildung 6.51 auch wieder den Vorteil lokaler Darstellung in den Vordergrund. Gerade in Bezug auf die Wichtigkeit einzelner Wandstärken – hier insbesondere R1 – können leichte Unterschiede zur globalen Sensitivität festgestellt werden.



**Abbildung 6.51:** *ExPLAIN* Use Case III: Differenz-SHAP zwischen Referenz und einem Ausreißer

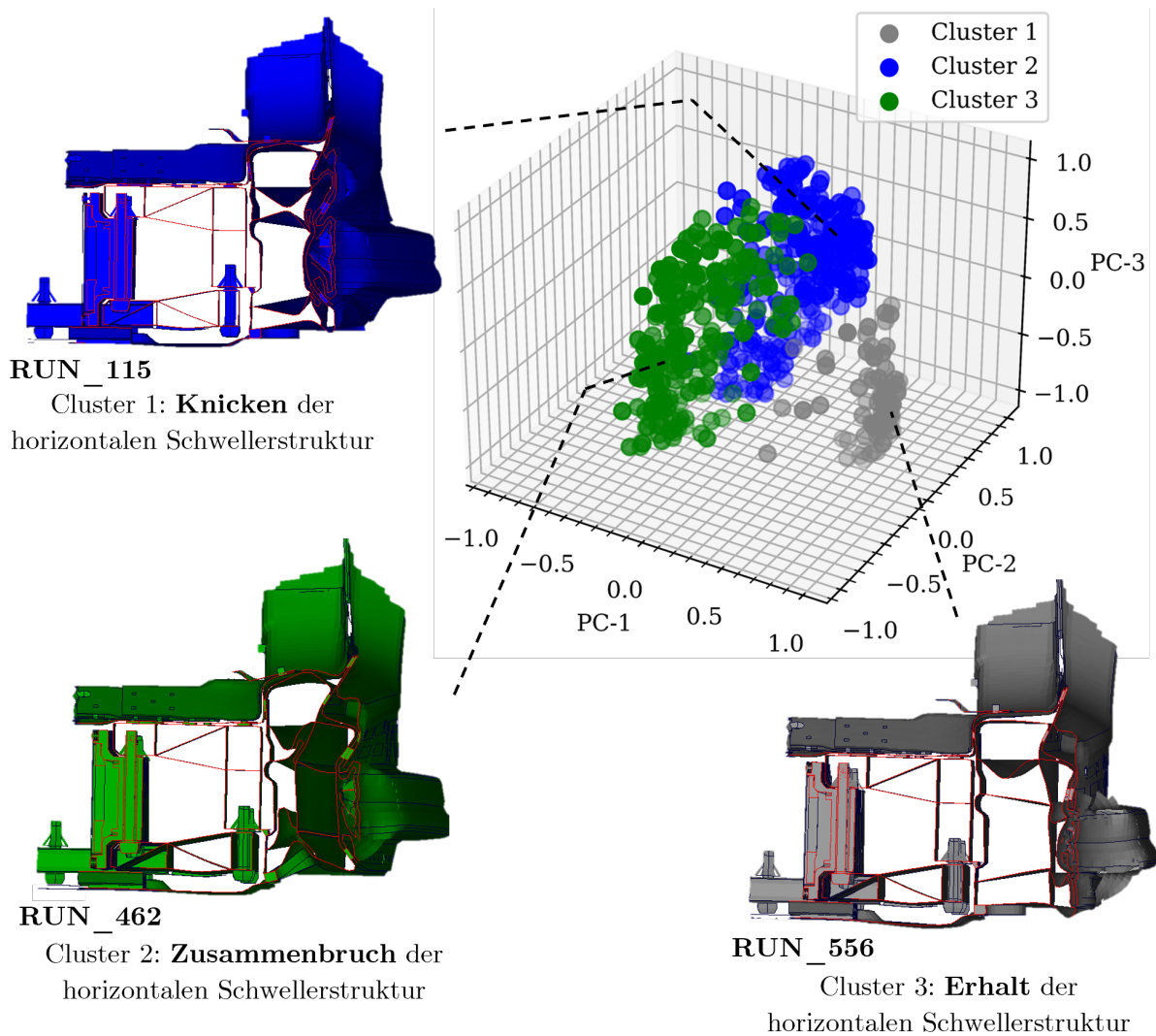
#### 6.4.3.2 Analyse von Clusterzugehörigkeiten

Neben der Betrachtung einzelner Komponenten des Einbettungsraums können auch Clusterzugehörigkeiten näher analysiert werden. Dies ist besonders interessant, wenn mehrdimensionale Einbettungsräume in zwei oder drei Dimensionen betrachtet werden. Während durch die erste Hauptkomponente 16.5% der Varianz in den Daten von Use Case III erklärt wird, bilden die ersten drei Komponenten zusammen 37.4% ab. Der Informationsgehalt ist hier entsprechend höher. Mit *ExPLAIN* kann die nähere Analyse eines solchen Einbettungsraums durch die Verwendung eines Klassifikationsalgorithmus realisiert werden. Der (i-o)-Zusammenhang in Use Case III kann dann zwischen Clusterzugehörigkeiten und den Wandstärken analysiert werden.

Im ersten Schritt werden dazu die Clusterzugehörigkeiten im Einbettungsraum der ersten drei Hauptkomponenten durch ein k-Means Clustering bestimmt. Die Anzahl drei entstammt der Optimierung der Silhouette Score [Rou87] für die so ein Wert von 0.40 erzielt wird. Das Vorgehen zur Identifikation der Anzahl von Clustern ist damit analog zu *DAYSi*. Die dimensionsreduzierte Darstellung des voxelisierten Seitenschweller im letzten Zeitschritt inklusive der Clusterzugehörigkeiten ist in Abbildung 6.52 gezeigt. Im dreidimensionalen Raum wird nun das Schwellerverhalten feingranularer aufgelöst, als das für die erste Hauptkomponente in Abbildung 6.48 auf S. 174 der Fall war. Wäh-

rend Cluster 1 weiterhin die Simulationen beinhaltet, bei denen die horizontale Struktur nicht versagt, wird nun genauer zwischen einem vollständigen Zusammenbruch wie in RUN\_462 und einem Knicken der Struktur wie in RUN\_115 differenziert. Das liefert, wie in Abbildung 6.52 dargestellt, zusätzliche Einblicke.

Entsprechend dem Ablauf auf *ExPLAIN* wird nun im Folgenden das Metamodell – hier ein XGBoost Klassifikator – zum Abbilden der Zusammenhänge trainiert.



**Abbildung 6.52:** *ExPLAIN* Use Case III: Cluster im 3D-Einbettungsraum der PCA

Der Trainingsfortschritt und die Fehlermetriken des trainierten Modells sind in Abbildung 6.53 dargestellt. Da es sich um ein Klassifikationsproblem handelt, wird hier eine logarithmische Verlustfunktion verwendet (vgl. Gleichung (2.6) auf S. 19). Für die ungesehenen Validierungsdaten beträgt die Accuracy 0.87, die Precision 0.90, der Recall 0.87 und die F1-Score 0.86 für den ersten Fold. Die Accuracy misst in diesem Kontext den Anteil der korrekt vorhergesagten Clusterzugehörigkeiten pro Instanz im Verhältnis zur Gesamtzahl

der Cluster. Die etwas geringeren Werte in der Validierungsphase deuten darauf hin, dass das Modell hier größere Herausforderungen hat, alle Clusterzugehörigkeiten präzise zu erkennen. Der Rückgang des Recall im Vergleich zur Precision in der Validierung zeigt, dass das Modell eher dazu neigt, relevante Clusterzugehörigkeiten zu übersehen, anstatt falsch positive Vorhersagen zu erzeugen. Der F1-Score fasst diese beiden Aspekte zusammen und bestätigt eine insgesamt hohe Modellgüte mit einem leichtem Overfitting.

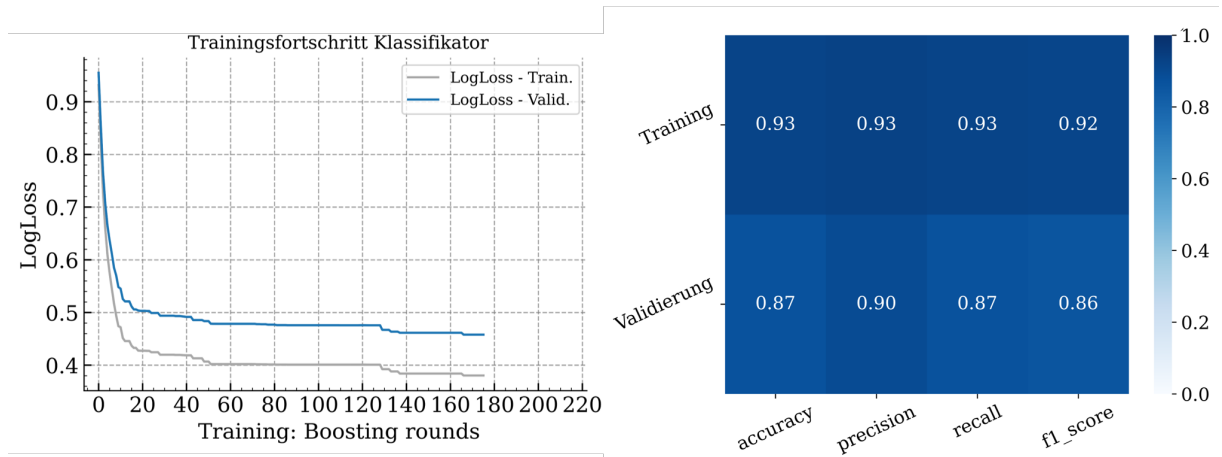
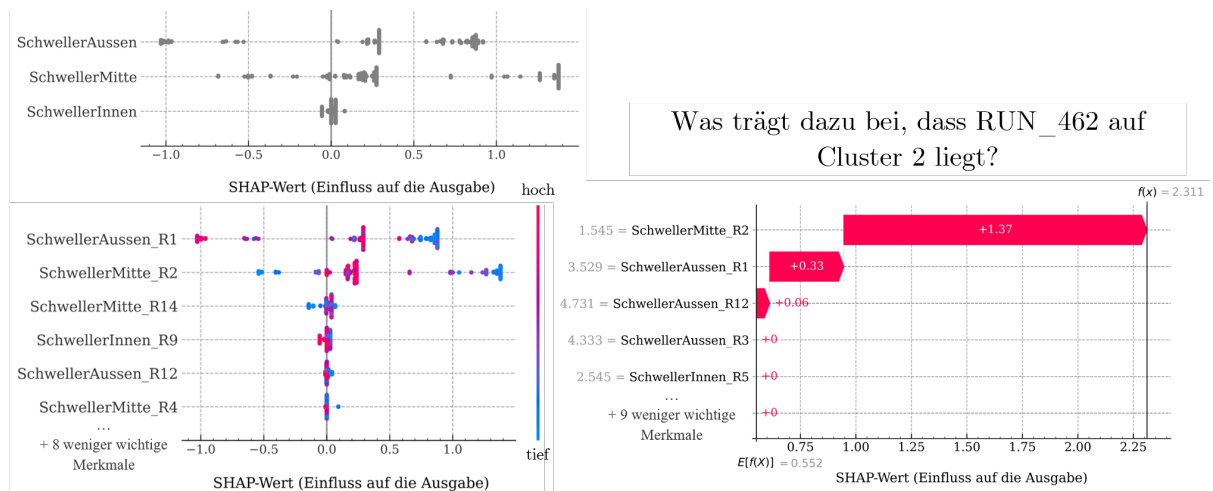


Abbildung 6.53: *ExPLAIN* Use Case III: Trainingsfortschritt und Güte Metamodell in Fold 1



a) SHAP-Werte auf Subsystem-/Merkmalsebene    b) Lokale Beiträge Clusterzugehörigkeit

Abbildung 6.54: *ExPLAIN* Use Case III: Globale Sensitivität auf Subsystem- und Merkmalsebene und lokale Erklärung

Die SHAP-Analyse im Rahmen von *ExPLAIN* erlaubt nun Einflüsse zu identifizieren, die zu der jeweiligen Clusterzugehörigkeit beitragen. In Abbildung 6.54 ist jedoch die Logit-Skala zu beachten, in der die SHAP-Werte berechnet werden. Diese repräsentiert

die unskalierten, kontinuierlichen Vorhersagewerte des Metamodells für jede Klasse, aus denen die Wahrscheinlichkeitsverteilung über die Klassen durch Anwendung der Softmax-Funktion abgeleitet wird. Ein höherer Logitwert für eine Klasse geht mit einer höheren vorhergesagten Wahrscheinlichkeit für diese Klasse relativ zu den anderen Klassen einher. Eine einfache Umrechnung in Wahrscheinlichkeiten ist hier nicht möglich, da die Additivität nicht automatisch sicher gestellt ist.

Die Ergebnisse für das wahrscheinlichste Cluster pro Simulation ist Abbildung 6.54 a) zu entnehmen und erlaubt Rückschlüsse auf die globale Sensitivität. Maßgeblich für die Clusterzuordnung sind die Wandstärken R1 und R2. Das bestätigt die Betrachtungen mit dem Regressionsmodell anhand der ersten Hauptkomponente in Abbildung 6.49 auf S. 175.

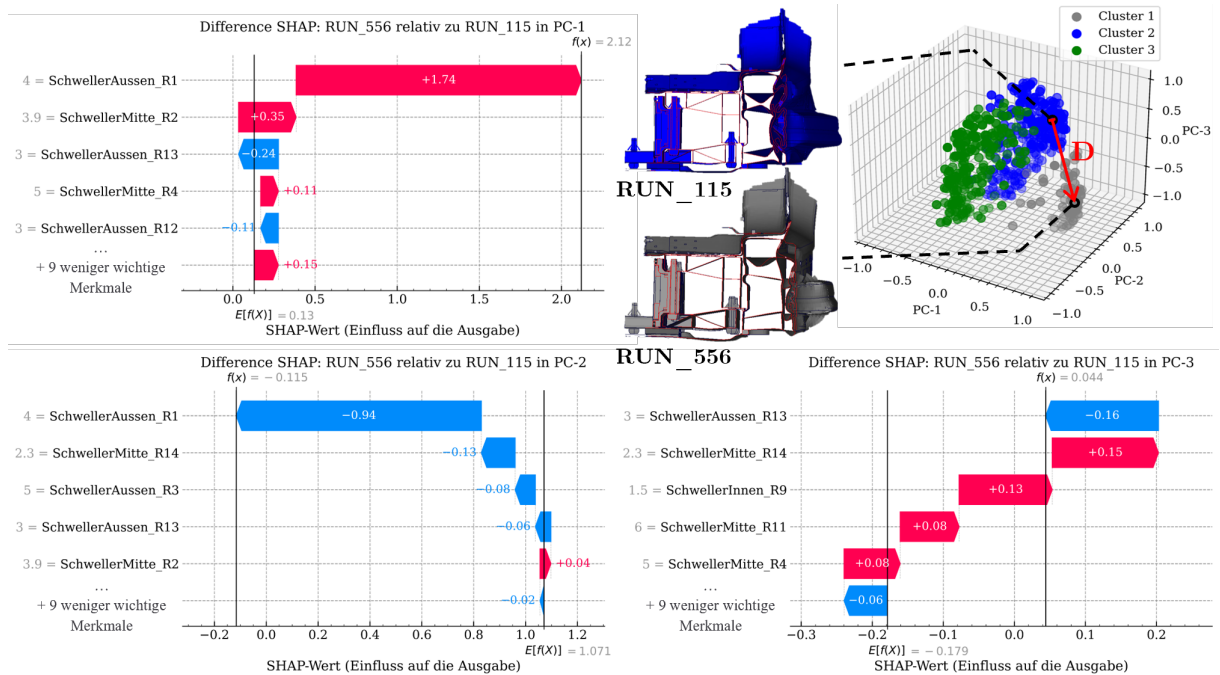
Warum RUN\_531 gerade dem dritten Cluster zugeordnet wird, ist in Abbildung 6.54 b) gezeigt. Maßgeblich ist hier R2. Der Einfluss der übrigen Wandstärken ist nahezu vernachlässigbar.

Folglich können mit *ExPLAIN* auch mehrdimensionale Einbettungsräume über Clusterzugehörigkeiten einfach analysiert werden. Zudem sind die Ergebnisse konsistent mit den Analysen in einer Dimension.

### 6.4.3.3 Differenz-SHAP im 3D-Einbettungsraum

Das eingeführte Konzept Differenz-SHAP ermöglicht einen intuitiven Vergleich zweier Simulationen. Dies wurde bereits anhand des Vergleichs der initialen und optimalen Systemkonfiguration in Abbildung 6.41 auf S. 168 sowie anhand eines Ausreißers zu einer weniger auffälligen Variante in Abbildung 6.51 auf S. 177 verdeutlicht. Diese Beispiele beschränkten sich bisher jedoch ausschließlich auf einen eindimensionalen Ausgaberaum. Nun wird anhand des 3D-Einbettungsraums für den voxelisierten Seitenschweller aus Use Case III untersucht, inwiefern ein Vergleich auch in diesem Kontext realisiert werden kann.

Abbildung 6.55 zeigt den Vergleich von RUN\_115 und RUN\_556 im 3D-Einbettungsraum. Die beiden Simulationen sind dem Cluster 1 und Cluster 2 zugehörig. Die Abstände können nun komponentenweise erklärt werden. Die größten Unterschiede von 1.99, sind in der ersten Hauptkomponente zu finden und werden vor allem durch R1 in positiver Richtung verursacht. Bezogen auf die zweite Hauptkomponente ist ebenfalls R1 maßgebend. Die geringste Differenz von 0.223 ist in der dritten Hauptkomponente festzustellen. Unterschiede werden hier vor allem auf R13, R14 und R9 zurückgeführt. In diesem Schema können beliebige Datenpunkte entsprechend auch im 3D-Einbettungsraum verglichen werden. Ein Vergleich über absolute Abstände, beispielsweise mit dem euklidischen Distanzmaß, ist hier nicht trivial möglich, da eine beitragsweise Aufteilung nötig wäre.



**Abbildung 6.55:** *ExPLAIN* Use Case III: Vergleich Simulationen im 3D-Einbettungsraum

Anhand der betrachteten Use Cases generiert *ExPLAIN* neue Einblicke, um vorliegende Daten besser zu verstehen. Spezifische Handlungsempfehlungen, welche Änderungen am System vorzunehmen sind, um ein bestimmtes Verhalten zu erreichen, wie das beispielsweise bei *DrAGOS* der Fall ist, werden hier nicht näher betrachtet. Die Entscheidungsfindung, auch unter direkter Einbindung der Ingenieure, um die Systeme auf Basis der Analyseergebnisse zu verbessern kann Teil von zukünftigen Arbeiten sein.

#### 6.4.4 Analyse Verfahrensparameter

Die in Abschnitt 5.4 auf S. 107 eingeführten Verfahrensparameter sollen nun auch hier weiter untersucht werden. Im Vorangegangenen wurde bereits der Mehrwert durch neue Einblicke in Simulationsdaten von *ExPLAIN* mit demselben Standardverfahrensparametersatz anhand von drei unterschiedlichen Use Cases ausführlich dargelegt.

##### 6.4.4.1 Vergleich Metamodelle

Ähnlich wie bereits für *DrAGOS* (vgl. Tabelle 6.1 auf S. 130) werden auch für *ExPLAIN* verschiedene Metamodelle verglichen. Verwendet werden dabei TabPFN [Hol+23; Hol+25], XGBoost [CG16], Support Vector Regression [Dru+96], Multilayer Perceptron<sup>2</sup> [RHW86], Decision Trees [Bre+84] und Lasso Regression [Tib96]. Die mittleren Fehlerkennwerte  $R^2$  und RMSE aus einer 5-fold Cross Validation gegenüber ungesehenen Daten sind Tabelle 6.2

<sup>2</sup>Die Architektur wird im Anhang in Tabelle A.11 auf S. 273 genau erläutert.

zu entnehmen. Die Tabelle differenziert dabei zwischen den drei betrachteten Use Cases. Die jeweils vorliegenden Daten unterscheiden sich dabei deutlich. In Use Case I liegen 1,473 Simulationen und 38 Eingaben vor, während Use Case II deutlich kleiner ist. Hier werden lediglich 98 Datenpunkte und 5 Eingaben betrachtet. Der dritte Use Case bildet mit 556 Datenpunkten und 14 Eingaben die mittlere Größe ab. Ein- und Ausgaben sind aus der Sicht des Metamodells skalare Größen. Des Weiteren unterscheiden sich die Eingaben in ihren Abhängigkeiten. Während in Use Case II stark korrelierte Eingaben vorliegen – Pearson Korrelation bis 0.8 (vgl. Abbildung 6.43 auf S. 170) – sind die Eingaben aus Use Case I und III mit einer Pearson Korrelation  $< 0.1$  kaum abhängig voneinander.

**Tabelle 6.2:** *ExPLAIN* Vergleich Metamodelle für Use Cases I-III

	TabPFN	XGBoost	Support Vector Regr.	Neural Network	Decision Trees	Lasso Regr.
Use Case I – Prognose Optimierungsgröße SwM auf Basis von Rohbauwandstärken						
R <sup>2</sup>	<b>0.95</b>	0.90	0.83	0.72	0.14	0.80
RMSE [-]	<b>0.04</b>	0.06	0.08	0.10	0.18	0.09
Use Case II – Prognose Brustbeschleunigung a3ms auf Basis von Zeitserienfeatures des Rückhaltesystems						
R <sup>2</sup>	0.25	<b>0.36</b>	0.24	0.12	0.12	< 0
RMSE [-]	13.0	<b>12.4</b>	13.9	15.6	16.5	16.1
Use Case III – Prognose Hauptkomponente auf Basis von Schwellerwandstärken						
R <sup>2</sup>	<b>0.93</b>	0.86	0.81	0.85	0.66	0.69
RMSE [-]	<b>0.24</b>	0.33	0.39	0.35	0.52	0.50

Ähnlich zu den Ergebnissen, die auch schon für *DrAGOS* (vgl. Tabelle 6.1 auf S. 130) ermittelt wurden, liefert TabPFN in den Use Cases I und II mit einem mittleren R<sup>2</sup> von deutlich über 0.90 eine hohe Prognosegüte. Lediglich in Use Case II, dessen Daten korreliert sind, werden schlechtere Ergebnisse mit einem mittleren R<sup>2</sup> von lediglich 0.25 erreicht. Das ist besser als alle anderen Modelle, wird jedoch durch XGBoost, das durch die selektive Nutzung der Merkmale besser mit korrelierten Eingaben umgehen kann, mit einem mittleren R<sup>2</sup> von 0.33 überboten. Trotzdem ist ersichtlich, dass die ML-Modelle Schwierigkeiten haben in diesem Use Case die Daten mit hoher Güte zu prognostizieren. XGBoost weist mit einem mittleren R<sup>2</sup> von 0.90 bzw. 0.86 und RMSE von 0.06 bzw. 0.33 in Use Case I und Use Case III die zweitbeste Prognosegüte auf. Die Decision Trees schneiden mit einem mittleren R<sup>2</sup> von 0.12 bis 0.66 generell vergleichsweise schlecht ab. Zusätzlich weisen diese überall den höchsten mittleren RMSE auf. Entsprechend ist das Modell aufgrund der einfachen Architektur nur bedingt in der Lage, die nichtlinearen

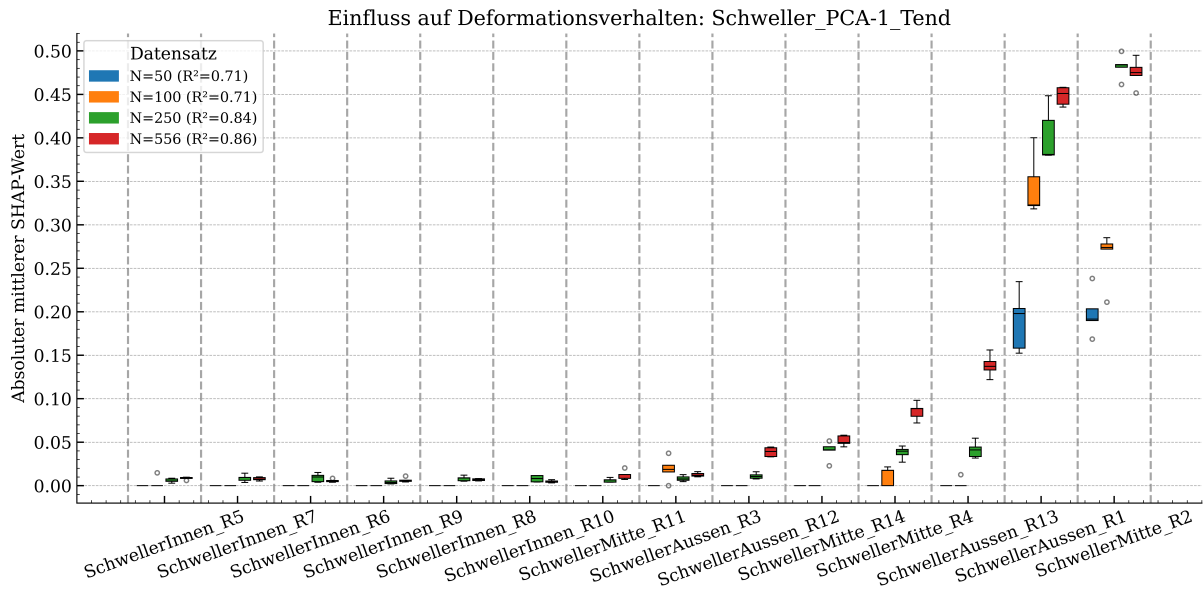
Zusammenhänge abzubilden. Im Gegensatz hierzu schneidet jedoch die ebenfalls einfache Lasso Regression mit einem mittleren  $R^2$  von 0.80 bzw. 0.69 in Use Case I und Use Case III vergleichsweise gut ab. Lediglich mit korrelierten Daten kommt das Modell nicht gut zurecht, was in einen negativen  $R^2$  zur Folge hat. Ein negativer  $R^2$  bedeutet, dass die Prognose schlechter ist als eine konstante Ausgabe des Mittelwerts. Das Multilayer Perceptron erreicht ebenfalls robuste und gute Ergebnisse mit einem mittleren  $R^2$  von bis zu 0.85. Bezogen auf die Güte ist jedoch die Transformerarchitektur von TabPFN vorzuziehen.

Für einen performanten Einsatz in der Praxis, gerade in Bezug auf die Bestimmung von Sensitivitäten, ist die Inferenzzeit ein relevanter Faktor. Während XGBoost die effizienteste Inferenz ermöglicht, sind am Beispiel von Use Case I die Modelle TabPFN um 29 %, Support Vector Regression um 27 %, Multilayer Perceptron um 9.3 %, Decision Trees um 23 % sowie die Lasso Regression um 12 % langsamer. Dies ist neben der Anwendung in Echtzeit und der Optimierung auch für die Berechnung der SHAP-Werte relevant, da hier theoretisch bis zu  $2^M$  Modellaufrufe getätigt werden müssen.  $M$  entspricht der Anzahl der Eingaben des Problems.

Folglich bestätigt sich auch hier XGBoost als ein guter Mittelweg aus zeitlicher Effizienz in der Inferenz und Prognosegüte. Hinzu kommt das extra für baumbasierte Verfahren entwickelte TreeSHAP [Lun+20], das die exakte Berechnung von SHAP in polynomieller Zeit ermöglicht. Die zügige Ergebnisgenerierung und -darstellung ist gerade in Bezug auf die Prozessintegration relevant, um Akzeptanz neuer Methoden zu schaffen.

#### 6.4.4.2 SHAP-Werte für unterschiedlich große Datensätze

Da im Vorhinein nicht eindeutig bestimmt werden kann, wie viele Daten zur Lösung einer Problemstellung erforderlich sind, muss der Datenbedarf typischerweise empirisch durch Experimente ermittelt werden. Anhand von Use Case III (vgl. Abbildung 6.47 auf S. 173) werden verschieden große Datensätze betrachtet. Diese enthalten 50, 100, 250 sowie die insgesamt generierten 556 Simulationen. Zur Definition der Datensätze, ausgehend von dem vollständigen Datensatz mit 556 Simulationen, wird ein Farthest Point Sampling Verfahren verwendet. Dieses wählt sukzessive den jeweils am weitesten entfernten Punkt vom bereits gewählten Set aus. Das stellt in jedem der vier Subsets eine möglichst ganzheitliche Ausfüllung des Designraums sicher. Anschließend werden fünf XGBoost Modelle trainiert und die SHAP-Werte für den Einfluss auf das Deformationsverhalten des Seitenschwellers entsprechend *ExPLAIN* berechnet. Die absoluten mittleren SHAP-Werte für die 14 betrachteten Wandstärken sind in Abbildung 6.56 gezeigt. Diese sind jeweils mittels Boxplot als Verteilung über die fünf Folds der k-fold Cross Validation angegeben, um zusätzlich die Unsicherheit hinsichtlich dem wiederholten Training mit unterschiedlichen Subsets des Datensatzes herauszustellen.



**Abbildung 6.56:** *ExPLAIN* Robustheit: Use Case III mit abweichender Datensatzgröße [50, 100, 250, 556] – Mittlere absolute SHAP-Werte

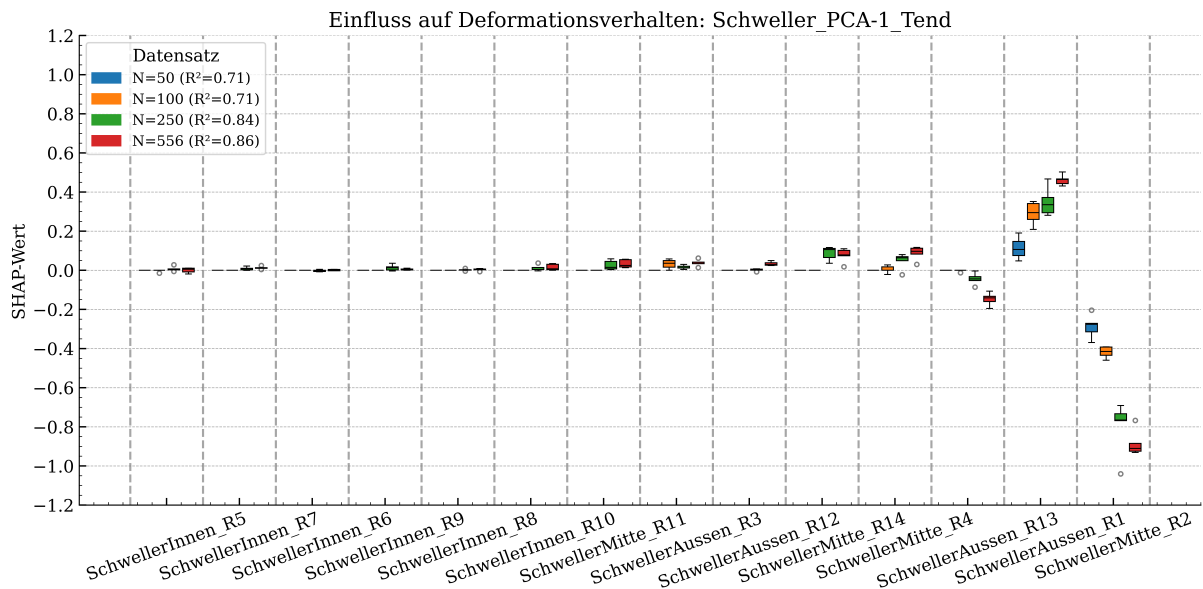
Die in Abbildung 6.56 gezeigten absoluten mittleren SHAP-Werte sind als Maß für die *globale Sensitivität* zu interpretieren, da sie ausgehend von den lokalen Einflüssen eine Aussage bezogen auf den gesamten Designraum treffen. Während für die Datensatzgröße 50 und 100 eine vergleichbare mittlere Prognosegüte von  $R^2=0.71$  vorliegt, sind die übrigen beiden Modelle mit einem mittleren  $R^2$  von über 0.8 deutlich genauer. Die Wandstärken R2 und R1 aus der Mitte und dem äußeren Bereich des Schwellers sind besonders relevant für die Entscheidungsfindung aller trainierten XGBoost Modelle. Ebenfalls relevant für die Prognose der Modelle mit den größeren beiden Datensätzen sind die Eingaben R13, R4, R14, R12 und R3. Alle weiteren Eingaben weisen nur einen marginalen Einfluss auf die Ausgabe auf. In Abbildung 6.56 ist ersichtlich, dass die Modelle, die auf den größeren Datensätzen aufbauen, gegenüber mehreren Eingaben sensitiv sind. Das deutet auf eine komplexere Entscheidungsfindung hin.

Alle Modelle extrahieren ein vergleichbares Gesamtbild aus den Daten bezogen auf die Relevanz einzelner Eingaben. Während die größeren beiden Datensätze vergleichbare Magnituden für die mittleren absoluten SHAP-Werte aufweisen, sind diese gerade für 50 Datenpunkte deutlich geringer. Das kann unter anderem daran liegen, dass weniger konträr wirkende Eingaben in die Entscheidungsfindung einbezogen werden. Die Streubreite der SHAP-Werte verursacht durch die einzelnen Folds nehmen mit zunehmender Datenanzahl ab. Für 50 Datenpunkte beträgt diese maximal 0.09 (R1) und für  $N=556$  maximal 0.05 (R2). Ebenfalls ist in Abbildung 6.56 zu sehen, dass die Magnitude von der Datensatzgröße

abhängt. Für steigende Datenzahl ist jedoch eine Konvergenz, gerade für die wichtigsten beiden Eingaben, feststellbar.

Die einzeln dargestellten Ergebnisse, wie bisher in den Use Cases, erlauben eine intuitive Darstellung. Unsicherheiten sind jedoch stets mit abzubilden, um eine Fehlinterpretation zu vermeiden.

Die SHAP-Werte ermöglichen zudem die Betrachtung der *lokalen Sensitivität* bezogen auf die Einflüsse der Eingaben innerhalb einer Simulation. Diese sind für drei repräsentative Simulationen RUN\_006, RUN\_542 und RUN\_554 aus unterschiedlichen Teilen des Wertebereichs der Hauptkomponente I im letzten Zeitschritt (Schweller\_PCA-1\_Tend) in Abbildung 6.57 bis Abbildung 6.59 dargestellt. In Abbildung 6.57 liefern die einzelnen XGBoost-Modelle bezogen auf Reihenfolge und Richtung der Beeinflussung identische Ergebnisse.

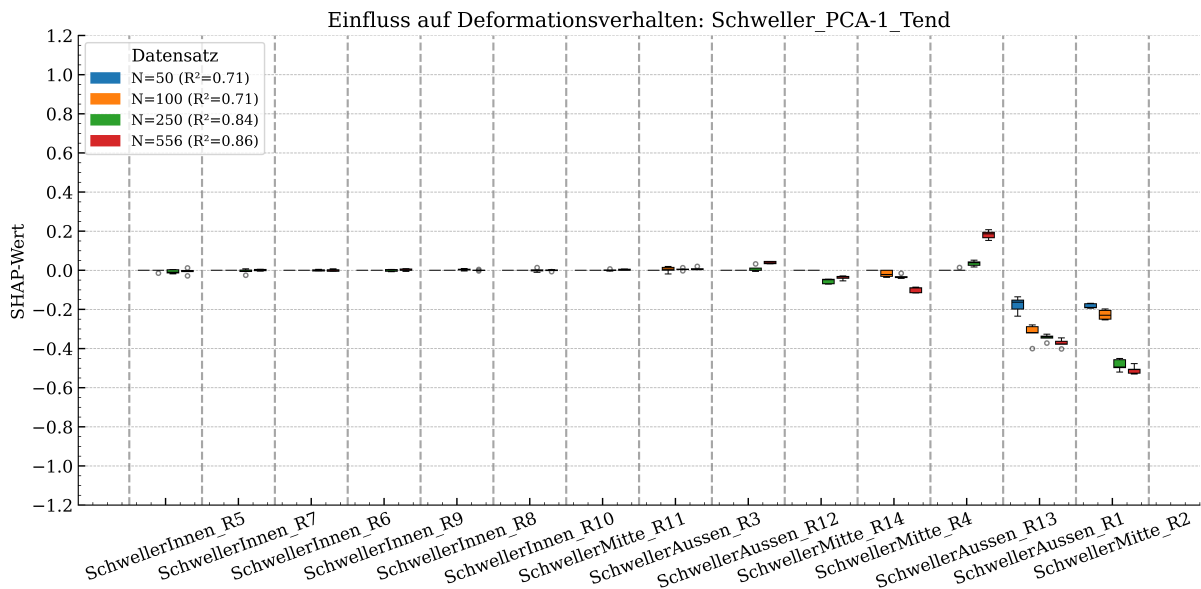


**Abbildung 6.57:** *ExPLAIN* Robustheit: Use Case III mit abweichender Datensatzgröße – SHAP-Werte für RUN\_006 mit geringer Schweller\_PCA-1\_Tend

RUN\_006 weist dabei eine geringere Ausgabe als der Referenzwert  $\mathbb{E}[f(x)]$  auf. Die beiden kleineren und größeren Datensätze sind jeweils bezogen auf die Magnitude der SHAP-Werte vergleichbar. Die größten Unterschiede ergeben sich in der Wichtigkeit von R2. Analog der Erkenntnisse der globalen Betrachtung stützten sich die XGBoost-Modelle der beiden kleineren Datensätzen lediglich auf die beiden wichtigsten Eingaben R2 und R1 zur Entscheidungsfindung.

Verglichen mit RUN\_006 ist die maximale Magnitude der SHAP-Werte von RUN\_542 in Abbildung 6.58 mit bis lediglich 0.5 um die Hälfte geringer. RUN\_542 weist eine geringe

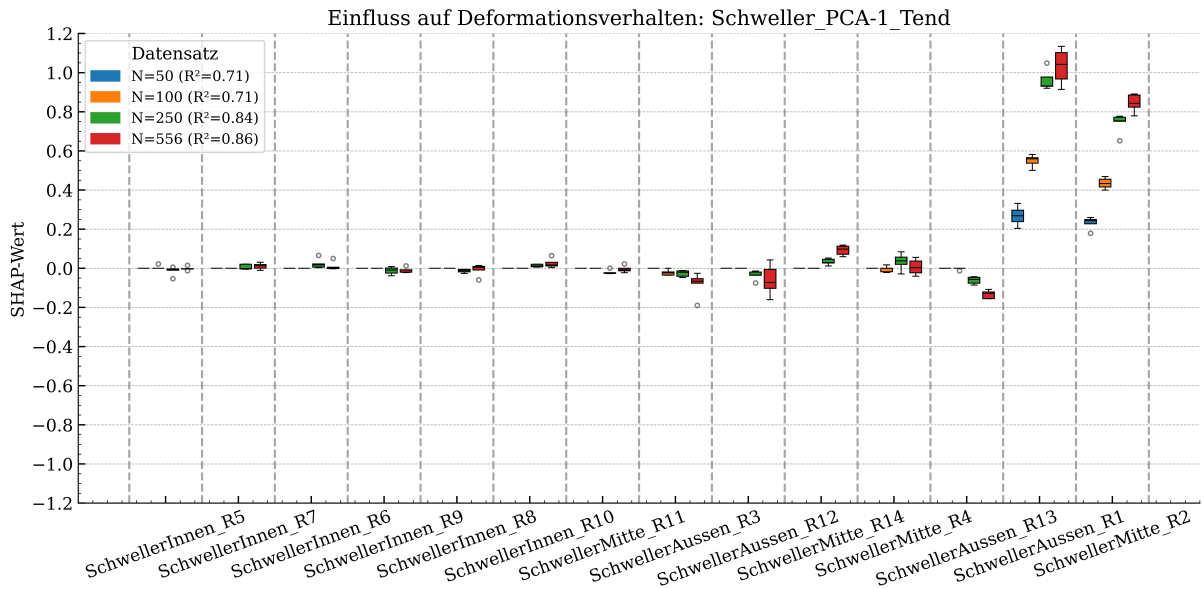
negative Ausgabe auf. Ebenfalls ist ersichtlich, dass die Unterschiede zwischen den Größen der Datensätze für R1 und R2 geringer sind als in dem bisherigen Beispiel. Die Aussage, dass R2 einen marginal größeren Einfluss als R1 hat, ist lediglich mit den beiden größeren Datensätzen möglich.



**Abbildung 6.58:** *ExPLAIN* Robustheit: Use Case III mit abweichender Datensatzgröße – SHAP-Werte für RUN\_542 mit Schweller\_PCA-1\_Tend nahe Null

Das letzte Beispiel – gezeigt in Abbildung 6.59 – weist einen hohen Wert für die Ausgabe auf. Von allen XGBoost-Modellen wird festgestellt, dass R1 bedeutender ist als R2. Die Streubereiche überlappen sich dabei nicht. Auch hier ist wieder ersichtlich, dass die kleineren beiden Datensätze geringere Magnituden aufweisen. Für die größeren beiden sind die Magnituden vergleichbar. Die Richtung der Einflussnahme ist auch in diesem Beispiel eindeutig bestimmt. Lediglich weniger bedeutsame Eingaben wie R12 oder R4 streuen marginal zwischen leicht positiver und leicht negativer Einflussnahme. Bei den vorangegangenen Beispielen war die Richtung auch bei den weniger wichtigen Eingaben sehr eindeutig bestimmt.

Zusammenfassend können die Ergebnisse der globalen Sensitivität auch im Rahmen der lokalen Sensitivitätsbetrachtung bestätigt werden: Tendenzen, welche Eingaben in welche Richtung die Zielvariable beeinflussen, können unabhängig der Datensatzgröße klar herausgestellt werden. Die Ergebnisse weisen weniger Streuung auf, wenn mehr Daten vorhanden sind. Ebenso ist eine Konvergenz der Magnitude zu sehen. Diese ist entsprechend den Unsicherheiten mit Vorsicht zu betrachten.



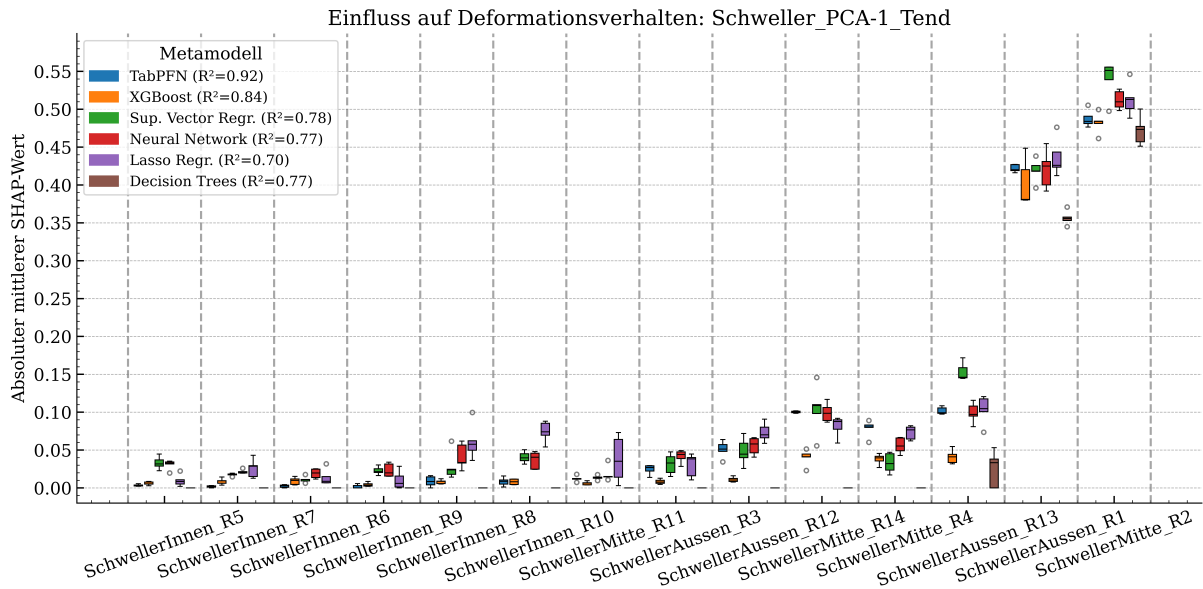
**Abbildung 6.59:** *ExPLAIN* Robustheit: Use Case III mit abweichender Datensatzgröße – SHAP-Werte für RUN\_554 mit hoher Schweller\_PCA-1\_Tend

#### 6.4.4.3 SHAP-Werte für unterschiedliche Metamodelle

Neben dem Datensatz ist auch das verwendete Metamodell eine relevante Stellgröße, deren Einfluss nun weiter untersucht werden soll. Aus Gründen der Rechenzeit wird der Einfluss an einem reduzierten Datensatz mit 250 Simulationen für Use Case III (vgl. Abbildung 6.47 auf S. 173) betrachtet.

Die *globale Sensitivität* in Bezug auf die Ausgabegröße Schweller\_PC-1\_Tend ist in Abbildung 6.60 gezeigt. Die Ergebnisdarstellung ist analog zum Vergleich der Datensatzgröße. Zuerst werden globale Sensitivitäten und dann beispielhaft lokale Sensitivitäten diskutiert. Auch hier werden Unsicherheiten durch die Verteilung der SHAP-Werte über die 5-fold Cross Validation hervorgehoben. Verglichen werden verschiedene Metamodelle, deren Prognosegüte bereits in Tabelle 6.2 auf S. 182 für den gesamten Datensatz mit 556 Simulationen diskutiert wurde.

Bezogen auf die globale Sensitivität in Abbildung 6.60 ist ersichtlich, dass die Streubänder mit maximal 0.08 für die Lasso Regression (R11) vergleichsweise gering sind. Tendenziell sind die Streubänder der Modelle mit hoher Güte, insbesondere TabPFN, noch geringer. Folglich können im Rahmen der 5-fold Cross Validation gut reproduzierbare Ergebnisse erzielt werden. Des Weiteren lässt sich feststellen, dass die Magnitude der SHAP-Werte, insbesondere der beiden wichtigsten Eingaben R1 und R2, unabhängig der Wahl des Metamodells sind. Dies ist insbesondere in Bezug auf die vergleichsweise großen Unterschiede bei der Verwendung von kleineren Datensätzen deutlich zu sehen. Decision Trees und Support Vector Regression weisen hier die deutlichsten Unterschiede auf.

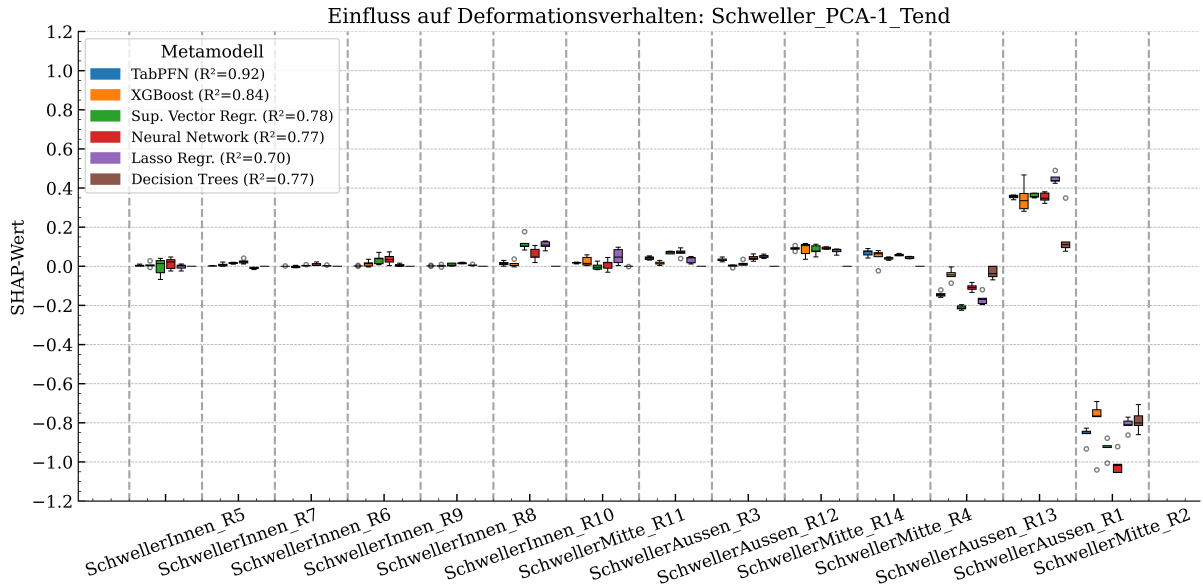


**Abbildung 6.60:** *ExPLAIN* Robustheit: Use Case III mit verschiedenen Metamodellen für  $N=250$  – Absolute mittlere SHAP-Werte

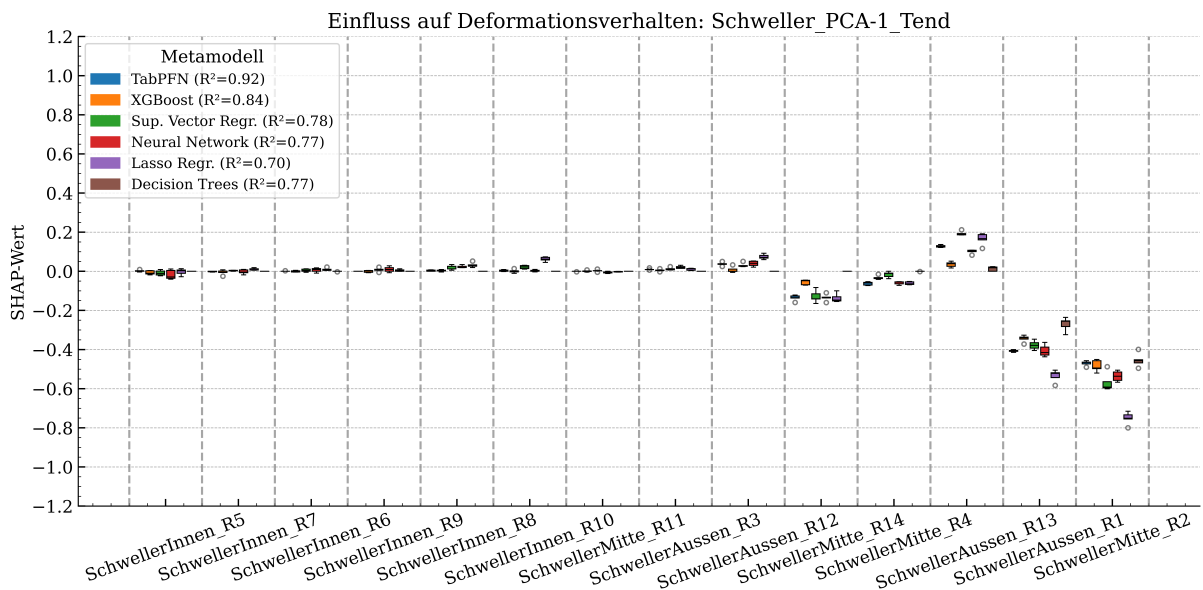
Die übrigen Modelle liefern vergleichbare Ergebnisse. Eine eindeutige Differenzierung zwischen wichtigen Eingaben (R2, R1), und Eingaben mit reduzierter Wichtigkeit (R13 bis R11 v.r.n.l. in Abbildung 6.60) und Eingaben mit marginaler Wichtigkeit (R9 bis R5 v.r.n.l. in Abbildung 6.60) wird entsprechend bereitgestellt. Die Decision Trees stützen ihre Prognose lediglich auf R2, R1 und R13. Den übrigen Wandstärken wird von diesem Modell keine Bedeutung zugewiesen. Einen vernachlässigbaren Einfluss für so viele Eingaben, ist auch aus physikalischer Sicht nicht sinnvoll erklärbar.

Die *lokalen Sensitivitäten* sind für drei repräsentative Simulationen RUN\_006, RUN\_542 und RUN\_554 aus verschiedenen Wertebereichen der Zielvariable Schweller\_PCA-1\_Tend in Abbildung 6.61 bis Abbildung 6.63 dargestellt.

Für RUN\_006 in Abbildung 6.61 ist die Ausgabe tiefer als der Referenzwert  $\mathbb{E}[f(x)]$ . Die bereits in der globalen Betrachtung herausgestellten Erkenntnisse für die Decision Trees bestätigen sich auch lokal – lediglich R2, R1 und R13 werden für die Entscheidungsfindung verwendet. Die Reihenfolge von R2, R1 und R13 als die drei wichtigsten Eingaben sowie deren Wirkrichtung wird auch hier einheitlich spezifiziert. Ebenso sind die Magnituden der Modelle miteinander vergleichbar. Den einzigen Ausreißer bildet erneut das Decision-Tree-Modell mit einem um etwa 59% geringeren Einfluss von R1 auf die Ausgabe Schweller\_PCA-1\_Tend.



**Abbildung 6.61:** *ExPLAIN* Robustheit: Use Case III mit verschiedenen Metamodellen für  $N=250$  – SHAP-Werte für RUN\_006 mit geringer Schweller\_PCA-1\_Tend

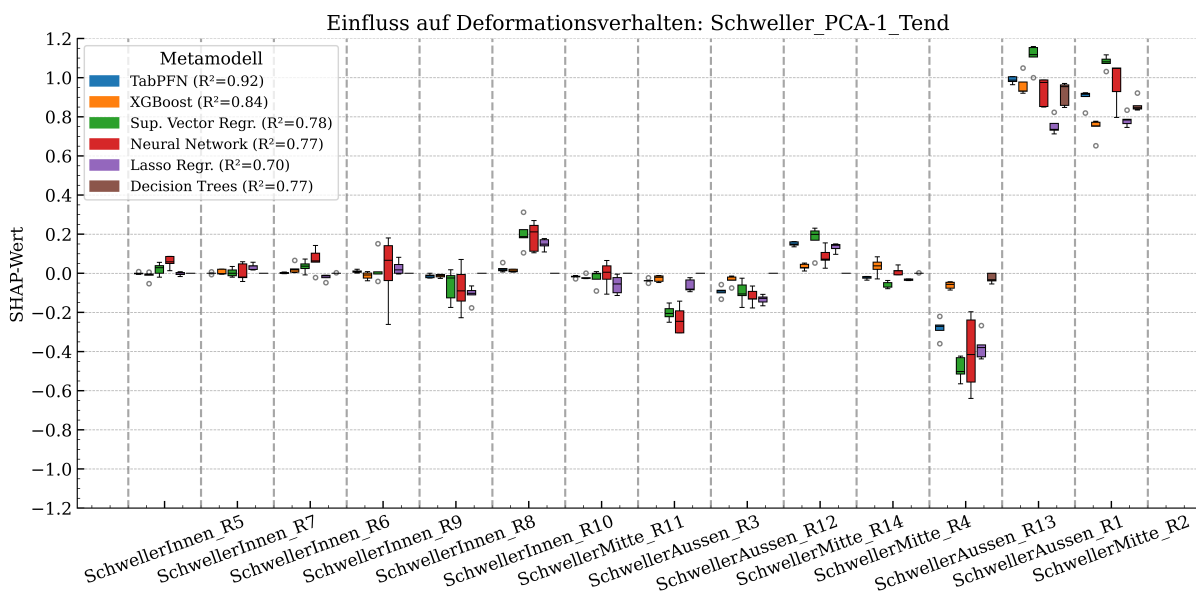


**Abbildung 6.62:** *ExPLAIN* Robustheit: Use Case III mit verschiedenen Metamodellen für  $N=250$  – SHAP-Werte für RUN\_542 mit Schweller\_PCA-1\_Tend nahe Null

In Abbildung 6.62 befindet sich die Ausgabe näher am Referenzwert  $\mathbb{E}[f(x)]$ . Auch hier ist die Magnitude der SHAP-Werte geringer als in dem vorangegangenen Beispiel. Die Streubreite der einzelnen Modelle ist mit maximal  $\pm 0.04$  sehr gering. Die größten Abweichungen gegenüber den übrigen Modellen, bezogen auf die Magnitude der drei wichtigsten

Eingaben, weisen auch hier wieder die Decision Trees und die Lasso Regression auf. Die Tendenz, dass auch hier R2 einflussreicher ist als R1 und R13 in die Gegenrichtung wirkt, lässt sich jedoch aus allen Ergebnissen ablesen.

In Abbildung 6.63 ist die Ausgabe höher als der Referenzwert  $\mathbb{E}[f(x)]$ . TabPFN, XGBoost und Support Vector Regression identifizieren R1 wichtiger als R2. Die übrigen drei Modelle treffen durch überlappende Streubereiche keine eindeutige Entscheidung. Eine hohe Streubreite von bis zu 0.42 für R13 tritt in den Ergebnissen des neuronalen Netzwerks auf. Die übrigen Modelle weisen höchstens  $\pm 0.17$  (R8, Support Vector Regression) auf. Die Richtung der Einflussnahme ist auch in diesem Beispiel eindeutig bestimmt. Analog zu der Betrachtung unterschiedlicher Datensätze, beispielsweise in Abbildung 6.59 auf S. 187, streuen auch hier einige weniger wichtige Eingaben, wie R4 und R9, zwischen leicht positiver und negativer Beeinflussung. Bei den vorangegangenen beiden Beispielen für verschiedene Metamodelle war die Richtung auch bei den weniger wichtigen Eingaben eindeutig bestimmt.



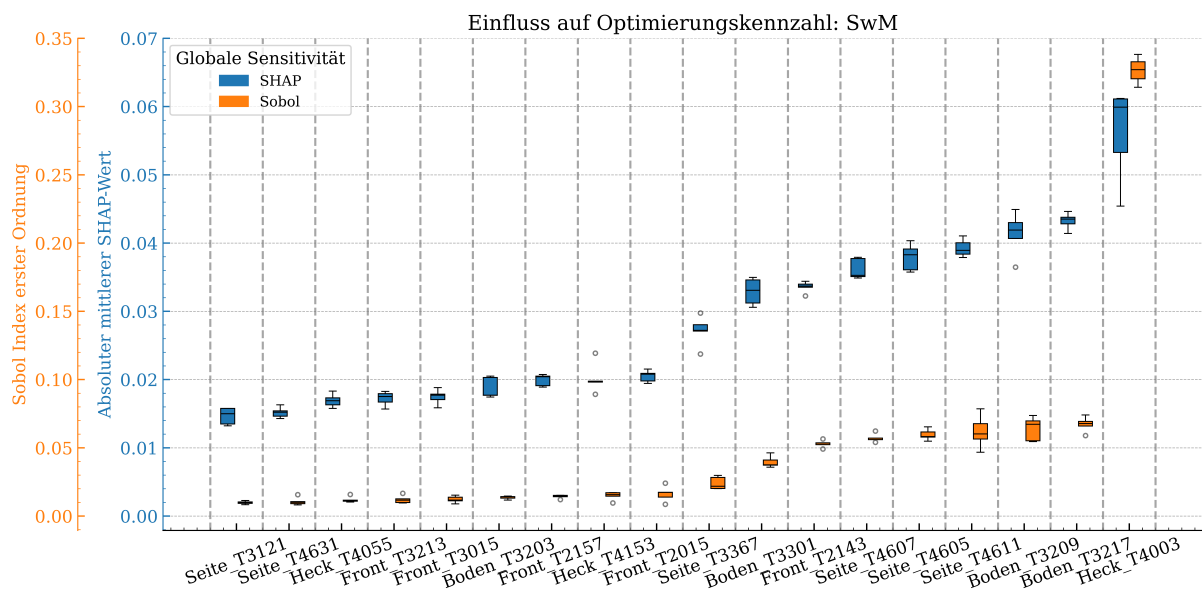
**Abbildung 6.63:** *ExPLAIN* Robustheit: Use Case III mit verschiedenen Metamodellen für  $N=250$  – SHAP-Werte für RUN\_554 mit hoher Schweller\_PCA-1\_Tend

Zusammenfassend werden hier die Ergebnisse, die bereits auch durch die verschiedenen großen Datensätze erreicht wurden bestätigt: Tendenzen, welche Eingaben die Zielvariable beeinflussen, können unabhängig vom Metamodell deutlich herausgestellt werden. Gerade im letzten Beispiel wird ersichtlich, dass die Magnituden auch abweichen können, weshalb diese mit Vorsicht zu betrachten sind. Gerade die Modelle TabPFN und XGBoost, die auch in der Lage sind hohe Prognosegüten auf den Daten zu erreichen, liefern SHAP-

Werte mit vergleichsweise geringen Streuungen. Support Vector Regression und auch die Lasso Regression sind mögliche Alternativen und könnten, beispielsweise zur Einordnung und Überprüfung von Ergebnissen, mit anderen Modellen herangezogen werden. Von den Decision Trees und auch den neuronalen Netzwerken ist auf Basis der ermittelten Ergebnisse abzuraten.

#### 6.4.4.4 Vergleich globaler und lokale Sensitivitätsmaße

Für *ExPLAIN* wurde aufgrund der eingeführten Ergebnisdarstellungen mittels System- und Differenz-SHAP die SHAP-Methode [LL17] zugrunde gelegt. Wie im Rahmen der Grundlagen in Abschnitt 2.2.2 auf S. 27 angesprochen gibt es jedoch auch weitere Verfahren. Um die Ergebnisse einzuordnen werden die Sobol Indizes [Sob01] und LIME [RSG16] verwendet. Dies wird anhand von Use Case I (Abbildung 6.37 auf S. 165) untersucht. Zum Vergleich der Ergebnisse in Bezug auf die *globale Sensitivität* werden die Einflüsse der 18 wichtigsten der insgesamt 38 verwendeten Wandstärken in Abbildung 6.64 dargestellt.

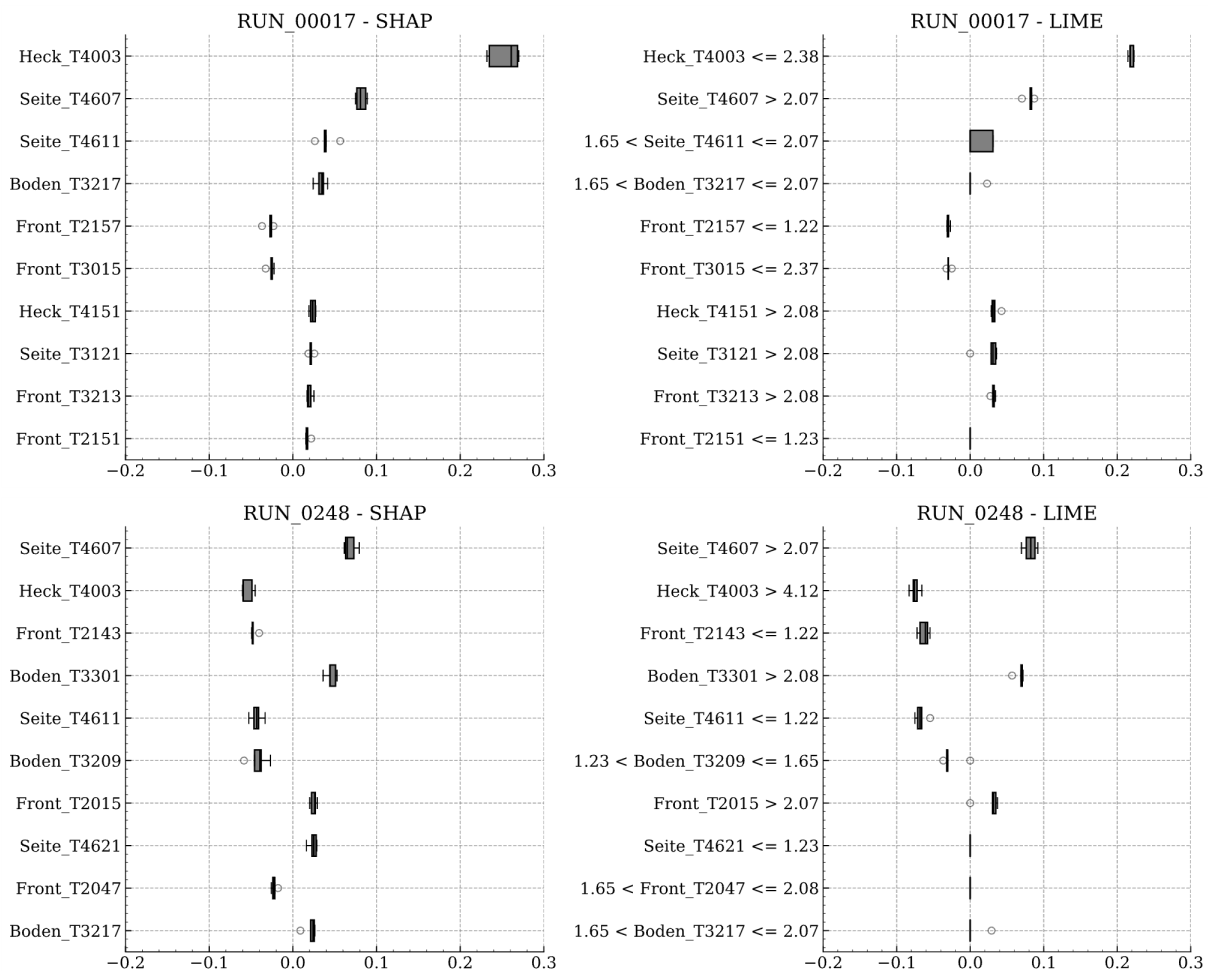


**Abbildung 6.64:** *ExPLAIN* Robustheit: Use Case I mit zwei globalen Sensitivitätsmaßen – Sobol Indizes erster Ordnung und absolute mittlere SHAP-Werte

Ein Vergleich der Magnituden ist hier nicht zielführend, da SHAP den Einfluss der Eingabe relativ zum Erwartungswert der Ausgabe misst, während Sobol Varianzanteile der Ausgabe auf die Eingaben zurückführt. Entscheidend sind daher vor allem die Robustheit der Verfahren sowie die Reihenfolge und Abgrenzung der einzelnen Eingaben. Für die Sobol Indizes wird die erste Ordnung betrachtet, die keine Interaktionseffekte berücksichtigt. Beide Verfahren identifizieren die identischen ersten zehn Wandstärken, die relevant für die Ausgabe – hier die Optimierungskennzahl SwM – sind. Für Eingaben mit geringerer

Bedeutung als T4003 zeigen die Boxplots für SHAP und Sobol überlappende Bereiche. Die Reihenfolge in Abbildung 6.64 wird daher auf Basis des Medians festgelegt. Entsprechend können in den wichtigsten zehn Wandstärken drei Bereiche mit hoher mittlerer und geringer Wichtigkeit identifizieren werden, T4003, T3217 bis T3301 und T3367 bis T3121, die sich nicht überschneiden. Diese lassen sich unabhängig von der Methode identifizieren. Auffällig ist, dass SHAP, insbesondere für T4003, eine größere Streubreite aufweist. Ebenfalls wird T4003 nicht so deutlich hervorgehoben, wie das mit den Sobol Indizes der Fall ist. Grundsätzlich sind die Ergebnisse unabhängig der verwendeten Methode und sind dementsprechend reproduzierbar.

Ein Beispiel für die *lokale Sensitivität* ist anhand von zwei Simulationen RUN\_0017 und RUN\_0248 in Abbildung 6.65 gezeigt. Abgebildet sind dort die zehn wichtigsten Eingaben bezogen auf SwM. Analog zu den bisherigen Darstellungen wird auch hier die Streuung über die 5-fold Cross Validation abgebildet. Sowohl SHAP als auch LIME bestimmen identisch die beiden einflussreichsten Eingaben.



**Abbildung 6.65:** *ExPLAIN* Robustheit: Use Case I mit zwei lokalen Sensitivitätsmaßen – LIME und SHAP-Werte

Auch hier ist ersichtlich, dass insbesondere Tendenzen und die ermittelten Richtungen der Beeinflussung vergleichbar sind. Lediglich ein Ausreißer lässt sich in RUN\_0017 mit T4611 finden. Dieser taucht beispielsweise in den zehn wichtigsten Einflüssen von LIME gar nicht auf. LIME verwendet ein lokales Modell, hier ein Entscheidungsbaum, um die Entscheidungslogik des komplexen Modells in der unmittelbaren Umgebung des Datenpunkts widerzuspiegeln. Entsprechend werden in Abbildung 6.65 auch Wertebereiche der Eingaben vorgegeben. SHAP hingegen legt für jede Eingabe einen spezifischen Wert fest. Im Gegensatz zu SHAP liefert jedoch LIME nicht die Möglichkeit direkt auch globale Sensitivitätsbetrachtungen durchzuführen.

Zusammenfassend zeigt SHAP die Möglichkeit sowohl global als auch lokal Ergebnisse im Einklang mit alternativen Methoden bereitzustellen. Ebenfalls relevant für die Praxis ist die Berechnungsdauer von Sobol, SHAP und LIME. Die Berechnung von LIME und Sobol gemeinsam nimmt mit 200 s 92 % mehr Zeit in Anspruch als die Berechnung von SHAP mittels TreeSHAP am Beispiel des hier verwendeten XGBoost-Modells. Der größte Teil der Zeit entfällt jedoch auf die Berechnung von LIME. Lediglich 10 s und damit 5 % entfallen auf die Berechnung der Sobol Indizes.

Für die Darstellungen mittels System und Differenz-SHAP sind allerdings LIME und Sobol nicht geeignet. Eine analoge Darstellung für Systems SHAP wäre nur durch das separate Zusammenfassen von Eingaben und eine anschließende Neuberechnung realisierbar. Das ist für die perspektivische Integration in interaktive und agile grafische Oberflächen hinderlich. Für Differenz-SHAP hingegen ist eine konsistente Additivität der Wichtigkeiten einzelner Eingaben erforderlich, was für LIME nicht der Fall ist.

#### 6.4.4.5 Umgang mit korrelierten Eingaben

Für Use Cases I und III sind die einzelnen Eingaben nicht voneinander abhängig. Die maximale Pearson Korrelation beträgt in beiden Fällen weniger als 0.1. Für Use Case II hingegen wurden die Eingaben aus der Simulationsausgabe extrahiert, um den (o-o)-Zusammenhang auch unparametrisiert abbilden und erklären zu können. Die resultierende Pearson Korrelation ist mit bis zu 0.8 vergleichsweise hoch. Genauere Angaben liefert die Korrelationsmatrix in Abbildung 6.43 auf S. 170.

Auch wenn gezeigt wurde, dass XGBoost vergleichsweise gut mit korrelierten Eingaben umgehen kann, gilt dies nicht automatisch für die darauf aufbauende Modellagnostik. Die in Abschnitt 2.2.2 auf S. 27 eingeführten SHAP-Werte basieren auf der Annahme, dass die marginale Verteilung  $P(x_S)$  durch Marginalisierung der gemeinsamen Verteilung  $P(x)$  über die nicht betrachteten Eingaben gewonnen werden kann. Das impliziert, dass die

Eingaben als unabhängig voneinander betrachtet werden und dadurch

$$P(x_S) = \prod_{j \in S} P(x_j)$$

gilt.  $S$  ist dabei die eine Teilmenge der Eingaben. Dementsprechend vereinfacht sich auch das bedingte Erwartungswertmodell

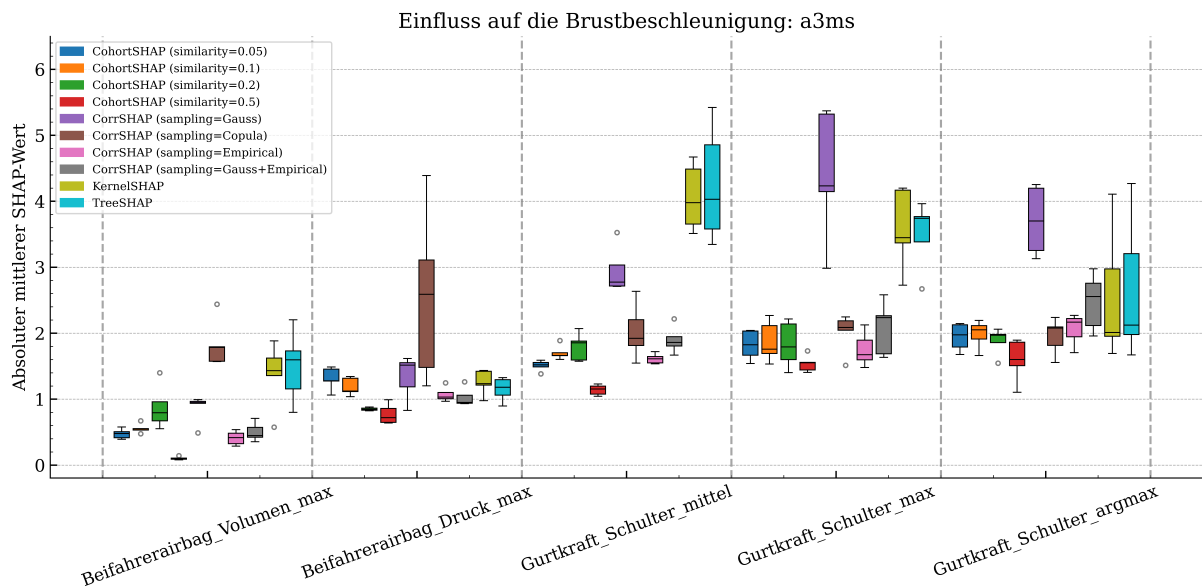
$$f_S(x_S) = \mathbb{E}[f(x) \mid x_S],$$

da keine Abhängigkeiten zu nicht beobachteten Merkmalen berücksichtigt werden müssen. Sind die Eingaben korreliert, sorgt jedoch das Entfernen von  $x_i$  für eine Veränderung der Verteilung der übrigen Eingaben  $x_j \in S$ , sodass

$$P(x_S) \neq P(x_S \mid x_i).$$

Die resultierenden SHAP-Werte  $\phi$  spiegeln folglich nicht die tatsächlichen Effekte der Merkmale auf die Modellvorhersage wider [AJL21].

Als Erweiterungen gegenüber Tree bzw. KernelSHAP werden CohortSHAP [MOS19; MOS21] und CorrSHAP [AJL21] verwendet. CohortSHAP verwendet ausschließlich tatsächlich beobachtete Datenpunkte, um das Problem unmöglicher oder unwahrscheinlicher Eingabekombinationen zu vermeiden. Dabei werden für jede Eingabe über eine Ähnlichkeitsmaß Gruppen (Cohorts) von Datenpunkten gebildet, die ähnliche Werte dieser Eingabe aufweisen. Die SHAP-Werte werden anschließend basierend auf den Mittelwerten der Ausgaben innerhalb dieser Gruppen berechnet. CorrSHAP ist der in [AJL21] erweiterte KernelSHAP-Ansatz und berücksichtigt Abhängigkeiten zwischen Eingaben, indem er die bedingte Verteilung der Eingaben schätzt und daraus Stichproben zieht, anstatt Unabhängigkeit anzunehmen. Hierfür werden zur Schätzung parametrische (multivariate Gauß-Verteilung, Gaußsche Copula) und nichtparametrische (empirische bedingte Verteilung) Verteilungen sowie ein Kombination der Ansätze verwendet. Dies führt zu realistischeren Datenkombinationen und damit zu genaueren und verlässlicheren Erklärungen. Abbildung 6.66 zeigt die absoluten mittleren SHAP-Werte für CohortSHAP mit verschiedenen Grenzwerten für die Ähnlichkeit zur Bildung der Gruppen sowie CorrSHAP mit vier verschiedenen Stichprobenverteilungen, den sog. Samplingstrategien. Ebenfalls sind zur Einordnung die mittels Tree und KernelSHAP berechneten Werte dabei. Die Verteilung gezeigt als Boxplot gibt auch hier wieder die mittleren absoluten SHAP-Werte über die 5-fold Cross Validation an. Abbildung 6.66 zeigt, dass die mittels Tree- und KernelSHAP berechneten Werte vergleichbar und mit zwei Verfahren reproduzierbar sind. Die Streubreiten sind nahezu identisch und der größte Unterschied bezogen auf den Median mit



**Abbildung 6.66:** *ExPLAIN* Robustheit: Use Case II mit korrelierten Eingaben – Absolute Mittlere SHAP-Werte mit CorrSHAP, CohortSHAP und TreeSHAP

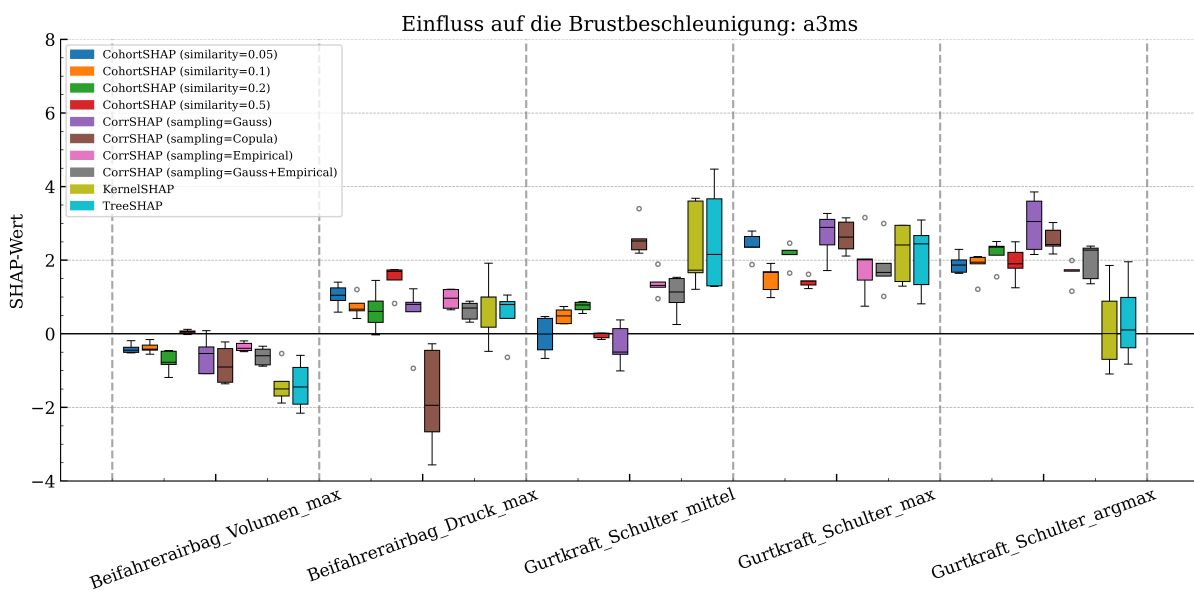
lediglich 0.2 tritt für die maximale Gurtkraft an der Schulter auf. Gerade im Vergleich mit den übrigen SHAP-Berechnungsverfahren ist die Streuung für beide Verfahren jedoch hoch und die Streubereiche von dem Volumen des Beifahrerairbags sowie dem Argmaximum der Schulterkraft überlappen. Schlussfolgern lässt sich daraus, dass die Annahme der Unabhängigkeit der Eingaben hier nicht korrekt ist. Noch größere Unsicherheiten lassen sich für CorrSHAP feststellen. Bei Verwendung der Copula Samplingstrategie streuen die SHAP-Werte des maximalen Drucks im Beifahrerairbag zwischen 1.1 und 4.2. Folglich gibt es Durchläufe in denen diese Eingabe die höchsten aber auch den geringsten Einfluss von allen Eingaben aufweist.

Zusätzlich überlappt der Einfluss aller Eingaben. Durch die hohe Unsicherheit ist diese Information nicht konsistent und daher auch wenig hilfreich. Ähnliche Ergebnisse bezogen auf die Streubreiten werden auch für die Gauß Samplingstrategie für CorrSHAP ermittelt. Während der maximale Druck des Beifahrerairbags hier zuverlässiger bestimmt werden kann, weist die maximale Gurtkraft eine erhöhte Streuung auf. Ein Unterschied besteht jedoch darin, dass nun die Einflüsse zwischen Airbag und Gurt klar voneinander getrennt sind, da die Streubänder nicht überlappen. Ein Grund für das schlechte Abschneiden kann die fehlerhafte Annahme der Gaußverteilung sein. Dies wird auch bei Betrachtung des Histogramms in Abbildung 6.43 auf S. 170 klar.

Wird für CorrSHAP jedoch die empirische Samplingstrategie oder eine Kombination aus Gauß und empirischer Samplingstrategie eingesetzt, verringert das Unsicherheiten deutlich.

Das bestärkt die Annahme, dass die vorhandenen Daten am besten empirisch abgebildet werden können. Die kombinierte Samplingstrategie liefert vergleichbare Ergebnisse, weist aber bezogen auf Maximum und Argmaximum eine höhere Streubreite auf. Die maximale Streubreite beträgt 1.0 für das kombinierte und 0.7 für die empirische Samplingstrategie. Beide Verfahren trennen die Einflüsse von Airbag und Gurt deutlich. Die Ergebnisse, insbesondere aus der empirischen Samplingstrategie, sind mit CohortSHAP reproduzierbar. Die CohortSHAP-Werte für eine Ähnlichkeit von 0.05 bis 0.2 sind in guter Übereinstimmung. Lediglich für eine Ähnlichkeit von 0.5 werden bezogen auf die Amplitude tendenziell geringere Werte ermittelt. Die Reihenfolge bleibt jedoch gleich. Eine eindeutige Trennung der Einflüsse von Airbag und Gurt erfolgt lediglich bei einer Ähnlichkeitskoeffizient von 0.05. Die beste Übereinstimmung mit der empirischen Samplingstrategie von CorrSHAP weist CohortSHAP mit einer Ähnlichkeit von 0.1 auf.

Aus Sicht der Domäne ist durch den physischen Kontakt zwischen Gurt und ATD nahelegend, dass dieser bezogen auf die Brustbeschleunigung eine höhere Relevanz aufweist. Insbesondere die Schaltzeit, die in Teilen durch das Argmaximum der Gurtkraft abgebildet wird, ist hier besonders wichtig. Bezogen auf den Airbag ist zu erwarten, dass gerade der Druck relevanten Einfluss hat. Sinkt der ATD aufgrund eines weicheren Airbags tiefer ein, erhöht sich die Last, die auf die Brust wirkt. Im Kontext des Ingenieurverständnisses des betrachteten Systems, liefern die durch durch CohortSHAP (0.1) und CorrSHAP (empirische Samplingstrategie) erzielten Ergebnisse in Abbildung 6.66 die schlüssigste Erklärung.



**Abbildung 6.67:** *ExPLAIN* Robustheit: Use Case II mit korrelierten Eingaben – SHAP-Werte mit CorrSHAP, CohortSHAP und TreeSHAP

Abbildung 6.67 zeigt die SHAP-Werte für eine Simulation. CorrSHAP mit Copula Samplingstrategie liefert als einziges Verfahren eine andere Wirkrichtung bezogen auf den maximalen Airbagdruck. TreeSHAP und KernelSHAP sind nahezu identisch, streuen aber stark. Für das Argmaximum der Gurtkraft wird zudem keine eindeutige Richtung identifiziert.

Alle Verfahren identifizieren einen negativen Einfluss des Volumens. Geringe Streuungen und übereinstimmende Ergebnisse weisen die empirischen Samplingstrategie für CorrSHAP und CohortSHAP mit einem Ähnlichkeitskoeffizienten von 0.1 auf. Entsprechend bestätigt das lokale Beispiel die bereits global herausgestellten Tendenzen. Das Beispiel verdeutlicht, dass hier der Airbag eine untergeordnete Rolle spielt. Wohingegen gerade die Maximale und das Argmaximum der Gurtkraft besonders relevant sind. Bezogen auf die Eingaben ist gerade die Maximalkraft hoch und die Schaltzeit spät.

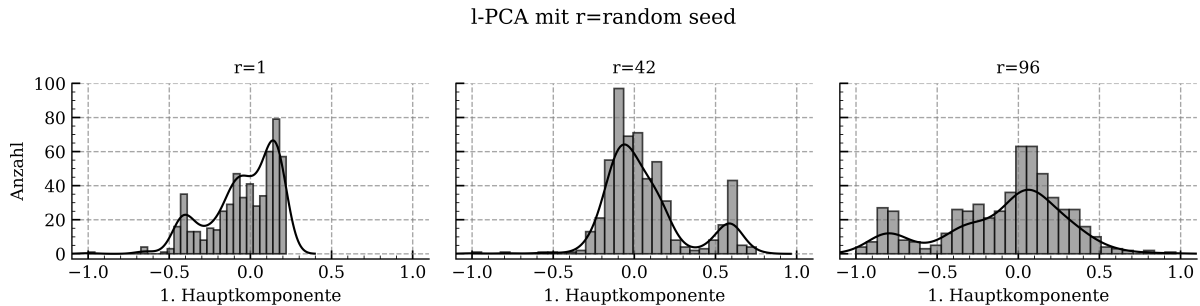
Diese Betrachtung zeigt, dass gerade für abhängige Eingabegrößen Vorsicht geboten ist. Auch wenn durch den Einsatz von CorrSHAP oder CohortSHAP das Streuband begrenzt werden kann, verbleiben Unsicherheiten, die es in der praktischen Anwendung abzuklären und zu diskutieren gilt.

#### 6.4.4.6 Vergleich Dimensionsreduktionsmethoden

In *ExPLAIN* nach Abbildung 5.9 auf S. 108, wurde erwähnt, dass es verschiedene Ansätze zur Dimensionsreduktion gibt, die auf das Voxelgrid angewendet werden können. Aufgrund linearer und deterministischer Eigenschaften wurde in der Use Case Betrachtung die PCA verwendet. Das Histogramm der ersten Hauptkomponente wurde bereits in Abbildung 6.48 auf S. 174 gezeigt. Deutlich ersichtlich ist, dass zwei Cluster herausgestellt werden, die sich im Rahmen der Use Case Betrachtung in Abschnitt 6.4.3 ab S. 173 auch als sinnvoll erwiesen haben. Mit Blick auf die Literatur wurde beispielsweise in [Tho+10; IG19; KGS20; Kra+23] mit direktem Bezug auf Crashsimulationen PCA ebenfalls als ein geeignetes Verfahren identifiziert.

Um diese Ergebnisse nun einzuordnen, werden vier weitere Dimensionsreduktionsmethoden auf das Voxelgrid angewendet. Hierzu zählen die logPCA [LL20], DiffMaps [CL06] und UMAP [MHM18]. Des Weiteren wird die in [IG19] vorgeschlagene Methode auf Basis des diskreten *Laplace-Beltrami-Operators* auf dem FE-Oberflächennetz angewendet. Ziel dieser Betrachtung ist es nun herauszustellen, welche Ergebnisse durch andere Verfahren ermittelt werden und wie robust diese Ergebnisse bezogen auf deren Hyperparameter sind. Die Dimensionsreduktion erfolgt für eine Dimension, die dann mittels einem Histogramm visualisiert wird. Der Verlauf der Dichte wird über eine Kernel Density Estimation ermittelt. Zur verbesserten Darstellung sind die Skalen der Einbettungsräume mit dem absoluten Maximum skaliert und auf den Median zentriert.

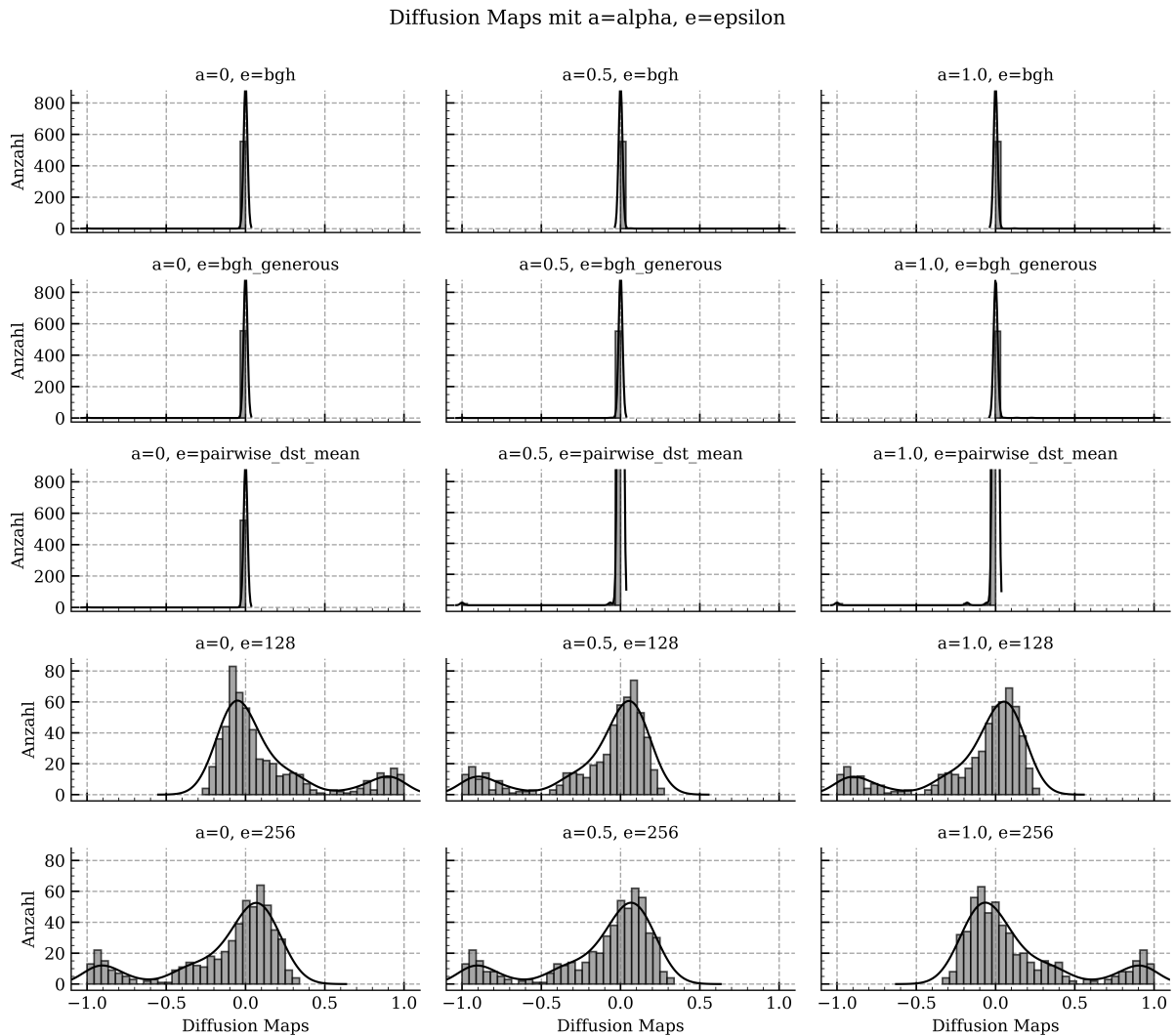
Die Ergebnisse für die **logPCA** sind Abbildung 6.68 zu entnehmen. Für den Optimierungsvorgang im Rahmen der logPCA kann ein Seed zur Initialisierung angegeben werden. Drei verschiedene Seeds [1, 42, 96] wurden dabei untersucht.



**Abbildung 6.68:** *ExPLAIN* Robustheit: Use Case III mit Dimensionsreduktion des Voxelgrids mit logPCA nach  $\mathbb{R}^1$

Insbesondere  $r=42$  zeigt mit der Basis-PCA vergleichbare Ergebnisse.  $r=1$  und auch  $r=96$  prägen ebenfalls zwei Cluster aus.  $r=1$  separiert das Knicken der horizontalen Stege im Schweller weniger deutlich. Im Fall von  $r=96$  sind die Cluster deutlich schlechter separiert. Das zweite Cluster ist als solches kaum zu erkennen. Eine leichte Abhängigkeit gegenüber dem Seed ist also festzustellen. Für die vorliegende Problemstellung bietet jedoch die extra für binäre Daten optimierte logPCA keinen wirklichen Vorteil, da sie identische Strukturen in den Daten wie die PCA herausstellt.

Die Histogramme für die **DiffMaps** sind in Abbildung 6.69 dargestellt. Im Gegensatz zur linearen PCA ermöglicht DiffMaps auch das Auffinden nichtlinearer Strukturen und Zusammenhänge in den Daten. Zwei relevante Hyperparameter,  $\alpha$  und  $\epsilon$ , werden untersucht, um die Robustheit zu bewerten. Der Parameter  $\epsilon$  definiert die Skalierung des Gauß-Kerns und legt damit die räumliche Skala fest, auf der die Ähnlichkeiten zwischen Datenpunkten quantifiziert werden. Kleinere Werte heben dabei die lokale Struktur hervor, während größere Werte zu einer globaleren Glättung führen. Der Parameter  $\alpha$  steuert die Normalisierung des Gauß-Kerns zur Korrektur von Punktdichteverzerrungen.  $\alpha=0$  bedeutet keine Korrektur, und  $\alpha=1$  sorgt für eine vollständige Dichteanpassung, um die zugrundeliegende Geometrie der Daten besser abzubilden. Zur Schätzung von  $\epsilon$  können verschiedene Heuristiken verwendet werden, die dabei helfen, problemunabhängig sinnvolle Einbettungsräume zu erzeugen. Für  $\epsilon$  werden hier die Methode nach [BH16] sowie dem Mittelwert der paarweisen Distanzen aller Datenpunkte herangezogen. Der Parameter  $\alpha$  wird dabei in den Werten [0, 0.5, 1.0] variiert. Die Breite für den Gauß-Kern ermittelt mit der Methode nach [BH16] (bgh) beträgt  $\epsilon_{\text{bgh}}=16$ . Dieser wird im Fall von (bgh\_generous) verdoppelt auf  $\epsilon_{\text{bgh\_generous}}=32$ .

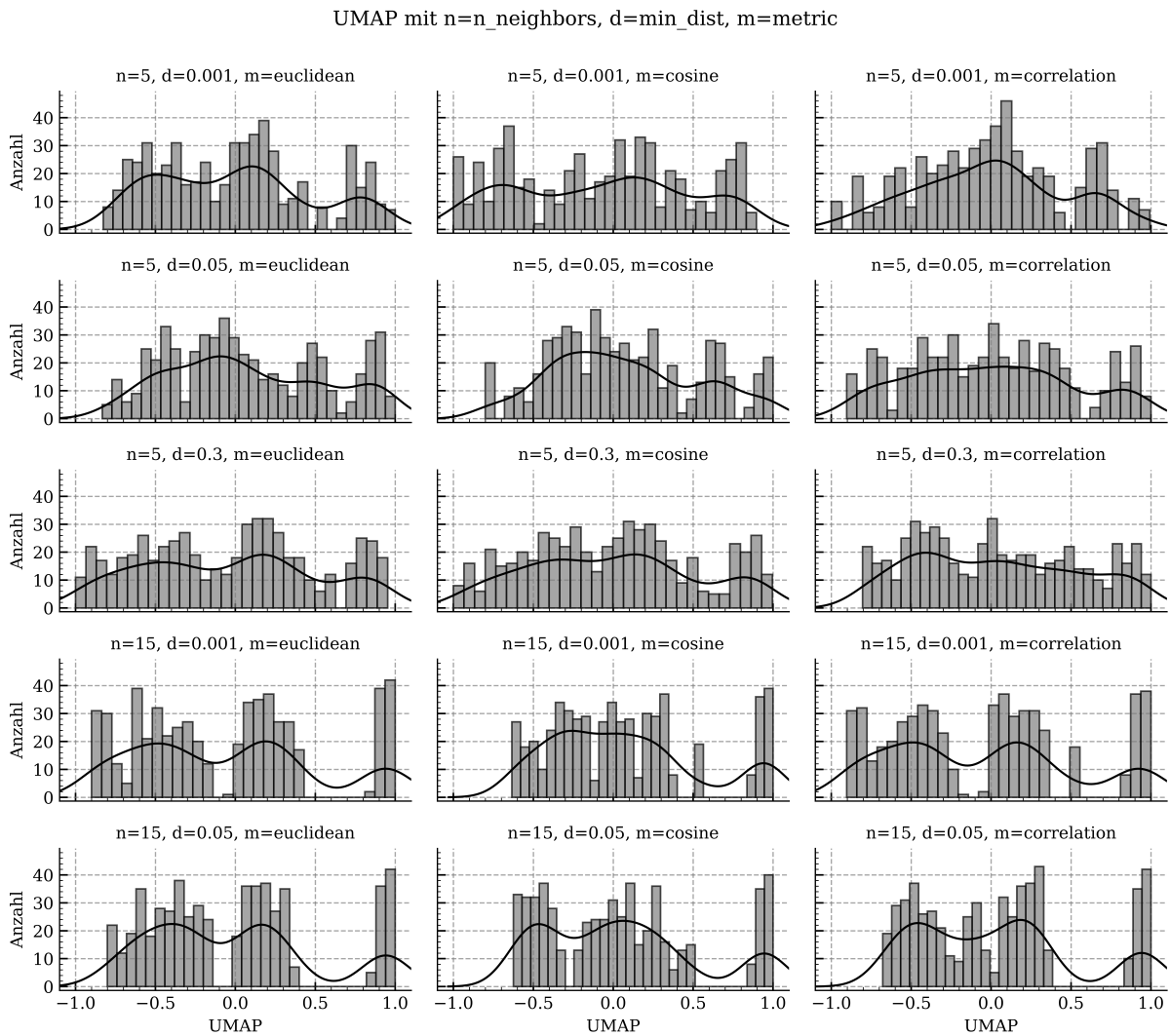


**Abbildung 6.69:** *ExPLAIN* Robustheit: Use Case III mit Dimensionsreduktion des Voxelgrids mit DiffMaps nach  $\mathbb{R}^1$

Bei Verwendung des Mittelwerts der paarweisen Distanzen ergibt sich  $\epsilon_{\text{pairwise\_dst\_mean}}=20.4$ . Beide Methoden liefern vergleichbare Ergebnisse bezogen auf die Breite des Gauß-Kerns. Bei Analyse der Ergebnisse in Abbildung 6.69 fällt jedoch auf, dass unabhängig von  $\alpha$  kein sinnvoller Einbettungsraum ermittelt wird. Die Clusterbildung ist aus Sicht der Domäne nicht nachvollziehbar und die Abstände nicht sinnvoll gewählt. Nahezu alle Simulationen fallen auf einen Wert, was keine sinnvolle Differenzierung und Interpretation erlaubt. Es zeigt sich, dass bei einer Vergrößerung von  $\epsilon$  – ebenfalls unabhängig von  $\alpha$  – identische Ergebnisse wie mit der PCA erzielt werden können. Auch hier bilden sich eindeutig zwei Cluster aus und trennen dabei die Simulationen von denen, deren horizontalen Stege versagen. Wichtig ist, dass die Ergebnisse damit nicht unabhängig der Hyperparameter sind. In der Praxis verkompliziert das den Einsatz, da erst ein sinnvoller Einbettungsraum

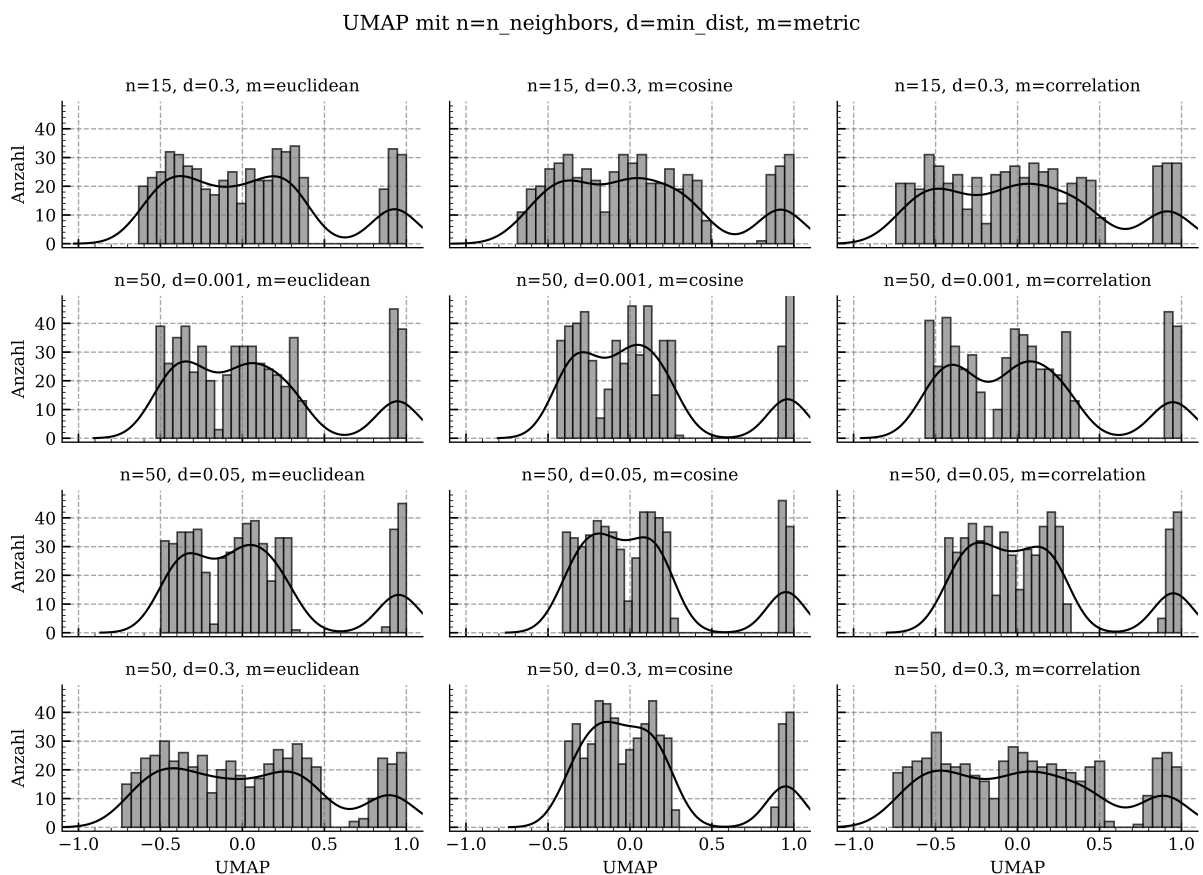
manuell ermittelt werden muss. Die Heuristiken können dabei als erste Basis dienen, auch wenn damit zu rechnen ist, dass für ähnliche Problemstellungen nicht direkt gut interpretierbare Einbettungsräume ermittelt werden.

Während DiffMaps die Datenstruktur als Diffusionsprozess auf einem Graphen modellieren, basiert **UMAP** auf topologischen Prinzipien und optimiert die Übereinstimmung der Nachbarschaften im hoch- und niedrigdimensionalen Raum. Die Histogramme für UMAP sind in Abbildung 6.70 und Abbildung 6.71 gezeigt. Untersucht werden hier die Anzahl einbezogener Nachbarn, minimale Distanz im Einbettungsraum sowie das Distanzmaß im ursprünglichen Raum. Die Anzahl der Nachbarn (`n_neighbors`), spezifiziert die Zahl der nächsten Datenpunkte, die zur Erfassung der lokalen Datenstruktur herangezogen werden.



**Abbildung 6.70:** *ExPLAIN* Robustheit: Use Case III mit Dimensionsreduktion des Voxelgrids mit UMAP nach  $\mathbb{R}^1$  – Teil 1

Kleinere Werte legen die Betonung auf lokale Details und größere Werte auf globale Zusammenhänge. Untersucht werden daher [5, 15, 50] Nachbarn. Der Minimalabstand im Einbettungsraum (`min_dist`) reguliert den minimalen Abstand zwischen eingebetteten Datenpunkten und beeinflusst folglich die Dichte und Trennung von Clustern. Die Distanzmetrik (`metric`) legt fest, wie Abstände im ursprünglichen Raum gemessen werden. Die euklidische Distanz (`eukclidean`) misst dabei die geometrische Entfernung zwischen zwei Vektoren und berücksichtigt sowohl den Betrag als auch die Richtung. Im Gegensatz dazu bewertet die Kosinus-Distanz (`cosine`) ausschließlich den Winkel zwischen den Vektoren. Die Korrelations-Distanz (`correlation`) kompensiert Mittelwertverschiebungen und erfasst mithilfe des Pearson-Korrelationskoeffizienten ausschließlich lineare Zusammenhänge.



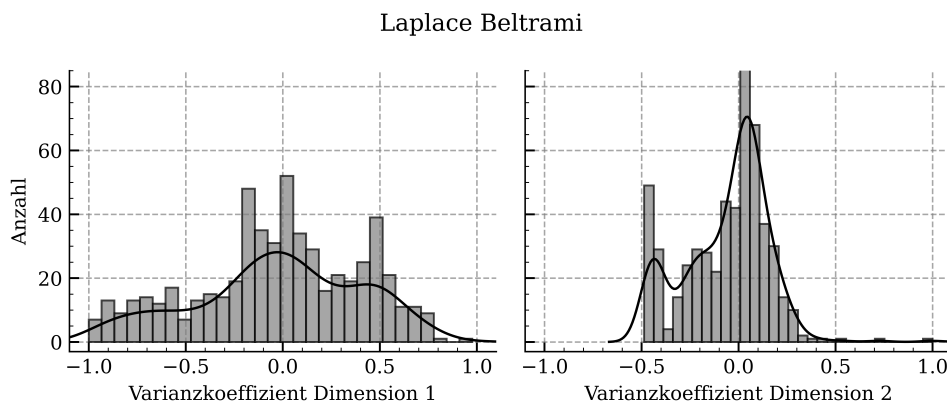
**Abbildung 6.71:** *ExPLAIN* Robustheit: Use Case III mit Dimensionsreduktion des Voxelgrids mit UMAP nach  $\mathbb{R}^1$  – Teil 2

In Abbildung 6.70 und Abbildung 6.71 ist ersichtlich, dass auch hier verschiedene Einbettungen in Abhängigkeit der Hyperparameter entstehen. Bei der Mehrheit ist jedoch eine deutlichere Trennung der Cluster sowie die Spezifikation von drei Clustern zu beobachten. Das deutet darauf hin, dass durch die Nichtlinearität neue Strukturen hervorgehoben werden können, die durch die anderen Verfahren nicht offengelegt werden konnten. Ergeb-

nisse mit einer geringeren Anzahl Nachbarn (5) liefern ein stark geglättetes Ergebnis und sind daher nur schwer deutbar. Für  $n=5$  liefert lediglich die Kombination mit  $d=0.001$  und einem Euklidischen Distanzmaß eine deutlichere Clusterausprägung. Hier sorgt die Verwendung der Kosinus- oder Korrelationsdistanz für eine zusätzliche Glättung des Einbettungsraums. Bei Erhöhung der Anzahl berücksichtigter Nachbarn ( $n=15$  und  $n=50$ ) werden die Clusterausprägungen deutlicher. Ebenso schwindet die Abhängigkeit zur verwendeten Distanzmetrik. Tendenziell ist für das euklidische Distanzmaß eine etwas bessere Separierung der Cluster erkennbar, beispielsweise für  $n=15$  und  $d=0.001$  oder  $d=0.05$ . Die deutlichste Trennung von Clustern ist für  $n=50$  zu erkennen.

Folglich sind auch die Ergebnisse für UMAP von den verwendeten Hyperparametern abhängig. Für jede Problemstellung ist entsprechend eine separate Ermittlung der Hyperparameter erforderlich. Verglichen mit den Ergebnissen der DiffMaps wird hier kein komplettes Versagen des Verfahrens festgestellt, das alle Datenpunkte einem Wert im Einbettungsraum zuordnet.

Zur weiteren Einordnung der Einbettungsräume, die aus der Dimensionsreduktion des Voxelgrids resultieren, wird auch der **Laplace-Beltrami-Operator** aus [IG19] betrachtet. Die Methode wird aber nicht auf das Voxelgrid angewendet, sondern auf das FE-Netz direkt. Abrufbar ist diese in dem kommerziellen Software-Tool SimExplore von Fraunhofer SCAI [Fra24]. Die Ergebnisse für die ersten beiden Dimensionen sind in Abbildung 6.72 gezeigt.



**Abbildung 6.72:** *ExPLAIN* Robustheit: Use Case III mit Dimensionsreduktion des FE-Netzes mit Laplace-Beltrami nach  $\mathbb{R}^1$

Die erste Dimension des Einbettungsraums ähnelt der, die mittels UMAP (mit  $n = 5$  und euklidischem Abstandsmaß) bestimmt wurde. Drei Bereiche zeichnen sich ab, wobei keine klaren Ausprägungen der Cluster erkennbar sind. Das kann darauf hindeuten, dass Effekte vorwiegend lokal abgebildet werden. Interessant ist, dass die zweite Dimension des

Einbettungsraums eine gute Übereinstimmung mit den Ergebnissen der PCA aufweist. Hier sind die beiden Cluster deutlich voneinander getrennt. Folglich ist hier eine saubere Trennung linearer und nichtlinearer Effekte möglich.

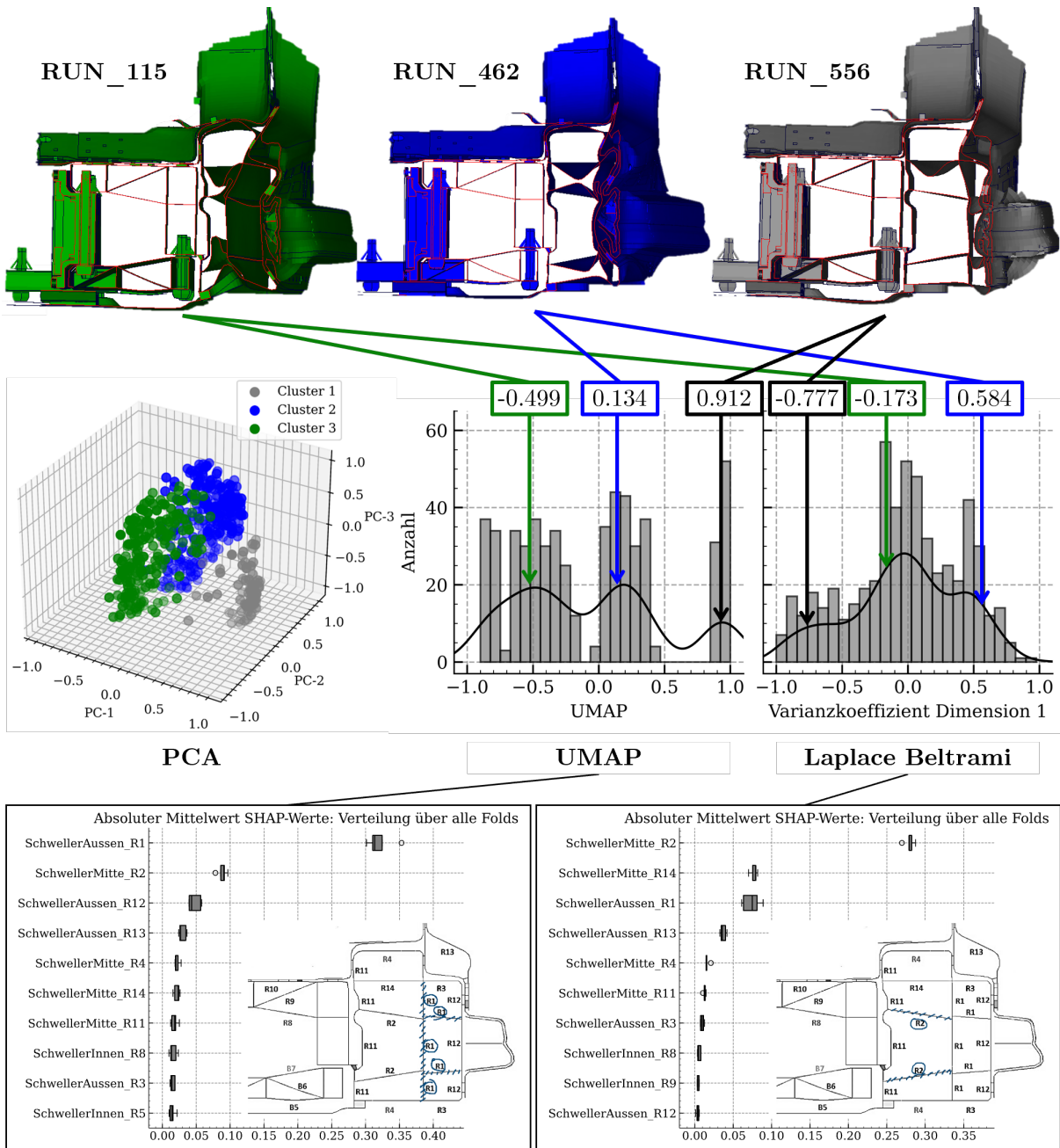
Zusammenfassend lässt sich feststellen, dass die Dimensionsreduktion mit verschiedenen Methoden abseits der initial verwendeten PCA realisiert werden kann. Insbesondere die nichtlinearen Verfahren wie UMAP oder DiffMAP weisen eine Abhängigkeit zu den gewählten Hyperparametern auf. Für den Fall einer realen Anwendung ist hier eine automatische Optimierung oder interaktive Darstellung der Verteilungen denkbar, da Ergebnisse wie die DiffMaps bei Schätzung der Breite des Gauß-Kerns mit bgh keinen praktischen Mehrwert bieten. Die Ergebnisse aus der Dimensionsreduktion des Voxelgrids wurden zudem mit einem grundlegend anderen Ansatz auf Basis von Laplace-Beltrami auf dem FE-Oberflächennetz reproduziert.

Automatische Methoden zur Ausreißerererkennung in FE-Daten wie in [Kra+23] müssen daher sicherstellen, dass die Ergebnisse der Dimensionsreduktion schlüssig sind. Ansonsten ist eine Bestimmung auffälliger Datenpunkte, beispielsweise des mittels MAD, nicht mehr sinnvoll möglich.

#### 6.4.4.7 Vergleich Einbettungsräume

Zur Anknüpfung an den vorangegangenen Abschnitt wird nun untersucht, inwiefern die Ergebnisse auch aus Sicht der Domäne vergleichbar sind. Hierfür wird weiterhin Use Case III nach Abbildung 6.47 auf S. 173 betrachtet. In Abbildung 6.73 sind die drei Simulationen dargestellt, die bereits schon Teil der Betrachtung von Clustern im 3D-Einbettungsraum in Abschnitt 6.4.3.2 ab S. 177 waren.

Der dort betrachtete Einbettungsraum basierte auf einer linearen PCA, wobei drei separate Cluster identifiziert werden konnten. Im Gegensatz dazu waren im 1D-Einbettungsraum der PCA (vgl. Abbildung 6.48 auf S. 174) lediglich zwei Cluster ersichtlich. Im Kontext der Ergebnisse aus dem Vergleich verschiedener, auch nichtlinearer, Dimensionsreduktionsmethoden ist zu sehen, dass bereits im 1D-Einbettungsraum die Information der 3D PCA-Einbettung abgebildet werden kann. Abbildung 6.73 zeigt, dass die drei Verhaltensweisen auch in den nichtlinearen Einbettungsräumen dargelegt werden. Hierfür sind 1D-Einbettungen mittels UMAP (Hyperparameter  $n=15$ ,  $d=0.001$  und  $m=euclidean$ ) und des Laplace-Beltrami-Operators sowie die 3D Referenz mittels PCA gezeigt.



**Abbildung 6.73:** *ExPLAIN* Robustheit: Use Case III Vergleich der 1D-Einbettungsräume aus PCA, UMAP und Laplace-Beltrami – global

Für UMAP und Laplace-Beltrami ist jeweils die Position der drei Beispielsimulationen – repräsentativ für drei Verhaltensweisen – farbig hervorgehoben. Auch wenn die Methoden in der Lage sind identisches Verhalten in Clustern zu aggregieren, ist die Reihenfolge und Anordnung im Raum unterschiedlich. Die deutlichste Trennung wird durch UMAP realisiert. Allerdings ist für dieses Verfahren die Wahl der Hyperparameter nicht trivial und muss im Zweifelsfall manuell erfolgen. Die Dimensionsreduktion mit dem Laplace-Beltrami-Operator ist weniger trennscharf, dafür jedoch deterministisch ermittelbar.

Um die Einbettungsräume besser zu verstehen, kann eine Erklärung mit *ExPLAIN* bereitgestellt werden. Das wurde bereits für den linearen PCA-Einbettungsraum ausführlich erläutert, und wird nun auf nichtlineare Dimensionsreduktionsverfahren untersucht. Im unteren Teil von Abbildung 6.73 sind die globalen Sensitivitäten auf Basis von absoluten mittleren SHAP-Werten für die zehn wichtigsten der 14 Eingaben in Use Case III dargestellt. Die Verteilung ist hier erneut über die fünf Folds der k-fold Cross Validation gegeben. Während die Zusammenhänge in UMAP vergleichbar mit denen der PCA-Einbettung sind, liegen für Laplace-Beltrami andere Zusammenhänge vor, die die Anordnung begründen.

Für UMAP ist R1 nun maßgeblich, wobei R2 weiterhin als relevant identifiziert wird. Die für den Einbettungsraum des Laplace-Beltrami-Operators wichtigste Eingabe ist R2. R1 wird hier erst an dritter Stelle gelistet. Zudem ist die Wandstärke R14 an zweiter Stelle aufgeführt und damit bedeutsam. Für die UMAP und PCA Einbettung hatte diese nur mittleren Einfluss. Bei Betrachtung des horizontalen Deformationsverhaltens, das maßgeblich durch R2 getrieben wird, sind also die mittels Laplace Beltrami erzielten Ergebnisse am repräsentativsten. Das geht zudem mit den im Verfahren getroffenen Annahme der isometrischen Deformation einher, wohingegen die voxelbasierten Einbettungen nur räumliche Information darstellen. *ExPLAIN* kann folglich auf Knopfdruck Informationen über den Aufbau des sonst intransparenten Einbettungsraum geben.

Abbildung 6.74 zeigt den Vergleich von RUN\_556 und RUN\_462 relativ zu RUN\_115 mittels Differenz-SHAP. Wichtig ist hierbei, dass entsprechend der globalen Sensitivitäten auch die lokale Betrachtung Unterschiede aufweist, trotz des Vergleichs derselben Simulationen. Es gilt also zu beachten, dass die Erklärung auch an den Begebenheiten des Einbettungsraums abhängt.

Zur Nachvollziehbarkeit, beispielsweise einer Ausreißerdetektion, ist es relevant, unterschiedliche oder identische Zusammenhänge in den Einbettungsraums darlegen zu können. Wird bezogen auf eine Referenzsimulation ein Ausreißer im Einbettungsraum identifiziert, stellt *ExPLAIN* Einflüsse bereit, die erklären, warum die Simulation als Ausreißer identifiziert wird.

Im Fall von nicht plausiblen Ergebnissen, ist es nun auch möglich den Einbettungsraum und die dort abgebildeten Zusammenhänge zu hinterfragen. Angesichts der nicht garantiert sauberen und intuitiven Verteilung, wie beispielsweise für DiffMaps in Abbildung 6.69 auf S. 199, erlaubt die nun hergestellte Transparenz ein besseres Verständnis. Das Verständnis über die Zusammenhänge in den Einbettungsräumen der Simulationsdaten war bisher nur aufwändig manuell möglich.

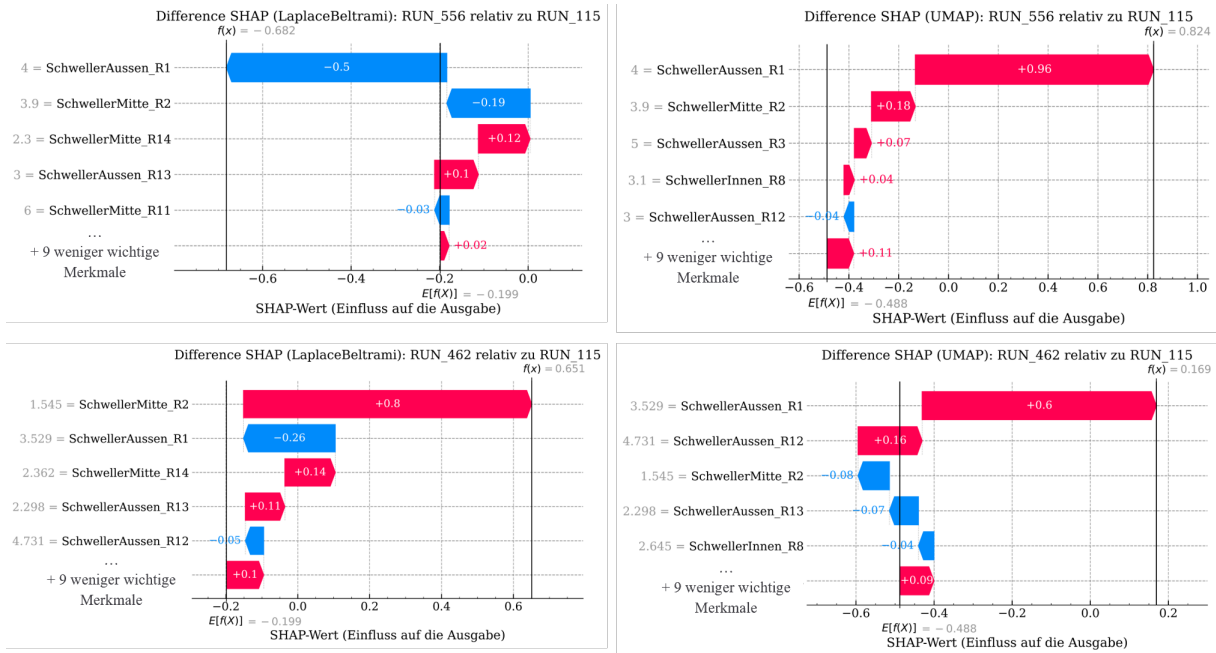


Abbildung 6.74: *ExPLAIN* Robustheit: Use Case III Vergleich der 1D-Einbettungsräume aus UMAP und Laplace-Beltrami – lokal

## 6.5 Zusammenfassung

Im Rahmen von Kapitel 6, das der Deskriptiven Studie II (Phase VI) des Aufbaus nach der DRM in Abbildung 1.1 auf S. 6 entspricht, wurden die neu eingeführten entwicklungsunterstützenden Maßnahmen – *RAMEn*, *DrAGOS*, *DAYSi* und *ExPLAIN* – an repräsentativen Use Cases aus der Entwicklung passiver Sicherheitssysteme evaluiert.

Die Erkenntnisse dieses Kapitels, insbesondere in Bezug auf den praktischen Mehrwert, können folgendermaßen zusammengefasst werden:

- *RAMEn* wurde anhand der strukturierten Bereitstellung von Wissen für den effektiven Einsatz datengetriebener Methoden evaluiert. Dies wurde mittels der drei im Rahmen dieser Arbeit eingeführten Methoden sowie weiteren Beispielen aus dem Stand der Technik gezeigt. Der praktische Mehrwert hiervon wurde zusätzlich durch eine Umfrage untermauert. Diese legt offen, dass das Interesse und die Motivation für den Einsatz datengetriebener Methoden vorhanden ist, da deren Potenzial als hoch eingeschätzt wird. Andererseits fehlt jedoch das nötige Wissen über Methoden und deren Verwendung zum aktuellen Zeitpunkt. Ebenfalls wurde gezeigt, dass sich die zugrundeliegende Idee von *RAMEn* für die Leichtbauentwicklung generalisieren lässt.
- Die Auslegung von Rückhaltesystemen ist ein repräsentatives Beispiel für die Produktgenerationsentwicklung. Hieran wurde verdeutlicht, dass durch den Einsatz von *DrAGOS* dateneffizient und automatisch Verbesserungen im Systemverhalten – auch für ungesehene Problemstellungen wie die Verwendung anderer ATD-Typen oder Sitzpositionen – vorgenommen werden können. Dies gilt insbesondere auch, wenn rechenintensive FE-Simulationen verwendet werden. Die Evaluierung zeigt, dass die Optimierung mittels *DrAGOS* weniger als zehn Schritte benötigt und sogar in der Lage ist bessere optimale Lösungen als ein konventioneller globaler Optimierungsalgorithmus aufzufinden. Die durchgeführten Robustheitsuntersuchungen attestieren zudem die numerische Stabilität des RL-Agenten in der Rückhaltesystemauslegung.
- Die Evaluierung von *DAYSi* anhand von FE-Simulationen für den Front- und Seitenaufprall zeigt, dass für die Auswertung detaillierte Informationen über das Verhalten des Systems auch über die 1D-Zeitreihendaten ermittelt werden können. In allen drei Use Cases wurden auffällige Datenpunkte korrekt identifiziert und die dafür verantwortlichen Ursachen hervorgehoben. Gerade im Vergleich zu den mehrdimensionalen FE-Daten sind die ISO-MME-Daten speichereffizient und die mittels *DAYSi* berechneten Ergebnisse sind innerhalb weniger Minuten verfügbar. Das senkt die Hemmschwelle zur Verwendung der Methode signifikant. Zudem bleiben Ergebnisse auch bei der Verwendung unterschiedlich vieler Simulationen oder Sensoren konsistent. Für den Anwender wurde ein Satz stabiler Verfahrensparameter ermittelt.

- *ExPLAIN* wurde mithilfe von drei verschiedenen Use Cases aus Fahrzeugstruktur- und Rückhaltesystemauslegung evaluiert. Untersucht wurde dabei insbesondere die Modularität von Skalar-, 1D-Zeitreihen- und FE-Datenverarbeitung. Es wurde gezeigt, dass die intuitive Darstellung auf Basis von SHAP auf verschiedenen Systemebenen sowie der A/B-Vergleich neue Einblicke in die Simulationsergebnisse ermöglicht. Dies gilt insbesondere für die Erklärung von Einbettungsräumen und Clusterlabeln von Deformationsverhalten einzelner Subsysteme. *ExPLAIN* eignet sich zudem für die interaktive Auswertung, was die Integration in ein interaktives Tool vereinfacht. Der Vergleich verschiedener Metamodelle und Verfahren der Sensitivitätsanalyse attestiert auch hier robustes Verhalten bezogen auf die Reihenfolge und Richtung der Wichtigkeit. Sind die Eingaben abhängig können Unsicherheiten mit Corr- und CohortSHAP reduziert werden. Vorsicht ist im Hinblick auf die quantitative Ausprägung geboten.

Alle vier Konzepte tragen damit auf unterschiedliche Weise zur Optimierung der Entwicklungsprozesse in der passiven Fahrzeugsicherheit bei. Zudem ergänzen sich die Ansätze: Beispielsweise können die durch *DrAGOS* durchgeführten Simulationen zur optimalen Auslegung des Rückhaltesystems anschließend mit *DAYSi* auf Auffälligkeiten überprüft werden. Für den Fall, dass auffällige Signale ermittelt werden, kann durch *ExPLAIN* der Einfluss geänderter Systeme – beispielsweise auf Maximalwerte des Signals – genauer untersucht werden.

Teil der Evaluierung waren außerdem Aspekte wie Generalisierbarkeit und Robustheit, um die Übertragung auf ähnliche Problemstellungen für den Anwender so einfach wie möglich zu gestalten.

Die theoretischen Aspekte aus Kapitel 5 ab S. 85 und die gerade durchgeführte Evaluierung liefern die Grundlage für die im Folgenden stattfindende Diskussion unter Einbindung des Stands der Technik aus Kapitel 3 ab S. 43.

# 7 Diskussion

Die in diesem Kapitel abgehandelten Aspekte entsprechen ebenfalls der Deskriptiven Studie II (Phase IV) des Aufbaus nach der DRM in Abbildung 1.1 auf S. 6. Die neu eingeführten entwicklungsunterstützenden Maßnahmen – *RAMEn*, *DrAGOS*, *DAYSi* und *ExPLAIN* – werden nun anhand bestehender Literatur und der Ergebnisse der Evaluierung in Kapitel 6 eingeordnet und diskutiert. Ebenfalls werden Schwachstellen und Potenziale mit dem Blick auf zukünftige Arbeiten abgeleitet.

## 7.1 *RAMEn* – Methodenauswahl

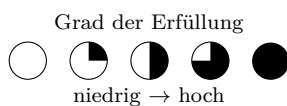
Nach der Einführung von *RAMEn* in Abschnitt 5.1 ab S. 87 und der Evaluierung in Abschnitt 6.1 ab S. 117 erfolgt nun die Diskussion anhand der Literatur aus Abschnitt 3.1 ab S. 44. Ebenfalls wird auf den Vergleich und die Einordnung bestehender Arbeiten in Tabelle 4.1 auf S. 73 verwiesen.

### 7.1.1 *RAMEn* im Kontext bestehender Literatur

Zunächst wird *RAMEn* in den Kontext bestehender Literatur anhand von Tabelle 7.1 eingeordnet.

Tabelle 7.1: *RAMEn* im Kontext bestehender Literatur

	Ebene des Prozessmodells	Typ des Prozessmodells	Aktivitäten- vs. Phasenbasiert	Auswahl von Methoden	Prozessintegration von Methoden	Einfache Anwendung	Generalisierbarkeit	Bezug zu einem Anwendungsfeld
<i>RAMEn</i>	ME	AB	P+A					



ME = meso, AB = abstrakt, A=aktivitätenbasiert, P=phasenbasiert

Das Vorgehensmodell selbst ist durch den zugrundeliegenden Problemlösezyklus und das Phasenmodell auf der *Mesoebene* verortet. Die Herleitung der Zusammenhänge über die Schnittfeldbetrachtung ist *abstrakt*. Ein Mehrwert wird insbesondere durch die Auswahl

von Methoden im Prozessmodell und die ausführliche Betrachtung der Prozessintegration – wie lassen sich datengetriebene Methoden wo anwenden – gegeben. Entsprechend bietet das Prozessmodell einen bisher nicht realisierten Blickwinkel auf die Entwicklung in der passiven Fahrzeugsicherheit.

Darüber hinaus wird durch *einfache Dokumentationsvorschriften*, die sich beispielsweise in Confluence [Atl25] mit geringem Aufwand umsetzen lassen, die Hemmschwelle für die Verwendung eines solchen Ansatzes innerhalb einer Organisationseinheit abgesenkt. Gerade dieser Aspekt stellt für ontologiebasierte Ansätze wie in [GGW23] eine wesentliche Hürde dar, da deren Einführung und Wartung mit erheblichem Aufwand verbunden sind.

Wie durch die Umfragen verdeutlicht, ist zu erwarten, dass die strukturierte *Bereitstellung von Wissen* einen großen Hebel aufweist, da grundsätzliches Interesse und ungenutztes Potenzial vorhanden ist. Eine zusätzliche Motivation stellen die in *RAMEn* klar formulierten Ziele und Ergebnisse dar. Durch die abstrakten Überlegungen, die die Grundlage für *RAMEn* bilden, konnte gezeigt werden, dass eine Generalisierung möglich ist, und so zusätzlicher Mehrwert generiert werden kann.

Anhand der spezifischen Formulierung für das *Anwendungsfeld* der Entwicklung crashbelasteter Systeme in der passiven Fahrzeugsicherheit wurde die Flexibilität gezeigt, lokal relevante Aspekte, wie die Systemebene, mit einzubeziehen. Diese Flexibilität ist in bestehenden Prozessmodellen nur bedingt gegeben und wird nicht mit Bezug zur Anwendung vertieft diskutiert.

In den betrachteten Ansätzen [Kle+18; Meh+21] im Stand der Technik bildet der in den *Datenwissenschaften* weit verbreitete CRISP-DM [WH00] die Grundlage für die Überlegungen. Ursprünglich ist dieses Modell zum Entwickeln datengetriebener Methoden selbst gedacht und dient daher zur Strukturierung von Softwareentwicklungsprojekten. *RAMEn* hingegen fokussiert sich auf die Einordnung, Auswahl und Prozessintegration bereits trainierter ML-Modelle bzw. den darauf aufbauenden Entwicklungsmethoden. Folglich ergänzen sich die beiden Modelle und können gemeinsam verwendet werden. Basierend auf identifizierten Leerstellen in *RAMEn* können Ideen generiert werden, die als Eingabe für den CRISP-DM zum Aufbau eines Geschäftsverständnis verhelfen können. Die Hauptmerkmale, die die lokale Situation in der Anwendung abbilden, können zudem ergänzend genutzt werden. Aspekte im CRISP-DM, wie das Verständnis von Markt und Kunden, müssen jedoch aus weiteren Quellen aggregiert werden. Eine auf Basis von CRISP-DM entwickelte Methode kann dann in *RAMEn* integriert und in der Praxis effektiv eingesetzt werden.

### 7.1.2 Beantwortung der Forschungsfrage

Die in Abbildung 1.1 auf S. 6 gestellte Frage – *Wie lässt sich die situative Auswahl und Integration von Methoden in Entwicklungsprozessen unterstützen?* – kann mit den gewonnenen Erkenntnissen folgendermaßen beantwortet werden:

- Die theoretische Grundlage dafür, wie datengetriebene Methoden ausgewählt und in Prozesse integriert werden können, um Entwicklungsprozesse zu unterstützen, wird durch die eingeführte Betrachtungsweise zur *Orientierung komplexer Schnittfelder* bereitgestellt. Das Auftrennen der Felder innerhalb der Aktivitäten ermöglicht eine systematische Darstellung, welche Unterstützung in welchem Bereich erfolgt. Zudem wird die Anwendung durch die Einführung von Hauptmerkmalen feiner aufgelöst. Diese erlauben es, neben der in [VDI 2221-1; VDI 2221-2] berücksichtigten Zeit, auch weitere Aspekte – wie Systemebenen, Projekte und Disziplinen – mit einzubeziehen. Dies ist gerade für die praktische Anwendung relevant, da so auch ein Übertrag von Erkenntnissen darstellbar wird.
- Um sowohl bestehende Entwicklungsprozesse, wie den in der Automobilentwicklung weit verbreiteten *Stage-Gate-Prozess*, als auch die Ingenieursaktivitäten selbst einzubeziehen, erfolgt eine Auftrennung zur Berücksichtigung beider Aspekte. Zusammenhänge werden dabei durch die Durchführung einer Aktivität in der Anwendung zur Lösung einer Aufgabe sichergestellt. Das sorgt für Kompatibilität und Akzeptanz in der Praxis. Festgehalten wird dieser Zusammenhang über das eingeführte Dokumentationsschema. Untermuert wird dies durch eine Umfrage, die einerseits das Potenzial, aber auch den Bedarf an einer besseren Abrufbarkeit und Verbreitung datengetriebener Methoden hervorhebt.
- Um eine bestmögliche *Prozessintegration* datengetriebener Methoden zu gewährleisten, wurde der Aspekt der Anwendung – also wie die Methoden eingesetzt werden – praxisnah an der Auslegung crashbelasteter Systeme in der passiven Fahrzeugsicherheit diskutiert. Dafür wurden im Rahmen einer ausführlichen Literaturrecherche drei Hauptmerkmale identifiziert: Systemebene, Zeit und die Organisationsstruktur (Projekt, Disziplin, Aufgabe). Aufbauend auf diesen Überlegungen wird dies durch die erwähnte Dokumentation, sowie die Bereitstellung von Ziel und Ergebnis unterstützt. Dies bildet die Grundlage für eine zielgerichtete Kommunikation und fördert somit Akzeptanz und Motivation im praktischen Einsatz.
- Zuletzt konnte gezeigt werden, dass die abstrakten Überlegungen einfach generalisiert werden können. Dies wurde am Beispiel der datengetriebenen Methoden für die Leichtbauentwicklung diskutiert, sodass auch ein Mehrwert für andere Anwendungsfelder geschaffen werden kann.

### 7.1.3 Einfluss auf Bewertungskriterien

Zu Beginn der Arbeit wurden in Abschnitt 1.3.1 auf S. 4 globale Bewertungskriterien eingeführt. Basierend auf der Evaluierung lässt sich insbesondere der positive Einfluss auf das *Wissen* bei beteiligten Ingenieuren und Organisationen hervorheben. *RAMEn* leistet einen Beitrag zur Wissensgenerierung und -bewahrung, was in Konsequenz zur Steigerung der Effizienz hinsichtlich *Entwicklungszeit* und *Entwicklungskosten*, aber auch der *Innovationsfähigkeit* beiträgt [GB18]. Ebenfalls wird die Grundlage für den Einsatz von datengetriebenen Methoden wie *DrAGOS*, *DAYSi* und *ExPLAIN*, die bei der Optimierung der *technischen Wertigkeit* oder der Sicherung von (*Prozess- und Produkt-*) *Qualität* unterstützen, geschaffen.

### 7.1.4 Schwachstellen und zukünftige Potenziale

Da das Vorgehensmodell auf dem Problemlösezyklus und dem Phasenmodell der [VDI 2221-1; VDI 2221-2] aufbaut, sind die Schwachstellen der Grundlage zu beachten. Hierzu zählt der starke Fokus auf mechanische Produkte und die folglich zurückbleibenden mechatronischen oder Software-Produkte (vgl. [VDI 2206]). Ebenso fehlen dem Modell Aspekte einer *agilen und konvergenten Entwicklung*, sowie die Überprüfung des Problemverständnisses, das separat in einer vorgelagerten Phase behandelt wird [KW23].

Innerhalb der vorliegenden Arbeit wurde das vorgeschlagene Vorgehensmodell anhand exemplarischer Beispiele zur systematischen Einordnung und Auswahl von Methoden, der Diskussion hinsichtlich der *Bereitstellung von Information* zur Wissensgenerierung sowie der Generalisierungsbetrachtungen evaluiert. Eine endgültige Validierung von *RAMEn* stellt jedoch eine große Herausforderung dar und ist mit erheblichem Aufwand verbunden. Dies ist mit den langen Entwicklungszyklen, die in der Automobilindustrie typischerweise mehrere Jahre dauern, zu begründen. Hinzu kommt die für die Validierung notwendige Kontrollgruppe in derselben Organisationseinheit. Dieser müssten Methoden und Informationen vorenthalten werden.

Weiterhin offen bleiben die in Tabelle 7.1 nicht vollständig ausgefüllten Kreise. Hinsichtlich der Auswahl von Methoden gilt es, *automatisierte und intelligente Ansätze* – beispielsweise LLMs – hinzuzuziehen und eine noch einfachere Auswahl zu realisieren. Ein erstes Konzept wurde hierfür bereits in [Mat+24] dargelegt. Erste vielversprechende Perspektiven wurden in [SLN25] sowie in eigenen Folgearbeiten [Mat+25] identifiziert. Der Vorteil hierbei ist die interaktive Darstellung der Information, die jederzeit durch den Ingenieur hinterfragt werden kann. Ebenso wird die Dokumentationserstellung signifikant vereinfacht, da Erklärungen automatisch hinzugefügt werden können [Mat+25]. Folglich besteht in Bezug auf den in Tabelle 7.1 auf S. 209 aufgeführten Punkt der einfachen Anwendung noch

---

ungenutztes Potenzial, das zukünftig gehoben werden kann. Ebenfalls könnten bisher zu aufwändige Ansätze im Bereich der Ontologien dadurch besser praktikabel werden.

## 7.2 *DrAGOS* – Datengetriebene Systemauslegung

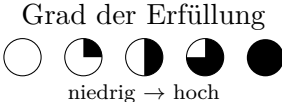
Nach der Einführung von *DrAGOS* in Abschnitt 5.2 ab S. 94 und der Evaluierung in Abschnitt 6.2 ab S. 126 erfolgt nun die Diskussion anhand der Literatur aus Abschnitt 3.3 ab S. 53. Außerdem wird auf den Vergleich und die Einordnung bestehender Arbeiten in Tabelle 4.2 auf S. 76 verwiesen.

### 7.2.1 *DrAGOS* im Kontext bestehender Literatur

Zunächst wird *DrAGOS* in den Kontext bestehender Literatur anhand von Tabelle 7.2 eingeordnet.

**Tabelle 7.2:** *DrAGOS* im Kontext bestehender Literatur

	Verwendung von RL-Agenten	Optimierer steuert Metamodell	Optimierer steuert FE-Simulation	Wissenstransfer im Entwicklungsprozess	Integration in den Entwicklungsprozess	Industrienahes Evaluationszenario
<i>DrAGOS</i>	●	●	●	◐	◐	●

Grad der Erfüllung  
  
 niedrig → hoch

Im Kontext der Literatur beschäftigt sich lediglich *DrAGOS* mit der Verwendung von RL-Agenten und einer *Kombination aus Metamodell und FE-Simulation*.

Anhand der Use Cases wurde gezeigt, dass eine vortrainierte Strategie aufgaben- und produktgenerationsübergreifend eingesetzt werden kann. Das entspricht einem *Wissenstransfer* auf andere Problemstellungen, was neben einer Effizienzsteigerung auch das Binden von Wissen an Organisationen ermöglicht.

Die Integration von *DrAGOS* in Entwicklungsprozesse wird durch die Einbettung in die *Produktgenerationsentwicklung* vorgenommen.

Ebenfalls werden die Erkenntnisse in einem *industrienahen Szenario* diskutiert, was einen Mehrwert in Bezug auf das praktische Potenzial bietet, da bisher vorrangig akademische Beispiele im Kontext RL-basierte Strukturoptimierung betrachtet wurden.

Die direkte Interaktion mit Simulationen, wie beispielsweise in der Arbeit von [HO20; HO22; Bro+22] während des Trainingsprozesses des RL-Agenten, ist in diesem Szenario nicht realisierbar, da pro Schritt ein umfangreicher Rechenaufwand von bis zu 20 Stunden

auf einem Hochleistungscluster erforderlich ist. Dies würde hier zu Trainingszeiten von mehreren Jahren führen.

Während ausgehend von eigenen Arbeiten [Mat23] mittlerweile auch weitere Literaturquellen [BGS23; BGS24a; BGS24b] die Verwendung von Metamodellen im Kontext des RL-Agenten Trainings verwenden, fehlt in diesen der Bezug zur Anwendung auf FE-Simulationen. Darüber hinaus bieten die im Rahmen dieser Arbeit durchgeführten Untersuchungen zur Robustheit des PPO-RL-Agenten, sowie die Durchführung des Benchmarks mit einem klassischen Optimierer einen weiteren Mehrwert. Diese Benchmarks wurden bisher in erster Linie an statischen Problemstellungen wie in [HO20; HO22; Bro+22] durchgeführt.

Nebenbei wurde im Rahmen der Evaluierung von *DrAGOS* auch gezeigt, dass die generalisierte Trainingsbasis und der *In-Context-Learning-Ansatz* von TabPFN [Hol+25] auch für Use Cases im Bereich der Fahrzeugsicherheit einen Mehrwert hinsichtlich der Genauigkeit bieten können. Gegenüber dem besten klassischen Metamodell (XGBoost) ist die Güte im Bestimmtheitsmaß  $R^2$  um weitere zwei bis fünf Prozent besser. Die Auswahl der Metamodelle ist in [BGS23; BGS24a; BGS24b] bisher lediglich am Rande betrachtet worden.

Bei Rückkopplung auf die drei *Grundprinzipien der Produktgenerationsentwicklung* nach Abschnitt 2.1.2 auf S. Abschnitt 2.1.2 ist festzustellen, dass durch *DrAGOS* insbesondere das Grundprinzip der Variation von Lösungsprinzipien adressiert wird. Die übrigen beiden Grundprinzipien (Wiederverwendung und Einführung von Lösungsprinzipien) treffen nicht zu. *DrAGOS* nimmt eine Anpassung des Systems in neuen Randbedingungen vor, fügt jedoch keine neuen Systembestandteile hinzu, wie beispielsweise die Einführung eines Gurtendbeschlagsstraffers als einen weiteren Stellhebel für die Belastungsreduktion.

## 7.2.2 Beantwortung der Forschungsfrage

Die in Abbildung 1.1 auf S. 6 gestellte Frage – *Wie lässt sich die generations- und aufgabenstellungsübergreifende Systemauslegung optimieren?* – kann mit den gewonnenen Erkenntnissen folgendermaßen beantwortet werden:

- Technisch komplexe Systeme lassen sich durch den Einsatz von *DrAGOS* erfolgreich optimieren. Gezeigt werden konnte, dass Systemauslegungsprozesse sowohl für gesehene als auch für ungesehene Entwicklungsaufgaben bei Verwendung unterschiedlicher ATD-Typen, Sitzpositionen oder Produktgenerationen im Kontext der *Rückhaltesystemoptimierung effizienter* gestaltet werden können. Im Benchmark ist der RL-Agent in der gleichen Umgebung in der Lage, eine 7,3% bessere Lösung als ein globaler Optimierungsalgorithmus zu identifizieren. Während der klassische Optimierer mehrere tausend Schritte benötigt, braucht der RL-Agent lediglich acht Schritte. Folglich kann

der RL-Agent Wissen darüber speichern, mit welcher Strategie die Systeme optimiert werden können. Dieses Wissen kann situativ abgerufen und in verschiedenen Szenarien im Produktentwicklungsprozess immer wieder eingesetzt werden. Folglich ist neben der Effizienz auch die Übertragung von Wissen an die Organisation festzustellen.

- Zur Integration der Methode in Entwicklungsprozesse erfolgt ein Konzept zur Effizienzsteigerung des Trainings sowie der Schulerschluss mit der Produktgenerationsentwicklung. Das *Vortraining* funktionaler Abhängigkeiten mittels eines Metamodells und die anschließende *Anwendung* auf eine FE-Simulation realisieren das Lernen von nur in aufwändigen und komplexen FE-Modellen abgebildetem Systemverhalten mit einem RL-Agenten. Dies wurde im Trainingsvorgang verdeutlicht, da der RL-Agent von verschiedensten Startpunkten aus das System optimieren muss und dabei die Belohnung ansteigt. Die durchgeführten Robustheitsuntersuchungen attestieren die numerische Stabilität des RL-Agenten. Das Vortraining erlaubt zudem ein schnelles Nachtrainieren – beispielsweise bei der Änderung einer Schrittweite oder dem Hinzufügen eines weiteren Belastungswertes in die Belohnungsfunktion. Im Sinne der Produktgenerationsentwicklung erlaubt *DrAGOS* die *Variation* einer bestehenden Lösung.
- Die eingeführte Grundidee von *DrAGOS* ist allgemein, wobei die beispielhafte *Übertragung und Anpassung* auf eine typische Problemstellung der Produktgenerationsentwicklung – der Auslegung von Rückhaltesystemen – demonstriert wurde. Zudem ist die praktische Relevanz der verwendeten FE-Modelle hervorzuheben, die Alltagsgegenstand in der Automobilentwicklung sind. Die Überlegungen sind jedoch auch auf andere rechenintensive FE-Modelle, beispielsweise der Strukturoptimierung, übertragbar.

### 7.2.3 Einfluss auf Bewertungskriterien

Zu Beginn der Arbeit wurden in Abschnitt 1.3.1 auf S. 4 globale Bewertungskriterien eingeführt. Basierend auf der Evaluierung lässt sich insbesondere der positive Einfluss auf die *Entwicklungszeit* und die *Entwicklungskosten*, bei der Optimierung der *technischen Wertigkeit* hervorheben. Bezogen auf die technische Wertigkeit ist der RL-Agent in der Lage, eines von zwei ungesehenen Szenarien besser zu lösen als ein Benchmark-Optimierer. Das hebt die Fähigkeit, optimale Lösungen zu finden, hervor, weist aber auch darauf hin, die ermittelten Lösungen stets zu hinterfragen. Des Weiteren kann eine positive Beeinflussung der Bewahrung von *Wissen* festgestellt werden. Tritt ein Ingenieur, der sich bisher um diese Entwicklungsaufgaben gekümmert hat, aus, können nun unter Hinzunahme eines trainierten RL-Agenten auch weiterhin Rückhaltesysteme optimiert werden. Durch die reine Variation der Lösungsprinzipien leistet *DrAGOS* keinen direkten Beitrag zur *Innovation* im Produkt. (*Prozess- und Produkt-*) *Qualität* kann durch automatische Prozesse systematisiert werden, ein spezifischer Beitrag bleibt jedoch aus Sicht des Autors aus.

### 7.2.4 Schwachstellen und zukünftige Potenziale

Schwachstellen von *DrAGOS* ergeben sich in der *sukzessiven Durchführung* der Simulationen. Bei einer metamodellbasierten Optimierung können Trainingsdaten parallel generiert werden. Dies kann bei sauber automatisierten Pipelines einen zeitlichen Vorteil bieten. Ebenfalls lässt sich hier einfacher sicherstellen, dass der gesamte Designraum betrachtet und durchsucht wird. Da der RL-Agent seine erlernte Strategie verfolgt, ist dies nicht zwingend sichergestellt. Um eine erste Empfehlung, basierend auf den erzielten Ergebnissen zu formulieren, ist der RL-Agent besonders geeignet, wenn kleinere Optimierungen durchgeführt werden sollen. Das kann beispielsweise in der Modellpflege der Fall sein. Bei Designexploration hingegen sind nach jetzigen Erkenntnissen metamodellbasierte Ansätze wie in [Thi+06; JGK21] vorzuziehen.

Zusätzlich ist zu berücksichtigen, dass durch die RL-Agenten-basierte Ansätze auch *suboptimale Lösungen*, wie durch *DrAGOS* für den H3-ATD ermittelt werden können. Das Auffinden von suboptimalen Lösungen bei dem Vergleich mit konventionellen Optimierern ist auch in der Literatur [HO20; Bro+22] bekannt.

Ebenfalls kann die *Untersuchung weiterer Problemstellungen* betrachtet werden, bei denen große FE-Simulationen zur Optimierung verwendet werden. Beispielsweise die Optimierung von Längsträgern im Frontal- oder von Schwellerstrukturen im Seitenaufprall. Diese sind ebenfalls relevante Bestandteile jeder modernen Fahrzeuggeneration. Hierfür müsste jedoch eine – möglicherweise geometrische – generationsübergreifende Beschreibung der Systeme eingeführt werden.

Bisher wurde die Interaktion zwischen Ingenieur und RL-Agent nur am Rande erwähnt. Daher gilt es, auch die *Mensch-Maschine-Kollaboration* weiter und genauer zu untersuchen. Dies kann auch dabei helfen, das Vertrauen in diese Methoden zu erhöhen.

In der Anwendungspraxis sind zudem *Problemstellungen meist gekoppelt* zu betrachten. Vorgegebene Insassenbelastungen müssen mit demselben Rückhaltesystem für verschiedene Lastfälle und auch demselben ATD-Typen gleichzeitig erfüllt werden. Wie *DrAGOS* mit solchen Problemstellungen umgeht, wurde im Rahmen dieser Arbeit nicht näher thematisiert. Inwiefern Multi-Agentensysteme oder die simultane Optimierung hier zielführend sind, kann Gegenstand von zukünftigen Arbeiten sein.

Interessant ist auch die Entwicklung von *Transfer-Learning-Methoden*. Diese könnten auf Basis von wenigen Simulationen auf neue Problemstellungen angepasst werden. Im nächsten Schritt kann hier wieder eine Metamodelloptimierung durchgeführt werden. Eine Sicherstellung der Glaubhaftigkeit und Güte der Modelle ist hier jedoch essenziell.

## 7.3 *DAYSi* – Analyse von Crashsensordaten

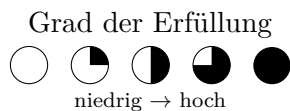
Nach der Einführung von *DAYSi* in Abschnitt 5.3 ab S. 99 und der Evaluierung in Abschnitt 6.3 ab S. 141 erfolgt nun die Diskussion anhand der Literatur aus Abschnitt 3.4 ab S. 60. Ebenfalls wird auf den Vergleich und die Einordnung bestehender Arbeiten in Tabelle 4.2 auf S. 80 verwiesen.

### 7.3.1 *DAYSi* im Kontext bestehender Literatur

Zunächst wird *DAYSi* in den Kontext bestehender Literatur anhand von Tabelle 7.3 eingeordnet.

**Tabelle 7.3:** *DAYSi* im Kontext bestehender Literatur

	Simulationstyp	Lernverfahren	$\mathcal{G}(t, \xi)$ – FE-Daten	1D – (Zeit-)Serientdaten	0D – Skalare	Cluster- und Ausreißeranalyse	Ursachenanalyse	Beitragsanalyse	Automatischer Prozess
<i>DAYSi</i>	DY	UL	○	●	○	◐	◑	◒	●



DY = Dynamisch, ST = Statisch, UL = Unsupervised Learning

*DAYSi* ist durch die Verarbeitung von Zeitserientdaten in erster Linie für *dynamische Simulationen* verwendbar.

Zur Bestimmung der Abweichungen wird ein Clustering-Algorithmus verwendet, der dem *Unsupervised Learning* zuzuordnen ist.

In Bezug auf die Datentypen werden ausschließlich *1D-Zeitreientdaten*, wie sie von Crashsensoren ausgegeben werden, verarbeitet. Skalare und FE-Daten werden nicht betrachtet. *DAYSi* fokussiert sich auf die Identifikation von *Abweichungen und Ausreißern* im Signal- und Zeitbereich, die auf verschiedenen Ebenen visualisiert werden. Das erlaubt, Ergebnisse systematisch und nachvollziehbar darzustellen.

Ebenfalls wurde gezeigt, dass durch die Rekonstruktion der *Eventkette* auf Basis von Kreuzkorrelationen Ingenieure bei der *Ursachenfindung* unterstützt werden können. Die grundlegende Annahme dabei ist, dass starke Auffälligkeiten, die früh in der Simulation auftreten, mit hoher Wahrscheinlichkeit für spätere Auffälligkeiten mitverantwortlich sind. Neben diesen vorgelagerten Ursachen können auch nachgelagerte sekundäre Effekte spezifiziert werden.

Eine nähere Betrachtung der *Beiträge* einzelner Signale ist mittels der Abweichungskoeffizienten grundsätzlich möglich. Allerdings sind diese Kennwerte nur bedingt intuitiv und dienen in erster Linie dazu, eine qualitative Reihenfolge festzulegen.

*DAYSi* ist ein vollständig *automatischer Prozess* und bedarf keiner interaktiven Bestimmung von Ausreißern. Die Bestimmung der Verfahrensparameter wurde in der Robustheitsanalyse der Methode ausführlich diskutiert.

Im Abgleich mit bestehenden Ansätzen in Tabelle 4.3 auf S. 80 ergibt sich der Mehrwert in Bezug auf die *Datenverarbeitung der 1D-Zeitreihendaten*, da nun ein Vorgehen, Signale unterschiedlicher Sensoren parallel in einer Analyse zu verarbeiten, existiert. In Bezug auf die Cluster- und Ausreißeranalyse werden mit der Literatur vergleichbare Ergebnisse erzielt, da die dort vorhandenen Schwachstellen, wie die *fehlende Interpretierbarkeit* der Abweichungen und der Cluster, auch hier verbleiben. Weiterer Mehrwert wird insbesondere in Bezug auf die Ursachenfindung in der Analyse von Crashsensoren geschaffen. Bisherige Ansätze wie [Iza+23; PGS23; PGS25] behandeln diese Problemstellung nicht zentral und Herangehensweisen aus der Analyse von Parameterdaten [Die+18; CK20] sind nicht direkt übertragbar.

Anhand von drei verschiedenen Use Cases und darauf aufbauenden Untersuchungen zur Bestimmung der Robustheit des Verfahrens, wurde gezeigt, dass *DAYSi* für verschieden komplexe FE-Modelle (Schlitten und Gesamtfahrzeug) sowie verschiedene Lastszenarien (Front- und Seitenaufprall) einsetzbar ist. Auch die Auswahl der Simulationen und Sensoren spielt eine Rolle, da sie den Kontext für die Analyse liefern. Es konnte aber gezeigt werden, dass diese Auswahl mit Ingenieurverständnis getroffen werden kann und Ergebnisse bei anderer Auswahl konsistent bleiben.

Das Aufsetzen auf dem *ISO-MME-Datenformat* [ISO 13499] ermöglicht einen übergreifenden Einsatz bezüglich Projekt und Disziplin, was ebenfalls die Integration in bestehende Prozesse erleichtert. Ebenso ist dieses Format die Schnittstelle zwischen Versuch und Simulation und ermöglicht einen automatischen Abgleich davon. Dieser übergreifende Einsatz ist für FE-Daten-basierte Ansätze nicht einfach realisierbar, da Knoten und Elementnummerierung, sowie die Zuordnung zu Bauteilen nicht einheitlich sind. Ein weiterer Vorteil, der sich insbesondere gegenüber FE-Daten-basierten Verfahren ergibt, ist die deutlich geringere *Rechenzeit* von weniger als fünf Minuten in allen betrachteten Beispielen. Die meisten Sensoren wurden für die Analyse in Abschnitt 6.3.4.2 auf S. 153 verwendet. Die Rechenzeit beträgt inklusive der Erstellung von Plots und PowerPoint-Reports 285 s auf einer CAE-Workstation.

Im Feld der KI werden Analysemethoden wie *DAYSi* der *Anomaliedetektion* zugeordnet. Da für den betrachteten Anwendungsfall nicht zwingend eine Anomalie vorliegt, sobald

Abweichungen erfasst werden, wurde hier der Begriff Ausreißer verwendet [Blá+21; Kra24]. In Bezug auf Anomalien allgemein unterscheidet Nassif et al. [Nas+21] drei Typen: *Punktanomalien* sind einzelne Werte, die deutlich vom erwarteten Muster abweichen, etwa ein plötzlicher Aktienkurseinbruch. *Kollektive Anomalien* entstehen durch ungewöhnliche Muster über einen Zeitraum, beispielsweise eine unerwartete Trendänderung in der Erdtemperatur. *Kontextabhängige Anomalien* erscheinen isoliert betrachtet normal, fallen jedoch im gegebenen Kontext auf – etwa ein ungewöhnlich hoher Stromverbrauch in einer Nacht.

Klassische *Anwendungsbeispiele* für die Anomaliedetektion in Zeitseriendaten finden sich insbesondere in der Erddatenanalyse, der Cyber-Security sowie der Medizin [Blá+21]. Die Erkennung von Naturkatastrophen fokussiert sich beispielsweise auf die Identifikation von Erdbeben durch die Analyse von Vorläuferdaten, welche Trendänderungen und hochfrequente Störungen in seismischen Sensoren aufzeigen. Im Bereich der Cyber Security dienen ML-Algorithmen der Intrusion Detection, indem diese böses Netzwerkverhalten anhand von Datenpaketgrößen, Übertragungsraten oder Verbindungsaufbauzeiten identifizieren. In der Medizin werden EKG- (Elektrokardiogramm) und EEG-Daten (Elektroenzephalogramm) analysiert, um durch Anomalien Rückschlüsse auf Krankheitsbilder wie Herzrhythmusstörungen oder Epilepsie ziehen zu können. Gemeinsam haben die Problemstellungen, dass wiederkehrende Muster in den Zeitreihen es ML-Algorithmen ermöglichen, Normalverhalten zu modellieren und Anomalien aufzuzeigen.

Bezogen auf Crashsensoren sind vor allem kollektive und kontextabhängige Anomalien relevant. Welche dieser Anomalien vorliegt, ist zweitrangig, da alle Auffälligkeiten im System relevant sind und identifiziert werden müssen, um sie manuell weiter zu prüfen.

Ein wesentlicher *Unterschied der vorliegenden Crashsensordaten* liegt in der Menge, der zeitlichen Dauer und dem Fokus auf den Abgleich, was die Anwendung klassischer Ansätze erschwert. In der Entwicklung stehen meist nur wenige Simulationen oder Versuche für den direkten Vergleich zur Verfügung, was den Einsatz komplexer neuronaler Netze erschwert, da nur sehr wenige Trainingsdaten vorhanden sind. Zudem steht vor allem der Vergleich der Sensoren im Fokus. Aufgrund der Problemstellung, die nur ein “Event” innerhalb weniger hundert Millisekunden betrachtet, sind Trends weniger relevant und Ereignisse kehren nicht wieder.

Im Übertrag auf die Medizindomäne entspricht dies dem Anwendungsfall, EKG-Daten von mehreren Patienten zu vergleichen und den auffälligsten daraus zu ermitteln. Daher ist *DAYSi* eine speziell auf die Anwendung zugeschnittene Methode für die identifizierte Problemstellung der bisher meist manuell durchgeführten Sensordatenanalyse.

### 7.3.2 Beantwortung der Forschungsfrage

Die in Abbildung 1.1 auf S. 6 gestellte Frage – *Wie lässt sich auffälliges Systemverhalten identifizieren und analysieren?* – kann mit den gewonnenen Erkenntnissen folgendermaßen beantwortet werden:

- Eine Möglichkeit, *auffälliges Systemverhalten* in Crashesimulationen zu identifizieren, ist die Verwendung von *1D-Zeitreihendaten*, die von Sensoren im Fahrzeug aufgezeichnet werden. Diese Daten sind unabhängig von möglichen Änderungen in den Bauständen und der Konfiguration von FE. Diverse Messpunkte können mittels ISO-MME [ISO 13499] für physische Crashtests eindeutig definiert werden. Zudem ist die direkte Übertragung auf physische Crashtests gegeben, was den Ansatz breit einsetzbar macht. *DAYSi* stellt für die Analyse dieser 1D-Zeitreihendaten eine automatische Methode bereit, die innerhalb weniger Minuten beliebig viele der bis zu 1000 Sensoren gemeinsam analysieren kann.
- Die in der Analyse ermittelten *Abweichungen* – sensorweise für jedes Signal in allen Simulationen – werden über Abstände zu einem Hauptclustermittelpunkt bestimmt, der automatisch auf Basis eines k-Means-Clusterings ermittelt wird. Abweichungen werden dabei sowohl für den gesamten Signalverlauf als auch für einzelne Zeitfenster bestimmt, was Rückschlüsse auf Abweichungen in der Magnitude und der Zeit erlaubt.
- Auf Basis von Kreuzkorrelationen der zeitlichen Abweichungen wird für jede Simulation eine *Eventkette* ermittelt, die Informationen zu zeitlichen Zusammenhängen und damit auch mögliche Ursachen von Ausreißern liefert.
- Das vorgeschlagene *Visualisierungskonzept* für *DAYSi* – durch die Ergebnisdarstellung auf verschiedenen Ebenen (global, zeitbezogen sowie auf Datenpunkt- bzw. Simulationsebene) – und der automatische, zeiteffiziente Prozess tragen zu einem schnelleren Verständnis bei. Zudem wird durch die anwenderfreundliche Ausgestaltung die Übernahme in die praktische Anwendung weiter vereinfacht.
- Im Rahmen der Evaluierung wurde das Systemverhalten in drei verschiedenen Use Cases aus der Entwicklung passiver Sicherheitssysteme analysiert. Hierbei konnten auffälliges Systemverhalten identifiziert und Gründe dargelegt werden, die sich in der manuellen Analyse bestätigen ließen. Anschließend durchgeführte Robustheitsuntersuchungen attestieren die Stabilität und Konsistenz der Ergebnisse. Relevante Verfahrensparameter wie die Größe der Zeitfenster, die Filterung und die Normalisierung wurden betrachtet und bestmögliche Werte festgesetzt. Besonders relevant ist der Kontext, der durch Sensoren und Datenpunkte (Simulationen) vorgegeben wird.

### 7.3.3 Einfluss auf Bewertungskriterien

Zu Beginn der Arbeit wurden in Abschnitt 1.3.1 auf S. 4 globale Bewertungskriterien eingeführt. Basierend auf der Evaluierung lässt sich insbesondere der positive Einfluss auf die *Entwicklungszeit* und die *Entwicklungskosten* feststellen. Neben der schnellen Identifikation von Auffälligkeiten, die dem System selbst zuzuordnen sind, wie beispielsweise die Änderung der Schaltzeit des Gurts, können auch Fehler identifiziert werden, die bei der Sicherung von (*Prozess- und Produkt-*) *Qualität* unterstützen und somit auch einen Beitrag zur *technischen Wertigkeit* des Produkts leisten. Zusätzlich werden Ingenieure dazu befähigt mehr Simulationen durchzuführen, wenn die Auswertzeit reduziert wird. Des Weiteren leistet *DAYSi* einen Beitrag zum *Wissen* beteiligter Ingenieure, da auch Ursachen und Wirkungen hervorgehoben werden, die zum Systemverständnis beitragen. Diese Ergebnisse sind jedoch stets kritisch zu hinterfragen, insbesondere im Hinblick auf den bereitgestellten Kontext. Aus Sicht des Autors wird der *Innovationsgrad* nicht direkt beeinflusst.

### 7.3.4 Schwachstellen und zukünftige Potenziale

Schwachstellen von *DAYSi* ergeben sich neben der *Interpretierbarkeit* von Abweichungskoeffizienten auch im weiteren Ausbau der Ursachenfindung. Die Abweichungskoeffizienten weisen keine physikalische Einheit auf. Das wird auch weiterhin ein Gegenargument beteiligter Ingenieure sein solche Methoden einzusetzen. Ebenso gilt es die Skalen und Abstände der Ausreißer genauer zu untersuchen.

Durch die Eventkette wird lediglich auf auffällige Sensoren verwiesen – eine weitere Beschreibung oder *Klassifizierung der Ursache* bleibt jedoch aus. Anhand des Beispiels aus Use Case II wird die Ursache im Gurt angegeben, sodass der Ingenieur dort nach weiteren Fehlern suchen kann. Die Ursache des Versagens, also ein fehlerhafter Kontakt zwischen Gurtschloss und -zunge, fehlt jedoch in der Auswertung und muss weiterhin durch den Ingenieur ermittelt werden. Ebenso fehlt die Ursache für den fehlerhaften Kontakt, die beispielsweise in fehlender Definition, zu geringer Kontaktdistanz, fehlender Elemente der Bauteile, in den im Kontakt angegebenen Knotensets oder auch in der Verwendung eines für das Fahrzeug oder darin befindlicher Bauteile wie dem Sitz nicht kompatiblen Gurtmodells liegen kann. Letzteres kann durch die fehlerhafte manuelle Zuweisung, aber auch durch fehlerhafte Attribute in einer automatischen Zuweisung verursacht werden. Diese mit jeder Stufe immer komplexer werdende Kette könnte maschinell nur durch ein generelles Verständnis von System und Infrastruktur, sowie unter Einbeziehung weiterer Schnittstellen nachvollzogen werden. In Bezug auf *DAYSi* verbleibt die Verfolgung der weiteren Kette außerhalb der 1D-Zeitseriendaten selbst entsprechend bei dem Ingenieur. In zukünftigen Arbeiten könnten daher übergeordnete Methoden – beispielsweise mittels

Retrieval Augmented Generation (RAG) – entwickelt werden, die *DAYSi* nutzen, um innerhalb der Simulationen auf Basis von Zeitserien Ursachen zu ermitteln und diese dann über Textverständnis Input key-Files, Solver-Reports oder Modellaufbau-Logs zu überprüfen, um auch außerhalb der Simulation Einblicke in Ursachen zu geben.

Auch für *DAYSi* gilt es die *Mensch-Maschine-Kollaboration* weiter und genauer zu untersuchen. Durch Umfragen könnte ermittelt werden, ob die Präferenz auf automatischen oder interaktiven Verfahren liegt. Ebenfalls könnte validiert werden, inwiefern die mit *DAYSi* eingeführte hierarchische Ergebnisdarstellung auch einen messbaren Mehrwert bringt. Solche Untersuchungen können dabei helfen, das Vertrauen in Methoden dieser Art generell zu erhöhen.

Im Hinblick auf Anwendungsfälle aus anderen Domänen im Bereich der Anomaliedetektion in Zeitseriendaten, können zukünftige Arbeiten breiter generalisierte Ansätze, auch auf Basis von Deep Learning, verfolgen. Solche Ansätze, wie in [KWB18; KB18; BRK21] zur Prognose von Bauteilschädigungen, Aufprallpunkten oder Brustbeschleunigungen beschrieben, können auf die Anomalieerkennung und Analyse von Simulations- und Versuchsdaten übertragen werden.

## 7.4 *ExPLAIN* – Erklärung von Systemverhalten

Nach der Einführung von *ExPLAIN* in Abschnitt 5.3 ab S. 99 und der Evaluierung in Abschnitt 6.3 ab S. 141 erfolgt nun die Diskussion anhand der Literatur aus Abschnitt 3.4 ab S. 60. Ebenfalls wird auf den Vergleich und die Einordnung bestehender Arbeiten in Tabelle 4.2 auf S. 80 verwiesen.

### 7.4.1 *ExPLAIN* im Kontext bestehender Literatur

Zunächst wird *ExPLAIN* in den Kontext bestehender Literatur anhand von Tabelle 7.4 eingeordnet.

Tabelle 7.4: *ExPLAIN* im Kontext bestehender Literatur

Grad der Erfüllung

niedrig → hoch

	Simulationstyp	Lernverfahren	$\mathcal{G}(t, \xi)$ – FE-Daten	1D – (Zeit-)Seriedaten	0D – Skalare	Cluster- und Ausreißeranalyse	Ursachenanalyse	Beitragsanalyse	Automatischer Prozess
<i>ExPLAIN</i>	DY	U/SL							

DY = Dynamisch, UL = Unsupervised Learning, SL = Supervised Learning

Der Fokus von *ExPLAIN* liegt vor allem auf *dynamischen Simulationen*. Ein direkter Übertrag auf Schwingungs- oder statische Festigkeitssimulationen ist jedoch möglich, da auch dort Skalare und FE-Daten vorhanden sind.

Das zentrale Lernverfahren in *ExPLAIN* ist vom Typ *Supervised Learning*, um einen (i-o)-Zusammenhang oder (o-o)-Zusammenhang abzubilden. Zur Dimensionsreduktion von FE-Daten wird jedoch ein Unsupervised Learning Verfahren vorgeschlagen.

Zentraler Bestandteil des Konzepts ist die modulare Verarbeitung von *Skalaren*, *1D-Zeitserien-* und *FE-Daten*. Bezogen auf die FE-Datenverarbeitung ergeben sich nach Tabelle 7.4 noch weitere Potenziale bezüglich der Integration von Material- und Versagensinformationen, beispielsweise im Voxelgrid selbst.

Eine *Cluster- und Ausreißeranalyse* kann durch die extrahierten Features sowie die Einbettungsräume manuell erfolgen. Das Auffinden und Herausstellen dieser ist jedoch kein essenzieller Bestandteil der Methode.

Der Fokus von *ExPLAIN* liegt insbesondere auf der *Ursachenanalyse*, sowie der intuitiven Darlegung einzelner *Beiträge* möglicher Ursachen. Die Hauptursachen sowie deren Wirkrichtung können robust herausgestellt werden. Die Magnitude einzelner Beiträge, die

mit SHAP ermittelt wurde, ist jedoch mit Vorsicht zu betrachten, da diese abhängig von den Randbedingungen variieren kann. In bestehender Literatur sind gerade die letzten beiden Aspekte bisher nicht hinreichend betrachtet worden. Weitere Potenziale ergeben sich zudem in der Ursachenanalyse mittels kausaler Modelle.

Ein vollständig *automatischer Prozess* wie in *DAYSi* wird nicht bereitgestellt. Insbesondere die Ergebnisse der Sensitivitätsanalyse erfordern die aktive Interpretation oder auch das Generieren weiterer Daten.

Im Abgleich mit bestehenden Methoden aus der Literatur in Tabelle 4.3 auf S. 80 ergibt sich der Mehrwert in Bezug auf die *modulare Datenverarbeitung* aller relevanten Datentypen in FE-Simulationen. Ansätze, die ein vergleichbares Ziel verfolgen, wie beispielsweise [CK20], fokussieren sich lediglich auf einzelne Datentypen. Zudem ist die Datenverarbeitung in *ExPLAIN* übergreifend in Bezug auf Entwicklungsstand und Produktgeneration aufgrund der Verwendung einer Voxelisierung zur Vorverarbeitung der FE-Daten und ISO-MME für 1D-Zeitserien. Gerade bei größeren Unterschieden in der Geometrie sind Mappingverfahren zur Vereinheitlichung der Repräsentation nicht mehr ausreichend und verzerren die Ergebnisse. Die Verwendung von Voxeln ist hier flexibler, erfordert allerdings einen zusätzlichen Rechenaufwand in der Vorverarbeitung.

Die in [DHS16; Die+16; Die+18] verwendeten Decision Trees sind inhärent interpretierbar. Dieses Modell ist jedoch zur *Modellierung nichtlinearer Zusammenhänge*, wie sie in den betrachteten Problemstellungen vorherrschen, nicht hinreichend performant. Der RMSE ist verglichen mit XGBoost oder TabPFN in Tabelle 6.2 auf S. 182 bis zu drei- bzw. viermal so hoch. Diese Ungenauigkeiten wirken sich negativ auf die SHAP-Werte aus, da deren Berechnung ausschließlich auf dem Metamodell aufsetzt. Diese sind für die intuitive Ergebnisdarstellung in *ExPLAIN* essenziell.

Anhand von drei verschiedenen Use Cases und darauf aufbauenden Untersuchungen zur Bestimmung der *Robustheit* der Methode wurde gezeigt, dass *ExPLAIN* für verschieden komplexe Modelle (Schlitten und Gesamtfahrzeug) sowie verschiedene Lastszenarien (Front- und Seitenaufprall) einsetzbar ist. Wirkrichtung und Reihenfolge von Einflüssen auf das Systemverhalten können lokal und global robust bestimmt werden.

Ein *(o-o)-Zusammenhang* ist in der identifizierten Literatur bisher nicht betrachtet worden. Dieser erlaubt eine neue Sichtweise und flexible Analyse auch unparametrisierter FE-Modelle. Bezogen auf latente Darstellungen in Einbettungsräumen von FE-Daten, wie in [Tho+10; IG19; Kra+23], zur Identifikation von Clustern und Ausreißern kann durch *ExPLAIN* genauer erläutert werden, was einen Einfluss auf die Repräsentation hat. Das trägt zur Transparenz und damit langfristig auch zur Akzeptanz solcher Methoden bei. Im Fall von Optimierung und der Erstellung entsprechender Reports ist nun ein direkter

Vergleich von initialer und optimierter Variante möglich, was auch Blackbox-Prozesse, wie in [CDP05; Büt+23], transparenter gestaltet.

Essentiell ist eine Interpretation der Ergebnisse aus der *Ingenieursdomäne* heraus, ebenso wie die Auswahl von Features, gerade wenn deren Einfluss auf das Systemverhalten untersucht werden soll.

Ein großer *zeitlicher Anteil* entfällt auf die Vorverarbeitung der FE-Daten. Dies dauert anhand von Use Case III für 556 Simulationen insgesamt etwa zwei Stunden. Hierbei entfallen 52 min auf das Laden der Simulationsdaten und 68 min auf das Voxelisieren selbst. Diese Zahlen gelten jedoch für das Laden und Voxelisieren aller Zeitschritte und wurden auf einer CAE-Workstation ermittelt. Das anschließende Trainieren der Metamodelle inklusive einer Hyperparameteroptimierung ist für alle Use Cases in etwa 10 min erfolgt. Folglich können durch zielgerichtetes Datenmanagement in der Praxis weitere Effizienzpotenziale gehoben werden.

### 7.4.2 Beantwortung der Forschungsfrage

Die in Abbildung 1.1 auf S. 6 gestellte Frage – *Wie lassen sich Einflussgrößen des Systemverhaltens identifizieren und analysieren?* – kann mit den gewonnenen Erkenntnissen folgendermaßen beantwortet werden:

- Die Grundlage, um diverse Einflussgrößen betrachten zu können, ist die *modulare Verarbeitung der Datentypen*, die in FE-Simulationen Abbildung 3.2 auf S. 50 vorhanden sind. ExPLAIN stellt für jeden Datentyp ein Verarbeitungskonzept bereit. Entsprechend kann der Anwender die für ihn relevanten Bereiche des Systems berücksichtigen und beliebige Daten – auch aus anderen Fahrzeugen oder Entwicklungsständen – einbeziehen.
- Um *Einflussgrößen* auf das Systemverhalten zu analysieren, erfolgt die Darlegung globaler und lokaler Sensitivitäten eines Metamodells, das die beliebigen Abhängigkeiten innerhalb der Systeme, die der Ingenieur näher betrachten will, abbildet. Die Identifikation von relevanten Einflussgrößen auf die durch den Anwender ausgewählte Zielgröße ist robust möglich, insbesondere auf die Spezifizierung der Reihenfolge sowie Wirkrichtungen bezogen. Mit Vorsicht zu beachten sind die quantitative Ausprägung der Effekte und abhängige Eingaben.
- Zur direkten Unterstützung beteiligter Ingenieure werden *System- und Differenz-SHAP* eingeführt. Diese ermöglichen eine flexible Analyse auf verschiedenen Systemebenen, sowie einen A/B-Vergleich, der in dieser Form bisher nicht möglich war. Durch den Vergleich mit Sobol und LIME wurde verdeutlicht, dass der SHAP-Ansatz die Ergebnisse aus Ingenieurssicht intuitiver darlegt. Durch die lineare Skala entspricht hier ein doppelter SHAP-Wert auch einem doppelten Einfluss, was in Bezug auf *DAYSi* weiterhin offen bleibt.

- Identifikation und Analyse von Einflüssen in Simulationsdaten wurde anhand von *drei Use Cases aus Fahrzeugstruktur- und Rückhaltesystemauslegung* erfolgreich demonstriert. Die Identifikation war innerhalb weniger Minuten in den ersten beiden Szenarien und in etwa zwei Stunden im dritten Szenario möglich. Das macht die Methode gut in den Arbeitsalltag integrierbar. Auch wenn die Grundzüge der Problemstellungen durch wenige Daten erfasst werden können, ist eine hinreichend hohe Anzahl (Faustregel:  $N \geq M^2$ ) sinnvoll.

### 7.4.3 Einfluss auf Bewertungskriterien

Zu Beginn der Arbeit wurden in Abschnitt 1.3.1 auf S. 4 globale Bewertungskriterien eingeführt. Basierend auf der Evaluierung lässt sich insbesondere der positive Einfluss auf das *Wissen* beteiligter Ingenieure durch neue Einblicke hervorheben. Das bessere Verständnis der Daten und des Systems ist die Grundlage für die Erhöhung von (*Prozess- und Produkt-*) *Qualität* und leistet damit auch einen indirekten Beitrag zur *technischen Wertigkeit* des Produkts. Werden ungewollte Deformationseigenschaften – wie ein Knicken der horizontalen Schwellerstrukturen – frühzeitig erkannt, ist ein zeitiges Gegensteuern effektiv möglich. Durch effiziente Analyseprozesse werden zudem Ingenieure dazu befähigt mehr Simulationen durchzuführen, was tiefgreifendere Optimierungen ermöglicht. In der Konsequenz kann so perspektivisch auch ein indirekter Beitrag zur Reduktion von *Entwicklungszeit* und *Entwicklungskosten* geleistet werden. Das Aufdecken von möglichen Ursachen und Beiträgen kann Kreativitätsprozesse zur Findung neuer Lösungen unterstützen. Allerdings werden durch die Methode selbst keine neuen Lösungsvorschläge generiert. Daher ist nur ein bedingter Einfluss auf den *Innovationsgrad* festzustellen.

### 7.4.4 Schwachstellen und zukünftige Potenziale

Schwachstellen von *ExPLAIN* ergeben sich in der Größe der benötigten Datenmenge, dem physikalischen Verständnis der verwendeten Metamodelle, dem idealen Einsatzzeitpunkt im Prozess sowie den Ergebnissen selbst.

Große Crashsimulationsmodelle, wie beispielsweise ein Seitenaufprall mit ATDs, können auf einem Hochleistungscluster mehr als einen Tag Rechenzeit benötigen. Für komplexe Problemstellungen müssten Hunderte solcher Simulationen generiert werden, was zum aktuellen Zeitpunkt aufgrund der Kosten nur begrenzt realisierbar ist. Auch wenn die Methode durch das Datenverarbeitungskonzept das Einbeziehen anderer Baureihen oder Entwicklungsstände zur Kontextualisierung ermöglicht, ist dieser Aspekt im Rahmen weiterer Untersuchungen genauer zu beleuchten.

Alle Metamodelle, über die die Einflüsse ermittelt wurden, besitzen kein direktes Bewusstsein für physikalische Prinzipien wie die Energieerhaltung. Trotz hoher Modellgüten,

insbesondere in Use Case I und Use Case III, und dem guten Umgang mit korrelierten Daten in Use Case II, kann die physikalische Validität der Vorhersagen nicht garantiert werden. Teil zukünftiger Arbeiten kann also die Verwendung von physikinduzierten ML-Modellen sein.

Schwächen im System, die genauer analysiert und erklärt werden müssen, werden oft erst schrittweise im Entwicklungsprozess aufgedeckt. Oft ist dann charakteristisches Verhalten bereits bekannt. Sofern der Fokus auf dem generellen Verständnis des Systems liegt – wie beispielsweise in Use Case III – liegt das größte Potenzial im möglichst *frühen Einsatz des Entwicklungsprozesses* mit großzügig abgestecktem Designraum. Bei ungünstiger Problemformulierung oder zu spätem Einsatz können Ergebnisse offensichtlich sein, beispielsweise dass der Vorderwagen in einem Frontalaufprall relevanter als der Hinterwagen ist. Die Grenze, wann wer Probleme wie gut versteht, hängt zusätzlich auch vom Individuum ab. Folglich bleiben der Zeitpunkt des Einsatzes von ExPLAIN, sowie die Auswahl relevanter Merkmale derzeit auch noch eine Frage der Ingenieurserfahrung.

Die Ergebnisse, insbesondere die der SHAP-Werte, sind stets *kritisch zu betrachten*. Im Zweifelsfall können zur zusätzlichen Sicherheit, wie auch im Rahmen dieser Arbeit gezeigt, alternative Metamodelle und Sensitivitätsanalysemethoden verwendet werden, um Ergebnisse zu bestätigen. Das hilft zudem bei der Akzeptanz, da klassische Methoden in den Prozessen meist etablierter sind. Wie bereits in [Ros+20] verdeutlicht, darf eine Interpretation der Ergebnisse nur im Kontext von Domänenwissen erfolgen. Je nach Komplexität der Problemstellung können Expertengespräche durchgeführt werden, um weiteren Kontext für die Ergebnisse bereitzustellen. Hierbei ist es wichtig, keine ungerechtfertigten kausalen Zusammenhänge anzunehmen [Mol+22]. Ohne kritisches Hinterfragen ist daher eine Fehlleitung durch die Ergebnisse möglich, was beispielsweise in [LB20] anhand einer Nutzerstudie hervorgehoben wird.

Wie auch anhand von Use Case II gezeigt, hat die *Abhängigkeit der Eingaben* einen Einfluss auf die Ergebnisse und erfordert die Verwendung geeigneter Modellagnostik-Methoden. Durch den Einsatz geeigneter Methoden können Unsicherheiten reduziert werden. Trotzdem sind Ergebnisse stets mit Vorsicht zu betrachten. Zurückführen lässt sich das unter anderem auf die Modellagnostik-Methoden selbst. Diese fokussieren sich hauptsächlich auf die Optimierung der Erklärung, ohne das darunterliegende Modell zu beachten [Mol+22]. Ungenauigkeiten oder Prognosefehler werden daher nicht berücksichtigt.

Auch wenn die eingeführten System- und Differenz-SHAP-Werte die Ergebnisse vereinfachen und dadurch *Komplexität reduzieren*, können sich zukünftige Arbeiten mit der weiteren Aufbereitung der Analyseergebnisse befassen. Trotz der intuitiven Darstellung von SHAP muss hierfür erst ein Grundverständnis bei beteiligten Personen etabliert werden. Die Aufbereitung der Ergebnisse durch die Kombination von interpretierbarem

---

ML und LLMs wurde bereits in eigenen Folgearbeiten betrachtet [Mat+25] und zeigt ein hohes Potenzial. Dieser Aspekt ist besonders vielversprechend im Kontext der Mensch-Maschine-Kollaboration. Leitfäden, wie mit maschinengenerierten Ingenieurserklärungen umgegangen wird, könnte das Vertrauen in solche Methoden erhöhen.



## 8 Zusammenfassung und Ausblick

Die vorliegende Arbeit widmete sich der datengetriebenen Optimierung von Entwicklungsprozessen in der Auslegung crashbelasteter Systeme am Beispiel der passiven Fahrzeugsicherheit. Als Teil einer systematischen Herangehensweise, wurde die Strukturierung dieser Arbeit mittels der DRM nach Abbildung 1.1 auf S. 6 vorgenommen. Allem vorangestellt ist nach der DRM die Klärung des Forschungsgegenstandes (Phase I) in Kapitel 1 ab S. 1. Dieser ist vorwiegend durch die Komplexität und den hohen Aufwand bei der Auslegung crashbelasteter Systeme motiviert. Gerade dort bieten datengetriebene Methoden vielversprechende Ansätze, um Entwicklungszeiten und -kosten zu reduzieren, die Qualität zu sichern sowie das Wissen beteiligter Personen und Organisationen zu steigern.

Dabei bilden die theoretischen Grundlagen und der Stand der Technik, die in Kapitel 2 ab S. 9 und Kapitel 3 ab S. 43 ausführlich diskutiert wurden, das Fundament und gehören zur Präskriptiven Studie I (Phase II) der DRM. Abgehandelt wurden einerseits die Produktentwicklung und KI, aber auch die virtuelle Systemauslegung mittels der FEM. Der Stand der Technik fokussiert auf datengetriebene Produktentwicklungsprozesse und Modelle, um übergeordnete Aspekte miteinzubeziehen. Des Weiteren werden aber auch datengetriebene Methoden zur Optimierung und Auslegung von crashbelasteten Systemen, sowie datengetriebene Methoden zur Analyse komplexer Simulationsdaten diskutiert.

Defizite wurden anschließend in Kapitel 4 ab S. 69 identifiziert, auf deren Basis die Beiträge der Arbeit in Abschnitt 4.2 ab S. 81 anhand der Forschungsfragen spezifiziert wurden. Defizite liegen insbesondere in der fehlenden Orientierung und systematischen Betrachtung komplexer Schnittfelder, sowie in der anwendungsgerechten Prozessintegration datengetriebener Methoden bezogen auf Entwicklungsphasen und -aktivitäten. Darüber hinaus sind Konzepte zur problemstellungsübergreifenden Optimierung komplexer Systeme, zum automatischen Vergleich von Crashesensorsignalen, sowie zum Einsatz interpretierbarer ML-Verfahren für die intuitive Ergebnisanalyse bisher nicht hinreichend betrachtet worden.

Im Hauptteil der Arbeit – der Präskriptiven Studie (Phase III) des Aufbaus nach der DRM – in Kapitel 5 ab S. 85 werden entsprechend der Defizite vier entwicklungsunterstützende Maßnahmen eingeführt. Durch das Vorgehensmodell *RAMEn* und die darin integrierten Methoden *DrAGOS*, *DAYSi* und *ExPLAIN* wird ein ganzheitliches Bild in Bezug auf den Einsatz datengetriebener Methoden für das betrachtete Anwendungsfeld gezeichnet. *RAMEn* ermöglicht es, die drei Methoden nicht nur isoliert voneinander zu betrachten, sondern als Teil eines Gesamtkonstrukts, in dem auch bestehende Ansätze und Methoden integriert werden können.

In der Evaluierung – der Präskriptiven Studie II (Phase IV) des Aufbaus nach der DRM – in Kapitel 6 ab S. 117 werden praxisnahe Use Cases aus der Automobilindustrie abgehandelt und diskutiert, um den generalisierten Einsatz des Vorgehensmodells und der Methoden sicherzustellen. Für die datengetriebenen Methoden wurde hierbei ein besonderes Augenmerk auf die Robustheit der Verfahrensparameter gelegt. In der anschließenden Diskussion in Kapitel 7 ab S. 209, die ebenfalls Teil der Präskriptiven Studie II ist, werden die Ergebnisse im Kontext der bestehenden Ansätze und Methoden aus der Literatur betrachtet.

Die vier entwicklungsunterstützenden Maßnahmen sind im Folgenden zusammengefasst.

- Das Vorgehensmodell *RAMEn* betrachtet die Produktentwicklung aus der Mesoebene heraus und schafft abstrakte, generalisierbare Rahmenbedingungen zur systematischen Einordnung, Auswahl und Integration von Methoden in Entwicklungsprozesse. Ein wesentlicher Mehrwert liegt in der Separierung von (Schnittfeld-)Aktivitäten und Phasen. Eine detaillierte Prozessintegration wird durch eine Dokumentationsvorschrift und identifizierte Hauptmerkmale realisiert. Die praktische Relevanz des Prozessmodells wurde anhand der Auslegung crashbelasteter Systeme demonstriert. Eine Generalisierung fand anhand datengetriebener Leichtbaukonstruktionen statt. Die Evaluierung hebt positive Effekte auf die Wissensgenerierung hervor, wobei die endgültige Validierung des Modells aufgrund mehrjähriger Entwicklungszyklen Teil zukünftiger Arbeiten ist. *RAMEn* ergänzt bestehende Modelle wie CRISP-DM. Schwachstellen bestehen im Fokus auf mechanische Produkte und noch nicht ausgeschöpfte Potenziale bei der Automatisierung der Methodenauswahl durch LLM-basierte Ansätze.
- *DrAGOS* nutzt RL-Agenten zur generations- und aufgabenübergreifenden Optimierung. Durch das Vortraining auf Metamodellen, die das Systemverhalten in aufwändigen FE-Simulationen abbilden, kann der RL-Agent auch komplexes Systemverhalten in weniger als einer Stunde erlernen. In der Anwendung überträgt der RL-Agent diese Strategie direkt auf FE-Simulationen. Die Evaluierung wird anhand der Rückhaltesystemauslegung durchgeführt. Im Vergleich mit klassischen Optimierern zeigt *DrAGOS* eine höhere Effizienz bei der Ermittlung von Optima in weniger als zehn Schritten und ist in einem Fall auch in der Lage, für eine ungesehene Problemstellung ein besseres Optimum zu spezifizieren. Robustheitsuntersuchungen bestätigen zudem auch die numerische Stabilität im Training. Das effiziente Vortraining sowie der Schulterschluss zur Produktgenerationsentwicklung erlauben eine saubere Prozessintegration. Schwächen liegen im Zeitaufwand durch die sequentielle Simulationsdurchführung und der möglichen Spezifikation suboptimaler Lösungen. Zukünftige Potenziale bestehen im Vergleich mit Transferlearning-Ansätzen und der Anwendung für verwandte Problemstellungen.

- *DAYSi* erlaubt die automatische Analyse von 1D-Zeitreihendaten, wie sie von Crashsensoren in FE-Simulationen oder physischen Crashtests ausgegeben werden. Mit dem Ziel, auffällige Datenpunkte und Sensoren im Rahmen des Abgleichs von Daten zu identifizieren, werden die in der Analyse ermittelten Abweichungen – sensorweise für jedes Signal in allen Simulationen – in Bezug auf die Magnitude und die Zeit bestimmt. Weitere Einblicke hinsichtlich der Ursachenfindung innerhalb einer Simulation liefert eine Eventkette, die durch Kreuzkorrelationen ermittelt wird. Die Rechenzeit ist mit unter fünf Minuten sehr gering, und der ISO-MME-Standard ermöglicht einen übergreifenden Einsatz, was die Prozessintegration erleichtert. In der Evaluierung werden grobe und leichte Ausreißer hervorgehoben und Ursachen korrekt auf ursächliche Sensoren zurückgeführt. Schwachstellen bestehen in der eingeschränkten Interpretierbarkeit der Abweichungskoeffizienten und der Ursachenklassifikation, die auch weiterhin Ingenieurwissen erfordert. Zukünftige Arbeiten könnten Methoden zur automatisierten Ursachenanalyse mittels Retrieval Augmented Generation (RAG) und eine vertiefte Untersuchung der Mensch-Maschine-Kollaboration umfassen.
- *ExPLAIN* ermöglicht die modulare Verarbeitung verschiedener Datentypen (Skalare, 1D-Zeitreihen- und FE-Daten) und fokussiert auf Ursachen- und Beitragsanalyse beliebiger Zusammenhänge in Crashsimulationen. Mittels SHAP werden Einflussgrößen, deren Reihenfolge und Wirkrichtung robust identifiziert. Die eingeführten System- und Differenz-SHAP-Werte erlauben die anwendungsbezogene Analyse auf verschiedenen Systemebenen und den A/B-Vergleich. Im Rahmen der Evaluierung wird ein Vergleich einer optimierten Rohbaukonfiguration gegenüber einer initialen Rohbaukonfiguration mit Einflüssen im Ausgaberaum durchgeführt. Ebenso werden in einem unparametrisierten Datensatz Einflüsse auf die Brustbeschleunigung identifiziert. Bisher nicht nachvollziehbare Einbettungsräume von FE-Daten, die zur Ausreißerdetektion verwendet werden, können nun erklärt werden. *ExPLAIN* verbessert das Verständnis der Daten und Systeme und erhöht die Transparenz in bestehenden Blackbox-Prozessen. Schwächen liegen in der benötigten großen Datenmenge, fehlendem physikalischem Bewusstsein der Metamodelle und der Notwendigkeit der domänenspezifischen Interpretation. Zukünftige Potenziale bestehen in der weiteren Aufbereitung von Ergebnissen, beispielsweise durch LLM-generierte Reports, sowie der Vertiefung der Untersuchung zu abhängigen Eingaben.

Die vier eingeführten Ansätze *RAMEn*, *DrAGOS*, *DAYSi* und *ExPLAIN* unterstützen Ingenieursaktivitäten an unterschiedlichen Stellen. Es wurde gezeigt, dass diese dazu beitragen, Entwicklungsprozesse in der Auslegung crashbelasteter Systeme effizienter, transparenter und wissensbasierter zu gestalten. Allem voran steht dabei weiterhin das Ziel, die Mobilität von morgen noch ein Stück sicherer als gestern zu gestalten.



# Literatur

- [AJL21] K. Aas, M. Jullum und A. Løland. “Explaining individual predictions when features are dependent: More accurate approximations to Shapley values”. In: *Artificial Intelligence* 298 (2021), S. 103502. ISSN: 0004-3702. DOI: 10.1016/j.artint.2021.103502.
- [Ack+08] S. Ackermann, L. Gaul, M. Hambrecht und M. Hanss. “Principal component analysis for detection of globally important input parameters in nonlinear finite element analysis”. In: *Weimar Optimization and Stochastic Days*. Weimar, Deutschland, 2008.
- [AB11] A. Albers und A. Braun. “A generalised framework to compass and to support complex product engineering processes”. In: *International Journal of Product Development* 15.1–3 (2011), S. 6–25.
- [ABW15] A. Albers, N. Bursac und E. Wintergerst. “Product Generation Development - Importance and Challenges from a Design Research Perspective”. In: *New developments in mechanics and mechanical engineering : Proceedings of the International Conference on Mechanical Engineering (ME 2015) : Proceedings of the International Conference on Theoretical Mechanics and Applied Mechanics (TMAM 2015)*. Hrsg. von N. E Mastorakis. Bd. 13. Vienna, Österreich, März 2015, S. 16–21. ISBN: 978-1-61804-288-0.
- [AR22] A. Albers und S. Rapp. “Model of SGE: System Generation Engineering as Basis for Structured Planning and Management of Development”. In: *Design Methodology for Future Products*. Hrsg. von D. Krause und E. Heyden. Cham, Schweiz: Springer International Publishing, 2022, S. 27–46. ISBN: 978-3-030-78368-6.
- [Alb+19] A. Albers, S. Rapp, M. Spadinger, T. Richter, C. Birk, F. Marthaler, J. Heimicke, V. Kurtz und H. Wessels. “The Reference System in the Model of PGE: Proposing a Generalized Description of Reference Products and their Interrelations”. In: *Proceedings of the Design Society: International Conference on Engineering Design (ICED19) 1.1 (Aug. 2019)*, S. 1693–1702. DOI: 10.1017/dsi.2019.175.
- [Alb+15] A. Albers, N. Reiß, N. Bursac, B. Walter und B. Gladysz. “InnoFox – Situations-spezifische Methodenempfehlung im Produktentstehungsprozess”. In: *Stuttgarter Symposium für Produktentwicklung*. Stuttgart, Deutschland, 2015.
- [Alb+16] A. Albers, N. Reiss, N. Bursac und T. Richter. “IPeM-Integrated Product Engineering Model in Context of Product Generation Engineering”. In: *Procedia CIRP* 50 (2016), S. 100–105. ISSN: 2212-8271. DOI: 10.1016/j.procir.2016.04.168.

- [AHC15] M. M. Andreasen, C. T. Hansen und P. Cash. *Conceptual design: interpretations, mindset and models*. 1. Aufl. Cham, Schweiz: Springer International Publishing, 2015. ISBN: 978-3319198385.
- [Ans24] Ansys, Inc. *Pressemeldung: Ansys Releases AI-Powered Virtual Assistant AnsysGPT*. Letzter Aufruf: 04.06.2025. 2024. URL: <https://www.ansys.com/de-de/news-center/press-releases/4-11-24-ansys-releases-ansysgpt>.
- [Ans25] Ansys, Inc. *Website: Ansys LS-DYNA (Multiphysik-Solver)*. Letzter Aufruf: 24.09.2025. 2025. URL: <https://www.ansys.com/de-de/products/structures/ansys-ls-dyna>.
- [Atl25] Atlassian, Corp. *Website: Atlassian Confluence*. Letzter Aufruf: 20.05.2025. 2025. URL: <https://www.atlassian.com/de/software/confluence>.
- [BRK21] M. K. Belaid, M. Rabus und R. Krestel. “CrashNet: An encoder–decoder architecture to predict crash test outcomes”. In: *Data Mining and Knowledge Discovery* 35 (2021), S. 1688–1709. DOI: 10.1007/s10618-021-00753-2.
- [BH16] T. Berry und J. Harlim. “Variable bandwidth diffusion kernels”. In: *Applied and Computational Harmonic Analysis* 40.1 (2016), S. 68–96. ISSN: 1063-5203. DOI: 10.1016/j.acha.2015.01.001.
- [BSS21] F. Beyer, D. Schneider und A. Schumacher. “Finding three-dimensional layouts for crashworthiness load cases using the graph and heuristic based topology optimization”. In: *Structural and Multidisciplinary Optimization* 63 (2021), S. 1–15. DOI: 10.1007/s00158-020-02768-0.
- [Bey19] T. Beysolow II. *Applied Reinforcement Learning with Python: With OpenAI Gym, Tensorflow, and Keras*. 1. Aufl. Berkeley, Kalifornien, Vereinigte Staaten von Amerika: Apress, 2019. ISBN: 978-1-4842-5127-0.
- [BGW23] S. Bickel, S. Goetz und S. Wartzack. “Detection of Plausibility and Error Reasons in Finite Element Simulations with Deep Learning Networks”. In: *Algorithms* 16.209 (2023). DOI: 10.3390/a16040209.
- [Bic+19] S. Bickel, T. C. Sprügel, B. Schleich und S. Wartzack. “How Do Digital Engineering and Included AI Based Assistance Tools Change the Product Development Process and the Involved Engineers”. In: *Proceedings of the Design Society: International Conference on Engineering Design (ICED19)* 1.1 (Aug. 2019), S. 2567–2576. DOI: 10.1017/dsi.2019.263.
- [IRCOBI98] *IRCOBI Proceedings 1998: International IRCOBI Conference on Biomechanics of Impact*. Göteborg, Schweden, Sep. 1998.
- [Blá+21] A. Blázquez-García, A. Conde, U. Mori und J.A. Lozano. “A Review on Outlier/Anomaly Detection in Time Series Data”. In: *ACM Computing Surveys* 54.3 (Apr. 2021). ISSN: 0360-0300. DOI: 10.1145/3444690.

- [BC09] L. T. M. Blessing und A. Chakrabarti. *DRM, a Design Research Methodology*. 1. Aufl. London, Vereinigtes Königreich: Springer, 2009. ISBN: 978-1-84882-587-1.
- [Boh+13] B. Bohn, J. Garcke, R. Iza Teran, A. Paprotny, B. Peherstorfer, B. Schepsmeier und C.A. Thole. “Analysis of Car Crash Simulation Data with Nonlinear Machine Learning Methods”. In: *Procedia Computer Science* 18 (2013), S. 621–630. ISSN: 1877–0509. DOI: 10.1016/j.procs.2013.05.226.
- [Böh+23] P. Böhringer, D. Sommer, T. Haase, M. Barteczko, J. Sprave, M. Stoll, C. Karadogan, D. Koch, P. Middendorf und M. Liewald. “A strategy to train machine learning material models for finite element simulations on data acquirable from physical experiments”. In: *Computer Methods in Applied Mechanics and Engineering* 406 (2023), S. 115894. ISSN: 0045–7825. DOI: 10.1016/j.cma.2023.115894.
- [Bom+24] R. Bommasani, E. Brunskill, E. Brynjolfsson, E. Capstick, J. Clark, L. Fattorini, T. Gertsenberg, N. Goodman, N. Haber, S. Koyejo, P. Liang, K. Ligett, S. Luccioni, N. Maslej, J.C. Niebles, S. Oak, V. Parli, R. Perrault, A. Shi, Y. Shoham und E. Williamson. “Chapter 2: Technical Performance”. In: *The AI Index 2024 Annual Report*. Hrsg. von AI Index Steering Committee. Stanford, Kalifornien, Vereinigte Staaten von Amerika, Apr. 2024.
- [BGS23] A. Borse, R. Gulakala und M. Stoffel. “Machine learning enhanced optimisation of crash box design for crashworthiness analysis”. In: *Proceedings in Applied Mathematics and Mechanics* 23.4 (Sep. 2023), e202300145. DOI: 10.1002/pamm.202300145.
- [BGS24a] A. Borse, R. Gulakala und M. Stoffel. “Development of a machine learning-based design optimization method for crashworthiness analysis”. In: *Archives of Mechanics* 76.1–2 (2024), S. 61–92. DOI: 10.24423/aom.4454.
- [BGS24b] A. Borse, R. Gulakala und M. Stoffel. “Multi-parameter design optimization of crash box for crashworthiness analysis”. In: *Proceedings in Applied Mathematics and Mechanics* 24.3 (2024), e202400096. DOI: 10.1002/pamm.202400096.
- [BHT22] T. Bougna, G. Hundal und P. Taniform. “Quantitative Analysis of the Social Costs of Road Traffic Crashes Literature”. In: *Accident Analysis & Prevention* 165 (2022), S. 106282. ISSN: 0001–4575. DOI: 10.1016/j.aap.2021.106282.
- [BB07] D. Braha und Y. Bar-Yam. “The statistical mechanics of complex product development: empirical and analytical results”. In: *Management Science* 53.7 (2007), S. 1127–1145.
- [BR03] D. Braha und Y. Reich. “Topological structures for modeling engineering design processes”. In: *Research in Engineering Design* 14.4 (2003), S. 185–199.

- [BL04] T. Braun und U. Lindemann. “Method adaption – A way to improve methodical product development”. In: *DS 32: Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik, Croatia*. Hrsg. von D. Marjanovic. 2004, S. 977–982.
- [Bre+84] L. Breiman, J. Friedman, R. A. Olshen und C. J. Stone. *Classification and Regression Trees*. 1. Aufl. New York: Chapman und Hall/CRC, 1984. ISBN: 9781315139470.
- [Bro+22] N. K. Brown, A. P. Garland, G. M. Fadel und G. Li. “Deep reinforcement learning for engineering design through topology optimization of elementally discretized design domains”. In: *Materials & Design* 218 (2022), S. 110672. ISSN: 0264-1275. DOI: 10.1016/j.matdes.2022.110672.
- [BFN06] T. R. Browning, E. Fricke und H. Negele. “Key concepts in modeling product development processes”. In: *Systems Engineering* 9.2 (2006), S. 104–128. DOI: 10.1002/sys.20047.
- [BMV07] H. Burr, M. Mueller und M. Vielhaber. “EIMS – A Framework for Engineering Process Analysis”. In: *DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France*. Hrsg. von J. C. Bocquet. 2007.
- [Büt22] J. Büttner. “Effiziente Lösungsansätze zur Reduktion des numerischen Ressourcenbedarfs für den operativen Einsatz der Multidisziplinären Optimierung von Fahrzeugstrukturen”. Dissertation. Wuppertal, Deutschland: Bergische Universität Wuppertal, Jan. 2022.
- [Büt+23] J. Büttner, A. Schumacher, T. Bäck, S. Schwarz und P. Krause. “Making multidisciplinary optimization fit for practical usage in car body development”. In: *Structural and Multidisciplinary Optimization* 66.62 (2023). DOI: 10.1007/s00158-023-03505-z.
- [CDP05] R. N. Cadete, J. P. Dias und M. S. Pereira. “Optimization in Vehicle Crashworthiness Design Using Surrogate Models”. In: *6th World Congress of Structural and Multidisciplinary Optimization*. Rio de Janeiro, Brasilien, 2005.
- [Can+20] M. Cantamessa, F. Montagna, S. Altavilla und A. Casagrande-Seretti. “Data-driven design: the new challenges of digitalization on product design and development”. In: *Design Science* 6 (2020), e27. DOI: 10.1017/dsj.2020.25.
- [CG16] T. Chen und C. Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, California, USA: Association for Computing Machinery, 2016, S. 785–794. DOI: 10.1145/2939672.2939785.
- [CKP02] M. J. Chung, P. Kwon und B. T. Pentland. “Making process visible: a grammatical approach to managing design processes”. In: *Journal of Mechanical Design* 124.3 (2002), S. 364–374.

- [Cim+25] P.-C. Cimpoesu, D. Toal, L. Wang, A. Keane, J. Gregory und M. Nunez. “Reusing Optimization Histories to Accelerate Engineering Design Optimization”. In: *AIAA Scitech 2025 Forum*. Orlando, Florida, Vereinigte Staaten von Amerika, Jan. 2025. DOI: 10.2514/6.2025-0775.
- [CMM10] CMMI Institute. *CMMI for Development, Version 1.3*. Pittsburgh, Pennsylvania, Vereinigte Staaten von Amerika: Carnegie Mellon University, 2010.
- [CL06] R. R. Coifman und S. Lafon. “Diffusion Maps”. In: *Applied and Computational Harmonic Analysis* 21.1 (2006), S. 5–30. DOI: 10.1016/j.acha.2006.04.006.
- [CK20] Z. Conti und S. Kaijima. “Explainable ML: Augmenting the interpretability of numerical simulation using Bayesian networks”. In: *The Routledge Companion to Artificial Intelligence in Architecture*. Hrsg. von I. As und P. Basu. 1. Aufl. Routledge, 2020.
- [Coo80] K. G. Cooper. “Naval ship production: a claim settled and a framework built”. In: *Interfaces* 10.6 (1980), S. 20–36.
- [Cra+02] K. Craig, N. Stander, D. Dooge und S. Varadappa. “MDO of Automotive Vehicle for Crashworthiness and NVH Using Response Surface Methods”. In: *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. Atlanta, Georgia, Vereinigte Staaten von Amerika, Sep. 2002, S. 5607. DOI: 10.2514/6.2002-5607.
- [Cuk+73] R. I. Cukier, C. M. Fortuin, K. E. Shuler, A. G. Petschek und J. H. Schaibly. “Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I. Theory”. In: *Journal of Chemical Physics* 59.8 (1973), S. 3873–3878.
- [DTK20] B. Delipetrev, C. Tsinaraki und U. Kostic. *Historical Evolution of Artificial Intelligence*. EUR Report 30221EN. Luxemburg, Luxemburg: Publications Office of the European Union, 2020.
- [DIN 60050] Deutsches Institut für Normung e.V. *Internationales Elektrotechnisches Wörterbuch – Teil 351: Leittechnik (IEC 60050-351:2013)*. DIN IEC 60050-351:2014-09. Berlin, Deutschland, 2014.
- [DIN 9000] Deutsches Institut für Normung e.V. *Qualitätsmanagementsysteme – Grundlagen und Begriffe (ISO 9000:2015)*. EN ISO 9000:2015-11. Berlin, Deutschland, 2015.
- [DHS16] C. Diez, L. Harzheim und A. Schumacher. “Effiziente Wissensgenerierung zur Robustheitsuntersuchung von Fahrzeugstrukturen mittels Modellreduktion und Ähnlichkeitsanalyse”. In: *VDI-Berichte*. Bd. 2279. Düsseldorf, Deutschland: VDI Verlag, 2016.

- [Die+18] C. Diez, P. Kunze, D. Toewe, C. Wieser, L. Harzheim und A. Schumacher. “Big-Data Based Rule-Finding for Analysis of Crash Simulations”. In: *Advances in Structural and Multidisciplinary Optimization*. Hrsg. von A. Schumacher, T. Vietor, S. Fiebig, K.-U. Bletzinger und K. Maute. Cham, Schweiz: Springer International Publishing, 2018, S. 396–410. ISBN: 978-3-319-67988-4.
- [Die+16] C. Diez, C. Wieser, L. Harzheim und A. Schumacher. “Automated Generation of Robustness Knowledge for selected Crash Structures”. In: *14th LS-DYNA Forum 2016*. Bamberg, Deutschland, Okt. 2016.
- [Div24] Divis GmbH. *Website: Divis – ClearVu Analytics*. Letzter Aufruf: 29.12.2024. 2024. URL: <https://www.divis-gmbh.de/clearvu-analytics>.
- [Don+24] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, T. Liu, B. Chang, X. Sun, L. Li und Z. Sui. “A Survey on In-Context Learning”. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Hrsg. von Y. Al-Onaizan, M. Bansal und Y.-N. Chen. Miami, Florida, Vereinigte Staaten von Amerika: Association for Computational Linguistics, Nov. 2024, S. 1107–1128. DOI: 10.18653/v1/2024.emnlp-main.64.
- [Dru+96] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola und V. Vapnik. “Support vector regression machines”. In: *Proceedings of the 10th International Conference on Neural Information Processing Systems (NIPS’96)*. Denver, Colorado, Vereinigte Staaten von Amerika, 1996, S. 155–161.
- [VMAP25] VMAP Standards Community e.V. *Website: VMAP Standard*. Letzter Aufruf: 14.05.2025. 2025. URL: <https://www.vmap-standard.org>.
- [EuroNCAP] Euro NCAP. *Abbildung: Euro NCAP Porsche Taycan 2019*. Letzter Aufruf: 12.09.2024. URL: <https://www.euroncap.com/de/results/porsche/taycan/39129>.
- [Fle+24] J. Fleischer, J. Scholz, S. Zeidler, S. Quirin, K. König, J. Mathieu, K. Mai, B. Kneidl, M. Betancourt, T. Arnold, M. Gwosdek, L. Auch, K. Bechteler, A. Schuh und W. Wenzel. *Handlungsleitfaden zur systemischen Entwicklung und der gezielten Hebung von Leichtbaupotenzialen*. Abschlussbericht. Jülich, Deutschland: Forschungszentrum Jülich, 2024. DOI: 10.5445/IR/1000173068.
- [FMVSS208] FMVSS 208. *Code of Federal Regulations, Title 49 Chapter V Part 571-208, Occupant Crash Protection*. <https://www.ecfr.gov/current/title-49/subtitle-B/chapter-V/part-571/subpart-B/section-571.208>. Aufruf: 16.06.2025.
- [FD04] H.-J. Franke und M. Deimel. “Selecting and combining methods for complex problem solving within the design process”. In: *DS 32: Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik, Croatia*. Hrsg. von D. Marjanovic. 2004, S. 213–218.

- [Fra24] Fraunhofer SCAI. Website: *Fraunhofer SCAI – SimExplore*. Letzter Aufruf: 29.12.2024, 2024. URL: <https://www.scai.fraunhofer.de/de/geschaeftsfelder/numerische-datenbasierte-vorhersage/produkte/simexplore.html>.
- [FS99] Y. Freund und R. E. Schapire. “A Short Introduction to Boosting”. In: *Journal of the Japanese Society for Artificial Intelligence* 14.5 (Sep. 1999), S. 771–780.
- [GB18] J. Gao und A. Bernard. “An overview of knowledge sharing in new product development”. In: *The International Journal of Advanced Manufacturing Technology* 94 (2018), S. 1545–1550.
- [GI15] J. Garcke und R. Iza Teran. “Machine Learning Approaches for Repositories of Numerical Simulation Results”. In: *10th European LS-DYNA Conference*. Würzburg, Deutschland, 2015.
- [GPP17] J. Garcke, M. Pathare und N. Prabakaran. “ModelCompare”. In: *Scientific Computing and Algorithms in Industrial Simulations: Projects and Products of Fraunhofer SCAI*. Hrsg. von M. Griebel, A. Schüller und M. A. Schweitzer. Cham, Schweiz: Springer International Publishing, 2017, S. 199–205. ISBN: 978-3-319-62458-7.
- [Ger+21a] K. Gericke, B. Bender, G. Pahl, W. Beitz, J. Feldhusen und K.-H. Grote. “Der Produktentwicklungsprozess”. In: *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*. Hrsg. von B. Bender und K. Gericke. Berlin/Heidelberg, Deutschland: Springer, 2021, S. 27–55. ISBN: 978-3-662-57303-7.
- [Ger+21b] K. Gericke, B. Bender, G. Pahl, W. Beitz, J. Feldhusen und K.-H. Grote. “Grundlagen methodischen Vorgehens in der Produktentwicklung”. In: *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*. Hrsg. von B. Bender und K. Gericke. Berlin/Heidelberg, Deutschland: Springer, 2021, S. 27–55. ISBN: 978-3-662-57303-7.
- [GB12] K. Gericke und L. Blessing. “An Analysis of Design Process Models Across Disciplines”. In: *DS 70: Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia*. Hrsg. von D. Marjanovic, M. Storga, N. Pavkovic und N. Bojcetic. 2012, S. 171–180.
- [GK14] J. S. Gero und U. Kannengiesser. “The function-behaviour-structure ontology of design”. In: *An anthology of theories and models of design: philosophy, approaches and empirical explorations*. Hrsg. von A. Chakrabarti und L. T. M. Blessing. London, Vereinigtes Königreich: Springer-Verlag, 2014, S. 263–283.
- [Gér19] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2. Aufl. Farnham, Vereinigtes Königreich: O’Reilly Media, 2019.

- [GGW23] B. Gerschütz, S. Goetz und S. Wartzack. “AI4PD—Towards a Standardized Interconnection of Artificial Intelligence Methods with Product Development Processes”. In: *Applied Sciences* 13.5 (2023), S. 3002. DOI: 10.3390/app13053002.
- [GSW21] B. Gerschütz, B. Schleich und S. Wartzack. “A Semantic Web Approach for Structuring Data-Driven Methods in the Product Development Process”. In: *Proceedings of the 32nd Symposium Design for X*. Hrsg. von D Krause, K Paetzold und S. Wartzack. Sep. 2021. DOI: 10.35199/dfx2021.15.
- [Ger+21c] B. Gerschütz, B. V. M. Spießl, B. Schleich und S. Wartzack. “An Adapted Method for Design Process Capturing to Meet the Challenges of Digital Product Development”. In: 1 (Aug. 2021), S. 365–374. DOI: 10.1017/pds.2021.37.
- [Get+22] C. F. Gethmann, P. Buxmann, J. Distelrath, B. G. Humm, S. Lingner, V. Nitsch, J. C. Schmidt und I. Spiecker. “Einführung”. In: *Künstliche Intelligenz in der Forschung: Neue Möglichkeiten und Herausforderungen für die Wissenschaft*. Bd. 48. Ethics of Science and Technology Assessment. Berlin/Heidelberg, Deutschland: Springer, 2022. ISBN: 978-3-662-63449-3.
- [Goe06] M. Goeken. “Entwicklung von Data-Warehouse-Systemen – Anforderungsmanagement, Modellierung, Implementierung”. Dissertation. Marburg, Deutschland: Philipps-Universität Marburg, 2006.
- [Gon+21] Mark Gonter, Peter Knoll, André Leschke, Ulrich Seiffert und Florian Weinert. “Fahrzeugsicherheit”. In: *Vieweg Handbuch Kraftfahrzeugtechnik*. Hrsg. von Stefan Pischinger und Ulrich Seiffert. Wiesbaden, Deutschland: Springer Fachmedien, 2021, S. 1073–1160. ISBN: 978-3-658-25557-2.
- [GW22] L. Greve und B.P. van de Weg. “Surrogate modeling of parametrized finite element simulations with varying mesh topology using recurrent neural networks”. In: *Array* 14 (2022), S. 100137. DOI: 10.1016/j.array.2022.100137.
- [Hah24] S. Hahner. “Low-dimensional Representations for Diverse Collections of 3D Surface Meshes”. Dissertation. Bonn, Deutschland: Rheinischen Friedrich-Wilhelms-Universität Bonn, Jan. 2024.
- [HG22] S. Hahner und J. Garcke. “Mesh Convolutional Autoencoder for Semi-Regular Meshes of Different Sizes”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, Vereinigte Staaten von Amerika, 2022, S. 2344–2353. DOI: 10.1109/WACV51458.2022.00240.
- [HIG20] S. Hahner, R. Iza Teran und J. Garcke. “Analysis and Prediction of Deforming 3D Shapes Using Oriented Bounding Boxes and LSTM Autoencoders”. In: *Proceedings of the 29th International Conference on Artificial Neural Networks (ICANN)*. Bratislava, Slowakei, Sep. 2020.

- [HKG22] S. Hahner, F. Kerkhoff und J. Garcke. “Transfer Learning using Spectral Convolutional Autoencoders on Semi-Regular Surface Meshes”. In: *Proceedings of the First Learning on Graphs Conference (LoG 2022)*. Hrsg. von R. Bastian und P. Razvan. Bd. 198. Proceedings of Machine Learning Research. PMLR, Dez. 2022, 18:1–18:19.
- [Ham08] U. Hammerschall. “Flexible Methodenintegration in anpassbare Vorgehensmodelle”. Dissertation. München, Deutschland: Technische Universität München, 2008.
- [HO20] K. Hayashi und M. Ohsaki. “Reinforcement Learning and Graph Embedding for Binary Truss Topology Optimization Under Stress and Displacement Constraints”. In: *Frontiers in Built Environment* 6 (2020). ISSN: 2297–3362. DOI: 10.3389/fbuil.2020.00059.
- [HO22] K. Hayashi und M. Ohsaki. “Graph-based reinforcement learning for discrete cross-section optimization of planar steel frames”. In: *Advanced Engineering Informatics* 51 (2022), S. 101512. DOI: 10.1016/j.aei.2021.101512.
- [HCK24] J. Heo, M. G. Cho und T. Kim. “Optimization of Occupant Restraint System Using Machine Learning for THOR-M50 and Euro NCAP”. In: *Machines* 12.1 (2024). ISSN: 2075–1702. DOI: 10.3390/machines12010074.
- [Hol+23] N. Hollmann, S. Müller, K. Eggenesperger und F. Hutter. “TabPFN: A transformer that solves small tabular classification problems in a second”. In: *The 11th International Conference on Learning Representations (ICLR’23)*. Kigali, Ruanda, 2023.
- [Hol+25] N. Hollmann, S. Müller, L. Purucker, A. Krishnakumar, M. Körfer, S. B. Hoo, R. T. Schirmeister und F. Hutter. “Accurate predictions on small data with a tabular foundation model”. In: *Nature* (Jan. 2025). DOI: 10.1038/s41586-024-08328-6.
- [HLM25] C. S. H. Hong, K. J. Lee und C. T. Mueller. “Deep Reinforcement Learning for Efficient Exploration of Combinatorial Structural Design Spaces”. In: *Proceedings of the IASS Annual Symposium 2025: The Living Past as a Source of Innovation*. Hrsg. von J. G. Oliva, J. I. del Cueto und E. Drago. Mexiko-Stadt, Mexiko, Sep. 2025.
- [Hor17] H. Horii. “Multi-objective optimization of vehicle occupant restraint system by using evolutionary algorithm with response surface model”. In: *International Journal of Computational Methods and Experimental Measurements* 5 (2017), S. 163–173. ISSN: 2.
- [Hor21] H. Horii. “Advancement of vehicle occupant restraint system design by integration of artificial intelligence technologies”. In: *International Journal of Transport Development and Integration* 5 (2021), S. 242–253. ISSN: 3.

- [Hua+15] Y. Huang, Q. Zhou, X. Zhang und C. Wang. “Optimization study of occupant restraint system concerning variations in occupant size and crash severity in frontal collisions”. In: *International Journal of Vehicle Safety* 8 (Nov. 2015), S. 299–313. DOI: 10.1504/IJVS.2015.074373.
- [HBS22] B. G. Humm, P. Buxmann und J. C. Schmidt. “Grundlagen und Anwendungen von KI”. In: *Künstliche Intelligenz in der Forschung*. Bd. 48. Ethics of Science and Technology Assessment. Berlin/Heidelberg, Deutschland: Springer, 2022. DOI: 10.1007/978-3-662-63449-3\_2.
- [IBM16] IBM, Corp. *Analytics Solutions Unified Method – Implementations with Agile Principles*. Letzter Aufruf: 16.04.2025. Armonk, New York, Vereinigte Staaten von Amerika, 2016. URL: <https://public.dhe.ibm.com/software/data/sw-library/services/ASUM.pdf>.
- [Ill22] B. Illmer. “Vorgehensmodelle für die virtuelle Inbetriebnahme von cyberphysischen Produktionssystemen”. Dissertation. Saarbrücken, Deutschland: Universität des Saarlandes, 2022.
- [ISO 23053] ISO 23053. *Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)*. 2. Aufl. Genf, Schweiz, 2023.
- [ISO 15288] ISO/IEC/IEEE 15288. *Systems and software engineering – System life cycle processes*. 2. Aufl. Genf, Schweiz, 2023.
- [ISO 13499] ISO/TS 13499. *Road Vehicles – Multimedia data exchange format for impact tests*. 3. Aufl. Genf, Schweiz, 2019.
- [Iza14] R. Iza Teran. “Enabling the analysis of finite element simulation bundles”. In: *International Journal for Uncertainty Quantification* (Jan. 2014), S. 95–110. DOI: 10.1615/Int.J.UncertaintyQuantification.2013005436.
- [IG19] R. Iza Teran und J. Garcke. “A Geometrical Method for Low-Dimensional Representations of Simulations”. In: *SIAM/ASA Journal on Uncertainty Quantification* 7.2 (2019), S. 472–496. DOI: 10.1137/17M1154205.
- [Iza+23] R. Iza Teran, D. Steffes-Lai, A.-D. Pham, M. Pathare und J. Garcke. “Event Detection Methods for Multi-Sensor CAE-Data”. In: *13th European LS-DYNA Users Conference*. Baden-Baden, Deutschland, 2023.
- [Jän07] J. Jänsch. “Akzeptanz und Anwendung von Konstruktionsmethoden im industriellen Einsatz: Analyse und Empfehlungen aus kognitionswissenschaftlicher Sicht”. Dissertation. Darmstadt, Deutschland: Technische Universität Darmstadt, 2007.

- [Joh+24] N. N. Johnson, V. Madhavadas, B. Asati, A. Giri, S. A. Hanumant, N. Shajan, K. S. Arora und S. K. Selvaraj. “Implementation of Machine Learning Algorithms for Weld Quality Prediction and Optimization in Resistance Spot Welding”. In: *Journal of Materials Engineering and Performance* 33 (2024), S. 6561–6585. DOI: 10.1007/s11665-023-08503-2.
- [JF88] R. E. Johnson und B. Foote. “Designing Reusable Classes”. In: *Journal of Object-Oriented Programming* 1.2 (1988), S. 22–35.
- [Jol02] I. T. Jolliffe. *Principal Component Analysis*. 2. Aufl. Springer Series in Statistics. New York, New York, Vereinigte Staaten von Amerika: Springer Science+Business Media, 2002. ISBN: 978-0-387-95442-4.
- [JGK21] H. Joodaki, B. Gepner und J. Kerrigan. “Leveraging machine learning for predicting human body model response in restraint design simulations”. In: *Computer Methods in Biomechanics and Biomedical Engineering* 24.6 (2021), S. 597–611.
- [Kaz+10] A. Kazakci, A. Hatchuel, P. Le Masson und B. Weil. “Simulation of design reasoning based on C-K theory: a model and an example application”. In: *DS 60: Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik, Croatia*. Hrsg. von D. Marjanovic, M. Storga, N. Pavkovic und N. Bojcetic. 2010, S. 59–68.
- [KW17] T. N. Kipf und M. Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In: *The 5th International Conference on Learning Representations (ICLR’17)*. Toulon, Frankreich, 2017.
- [KRH17] J. Kirchherr, D. Reike und M. Hekkert. “Conceptualizing the circular economy: An analysis of 114 definitions”. In: *Resources, Conservation and Recycling* 127 (2017), S. 221–232. ISSN: 0921-3449. DOI: 10.1016/j.resconrec.2017.09.005.
- [Kle15] B. Klein. *FEM: Grundlagen und Anwendungen der Finite-Element-Methode im Maschinen- und Fahrzeugbau*. 10. Aufl. Wiesbaden, Deutschland: Springer Vieweg, 2015. ISBN: 978-3-658-06054-1.
- [KG19] B. Klein und T. Gänsicke. *Leichtbau-Konstruktion: Dimensionierung, Strukturen, Werkstoffe und Gestaltung*. 11. Aufl. Wiesbaden, Deutschland: Springer Vieweg, 2019. ISBN: 978-3-658-26846-6.
- [Kle+18] S. Klement, B. Saske, S.G. Arndt und R. Stelzer. “Prozessmodell für Datenanalysen in der Produktentwicklung”. In: *16. Gemeinsames Kolloquium Konstruktions-technik*. Bayreuth, Deutschland, 2018, S. 168–179.

- [Klu+23] C. Klug, M. Schachner, A. Eggers, J. Galazka, S. Gargallo, C. Jimenez, J. Kirch, I. Levallois, U. Lobenwein, N. Meissner, V. Pardede, B. Pipkorn, J. Ellway und M. van Ratingen. “EURO NCAP Virtual Testing – Crashworthiness”. In: *Proceedings of the 27th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*. Yokohama, Japan: NHTSA, 2023, S. 693–705.
- [Kne+25] J. Kneifl, J. Fehr, S. L. Brunton und J. N. Kutz. “Multi-hierarchical surrogate learning for explicit structural dynamical systems using graph convolutional neural networks”. In: *Computational Mechanics* 75.5 (2025), S. 1115–1135. DOI: 10.1007/s00466-024-02553-6.
- [KB18] M. Koch und T. Bäck. “Machine Learning for Predicting the Impact Point of a Low Speed Vehicle Crash”. In: *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Orlando, Florida, Vereinigte Staaten von Amerika: IEEE, Dez. 2018, S. 1432–1437.
- [KWB18] M. Koch, H. Wang und T. Bäck. “Machine Learning for Predicting the Damaged Parts of a Low Speed Vehicle Crash”. In: *13th International Conference on Digital Information Management (ICDIM)*. Berlin, Deutschland: IEEE, Sep. 2018, S. 179–184.
- [Koh+21] C. P. Kohar, L. Greve, T. K. Eller, D.S. Connolly und K. Inal. “A machine learning framework for accelerating the design process using CAE simulations: An application to finite element analysis in structural crashworthiness”. In: *Computer Methods in Applied Mechanics and Engineering* 385 (2021), S. 114008. DOI: 10.1016/j.cma.2021.114008.
- [KW23] F. Koppenhagen und C. H. Wecht. “Warum das nächste iPhone wieder nicht aus Deutschland kommt: Der Zusammenhang zwischen Entwicklungsmethodik und radikalen Produktinnovationen – Ein Vergleich zwischen Stanford und Deutschland”. In: *Konstruktion* 6 (2023), S. 68–74. ISSN: 0720-5953. DOI: 10.37544/0720-5953-2023-06-68.
- [Kos+11] A. Kossiakoff, W. N. Sweet, S. J. Seymour und S. M. Biemer. *Systems Engineering Principles and Practice*. John Wiley & Sons, Inc., 2011. ISBN: 9780470405482.
- [Kra24] D. Kracker. “Automatisierte Auswertung von Crashesimulationen unterschiedlicher Fahrzeug-Entwicklungsstände mit Methoden des maschinellen Lernens”. Dissertation. Wuppertal, Deutschland: Bergische Universität Wuppertal, 2024.
- [Kra+23] D. Kracker, R. K. Dhanasekaran, A. Schumacher und J. Garcke. “Method for automated detection of outliers in crash simulations”. In: *International Journal of Crashworthiness* 28.1 (2023), S. 96–107. DOI: 10.1080/13588265.2022.2074634.

- [KGS20] D. Kracker, J. Garcke und A. Schumacher. “Automatic Analysis of Crash Simulations with Dimensionality Reduction Algorithms such as PCA and t-SNE”. In: *16th International LS-DYNA Users Conference*. Virtuelles Event, 2020.
- [KL11] M. Kreimeyer und U. Lindemann. *Complexity Metrics in Engineering Design: Managing the Structure of Design Processes*. 1. Aufl. Berlin/Heidelberg, Deutschland: Springer, 2011. ISBN: 978-3-642-20962-8.
- [KEW97] V. Krishnan, S. D. Eppinger und D. E. Whitney. “A model-based framework to overlap product development activities”. In: *Management Science* 43.4 (1997), S. 437–451.
- [LB20] H. Lakkaraju und O. Bastani. “How do I fool you?: Manipulating user trust via misleading black box explanations”. In: *Proceedings of the 2020 AAAI/ACM Conference on AI, Ethics, and Society (AIES '20)*. New York, New York, Vereinigte Staaten von Amerika, Feb. 2020, S. 79–85.
- [LL20] A. J. Landgraf und Y. Lee. “Dimensionality reduction for binary data through the projection of natural parameters”. In: *Journal of Multivariate Analysis* 180 (2020), S. 104668. ISSN: 0047–259X. DOI: 10.1016/j.jmva.2020.104668.
- [Lic+21] H. Lickert, A. Wewer, S. Dittmann, P. Bilge und F. Dietrich. “Selection of Suitable Machine Learning Algorithms for Classification Tasks in Reverse Logistics”. In: *Procedia CIRP* 96 (2021), S. 272–277. ISSN: 2212-8271. DOI: 10.1016/j.procir.2021.01.086.
- [Lun+20] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal und S.-I. Lee. “From local explanations to global understanding with explainable AI for trees”. In: *Nature Machine Intelligence* 2 (2020), S. 56–67.
- [LL17] S. M. Lundberg und S.-I. Lee. “A unified approach to interpreting model predictions”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Hrsg. von U. von Luxburg, I. Guyon, S. Bengio, H. Wallach und R. Fergus. Long Beach, Kalifornien, Vereinigte Staaten von Amerika, 2017, S. 4768–4777.
- [MOS19] M. Mase, A. B. Owen und B. Seiler. “Explaining black box decisions by Shapley cohort refinement”. In: *arXiv Preprint arXiv:1911.00467* (2019).
- [MOS21] M. Mase, A. B. Owen und B. Seiler. “Cohort Shapley value for algorithmic fairness”. In: *arXiv Preprint arXiv:2105.07168* (2021).
- [MW10] H. Mason und C. H. Wiggins. *A Taxonomy of Data Science*. Letzter Aufruf: 16.04.2025. 2010. URL: [https://introdatasci.dlilab.com/pdf/A\\_Taxonomy\\_of\\_Data\\_Science.pdf](https://introdatasci.dlilab.com/pdf/A_Taxonomy_of_Data_Science.pdf).

- [Mat23] J. Mathieu. *DE 10 2023 117 183 A1: Verfahren und System zur Verbesserung des Systemverhaltens einer technischen Entität*. Patentoffenlegungsschrift. München, Deutschland, Juni 2023.
- [MDV23] J. Mathieu, M. Di Roberto und M. Vielhaber. “A data-driven methodology for the automated analysis and explanation of system behavior in crash simulations”. In: *14th European LS-DYNA Conference*. Baden-Baden, Deutschland, 2023.
- [Mat+24] J. Mathieu, P. Gupta, M. Di Roberto und M. Vielhaber. “Minimizing occupant loads in vehicle crashes through reinforcement learning-based restraint system design: assessing performance and transferability”. In: *Proceedings of the Design Society: DESIGN Conference 4* (2024), S. 2139–2148. DOI: 10.1017/pds.2024.216.
- [Mat+25] J. Mathieu, S. Kronwitter, J. Pfahler, M. Di Roberto und M. Vielhaber. “Enhancing numerical simulation analysis with the use of explainable artificial intelligence and large language models: a case study on vehicle side crash optimization”. In: *Proceedings of the Design Society: International Conference on Engineering Design (ICED25)* 5 (2025), S. 1545–1554. DOI: 10.1017/pds.2025.10168.
- [MV24] J. Mathieu und M. Vielhaber. “A Framework for the Selection of AI-Based Methods to Support the Development of Passive Vehicle Safety Systems”. In: *Proceedings of NordDesign 2024*. Hrsg. von J. Malmqvist, M. Candi, R. J. Saeundsson, F. Bystrom und O. Isaksson. 2024, S. 313–322. ISBN: 978-1-912254-21-7.
- [McC+55] J. McCarthy, M. L. Minsky, N. Rochester und C. E. Shannon. *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*. Letzter Aufruf: 29.05.2025. 1955. URL: <https://raysolomonoff.com/dartmouth/boxa/dart564props.pdf>.
- [MHM18] L. McInnes, J. Healy und J. Melville. “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction”. In: *arXiv Preprint arXiv:1802.03426* (2018).
- [Meh+21] J. Mehlstäubl, S. Nicklas, B. Gerschütz, N. Sprogies, B. Schleich, T. Lohner, S. Wartzack, K. Stahl und K. Paetzold. “Voraussetzungen für den Einsatz datengetriebener Methoden in der Produktentwicklung”. In: *Proceedings of the 32nd Symposium Design for X*. Hrsg. von D. Krause, K. Paetzold und S. Wartzack. 2021. DOI: 10.35199/dfx2021.13.
- [Mer+89] H. Mertz, A. Irwin, J. Melvin, R. Stanaker und M. S. Beebe. *Size, weight and biomechanical impact response requirements for adult size small female and large male dummies*. SAE Report 890756. SAE, 1989. DOI: 10.4271/890756.

- [Mil13] C. Mildner. “Numerische und experimentelle Untersuchungen des Crashverhaltens von FVK-verstärkten Metallstrukturbauteilen”. Dissertation. München, Deutschland: Technische Universität München, 2013.
- [MN10] R. D. Moen und C. L. Norman. “Circling back”. In: *Quality Progress* 43.11 (2010), S. 22–28.
- [Moh+15] D. Moher, L. Shamseer, M. Clarke, D. Ghersi, A. Liberati, M. Petticrew, P. Shekelle, L. A. Stewart und PRISMA-P Group. “Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement”. In: *Systematic Reviews* 4.1 (Jan. 2015), S. 1. DOI: 10.1186/2046-4053-4-1.
- [Mol+22] C. Molnar, A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller und W. Samek. “General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models”. In: *xxAI – Beyond Explainable AI*. Hrsg. von A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller und W. Samek. Bd. 13200. Cham, Schweiz: Springer, 2022, S. 1–18. ISBN: 978-3-031-04082-5.
- [Mor91] M. D. Morris. “Factorial sampling plans for preliminary computational experiments”. In: *Technometrics* 33.2 (1991), S. 161–174.
- [Mül+23] S. Müller, F. Natter, T. Roth, H. Talaat und C.-A. Thole. *Data Compression for Simulation Results*. Presentation am NAFEMS World Congress 2023. Tampa, Florida, Vereinigte Staaten von Amerika, Mai 2023.
- [Mus+24] M. Muschalik, H. Baniecki, F. Fumagalli, P. Kolpaczki, B. Hammer und E. Hüllermeier. “shapiq: Shapley Interactions for Machine Learning”. In: *Proceedings of the 38th Conference on Neural Information Processing Systems (NIPS’24)*. Vancouver, Kanada, 2024.
- [Nas+21] A. B. Nassif, M. A. Talib, Q. Nasir und F. M. Dakalbab. “Machine Learning for Anomaly Detection: A Systematic Review”. In: *IEEE Access* 9 (2021), S. 78658–78700. DOI: 10.1109/ACCESS.2021.3083060.
- [Nat+24] D. Nath, Ankit, D. R. Neog und S. S. Gautam. “Application of Machine Learning and Deep Learning in Finite Element Analysis: A Comprehensive Review”. In: *Archives of Computational Methods in Engineering* 31 (2024), S. 2945–2984. DOI: 10.1007/s11831-024-10063-0.
- [Noz+17] N. Nozaki, E. Konno, M. Sato, M. Sakairi, T. Shibuya, Y. Kanazawa und S. Georgescu. “Application of artificial intelligence technology in product design”. In: *Fujitsu Scientific and Technical Journal* 53 (Juli 2017), S. 43–51.
- [Ope23] OpenAI, Inc. *ChatGPT (GPT-4), Large Language Model*. Letzter Aufruf: 12.08.2024. 2023. URL: <https://www.openai.com/chatgpt>.

- [Ort+21] C. Ortmann, J. Sperber, D. Schneider, S. Link und A. Schumacher. “Crashworthiness design of cross-sections with the Graph and Heuristic based Topology Optimization incorporating competing designs”. In: *Structural and Multidisciplinary Optimization* 64 (2021), S. 1063–1077. DOI: 10.1007/s00158-021-02927-x.
- [PH20] G. Paaß und D. Hecker. *Künstliche Intelligenz: Was steckt hinter der Technologie der Zukunft?* 1. Aufl. Wiesbaden, Deutschland: Springer Vieweg, 2020.
- [Pah+07] G. Pahl, W. Beitz, J. Feldhusen und K.-H. Grote. *Engineering Design: A Systematic Approach*. 3. Aufl. London Vereinigtes Königreich: Springer, 2007. ISBN: 978-1-84628-318-5.
- [PGS23] A. Pakiman, J. Garcke und A. Schumacher. “Knowledge discovery assistants for crash simulations with graph algorithms and energy absorption features”. In: *Applied Intelligence* 53 (2023), S. 19217–19236. DOI: 10.1007/s10489-022-04371-w.
- [PGS25] A. Pakiman, J. Garcke und A. Schumacher. “A modified SimRank++ approach for searching crash simulation data”. In: *Applied Intelligence* 55 (2025), S. 614. DOI: 10.1007/s10489-024-05945-6.
- [PT06] G. Parry und C. Turner. “Application of lean visual process management tools”. In: *Production Planning & Control* 17.1 (2006), S. 77–86.
- [Pas+19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf und et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS’19)*. Red Hook, New York, Vereinigte Staaten von Amerika, 2019.
- [Ped+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot und É. Duchesnay. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (Nov. 2011), S. 2825–2830. ISSN: 1532-4435.
- [Pfa+21] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez und P. W. Battaglia. “Learning mesh-based simulation with graph networks”. In: *The 9th International Conference on Learning Representations (ICLR’21)*. Virtuelles Event, 2021.
- [PW15] F. Pianosi und T. Wagener. “A simple and efficient method for global sensitivity analysis based on cumulative distribution functions”. In: *Environmental Modelling & Software* 67 (Mai 2015), S. 1–11.
- [Pic+24] J. Pickel, S. Bickel, S. Goetz und S. Wartzack. “Integration of product development data for further ontological utilization”. In: *Proceedings of the Design Society: DESIGN Conference 4* (2024), S. 463–472. DOI: 10.1017/pds.2024.49.

- [Pie17] M. Pieper. *Mathematische Optimierung: Eine Einführung in die kontinuierliche Optimierung mit Beispielen*. 1. Aufl. Wiesbaden, Deutschland: Springer Fachmedien, 2017. ISBN: 978-3-658-16974-9.
- [PBS13] E. Plischke, E. Borgonovo und C. L. Smith. “Global sensitivity measures from given data”. In: *European Journal of Operational Research* 226.3 (Mai 2013), S. 536–550.
- [Pon07] J. C. Ponn. “Situative Unterstützung der methodischen Konzeptentwicklung technischer Produkte”. Dissertation. München, Deutschland: Technische Universität München, 2007.
- [Pra96] B. Prasad. “Toward definitions of a concurrent product design, development, and delivery (PD3) system”. In: *Concurrent Engineering* 4.2 (1996), S. 102–109. DOI: 10.1177/1063293X9600400205.
- [PB25] M. Puttegowda und S. Ballupete Nagaraju. “Artificial intelligence and machine learning in mechanical engineering: Current trends and future prospects”. In: *Engineering Applications of Artificial Intelligence* 142 (2025), S. 109910. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2024.109910.
- [Qi+17] C. R. Qi, H. Su, M. Kaichun und L. J. Guibas. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Bd. Honolulu, Hawaii, Vereinigte Staaten von Amerika. 2017, S. 77–85.
- [Rab+22] M. Rabus, M. K. Belaid, S. A. Maurer und S. Hiermaier. “Development of a Model for the Prediction of Occupant Loads in Vehicle Crashes: Introduction of the Real Occupant Load Criterion for Prediction (ROLC)”. In: *Automotive and Engine Technology* 7 (2022), S. 229–244. DOI: 10.1007/s41104-022-00111-x.
- [Reu25] R. Reuter. *Safety Companion*. online: carhs GmbH, 2025. URL: <https://www.carhs.de/de/companion-poster/product/safetycompanion-2025.html>.
- [RSG16] M. T. Ribeiro, S. Singh und C. Guestrin. “Why Should I Trust You?: Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, New York, Vereinigte Staaten von Amerika: Association for Computing Machinery, 2016, S. 1135–1144. DOI: 10.1145/2939672.2939778.
- [Ric83] E. Rich. *Artificial Intelligence*. 1. Aufl. New York, New York, Vereinigte Staaten von Amerika: McGraw-Hill, 1983. ISBN: 9780070522619.
- [Rie+20] M. Riesener, C. Doelle, M. Mendl-Heinisch und N. Klumpen. “Identification of evaluation criteria for algorithms used within the context of product development”. In: *Procedia CIRP* 91 (2020), S. 508–515. ISSN: 2212–8271. DOI: 10.1016/j.procir.2020.02.207.

- [Ros+20] R. Roscher, B. Bohn, M. F. Duarte und J. Garcke. “Explainable machine learning for scientific insights and discoveries”. In: *IEEE Access* 8 (2020), S. 42200–42216. DOI: 10.1109/ACCESS.2020.2976199.
- [Rou87] P. Rousseeuw. “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), S. 53–65. DOI: 10.1016/0377-0427(87)90125-7.
- [Rud24] S. Rudolph. “On Some Artificial Intelligence Methods in the V-Model of Model-Based Systems Engineering”. In: *Proceedings of the 12th International Conference on Model-Based Software and Systems Engineering (MODELSWARD 2024)*. Hrsg. von F. J. D. Mayo, L. F. Pires und E. Seidewitz. Rom, Italien, 2024, S. 386–393. DOI: 978-1-7138-9753-8.
- [RHW86] D. E. Rumelhart, G. E. Hinton und R. J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (1986), S. 533–536. DOI: 10.1038/323533a0.
- [Saa87] R. Saaty. “The analytic hierarchy process—what it is and how it is used”. In: *Mathematical Modelling* 9.3 (1987), S. 161–176. DOI: 10.1016/0270-0255(87)90473-8.
- [SAE J211-1] SAE International. *SAE J211: Instrumentation for Impact Test – Part 1: Electronic Instrumentation*. Letzter Aufruf: 19.06.2025. 2022. URL: [https://www.sae.org/standards/content/j211/1\\_202208](https://www.sae.org/standards/content/j211/1_202208).
- [SDS13] C. M. Schlick, S. Duckwitz und S. Schneider. “Project dynamics and emergent complexity”. In: *Computational and Mathematical Organization Theory* 19.4 (2013), S. 480–495.
- [Sch+17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford und O. Klimov. “Proximal Policy Optimization Algorithms”. In: *arXiv Preprint arXiv:1707.06347* (2017).
- [Sch20] A. Schumacher. *Optimierung mechanischer Strukturen: Grundlagen und industrielle Anwendungen*. 3. Aufl. Wiesbaden, Deutschland: Springer Vieweg, 2020. ISBN: 978-3-662-60327-7.
- [SB25] A. Schumacher und F. Beyer. *Graphen- und heuristikbasierte Topologieoptimierung von 3D-Crash-Strukturen von Personenkraftwagen*. Bd. 388. FAT-Schriftenreihe. 2025.
- [Sch04] K. Schwaber. *Agile Project Management with Scrum*. Redmond, Washington, USA: Microsoft Press, 2004. ISBN: 978-0735619937.
- [SBV23] G. Schweitzer, M. Bitzer und M. Vielhaber. “Engineering Graph as an approach to support Design Decisions in Product Development”. In: 3 (2023), S. 1625–1634. DOI: 10.1017/pds.2023.163.

- [Sid+24] Y. Siddiqui, A. Alliegro, A. Artemov, T. Tommasi, D. Sirigatti, V. Rosov, A. Dai und M. Nießner. “MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, Washington, Vereinigte Staaten von Amerika, 2024, S. 19615–19625.
- [SK10] I. Sobol und S. Kucherenko. “Derivative based global sensitivity measures”. In: *Procedia – Social and Behavioral Sciences* 2.6 (2010), S. 7745–7746.
- [Sob01] I. M Sobol. “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates”. In: *Mathematics and Computers in Simulation* 55.1 (2001), S. 271–280. ISSN: 0378–4754. DOI: 10.1016/S0378–4754(00)00270–6.
- [Son+24a] H. Song, Y. Sun, H. Gao, L. Guo und T. Wu. “A novel HI construction method based on healthy-state data training for rotating machinery components”. In: *Structural Health Monitoring* 24 (2024), S. 3849–3867. ISSN: 6. DOI: 10.1177/14759217241279784.
- [SLN25] S. Sonntag, J. Luttmer und A. Nagarajah. “Can large language models support machine learning implementation in product development? A comparative analysis and perspectives”. In: *Proceedings of the Design Society: International Conference on Engineering Design (ICED25)* 5 (2025), S. 861–870. DOI: 10.1017/pds.2025.10100.
- [Son+23] S. Sonntag, J. Luttmer, R. Pluhnau und A. Nagarajah. “A Pattern Language Approach to Identify Appropriate Machine Learning Algorithms in the Context of Product Development”. In: 3 (2023), S. 365–374. DOI: 10.1017/pds.2023.37.
- [Son+24b] S. Sonntag, E. Pohl, J. Luttmer, J. Geldermann und A. Nagarajah. “A conceptual MCDA-based framework for machine learning algorithm selection in the early phase of product development”. In: 4 (2024), S. 2257–2266. DOI: 10.1017/pds.2024.228.
- [SHT09] P. Spethmann, C. Herstatt und S. H. Thomke. “Crash simulation evolution and its impact on R&D in the automotive applications”. In: *International Journal of Product Development* 8.3 (2009), S. 291–305. DOI: 10.1504/IJPD.2009.024202.
- [Spr+20] T. C. Sprügel, S. Bickel, B. Schleich und S. Wartzack. “Approach and Application to Transfer Heterogeneous Simulation Data from Finite Element Analysis to Neural Networks”. In: *Journal of Computational Design and Engineering* 8 (2020), S. 1–18. DOI: 10.1093/jcde/qwaa079.
- [SSW17] T. C. Sprügel, T. Schröppel und S. Wartzack. “Generic Approach to Plausibility Checks for Structural Mechanics with Deep Learning”. In: *Proceedings of the 21st International Conference on Engineering Design (ICED 17)*. Vancouver, Kanada, 2017, S. 299–308.

- [SW14] T. C. Sprügel und S. Wartzack. “Konzept zur automatischen Bauteilerkennung innerhalb der FE-Softwareumgebung mittels Künstlichen Neuronalen Netzen”. In: *Proceedings of the 24th Symposium Design for X*. Hrsg. von D. Krause, K. Paetzold und S. Wartzack. Bamberg, Deutschland, 2014, S. 47–56. ISBN: 978-3-941492-79-0.
- [Ste24] J. Steinhäuser. “Prognose von Belastungswerten mittels Machine Learning in den frühen Phasen der Entwicklung von passiver Sicherheit”. In: *VDI Berichte 2245, SIMVEC Auslegung und Absicherung von Fahrzeugsystemen, 21. VDI Kongress mit Fachaustellung*. Baden-Baden, Deutschland, 2024, S. 429–452.
- [SCF24] S. Steininger, H. Camci und J. Fottner. “Current State, Potentials and Challenges for the Use of Artificial Intelligence in the early Phase of Product Development: A Survey”. In: *2024 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 2024, S. 551–555. DOI: 10.1109/IEEM62345.2024.10857061.
- [SP97] R. Storn und K. Price. “Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces”. In: *Journal of Global Optimization* 11 (1997), S. 341–359.
- [ST12] S. Suss und V. Thomson. “Optimal design processes under uncertainty and reciprocal dependency”. In: *Journal of Engineering Design* 23.10–11 (2012), S. 826–848.
- [SB20] R. Sutton und A. Barto. *Reinforcement Learning: an introduction*. Cambridge, Vereinigtes Königreich: The MIT Press, 2020.
- [TK99] T. Taura und A. Kubota. “A study on engineering history base”. In: *Research in Engineering Design* 11.1 (1999), S. 45–54.
- [The+24] S. Thel, L. Greve, B. van de Weg und P. van der Smagt. “Introducing Finite Element Method Integrated Networks (FEMIN)”. In: *Computer Methods in Applied Mechanics and Engineering* 427 (2024), S. 117073. ISSN: 0045-7825. DOI: 10.1016/j.cma.2024.117073.
- [Thi+06] M. Thiele, H. Mullerschön, M. van den Hove und B. Mlekusch. “Optimization of an Adaptive Restraint System Using LS-OPT and Visual Exploration of the Design Space Using D-SPEX”. In: *9th International LS-DYNA Users Conference*. Detroit, Michigan, Vereinigte Staaten von Amerika, 2006.
- [TM03] C.-A. Thole und L. Mei. “Reasons for Scatter in Crash Simulation Results”. In: *NAFEMS Seminar: Use of Stochastics in FEM Analyses*. Wiesbaden, Deutschland: NAFEMS, Mai 2003.
- [Tho+10] C.-A. Thole, L. Nikitina, I. Nikitin und T. Clees. “Advanced Mode Analysis for Crash Simulation Results”. In: *9th LS-DYNA Forum*. Bamberg, Deutschland, 2010.

- [Tib96] R. Tibshirani. “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), S. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.
- [Tri24] J. Trilling. “Unterstützung der Graphen- und Heuristikbasierten Topologieoptimierung crashbelasteter Strukturen durch Reinforcement Learning”. Dissertation. Wuppertal, Deutschland: Bergische Universität Wuppertal, 2024.
- [TSZ22] J. Trilling, A. Schumacher und M. Zhou. “Einsatz von Reinforcement Learning zur lokalen Versteifung von Extrusionsprofilen in Crashlastfällen”. In: *Zeitschrift für numerische Simulation und angrenzende Gebiete, NAFEMS-Online-Magazin* 61 (Feb. 2022), S. 59–70.
- [TSZ24] J. Trilling, A. Schumacher und M. Zhou. “Reinforcement learning based agents for improving layouts of automotive crash structures”. In: *Applied Intelligence* 54 (2024), S. 1751–1769. DOI: 10.1007/s10489-024-05276-6.
- [Ull+24] L. Ullrich, M. Buchholz, K. Dietmayer und K. Graichen. “Expanding the Classical V-Model for the Development of Complex Systems Incorporating AI”. In: *IEEE Transactions on Intelligent Vehicles* (2024), S. 1–15. DOI: 10.1109/TIV.2024.3434515.
- [Stein+22] B. van Stein, E. Raponi, Z. Sadeghi, N. Bouman, R.C.H.J. van Ham und T. Bäck. “A Comparison of Global Sensitivity Analysis Methods for Explainable AI With an Application in Genomic Prediction”. In: *IEEE Access* 10 (2022), S. 103364–103381. DOI: 10.1109/ACCESS.2022.3210175.
- [VDI 2221-1] Verein Deutscher Ingenieure. *VDI 2221 Blatt 1: Entwicklung technischer Produkte und Systeme – Modell der Produktentwicklung*. Düsseldorf, 2019.
- [VDI 2221-2] Verein Deutscher Ingenieure. *VDI 2221 Blatt 2: Entwicklung technischer Produkte und Systeme – Gestaltung individueller Produktentwicklungsprozesse*. Düsseldorf, Deutschland, 2019.
- [VDI 2206] Verein Deutscher Ingenieure. *VDI/VDE 2206: Entwicklung mechatronischer und cyber-physischer Systeme*. Düsseldorf, Deutschland, 2021.
- [Vir+20] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright und et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17 (2020), S. 261–272.
- [Wag22] M. Wagner. *Lineare und nichtlineare FEM: Eine Einführung mit Anwendungen in der Umformsimulation mit LS-DYNA*. 3. Aufl. Wiesbaden, Deutschland: Springer, 2022. DOI: 10.1007/978-3-658-36522-6.

- [Web14] C. Weber. “Modelling products and product development based on characteristics and properties”. In: *An anthology of theories and models of design: philosophy, approaches and empirical explorations*. Hrsg. von A. Chakrabarti und L. T. M. Blessing. London: Springer-Verlag, 2014, S. 327–352.
- [WH00] R. Wirth und J. Hipp. “CRISP-DM: Towards a Standard Process Model for Data Mining”. In: *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*. Manchester, Vereinigtes Königreich, 2000, S. 29–39. ISBN: 1902426088.
- [Wor23] World Health Organization. *Global Status Report on Road Safety 2023*. World Health Organization, 13. Dez. 2023. URL: <https://www.who.int/publications/i/item/9789240086517>.
- [WCC14] D. C. Wynn, N. H. M. Caldwell und P. J. Clarkson. “Predicting change propagation in complex design workflows”. In: *Journal of Mechanical Design* 136.8 (2014), S. 081009.
- [WC18] D. C. Wynn und P. J. Clarkson. “Process models in design and development”. In: *Research in Engineering Design* 29.2 (2018), S. 161–202. DOI: 10.1007/s00163-017-0262-7.
- [Xue+12] Z. Xue, S. Parashar, G. Li und Y. Fu. “Optimization Strategies to Explore Multiple Optimal Solutions and Its Application to Restraint System Design”. In: *SAE International Journal of Passenger Cars – Mechanical Systems* 5.1 (2012), S. 540–551. DOI: 10.4271/2012-01-0578.
- [YH23] S. Yavuz und S. Himmetoglu. “Development of a Restraint System for Rear-Facing Car Seats”. In: *Machines* 11.12 (2023). ISSN: 2075-1702. DOI: 10.3390/machines11121076.
- [Zan99] W. Zanker. “Situative Anpassung und Neukombination von Entwicklungsmethoden”. Dissertation. München, Deutschland: Technische Universität München, 1999.
- [ZY09] Y. Zeng und S. Yao. “Understanding design activities through computer simulation”. In: *Advanced Engineering Informatics* 23.3 (2009), S. 294–308.
- [ZRD19] M. Ziegler, S. Rossmann und S. Danzer. *Leading the Way to an AI-driven Organization: A Practical Guide for Executives to Navigate Tectonic Shifts and Build an Enduring Competitive Advantage with AI*. Letzter Aufruf: 19.06.2025. 2019. URL: [https://www.porsche-consulting.com/sites/default/files/2023-04/leading\\_the\\_way\\_to\\_an\\_ai-driven\\_organization\\_2019\\_c\\_porsche\\_consulting-v2.pdf](https://www.porsche-consulting.com/sites/default/files/2023-04/leading_the_way_to_an_ai-driven_organization_2019_c_porsche_consulting-v2.pdf).

# A Anhang

Der Anhang enthält weitere Einzelheiten zu den verwendeten Daten, den durchgeführten Robustheitsbetrachtungen und den Hyperparametern der ML-Modelle bzw. Optimierungsalgorithmen. Die Gliederung erfolgt nach den drei Methoden *DrAGOS* in Abschnitt A.1, *DAYSi* in Abschnitt A.2 ab S. 262 und *ExPLAIN* in Abschnitt A.3 ab S. 272.

## A.1 *DrAGOS* – Metamodelle und RL-Agent

Dieser Teil des Anhangs legt die Daten und Architektur der Metamodelle dar, mit denen der RL-Agent vortrainiert wurde, um die Zusammenhänge der Problemstellung in akzeptabler Zeit zu trainieren. Ebenso werden Hyperparameter für die Metamodelle, den RL-Agent und den Differential Evolution Algorithmus tabellarisch aufgeführt.

### A.1.1 Architektur des Metamodells zum Vortraining

Abbildung A.1 zeigt einen Überblick über die Architektur und enthält Beispiele für die visuelle Interpretation der Prognosegüte. Zur generellen Einordnung der Daten, wurde eine Dimensionsreduktion mittels PCA über alle Merkmale durchgeführt. Ersichtlich ist hier, dass sich drei Cluster bilden, die auf unterschiedliche Fahrzeugvarianten zurückgeführt werden können. Da die einzelnen Fahrzeuge unterschiedliche Crashpulse haben, die das Verhalten des Insassen maßgeblich beeinflussen, wurde hier ebenfalls eine PCA angewendet. Der Begriff Crashpuls bezieht sich auf die Fahrzeugverzögerung [Rab+22]. Im linken unteren Quadranten bildet sich ein Cluster, das auf eine Optimierungskampagne zurückzuführen ist. Hier weisen 98 Crashsimulationen einen identischen Crashpuls auf. Die übrigen Crashpulse sind in der Ebene verteilt. Unterschiedliche Sitzpositionen werden durch die drei gezeigten Abstände beschrieben.

Die Architektur in Abbildung A.1 enthält skalare Eingabeparameter, die die Numerik, Fahrzeugkonfiguration und das Rückhaltesystem beschreiben. Die grau hinterlegten Eingaben werden durch den RL-Agenten angepasst. Die übrigen Größen werden für das RL-Training später fest gesetzt. Für jede Ausgabegröße wird ein Metamodell trainiert und die Ergebnisse im Anschluss zusammengeführt und gemeinsam ausgegeben. Die Plots zeigen die Prognosegüte der trainierten XGBoost-Modelle für HIC15, a3ms und RA. Die Konvergenz des RMSE ist darunter gezeigt.

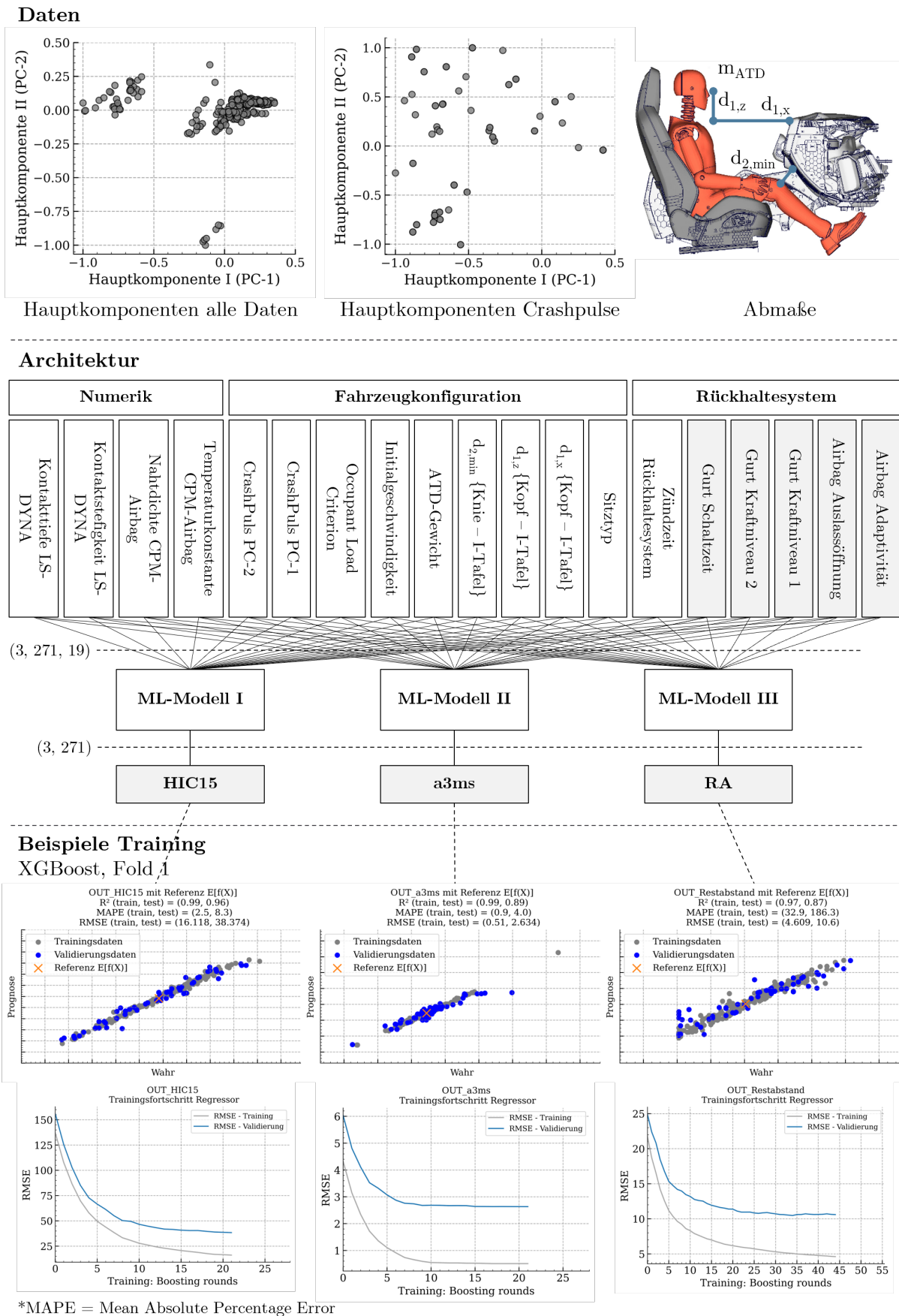


Abbildung A.1: *DrAGOS* Prognosemodell zum Vortraining der Optimierungsstrategie

## A.1.2 Hyperparameter der Metamodelle

Um das bestmögliche ML-Modell zum Vortraining des RL-Agenten auszuwählen, wurden verschiedene ML-Modelle auf der Daten trainiert. Alle Modelle wiesen dabei die Architektur aus Abbildung A.1 auf. Die Prognosegüten sind Tabelle 6.1 auf S. 130 zu entnehmen. Die zugehörigen Hyperparameter sind in den folgenden Tabellen angegeben. Im Trainingsvorgang wurden diese automatisch optimiert, sodass eine gewisse Anpassung des ML-Modells auf die vorliegenden Daten ermöglicht wird. Sofern nicht anders aufgeführt, wurde die Implementierung der klassischen ML-Modelle mit `scikit-learn` [Ped+11] und die des Neuronalen Netzwerks mit `torch` [Pas+19] vorgenommen.

**Tabelle A.1:** *DrAGOS* Vortraining: Hyperparameter TabPFN [Hol+25]

Pretrained Weights	Device	Ensembles	Eingaben	Fine-Tuning
Stand 28.02.2025, TabPFN GitHub	GPU	1	19	Nein

Die Hyperparameter der ML-Modelle für HIC15, a3ms und RA sind identisch.

**Tabelle A.2:** *DrAGOS* Vortraining: Hyperparameter XGBoost [CG16]

	HIC15	a3ms	RA	Beschreibung
<code>n_estimators</code>	22	22	45	Anzahl der Entscheidungsbäume im Ensemble
<code>learning_rate</code>	0.266	0.296	0.241	Schrittweite beim Update der Baumgewichte
<code>max_depth</code>	9	8	10	Maximale Tiefe der einzelnen Entscheidungsbäume
<code>min_child_weight</code>	3	1	10	Minimaler Gewichtssummenwert in einem Blattknoten
<code>subsample</code>	0.635	0.787	0.760	Anteil der Trainingsdaten für das zufällige Sampling pro Baum
<code>gamma</code>	1	1	3	Minimaler Verlustabfall für einen Split
<code>alpha</code>	2	0	5	L1-Regularisierungsterm für Blattgewichte
<code>lambda</code>	0	0	0	L2-Regularisierungsterm für Blattgewichte

**Tabelle A.3:** *DrAGOS* Vortraining: Hyperparameter Support Vector Regression

	HIC15	a3ms	RA	Beschreibung
<code>kernel</code>	<code>polynomial</code>	<code>polynomial</code>	<code>rbf</code>	Kernel-Funktion
<code>gamma</code>	0.053	0.053	0.053	Skalierung des Einflussradius des Kernels
<code>C</code>	1.95	3.61	8.67	Regularisierungsparameter
<code>epsilon</code>	0.346	0.133	0.100	Fehlertoleranzband

**Tabelle A.4:** *DrAGOS* Metamodell Vortraining: Hyperparameter Multilayer Perceptron

Architektur	Preprocessing	Training
<ul style="list-style-type: none"> <li>• 3 Hidden Layers, dim = 100</li> <li>• GELU-Aktivierung</li> <li>• LayerNorm (Layer 1 &amp; 2)</li> <li>• Dropout 0.1 (Layer 1 &amp; 2)</li> </ul>	<ul style="list-style-type: none"> <li>• Input: MinMaxScaler</li> <li>• Output: StandardScaler</li> <li>• Full-Batch Training</li> </ul>	<ul style="list-style-type: none"> <li>• Adam Optimizer, LR = 1e-3</li> <li>• Epochen = 100</li> <li>• MC-Samples = 10</li> <li>• KL-Gewicht = 1e-3</li> <li>• Heteroscedastic Loss</li> </ul>

Die Hyperparameter der ML-Modelle für HIC15, a3ms und RA sind identisch.

**Tabelle A.5:** *DrAGOS* Vortraining: Hyperparameter Decision Trees

	HIC15	a3ms	RA	Beschreibung
criteron	squared_error	squared_error	squared_error	Fehlerfunktion zur Bewertung der Splits
max_depth	17	20	3	Maximale Tiefe des Baums
min_samples_split	7	20	13	Anzahl Samples, ab der ein Knoten geteilt wird
min_samples_leaf	3	3	2	Minimale Anzahl an Samples, die ein Blatt haben muss
max_leaf_nodes	36	46	30	Maximale Anzahl an Blättern im Baum
ccp_alpha	2.81e-05	0.00223	0.04410	Parameter zur Steuerung der Baumkomplexität

**Tabelle A.6:** *DrAGOS* Vortraining: Hyperparameter Lasso Regression

	HIC15	a3ms	RA	Beschreibung
alpha	0.00203	0.00220	0.01056	Regularisierungsparameter
max_iterations	1000	1000	1000	Iterationen für den Optimierer

### A.1.3 Hyperparameter RL-Agent

Während des Trainings des PPO-Agenten geben verschiedene Kennzahlen Aufschluss darüber, ob der Lernprozess stabil und effektiv verläuft. Der *Value Loss* beschreibt, wie gut das Neuronale Netzwerk den zukünftigen Belohnungswert vorhersagt – idealerweise nimmt dieser Fehler im Verlauf des Trainings ab und konvergiert. Der *Entropy Loss* misst, wie explorativ der RL-Agent agiert. Im Verlauf des Trainings nimmt dieser Wert ab, da sich der Agent zunehmend auf gewinnbringende Aktionen festlegt. Die *approximierte KL-Divergence* gibt an, wie stark sich die Strategie bei einem Update verändert – Werte im Bereich von etwa 0.01 bis 0.03 deuten auf ein kontrolliertes Lernen hin; starke Ausreißer deuten auf instabile Updates hin. Die *Clip Fraction* gibt an, bei wie vielen Strategie-Updates das PPO-Clipping aktiv war – typische Werte liegen hier zwischen 0.1 und 0.3. Werte nahe 0 oder 1 weisen darauf hin, dass die Lernrate oder das Clipping nicht ideal spezifiziert sind. Schließlich beschreibt die *Explained Variance*, wie gut das Value Netzwerk die Variation in

den tatsächlichen Belohnungen erklären kann – ein Wert größer als 0.5 oder ein stetiger Anstieg spricht dafür, dass das Value Netzwerk die Umgebungsdynamik zunehmend besser erfasst. Als primäre Erfolgskennzahl dient jedoch die *mittlere Belohnung pro Episode*, da diese durchschnittliche Leistungsfähigkeit der Strategie über abgeschlossene Episoden hinweg widerspiegelt.

### A.1.4 Hyperparameter Differential Evolution

Zum Benchmarking von *DrAGOS* wird der Differential-Evolution-Algorithmus [SP97] verwendet. Dieser Optimierungsalgorithmus ist globaler und stochastischer Natur. Verwendet wird die Python-Implementierung in `scipy` [Vir+20].

**Tabelle A.7:** *DrAGOS* Benchmark: Differential Evolution Hyperparameter [SP97; Vir+20]

	<b>Wert</b>	<b>Beschreibung</b>
<b>strategy</b>	<b>best2bin</b>	Evolutionsstrategie
<b>maxiter</b>	100,000	Maximale Anzahl an Iterationen
<b>popsize</b>	16	Populationsgröße
<b>tol</b>	1e-9	Toleranz für Abbruch basierend auf Funktionswerten
<b>mutation</b>	(0.5, 1)	Mutationsfaktor oder Bereich für Mutationsfaktor
<b>recombination</b>	0.7	Rekombinationsrate
<b>init</b>	<b>latinhypercube</b>	Initialisierung der Population
<b>updating</b>	immediate	Update-Modus der Population ( <b>immediate</b> oder <b>deferred</b> )
<b>constraints</b>	()	Randbedingungen (linear oder nicht-linear)
<b>x0</b>	<b>None</b>	Startpunkt

## A.2 *DAYSi* – Robustheit Verfahrensparameter

Die Robustheit von *DAYSi* wurde bereits in Abschnitt 6.3 ab S. 141 ausführlich diskutiert. Zwei weitere Verfahrensparameter – Zeitfenstergröße und Filterung – wurden aus Platzgründen jedoch nicht im Hauptteil der Arbeit besprochen. Anschließend an die kurze Zusammenfassung der Ergebnisse in Abschnitt 6.3.4.6 auf S. 164 sind hier die vollständigen Ergebnisse aufgeführt.

### A.2.1 Zeitfenstergröße

Die Zeitfenstergröße wird zwischen 1 ms, 2 ms, 5 ms und 10 ms anhand von Use Case I (vgl. Abbildung 6.16 ab S. 141) variiert. Dies wird auch in Abbildung A.2 anhand einer beispielhaften Brustbeschleunigung in X-Richtung verdeutlicht. Während die durch *DAYSi* global bestimmte Reihenfolge für alle vier Zeitfenstergrößen identisch ist, ergeben sich für den zeitlichen Verlauf Unterschiede. Das ist insbesondere an der Feinheit der Auflösung festzustellen. Grundlegende Tendenzen sind jedoch identisch.

Bei Betrachtung von RUN\_1010 ist beispielsweise ersichtlich, dass bei einer Zeitfenstergröße von 1 ms und 2 ms im Zeitbereich 60 ms bis 70 ms ein lokales Minimum der Auffälligkeit festgestellt werden kann. Bei 5 ms und 10 ms ist die Auflösung dafür zu gering. Bei kleinen Zeitfenstergrößen ist jedoch zu beachten, dass die Rechenzeit linear mit der Verkleinerung skaliert. Die Berechnung der Abweichungskoeffizienten für den Zeitbereich dauern für 1 ms folglich zehn Mal so lange wie für eine Zeitfenstergröße von 10 ms.

Darüber hinaus sind in Abbildung A.3 und Abbildung A.4 die Eventketten für RUN\_1043 dargelegt. Hierbei ist besonders interessant, dass *DAYSi* bei der Verwendung der kleinen Zeitfenster 1 ms und 2 ms die frühen Abweichungen in den Airbagdrücken nicht rekonstruieren kann, obwohl diese auch in der Heatmap (vgl. Abbildung 6.18 auf S. 143) deutlich erkennbar sind.

Dies kann auf Rauschen zurückgeführt werden, was einen Einfluss auf die Berechnung der Zeitlags aufweist. Ob eine Filterung hier Abhilfe schafft, kann in zukünftigen Betrachtungen näher untersucht werden. Zudem werden in Abbildung A.3 einige der Signale des ATD als vorgelagert aufgeführt. Bei der Verwendung größerer Zeitfenster werden immer noch ATD Signale wie die Kopfbeschleunigung in Y-Richtung für eine Zeitfenstergröße von 5 ms und die Brusteingedrücke in X-Richtung für eine Zeitfenstergröße von 10 ms als vorgelagerte Effekte identifiziert. Grundlegend werden hier jedoch vor allem Größen des Rückhaltesystems als vorgelagert identifiziert, was für dieses Beispiel mit den manuellen Betrachtungen übereinstimmt.

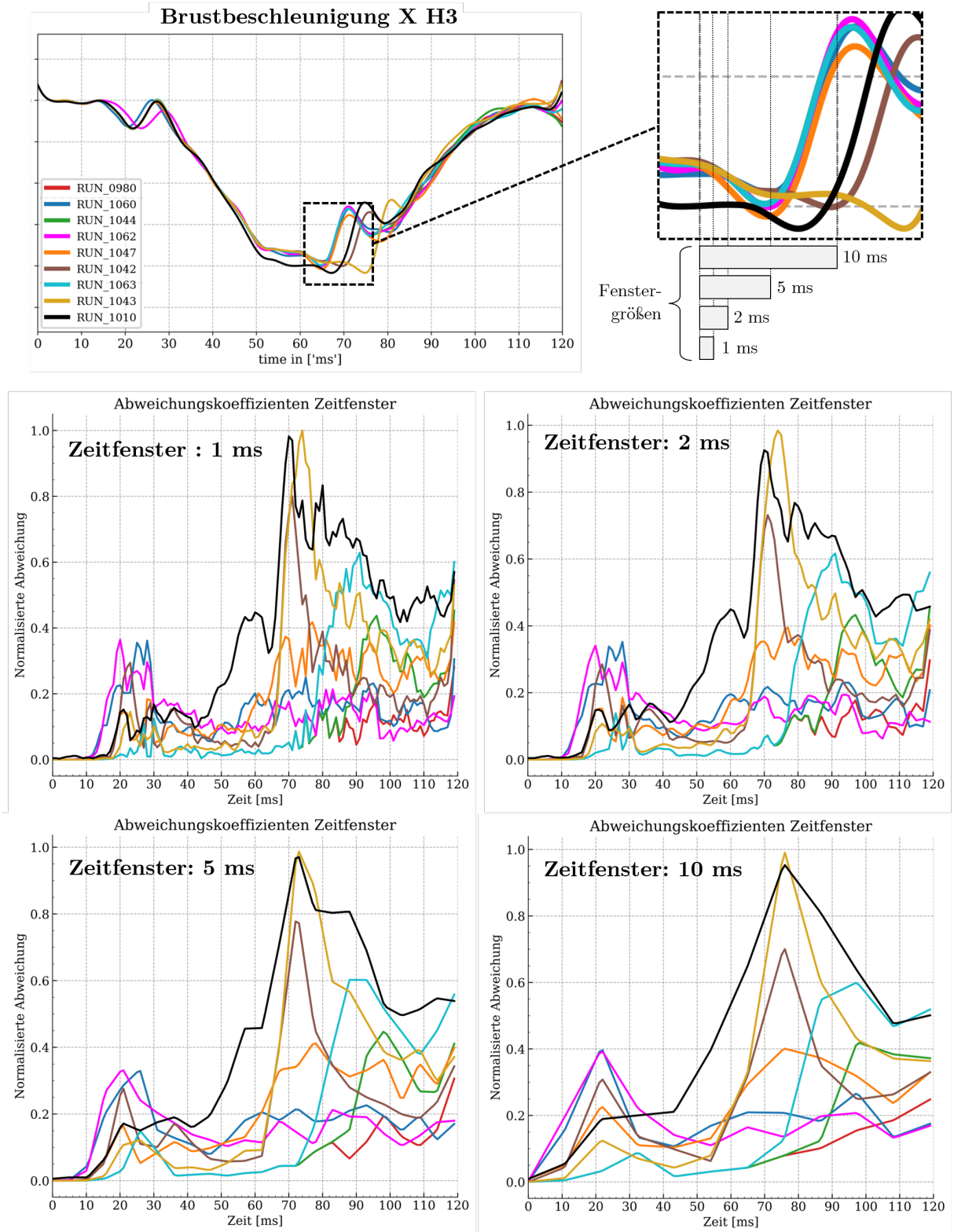


Abbildung A.2: *DAYSi* Robustheit: Use Case I mit anderer Zeitfenstergröße – Abweichung über die Zeit

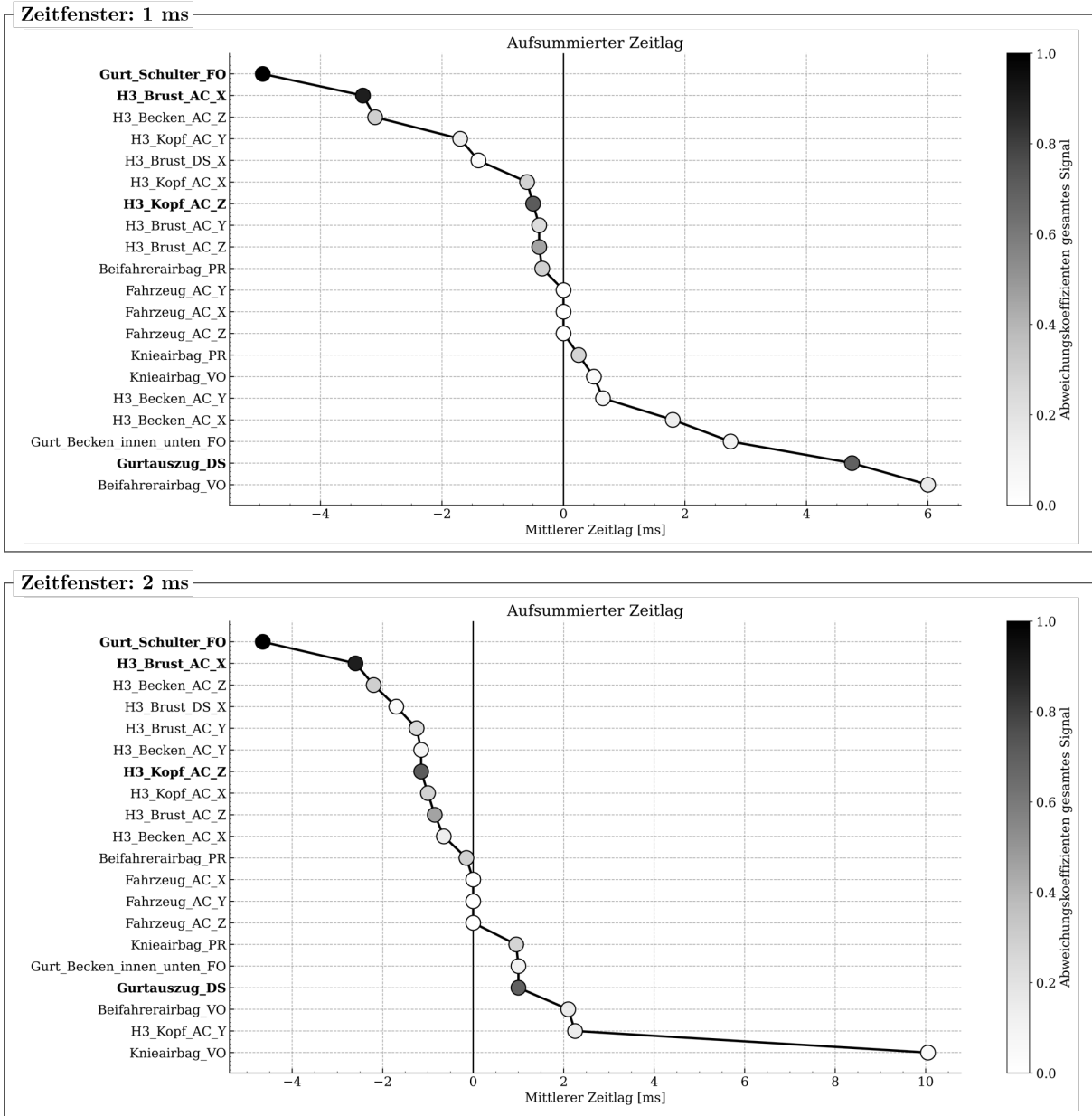


Abbildung A.3: *DAYSi* Robustheit: Use Case I mit anderer Zeitfenstergröße – Eventkette RUN\_1043 1 ms und 2 ms

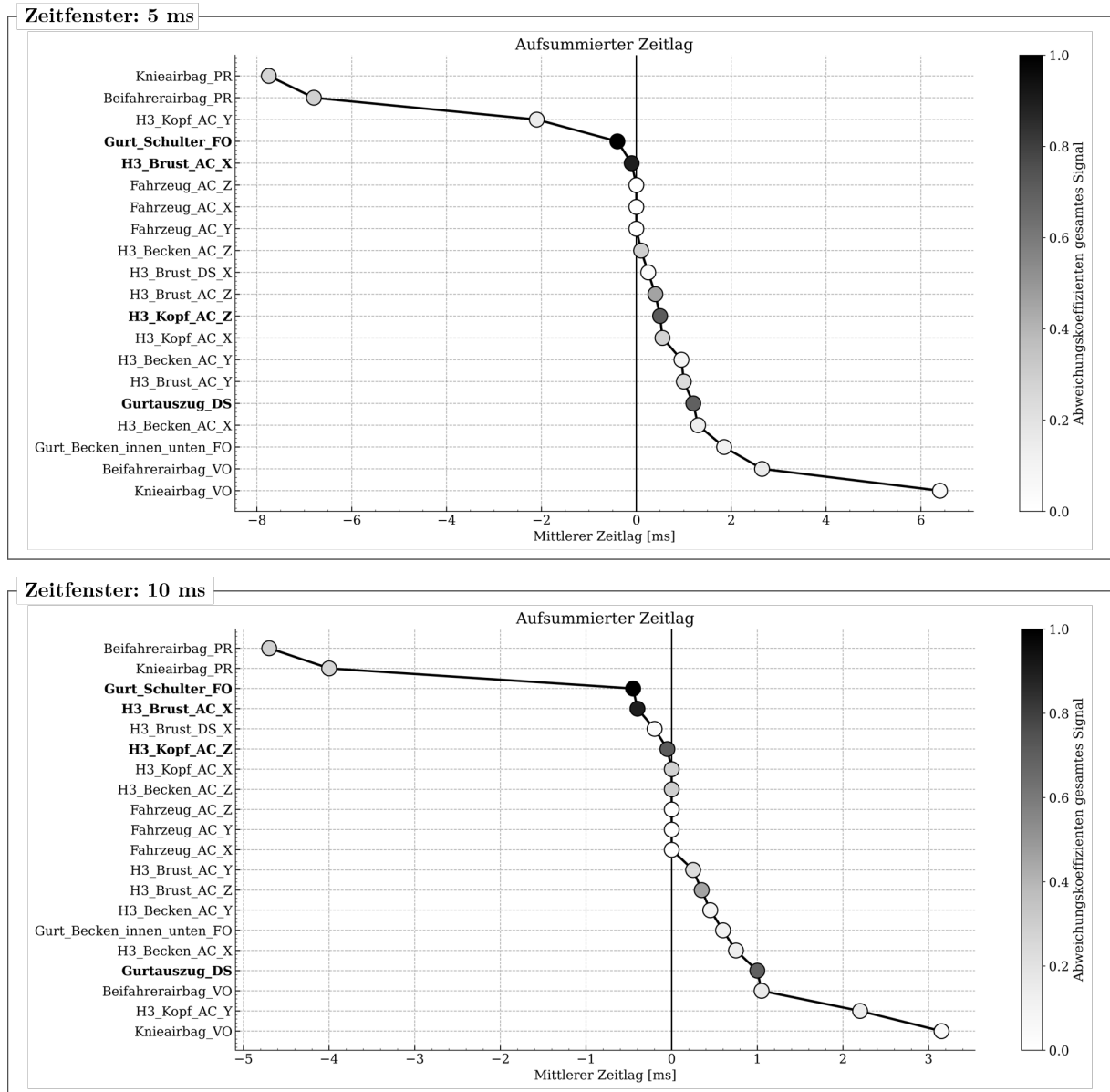


Abbildung A.4: *DAYSi* Robustheit: Use Case I mit anderer Zeitfenstergröße – Eventkette RUN\_1043 5 ms und 10 ms

Abschließend lässt sich also festhalten, dass insbesondere von zu kleinen Zeitfenstergrößen abzusehen ist. Für den weiteren Verlauf der Arbeit wird eine Zeitfenstergröße von 5 ms vorgeschlagen. Dies stellt einen guten Kompromiss aus Rechenzeit und Genauigkeit dar. Bei zu grober Auflösung ist davon auszugehen, dass Feinheiten nicht hinreichend genau abgebildet werden können.

## A.2.2 Filterung

Ein cfc-Filter ist ein genormter Tiefpassfilter zur Glättung von Sensorsignalen, wobei cfc60, cfc180, cfc600 und cfc1000 sich durch ihre unterschiedliche Grenzfrequenz und damit Filterstärke unterscheiden. Je höher die Zahl, desto höher die Grenzfrequenz und desto weniger stark wird das Signal geglättet. Weitere Informationen sind der Norm SAE 211-1 [SAE J211-1] zu entnehmen.

Die Filterung wird analog zur Zeitfenstergröße anhand des ersten Use Cases (vgl. Abbildung 6.16 ab S. 141) betrachtet. Die unterschiedliche Stärke der Filter ist auch eindeutig in dem Beispielsignal in Abbildung A.5 und Abbildung A.6 zu erkennen. Im Bereich zwischen 20 ms und 50 ms tritt bei cfc1000 Filterung immer noch sehr starkes Rauschen auf. Für cfc60 hingegen ist das Signal stark geglättet, wobei grundlegendes Verhalten erhalten bleibt. Im Gegensatz zur Zeitfenstergröße bleibt die globale Reihenfolge in Abbildung A.6 für schwächere nicht mehr ganz identisch. Eindeutig identifiziert werden die beiden auffälligsten Simulationen RUN\_1043 und RUN\_1010 sowie die unauffälligste Simulation RUN\_0980.

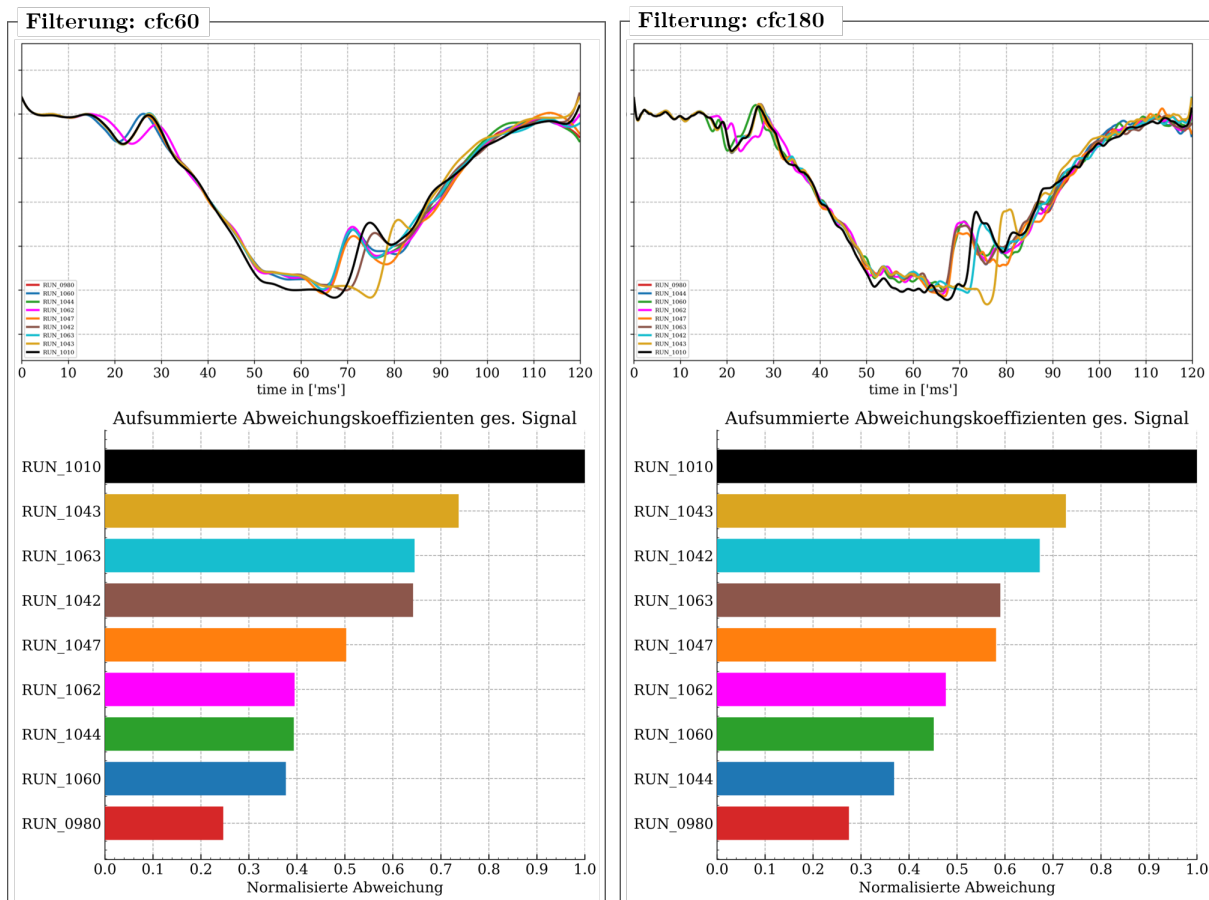
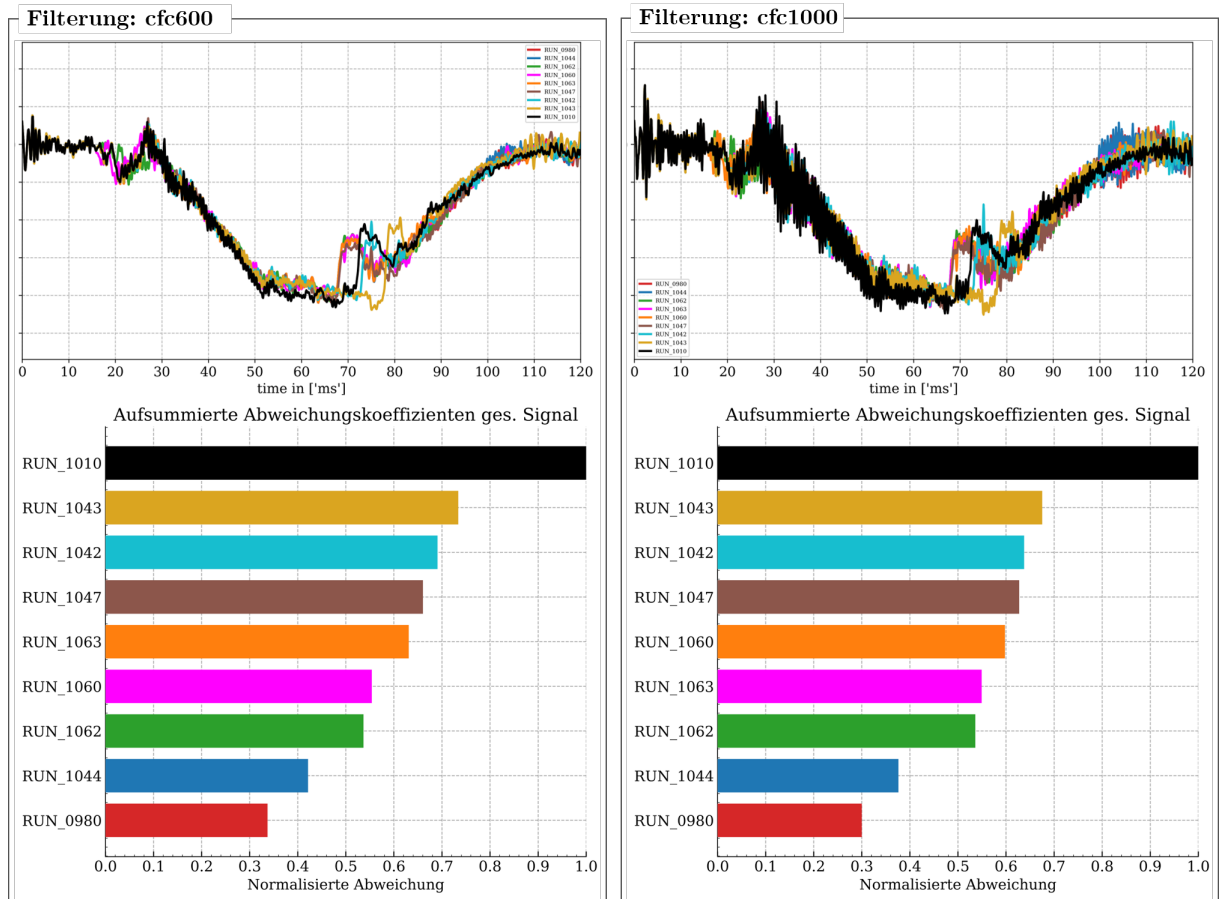


Abbildung A.5: *DAYSi* Robustheit: Use Case I mit Filterung cfc60 und cfc180



**Abbildung A.6:** *DAYSi* Robustheit: Use Case I mit Filterung cfc600 und cfc1000

Zur weiteren Betrachtung sind in Abbildung A.7 und Abbildung A.8 Detaildarstellungen für die Abweichungskoeffizienten für RUN\_1043 gezeigt. In den Abweichungskoeffizienten für das gesamte Signal sind kaum Unterschiede ersichtlich. Lediglich für die Brustbeschleunigung in Y- und Z-Richtung unterscheiden sich. Bezogen auf die Zeitdomäne werden diese beiden Signale ebenfalls deutlicher rot hervorgehoben. Dies lässt sich auf das besonders starke Rauschen des Signals zurück führen und verfälscht das eigentliche Verhalten den Signals deutlich. Dementsprechend ist eine eher starke Filterung hier von Vorteil, da jegliches Rauschen ausgefiltert wird. Die Glättung der Ergebnisse durch die Wahl größerer Zeitfenster hat ebenfalls zu einer Verbesserung der Ergebnisse geführt.

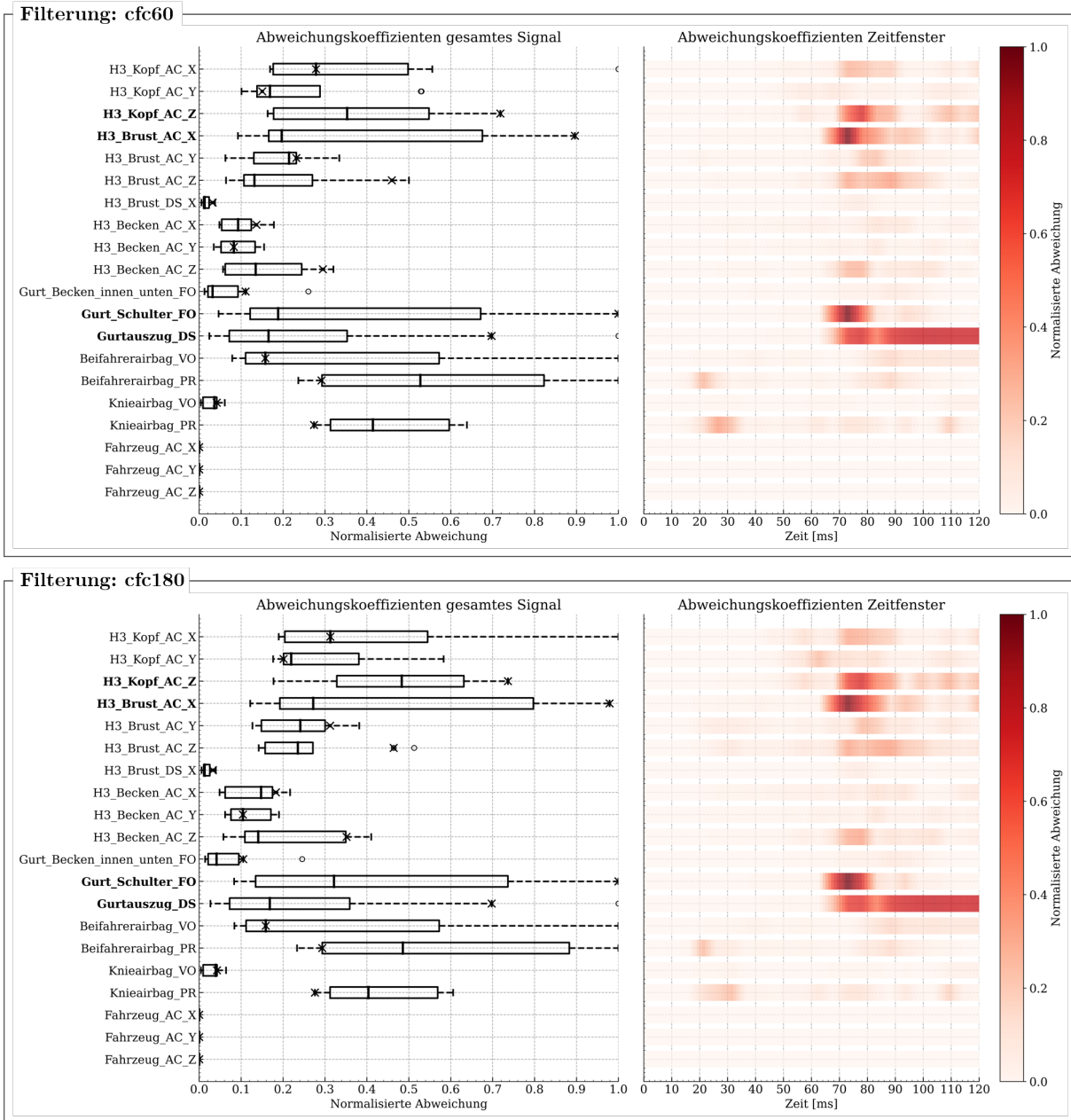
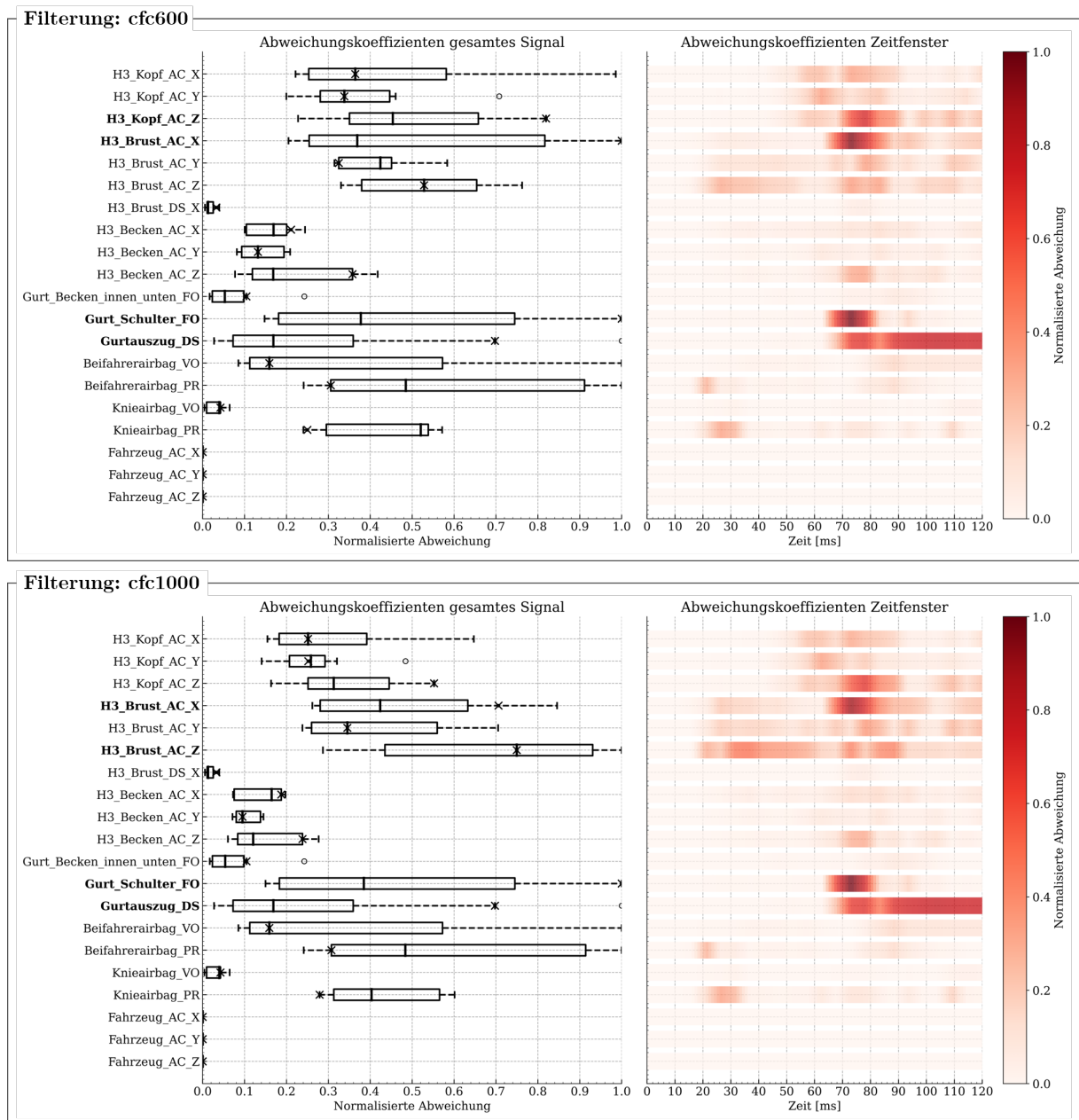


Abbildung A.7: *DAYSi* Robustheit: Use Case I mit anderer Filterung – Abweichungskoeffizienten RUN\_1043 cfc60 und cfc180



**Abbildung A.8:** *DAYSi* Robustheit: Use Case I mit anderer Filterung – Abweichungskoeffizienten RUN\_1043 cfc600 und cfc1000

Die zugehörigen Eventketten sind Abbildung A.9 und Abbildung A.10 zu entnehmen. Hier lassen sich deutlichere Unterschiede feststellen. Alle Eventketten identifizieren eindeutig die Airbagdrücke als vorgelagerte Effekte. Bei sehr schwacher Filterung mit cfc1000 ist ersichtlich, dass die Gurtkraft an der Schulter in die sekundären Effekte rutscht. Anders als bei cfc60 und cfc180 trennt cfc600 die Gurtkraft an der Schulter von der Brustbeschleunigung. Letztere wird bei cfc600 in den sekundären Effekten aufgeführt. Die Ergebnisse für cfc60, cfc180 und cfc600 sind darüber hinaus jedoch gut vergleichbar. Generell wird

an dieser Stelle vorgeschlagen mit *cfc60* eine eher stärkere Filterung vorzunehmen, um Rauschen zu vermeiden.

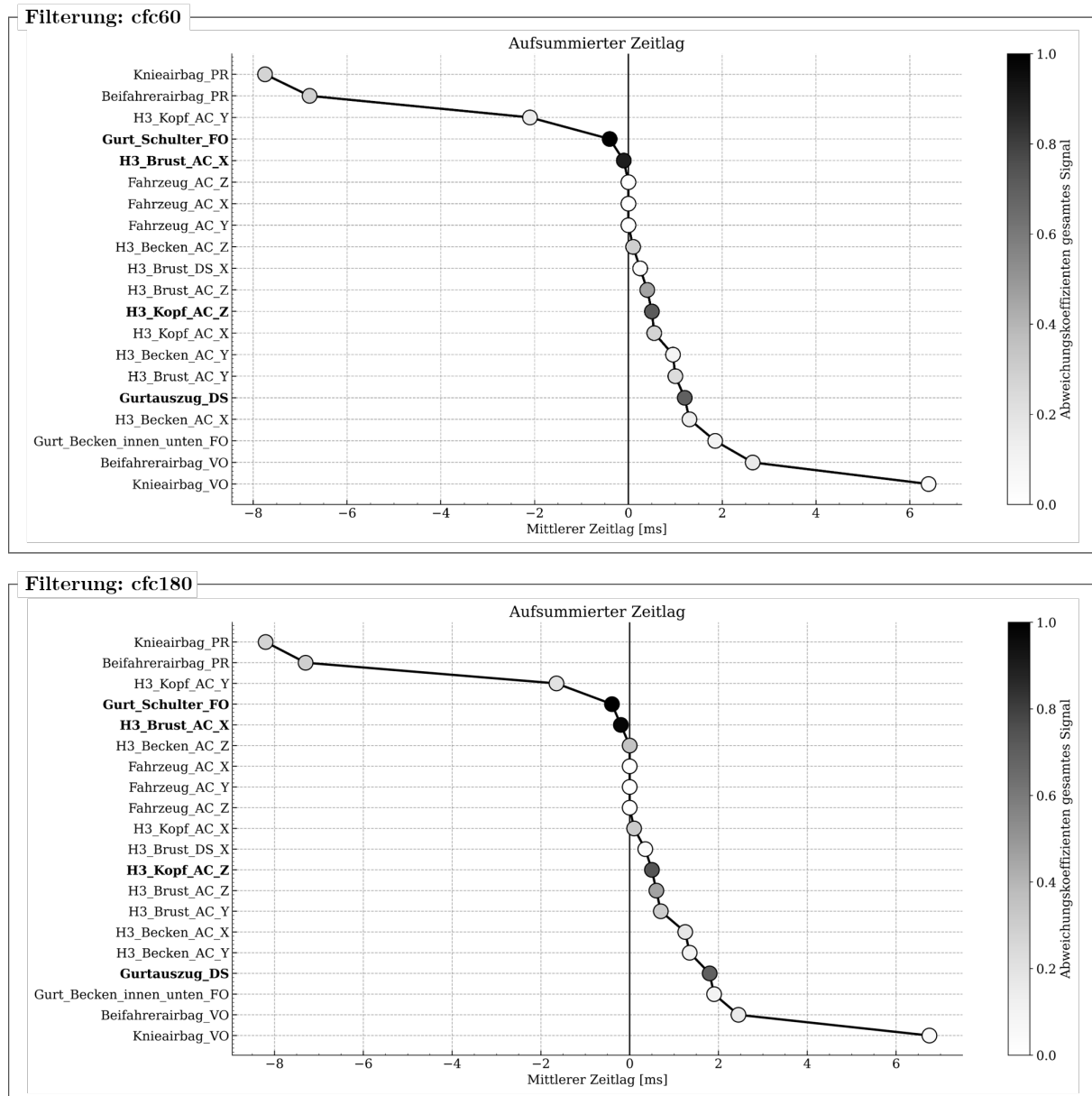


Abbildung A.9: *DAYSi* Robustheit: Use Case I mit anderer Filterung – Eventkette RUN\_1043 *cfc60* und *cfc180*

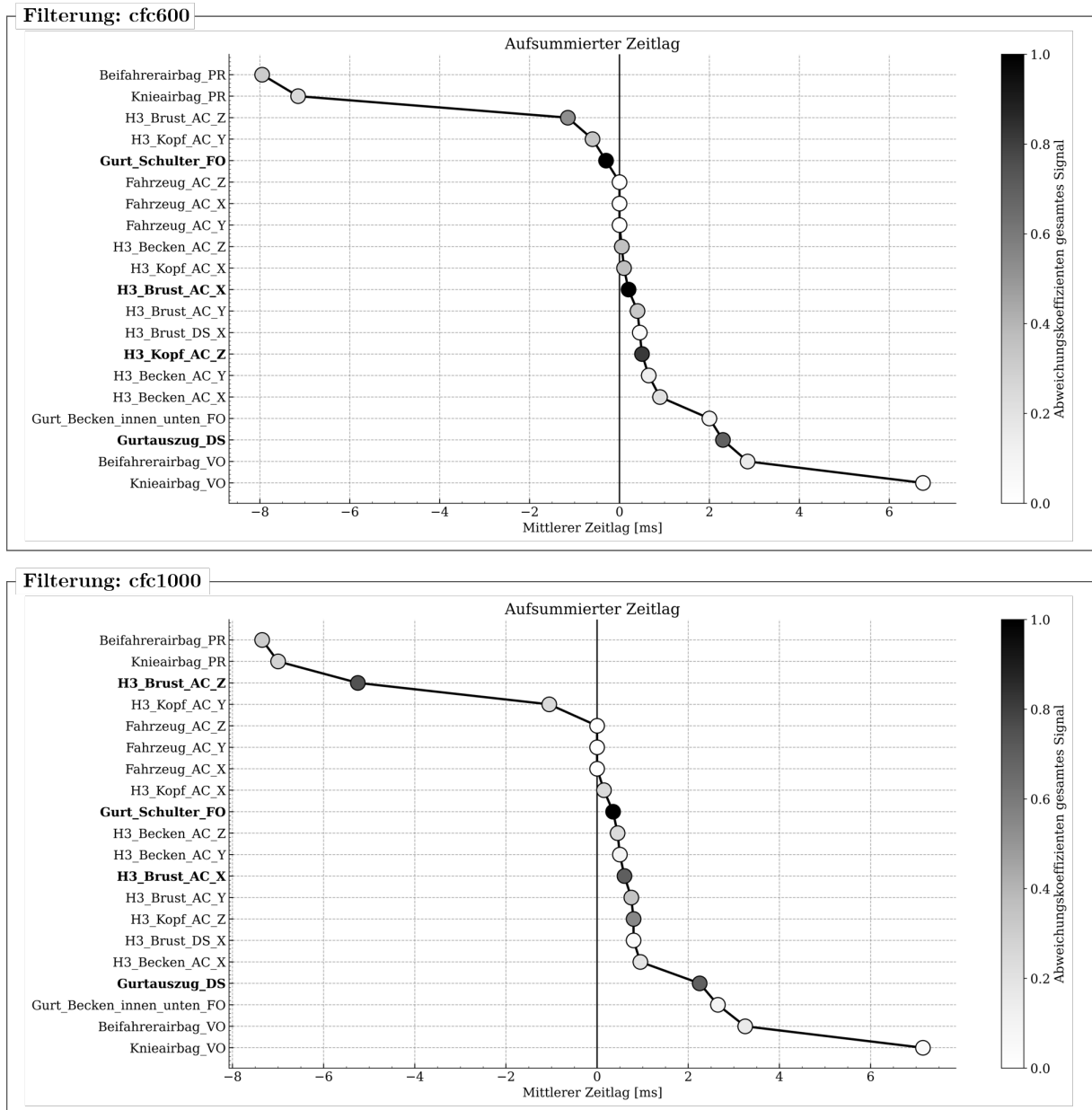


Abbildung A.10: *DAYSi* Robustheit: Use Case I mit anderer Filterung – Eventkette RUN\_1043 cfc600 und cfc1000

## A.3 *ExPLAIN* – Hyperparameter Metamodelle

Im Rahmen der Evaluierung von *ExPLAIN* in Abschnitt 6.4 ab S. 165 wurden verschiedene Metamodelle hinsichtlich ihrer Güte und der Reproduzierbarkeit in der Berechnung von SHAP-Werten untersucht. Die Güten für die Use Cass I-III sind Tabelle 6.2 auf S. 182 zu entnehmen. Die zugehörigen Hyperparameter sind in den folgenden Tabellen angegeben. Sofern nicht anders aufgeführt, wurde die Python-Implementierung der klassischen ML-Modelle mit `scikit-learn` [Ped+11] und die des Neuronalen Netzwerks mit `torch` [Pas+19] vorgenommen.

**Tabelle A.8:** *ExPLAIN* Metamodelle: Hyperparameter TabPFN [Hol+25]

Pretrained Weights	Device	Ensembles	Eingaben	Fine-Tuning
Stand 28.02.2025, TabPFN GitHub	GPU	1	19	Nein

Die Hyperparameter der ML-Modelle für SwM, a3ms und Schweller\_PCA-1\_Tend sind identisch.

**Tabelle A.9:** *ExPLAIN* Metamodelle: Hyperparameter XGBoost [CG16]

	SwM	a3ms	PC-1*	Cluster	Beschreibung
<code>n_estimators</code>	3,472	27	45	83	Anzahl der Entscheidungsbäume im Ensemble
<code>learning_rate</code>	0.12732	0.10926	0.22631	0.29076	Schrittweite beim Update der Baumgewichte
<code>max_depth</code>	3	5	5	15	Maximale Tiefe der einzelnen Entscheidungsbäume
<code>min_child_weight</code>	2	1	3	6	Minimaler Gewichtssummenwert in einem Blattknoten
<code>subsample</code>	0.51769	0.55705	0.72425	0.82459	Anteil der Trainingsdaten für das zufällige Sampling pro Baum
<code>gamma</code>	0	4	1	0.50	Minimaler Verlustabfall für einen Split
<code>alpha</code>	1	7	0	2	L1-Regularisierungsterm für Blattgewichte
<code>lambda</code>	3	5	2	3	L2-Regularisierungsterm für Blattgewichte

\*Schweller\_PCA-1\_Tend

**Tabelle A.10:** *ExPLAIN* Metamodelle: Hyperparameter Support Vector Regression

	SwM	a3ms	PC-1*	Beschreibung
<code>kernel</code>	<code>rbf</code>	<code>rbf</code>	<code>rbf</code>	Kernel-Funktion
<code>gamma</code>	0.02632	0.20	0.07143	Skalierung des Einflussradius des Kernels
<code>C</code>	3.34221	0.43859	1.11622	Regularisierungsparameter
<code>epsilon</code>	0.10007	0.29144	0.22682	Fehlertoleranzband

\*Schweller\_PCA-1\_Tend

**Tabelle A.11:** *ExPLAIN* Metamodelle: Hyperparameter Multilayer Perceptron

Architektur	Preprocessing	Training
<ul style="list-style-type: none"> <li>• 3 Hidden Layers, dim = 100</li> <li>• GELU-Aktivierung</li> <li>• LayerNorm (Layer 1 &amp; 2)</li> <li>• Dropout 0.1 (Layer 1 &amp; 2)</li> </ul>	<ul style="list-style-type: none"> <li>• Input: MinMaxScaler</li> <li>• Output: StandardScaler</li> <li>• Full-Batch Training</li> </ul>	<ul style="list-style-type: none"> <li>• Adam Optimizer, LR = 1e-3</li> <li>• Epochen = 100</li> <li>• MC-Samples = 10</li> <li>• KL-Gewicht = 1e-3</li> <li>• Heteroscedastic Loss</li> </ul>

Die Hyperparameter der ML-Modelle für SwM, a3ms und Schweller\_PCA-1\_Tend sind identisch.

**Tabelle A.12:** *ExPLAIN* Metamodelle: Decision Trees

	SwM	a3ms	PC-1*	Beschreibung
critierion	squared_error	squared_error	squared_error	Fehlerfunktion zur Bewertung der Splits
max_depth	6	14	12	Maximale Tiefe des Baums
min_samples_split	16	18	13	Anzahl Samples, ab der ein Knoten geteilt wird
min_samples_leaf	19	10	3	Minimale Anzahl an Samples, die ein Blatt haben muss
max_leaf_nodes	26	77	44	Maximale Anzahl an Blättern im Baum
ccp_alpha	0.0456	0.06355	0.01760	Parameter zur Steuerung der Baumkomplexität

\*Schweller\_PCA-1\_Tend

**Tabelle A.13:** *ExPLAIN* Metamodelle: Hyperparameter Lasso Regression

	SwM	a3ms	PC-1*	Beschreibung
alpha	0.00136	1.49858	0.01210	Regularisierungsparameter
max_iterations	1000	1000	1000	Iterationen für den Optimierer

\*Schweller\_PCA-1\_Tend



# Veröffentlichungen

- J. Mathieu, M. Di Roberto und M. Vielhaber. “A data-driven methodology for the automated analysis and explanation of system behavior in crash simulations”. In: *14th European LS-DYNA Conference*. Baden-Baden, Deutschland, 2023.
- K. König, J. Mathieu und M. Vielhaber. “A Sustainability-Driven Comparison of Methods for the Identification of Lightweight Design Potentials in Product Generation Engineering”. In: *Procedia CIRP: 31st CIRP Conference on Life Cycle Engineering 122* (2024), S. 145–150. ISSN: 2212-8271. DOI: 10.1016/j.procir.2024.01.021.
- K. König, J. Mathieu und M. Vielhaber. “Resource conservation by means of lightweight design and design for circularity—A concept for decision making in the early phase of product development”. In: *Resources, Conservation and Recycling* 201 (2024), S. 107331. ISSN: 0921-3449. DOI: 10.1016/j.resconrec.2023.107331.
- J. Mathieu, P. Gupta, M. Di Roberto und M. Vielhaber. “Minimizing occupant loads in vehicle crashes through reinforcement learning-based restraint system design: assessing performance and transferability”. In: *Proceedings of the Design Society: DESIGN Conference 4* (2024), S. 2139–2148. DOI: 10.1017/pds.2024.216.
- J. Mathieu und M. Vielhaber. “A Framework for the Selection of AI-Based Methods to Support the Development of Passive Vehicle Safety Systems”. In: *Proceedings of NordDesign 2024*. Hrsg. von J. Malmqvist et al. 2024, S. 313–322. ISBN: 978-1-912254-21-7.
- K. König, J. Mathieu und M. Vielhaber. “Identification of Lightweighting Potentials Towards More Sustainable Products via the Functional Lifecycle Energy Analysis (FLCEA)”. In: *Sustainable Manufacturing as a Driver for Growth*. Hrsg. von H. Kohl et al. Cham, Schweiz: Springer Nature, 2025, S. 137–145. ISBN: 978-3-031-77429-4.
- J. Mathieu, S. Kronwitter, J. Pfahler, M. Di Roberto, M. Vielhaber. “Enhancing numerical simulation analysis with the use of explainable artificial intelligence and large language models: a case study on vehicle side crash optimization”. In: *Proceedings of the Design Society: International Conference on Engineering Design (ICED25) 5* (2025), S. 1545–1554. DOI: 10.1017/pds.2025.10168.
- J. Mathieu, S. Kronwitter, F. Duddeck, J. Garcke und M. Vielhaber. “Explainable artificial intelligence for enhancing system understanding and interpretability of numerical crash simulations”. In: *Computers in Industry* 178 (2026), Art. 104466. ISSN: 0166-3615. DOI: 10.1016/j.compind.2026.104466.



# Betreute Abschlussarbeiten

- **P. Gupta**, *Use of Reinforcement Learning to Minimize Occupant Loads in Vehicle Crashes*, 08/2023, Masterarbeit, Lehrstuhl für Automatisierungstechnik, Technische Universität Ilmenau.
- **V. Jayadev**, *Use of Machine Learning to Predict Mechanical System Behavior*, 04/2025, Masterarbeit, Lehrstuhl für Automatisierungstechnik, Technische Universität Ilmenau.
- **S. Togrü**, *MetaCrash—A Predictive Modeling Framework for Sequential Vehicle Crash Data*, 10/2025, Masterarbeit, Lehrstuhl für Datenbanksysteme und Datenmining, Ludwig-Maximilians-Universität München.
- **A. Nagaraju**, *Integration of 3D Finite-Element Data into Machine-Learning-based Prediction Models*, 07/2025, Masterarbeit, Lehrstuhl für Materialfluss- und Handhabungstechnik, Hochschule Stralsund.
- **L. Alencar**, *Prognose von Kraft-Weg-Kurven in Felgendruckversuchen mit maschinellem Lernen*, 04/2025, Masterarbeit, Institut für Konstruktionstechnik und Technisches Design, Universität Stuttgart.
- **L. Pfeifer**, *Automatisierte Plausibilisierung von Crashtestdaten mittels Machine Learning*, 09/2025, Bachelorarbeit, DHBW-Stuttgart.



# Kurzlebenslauf Autor

<b>Name</b>	Janis Noah Mathieu
<b>Geboren</b>	15.01.1998 in Homburg (Saar)
<b>2022–2025</b>	<i>Doktorand Karosseriesystem CAE &amp; Fahrzeugsicherheit</i> Porsche Engineering Services GmbH, D-74321 Bietigheim-Bissingen
<b>2022–2025</b>	<i>Wissenschaftlicher Mitarbeiter Lehrstuhl für Produktentstehung</i> Universität des Saarlandes, D-66123 Saarbrücken
<b>2020–2022</b>	<i>Studentische Tätigkeiten Karosseriesystem CAE &amp; Fahrzeugsicherheit</i> Porsche Engineering Services GmbH, D-74321 Bietigheim-Bissingen
<b>2020–2022</b>	<i>M.Sc. Systems Engineering mit Vertiefung Systems Design</i> Universität des Saarlandes, D-66123 Saarbrücken
<b>2019–2020</b>	<i>HiWi Tätigkeiten</i> Universität des Saarlandes, D-66123 Saarbrücken
<b>2016–2020</b>	<i>B.Sc. Systems Engineering mit Vertiefung Maschinenbau</i> Universität des Saarlandes, D-66123 Saarbrücken
<b>2008–2016</b>	Allgemeine Hochschulreife (Abitur) Leibniz Gymnasium, D-66386 St. Ingbert

