AUTOMATED JOINING ELEMENT DESIGN FOR HIGH PRODUCT VARIETY IN THE MANUFACTURING INDUSTRY

DERK HENDRIK DOMINICK EGGINK

AUTOMATED JOINING ELEMENT DESIGN FOR HIGH PRODUCT VARIETY IN THE MANUFACTURING INDUSTRY

DERK HENDRIK DOMINICK EGGINK

AUTOMATED JOINING ELEMENT DESIGN FOR HIGH PRODUCT VARIETY IN THE MANUFACTURING INDUSTRY

Derk Hendrik Dominick Eggink

AUTOMATED JOINING ELEMENT DESIGN FOR HIGH PRODUCT VARIETY IN THE MANUFACTURING INDUSTRY

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. ir. A. Veldkamp,
on account of the decision of the Doctorate Board
to be publicly defended
on Thursday 7 September 2023 at 14.45 hours

by

Derk Hendrik Dominick Eggink
born on the 12th of March, 1992
in Lochem, the Netherlands

This dissertation has been approved by:

Promotors:
Prof. dr. ir. M.W. Groll
Prof. dr. ir. I. Gibson

Cover design: Derk Hendrik Dominick Eggink
Printed by: Ipskamp Printing
ISBN (print): 978-90-365-5767-2
ISBN (digital): 978-90-365-5768-9
DOI: 10.3990/1.9789036557689

Graduation Committee:

Chair / secretary:

Prof. dr. ir. H.F.J.M. Koopman

Promotors:

Prof. dr. ir. M.W. Groll
Universiteit Twente, Advanced Manufacturing,
Sustainable Products & Energy Systems

Prof. dr. ir. I. Gibson
Universiteit Twente, Advanced Manufacturing,
Sustainable Products & Energy Systems

Members:

Prof. dr. ir. G.M. Bonnema
Universiteit Twente, Systems Engineering &
Multidisciplinary Design

Dr. J. Griffin
Coventry University, Centre for Manufacturing and
Materials

Prof. R. Harrison
University of Warwick, Warwick Manufacturing Group

Prof. dr. ir. J. Henseler
Universiteit Twente, Product–Market Relations

Prof. dr. ing. M. Vielhaber
Universitat des Saarlandes, Systems Engineering

Abstract

Product variety and the manufacturing complexity that it induces are continuously increasing. This poses a challenge in the product development process and, consequently, the design of joints. Joining elements define the manner in which a permanent joint is created between parts. Joining element design is an ambiguous manual task with limited automation solutions. Thus, it can lead to long, iterative, error-prone development trajectories that may result in costly rework. Hence, automation solutions for joining element design must be intelligent. However, simply ensuring the intelligent automation of joining element design is insufficient. Modular design, through the approaches of modularization and commonalization, enables manufacturers to cope with the complexity induced by product variety. Unfortunately, modular design approaches have not yet considered joining elements. Hence, this dissertation study sought to answer the following research question: "*How can joining element design be automated for high-variety products?*"

This dissertation presents a framework for automating joining element design. The framework is structured into smaller design problems, which will guide designers through the process of designing joining elements. This structuring will also enable designers to evaluate and assess artificial intelligence (AI) for each design problem. Moreover, the design problems will enable designers to identify unused AI techniques, such as machine learning.

These techniques were conceptualized and several were implemented for validation in this study. A market-validated database with automotive Body-in-White structures was used. The validation included the use of decision trees to predict joining technologies and the number of joining elements. Both prediction tasks seem promising to implement in early design phases due to their simplicity.

Next, this study validated two approaches for predicting joining locations. The first was a straight-forward evolutionary algorithm for distributing spot welds over contact regions. However, this algorithm's performance fell off quickly for nontrivial design problems with high solution spaces.

The second approach was to use convolutional neural networks to predict spot weld locations, which provided robust and promising results. This study first explored a voxel-based regression and classification task by drawing locations in a grid-like structure. Classification with a segmentation approach produced more robust results due to the spatial dependencies of classes. Supervised machine learning enabled the consideration of knowledge of successful designs in new design problems. This concept was subsequently enriched with nongeometric data using a branding approach. However, these multimodal machine learning models did not improve the joining locations. Furthermore, the models could not extract the

additional information from the data samples.

The difficult task of predicting joining locations requires designers to remain in the loop. The computational cost and a lack of quality data prevent the models from fully carrying responsibility for this task. However, these models seem capable of being used in early design stages to speed up the design process.

In addition to the prediction of joining elements, the framework developed in this study conceptualizes their modular design. It defines joining elements as individual components, which enables the generation of interchangeable joining modules, similar to regular modules in products. This study validated two modular design concepts: (1) the reduction of joining locations through their clustering over multiple product variants, and (2) the addition of joining elements to other joints for commonalization. The results of both concepts indicated that they highly depend on the module interfaces. The geometric boundary conditions of these interfaces had major effects on the results. Consequently, these modular design approaches must be performed on a few strategic moments.

In short, automated joining element design for high-variety products requires the integration of multiple solutions. Companies and product development processes are dynamic and uncertain, which requires the appropriate methodology to be carefully selected. The framework proposed in this dissertation organizes and can guide designers through this selection. Furthermore, the framework identified new AI applications for various tasks in designing joining elements. Based on this, the study proposes and validates several novel methodologies. The validation experiments produced promising findings, which can be used in the automation of joining element design in manufacturing industries with high product variety.

Titel en samenvatting

GEAUTOMATISEERD ONTWERP VAN VERBINDINGSELEMENTEN BIJ GROTE PRODUCTVARIATIE IN DE MAAKINDUSTRIE

De variatie in producten en de daarmee gepaard gaande complexiteit in productie nemen voortdurend toe. Dit vormt een uitdaging in het proces van productontwikkeling en eveneens voor het ontwerpen van verbindingselementen. Verbindingselementen beschrijven de manier waarop blijvende verbindingen worden gemaakt tussen onderdelen. Het ontwerpen hiervan is een complex, handmatig proces met beperkte geautomatiseerde oplossingen. Dit leidt vaak tot langdurige, iteratieve en foutgevoelige ontwikkelingsfasen, wat kan resulteren in kostbaar herstelwerk. Hierdoor moeten geautomatiseerde oplossingen voor het ontwerp van verbindingselementen intelligent zijn. Niettemin is het eenvoudigweg zorgen voor intelligente automatisering van het ontwerp van verbindingselementen onvoldoende. Technieken als modularisatie en standaardisatie bieden fabrikanten de mogelijkheid om de complexiteit die wordt veroorzaakt door productvariatie aan te pakken. Helaas hebben modulaire ontwerpmethoden tot nu toe geen rekening gehouden met verbindingselementen. Hieruit volgt de onderzoeksvraag van dit proefschrift: "Hoe kan het ontwerp van verbindingselementen in de maakindustrie worden geautomatiseerd voor producten met een grote variëteit?"

Dit proefschrift presenteert een raamwerk voor het automatiseren van het ontwerp van verbindingselementen. Het raamwerk is gestructureerd rond kleinere ontwerpproblemen, die ontwerpers begeleiden gedurende het ontwerpproces. Deze structurering biedt tevens de mogelijkheid om de huidige methoden te evalueren in relatie tot de ontwerpproblemen. Bovendien helpen deze ontwerpproblemen bij het identificeren van ongebruikte maar toepasbare AI-technieken, zoals machine learning.

In dit onderzoek zijn deze technieken geconceptualiseerd en zijn er verschillende geïmplementeerd. Tijdens de validatie is gebruik gemaakt van een marktgevalideerde database met carrosseriestructuren van voertuigen. De validatie begint met het valideren van beslissingsbomen om verbindingstechnologieën en het aantal verbindingselementen te voorspellen. Beide benaderingen zijn vanwege hun eenvoud veelbelovend voor implementatie in de vroege ontwerpfases.

Vervolgens valideert dit onderzoek twee methoden voor het voorspellen van de locaties van verbindingselementen. De eerste methode is een eenvoudig evolutionair algoritme om puntlaslocaties over contactgebieden te verdelen. Echter, de prestaties van dit algoritme nemen snel af bij niet-triviale ontwerpproblemen met grote aantallen mogelijke oplossingen. De tweede methode voorspelt puntlaslocaties door gebruik te maken van convolutionele

neurale netwerken. Dit levert robuuste en veelbelovende resultaten op. Hierbij is een op voxel-gebaseerde regressie- en classificatietaak onderzocht, waarbij locaties worden ingetekend op een rasterachtige structuur. Het classificatiemodel presteert beter vanwege het expliciet meenemen van ruimtelijke afhankelijkheden in de data. Supervised machine learning integreert kennis van succesvolle ontwerpen in nieuwe voorspellingen. Dit concept is vervolgens verrijkt met niet-geometrische gegevens. Ondanks de toegevoegde informatie, bleken de modellen niet in staat om deze uit de data te extraheren, waarmee de prestaties niet verbeterden ten opzichte van de modellen zonder deze informatie.

Het voorspellen van de locaties van verbindingselementen blijkt zeer complex te zijn en vereist betrokkenheid van ontwerpers. Vanwege de rekenkracht en het gebrek aan data kunnen de modellen niet volledig verantwoordelijk zijn voor deze taak. Toch lijken deze modellen geschikt om te worden gebruikt in de vroege ontwerpfases om het ontwerpproces te versnellen.

Naast het voorspellen van verbindingselementen, conceptualiseert het raamwerk ook hun modulaire ontwerp. Het onderzoek definieert verbindingselementen als individuele componenten, wat het genereren van uitwisselbare verbindingsmodules mogelijk maakt, vergelijkbaar met reguliere modules in producten. Dit onderzoek heeft twee concepten voor modulair ontwerp gevalideerd: (1) het verminderen van het aantal verbindingslocaties door deze te clusteren over meerdere productvarianten, en (2) het toevoegen van verbindingselementen aan andere verbindingen voor een grotere gemeenschappelijkheid. De resultaten van beide concepten tonen aan dat ze sterk afhankelijk zijn van de interfaces tussen modules. Met name de geometrische randvoorwaarden van deze interfaces hadden een grote invloed op de resultaten. Hierdoor moeten dergelijke modulaire ontwerpmethoden alleen op strategische momenten worden toegepast.

Kortom, geautomatiseerd ontwerp van verbindingselementen voor producten met grote variatie vereist de integratie van meerdere oplossingen. Bedrijven en productontwikkelingsprocessen zijn dynamisch en onzeker, wat vraagt om zorgvuldige selectie van geschikte methoden. Het voorgestelde raamwerk organiseert methoden en kan ontwerpers begeleiden bij deze selectie. Bovendien presenteert het raamwerk nieuwe toepassingen van kunstmatige intelligentie en heeft deze gevalideerd. De experimenten hebben veelbelovende resultaten opgeleverd die kunnen worden ingezet voor verdere automatisering in het ontwerpen van verbindingselementen in de maakindustrie met grote productvariatie.

Acknowledgments

This dissertation explores the automation of joining element design in manufacturing industries with high product variety. Between autumn 2017 and summer 2021, I was a Ph.D. student at Mercedes-Benz AG and had the academic support of the University of Twente. The project started with a small idea, namely the belief of Patrick Papentin in implementing artificial intelligence in the design of automobiles. This idea transformed into not only multiple successful projects but also the present dissertation, and with that into new ways of thinking and working.

This work is the result of almost four years of investigation, exploration, comprehension, attempts, failures, and writing, and it is remarkably far from the expectations of the naive student who began this journey. This dissertation's output – this artifact that represents my professional career to date – is the result of discussions with and the ideas, support, trust, and love of many people.

As such, I first owe special thanks to Prof. dr. ir. Marco Groll for being my supervisor for the last few years. Your trust in me at the start and during this study gave me the motivation to persevere. Moreover, your valuable guidance and continuous support created an excellent opportunity for me to develop. My thanks also goes to Prof. dr. ir. Ian Gibson, and the other the members of the examination board, Prof. dr. ir. G.M. Bonnema, Dr. J. Griffin, Prof. dr. ir. J. Henseler, and Prof. dr. ing. M. Vielhaber.

Furthermore, I want to thank Patrick Papentin, Christoph Knödler, Johannes Biedert, Lina Longhitano, Felix Pauli, and Gunar Schier for their supervision and contributions, as well as for providing the conditions for me to prosper and complete this dissertation. Our discussions were always fruitful, including those that were off-topic over a cup of coffee. I thank you, my colleagues and friends, for making my time pleasant from the first day I arrived in Germany.

I also want to thank Daniel Felipe Perez-Ramirez, Hong-Duc Nguyen, Vivek Dhameliya, and Tobias Gerlach for conducting their master theses within the context of this dissertation. Their effort, hands-on experience, and curiosity validated the numerous ideas and concepts that we hypothesized. I greatly enjoyed my time supervising you.

Moreover, I want to thank Ben McKechnie for proofreading and editing the manuscript. Your work greatly improved the readability of the study.

Furthermore, I owe thanks to my family and friends for always making me feel welcome and heard, happy in the dark times, and amazing in the good times. I hope to enjoy your company for much time to come.

Last but not least, my gratefulness and special thanks go to my parents Henk and Erna and my siblings Rens and Lynn for their endless support throughout my life. Your love, trust, and energy have always been on my side. For this, I want so say thank you again – to all of you.

Zwolle, March 2022

Niek

Contents

Publications

This dissertation is grounded upon a collection of papers. The papers describe distilled results of research and will be references throughout the thesis. They highlight the areas of interest and the contributions to the scientific world.

1. **Joining element design and product variety in manufacturing industries.**
   D. Eggink et al. "Joining element design and product variety in manufacturing industries". In: *13th CIRP Conference on Intelligent Computation in Manufacturing Engineering*. Vol. 88. 2019, pp. 76–81. DOI: 10.1016/j.procir.2020.05.014

   Chapter: 2; **Status**: Published; **Purpose**: To report on the state of the art and to identify the research gap.

   This paper addresses the state of the art in joining element and modular design in industry and practice. It identifies the need for actively integrating product variety considerations into joining element design. The study concludes with a proposal to implement artificial intelligence methods as a solution pathway for identified issues.

2. **Towards automated joining element design.**
   D. H. D. Eggink et al. "Towards automated joining element design". In: *Procedia Computer Science CIRP*. Vol. 159. 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, 2019, pp. 87–96. DOI: 10.1016/j.procs.2019.09.163

   Chapter: 3; **Status**: Published; **Purpose**: To propose an AI toolbox with various concepts.

   This paper presents a novel artificial intelligence method toolbox for automated joining element design in the manufacturing industry considering product variety requirements. The method toolbox aims to solve issues and requirements from practice based on a target joining element design process. To the best of our knowledge, this is the first study to address holistic joining element design through the use of AI methods.

3. **Automated joining element design by predicting spot-weld locations using 3D convolutional neural networks.**
   D. Eggink et al. "Automated joining element design by predicting spot-weld locations using 3D convolutional neural networks". In: *2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. Piscataway, NJ: IEEE, 2020, pp. 1–9. ISBN: 978-1-7281-7037-4. DOI: 10.1109/ICE/ITMC49519.2020.9198601

   Chapters: 3 and 4; **Status**: Published; **Purpose**: To develop a methodology for predicting joining locations using 3D voxels and supervised machine learning.

   This work studies two previously presented concepts to predict spot weld locations. It describes the methodology, concepts, and results. Furthermore, it evaluates models using coordinate-based performance measurements as well as discusses challenges, limitations, and potentials.

4. **Modularization of joining elements in high variety manufacturing industries.**
   D. Eggink et al. "Modularization of joining elements in high variety manufacturing industries". In: *Procedia CIRP* 100 (2021), pp. 67–72. ISSN: 22128271. DOI: 10.1016/j.procir.2021.05.011

   Chapter: 3; **Status**: Published; **Purpose**: To develop a methodology for systematically commonalizing and modularizing joining elements.

   This work proposes a methodology for modularizing joining elements and introduces a trade-off to reduce the total cost by increasing the number of joining elements for increased modularization capacity.

5. **Generative Adversary Networks for spot weld design.**
   T. Gerlach et al. "Generative Adversarial Networks for spot weld design". In: *26th Annual Conference of the IEEE Industrial Electronics Society (IES)*. Västerås, Sweden: IEEE, 2021, pp. 1–8. DOI: 10.1109/ETFA45728.2021.9613282

   Chapter: 3; **Status**: Published; **Purpose**: To develop a methodology for predicting joining locations using 2D images and a Generative Adversarial Network.

   This work presents a methodology for predicting spot weld locations with 2D-based reference- and latent-guided approaches using StarGAN_v2.

Abbreviations

AI	Artificial Intelligence.
BIW	Body-In-White.
CAD	Computer-Aided-Design.
CBR	Case-Based Reasoning.
CC	Connection Case.
CNN	Convolutional Neural Network.
CR	Contact Region.
CV	Cross-Validation Set.
DR	Dimensionality reduction.
DSM	Design Structure Matrix.
ED	Element Densification.
EM	Expectation-Maximization.
EncDec	Encoder Decoder.
GAN	Generative Adversarial Network.
IoCR	Intersection Of Contact Regions.
IoU	Intersection Over Union.
JC	Joining Cluster.
JE	Joining Element.
JL	Joining Location.
JM	Joining Module.
JP	Joining Parameter.
JS	Joining Scenario.
JT	Joining Technology.
ML	Machine Learning.
MMML	Multimodal Machine Learning.
MPD	Modular product development.
NN	Neural Network.
OEM	Original Equipment Manufacturer.
PCA	Principal Component Analysis.
PDM	Product Data Management.
PMI	Product Manufacturing Information.
PP	Product Platform.
PROB	Probability Mapping Concept.
PV	Product Variety.
RI	Random Initialization
RBR	Rule-Based Reasoning.
RL	Reinforcement Learning.
RSW	Resistance Spot Weld.
S&O	Search & Optimization.
SA	Spatial Aggregation.
SEG	Segmentation Concept.
SML	Supervised Machine Learning.
TU	Technology Unification.
UML	Unsupervised Machine Learning.
UoCR	Union Of Contact Regions.

Nomenclature

Symbols

a	Weights.
A	Accurateness.
C	Complexity.
$\%CI$	Percent commonality index.
D, d	(Total) Depth.
E	Engineer preference.
G	Boolean matrix.
H, h	Height.
I	Implementation factor.
i	Index.
J	Jaccard index.
l	Length.
$\mathcal{L}$	Loss function.
m	Dataset size.
ΔN	Difference between the number of predicted and target joining locations.
n	Number (of).
p	Point, coordinate.
$\tilde{p}$	Valid coordinate pair.
P	Probability.
Q	Correctness.
r	Grid cell resolution.
R	Sales volume.
S	Similarity.
τ	Threshold.
$\tilde{T}$	Spatially-dependent matrix.
u	Technology.
V	Set with all valid point pairs.
W, w	(Total) Width.
$x^{(i)}$	Input data sample.
$y^{(i)}$	Target data sample.
$\hat{y}^{(i)}$	Predicted data sample.
Y	Target coordinates.
$\hat{Y}$	Predicted coordinates.

Subscripts

c	Class.
cc	Connection case.
cr	Contact region.
e	Edge.
i, j, k	Indexes.
je	Joining element.
nn	Nearest neighbor.
p	Joining location.
s	Joining scenario.
t	Technology.
v	Valid.
y	Target.
$\hat{y}$	Prediction.

Superscripts

c	Joining cluster centroid.
(i)	Data sample index.
I	Interface.
l	Location.
P	Component.
s	Stride.
T	Topology.
u	Unification.

List of Figures

List of Tables

Chapter 1

Introduction

Joining is a vital process in the manufacturing of products that accounts for up to 50% of the production time. Joining elements are the created, virtual objects that define the manner in which parts are joined. In other words, they describe the implementation of the joining process. In early product development, a focus on joining elements is essential for creating manufacturable, cost-effective products. However, joining element design is a multidisciplinary process that is complex, ambiguous, and sometimes repetitive. Multiple drivers in product development, such as greater customer demands and product variety, increase the difficulty of the process. Moreover, joining element design relies largely on manual trial-and-error approaches that prevent creating optimal designs. This situation causes unnecessary design iterations, reduced product quality, and increased lead time. Therefore, this dissertation study aimed to automate the production of joining elements for industries with high product variety.

This chapter introduces the study by first discussing the topic and context of joining element design, product variety, and artificial intelligence (AI); see Section 1.1. Next, Section 1.2 presents the content and objectives, including the research problem, the research aims and objectives, the research questions, and the significance of the research. Lastly, Section 1.3 concludes with addressing the structure of the dissertation.

1.1 Topic and context

This section starts with a generic explanation of product development and related processes. Then, the subsections 1.1.1, 1.1.2, and 1.1.3 introduce joining elements, product variety, and artificial intelligence (AI), respectively. Lastly, subsection 1.1.4 summarizes the topic and context of this study.

The demand for industries to better align their products with customer needs is continuously increasing [6]. Companies consistently need to deliver high-quality products on time to remain competitive in the market [6]. Multiple drivers affect product development when creating successful products, including the following:

- Shortening product life cycles combined with more frequently changing demands [6, 7];

- Globalization, which enables worldwide competitors and suppliers with different operating conditions to enter and change markets rapidly [6];
- Outsourcing, which involves specialized business partners taking over interests and functions to increase company effectiveness [8];
- Market saturation, which forces companies to explore new niches with state-of-the-art innovations and technology [6, 9, 10];
- Technological progress combined with innovations and performance increases, which enable state-of-the-art functionality to be retained in products;
- Customization of products to fulfill customer specific demands [6];
- Sustainability and legal effects pushed by societal interests [9].

The aforementioned drivers ensure an ever-continuing need for product development. Fig. 1.1 presents an overview of the product life cycle, which reveals the different phases of product development, from creating the initial product ideas up until actual production of the product.

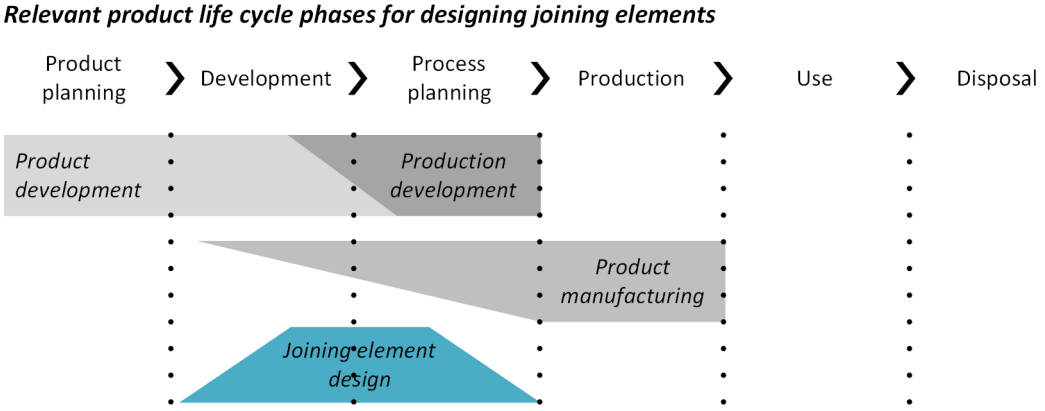


Fig. 1.1: An overview of the product life cycle with relevant phases for this study, taken from Eigner and Stelzer [11]. The image is extended with the joining element design phase.

The product development process addresses the question of what the product to be built actually is, whereas production development addresses how to build it. These processes partially overlap and have a significant influence on one another [11]. For example, choices regarding how to build a product influences its design. Fig. 1.1 also illustrates the product manufacturing process, which represents the actual manufacturing of the physical product. Here, an overlap with product development also exists, as product development requires not only digital but also physical prototypes to test designs and ideas [11]. Consequently, designs already affect production decisions early in the product life cycle.

Product development entails a decomposed sequence of the design, process, and production planning phases [12]. However, approaching product development as a sequential process hinders the exploitation of synergies between common systems without proper feedback [13]. Hence, collaboration is vital for developing common systems and their dependencies. In other words, product development cannot merely be a sequential process and

requires parallel processes. Said processes enable the synchronization and exploitation of the synergies of various stakeholders along the product life cycle. However, parallel life cycle processes lead to iterative collaboration due to requirements changing as new information is revealed.

Sequential and parallel development processes also affect joining elements. Fig. 1.1 also includes the joining element design process, which illustrates how the design of joining elements occupies a transitional phase between product development and production development as well as manufacturing. The joining element design process starts early in development, based on considerations such as the availability of resources, performance requirements, and cost. As the product matures, the design of joining elements becomes more detailed. Moreover, production ramp-up entails the analysis and evaluation of joining element designs. Production planners and engineers collaborate with designers to ensure manufacturability. In short, all of these overlapping processes in the product life cycle indicate the fact that numerous stakeholders are involved. Each stakeholder has their own preferences and requirements that products need to fulfill. To consider all their requirements, development needs to be multidisciplinary [14]. Moreover, the overlaps between the multidisciplinary processes necessitate impactful decisions early in the product life cycle, and these decisions significantly affect the fixed and change costs. Fig. 1.2 depicts the increasing fixed and change costs as well as the decreasing possibility of cost reduction during the product life cycle.

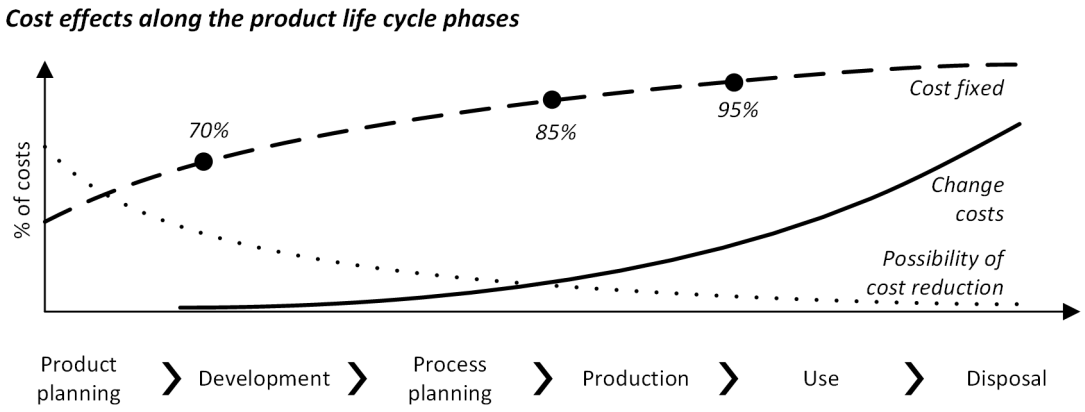


Fig. 1.2: A generic overview of the properties of costs along product life cycle phases, taken from Eigner and Stelzer [11].

As Fig. 1.2 presents, early phases of the product life cycle determine the fixed costs of the product, with decisions required on production facilities, product variants, and the product's positioning in the market [11]. The change costs rise rapidly once decisions have significant consequences for subsequent life cycle phases [11]. Every decision interlinks with multiple subsequent decisions, which over time rapidly increase the cost of revising early decisions. Furthermore, the possibility of reducing costs rapidly reduces. Hence, during development it is crucial to explore numerous potential solutions quickly as the change costs are still low during this phase.

The following subsection introduces joining elements, their importance in manufacturing,

and the major considerations in their design.

1.1.1 Joining elements

Joining is a critical process in manufacturing and includes welding, clinching, adhesive bonding, and riveting among others. Joining provides the function to a product as a whole and increases its manufacturability [15]. Joining processes smaller, cheaper parts to be assembled into more complex components and products [15]. The DIN 8593 standard [16] defines **joining** as follows: "*durably connecting or otherwise bringing together two or more geometrically defined work-pieces or the same kind of work-pieces with a shapeless substance. The cohesion is applied locally and increased in the whole respectively.*" Practically, joining defines a permanent connection without predefined objects, such as screws or nuts and bolts. Thus, the present study regarded a joining element as an individual entity that represents a joining process implementation. For example, when two components are spot welded, each spot weld is a joining element. Similarly, each line of glue is regarded as a joining element. In this context, **joining element design** refers to the task of defining and engineering the individual entities that represent joining processes. Joining has eight types of processes as per DIN 8593 [16], which are presented in Fig. 1.3. The DIN-standard subdivides joining in mechanical, thermal, and chemical processes.

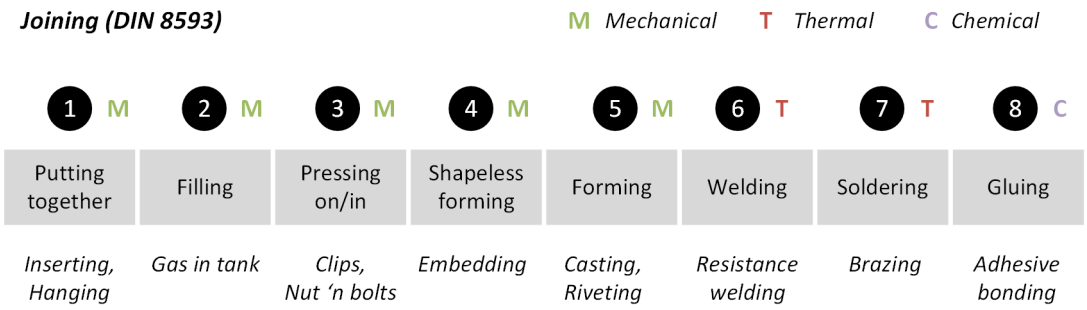


Fig. 1.3: Overview of joining processes according to DIN 8593 [16]

The figure shows an example for each process. This study focused on the forming, welding, soldering, and gluing processes. These processes create joining elements located on the contact regions (CRs) of components, and they do not merely depend on the interlocking of components' geometries. Furthermore, they describe nonphysical joining elements for connecting parts, such as through spot welds, adhesive bonds, or laser beam welds. Each process defined in DIN 8593 [16] has many variations and subprocesses, all of which have their own advantages and disadvantages. The standard DIN EN ISO 4063 [17] defines their nomenclature as well as the detailed process variants of joining technologies.

Each process variant and combinations thereof may occur numerous times in a product. Products can easily contain thousands of joining elements [18]. For example, in the automotive industry, a vehicle may contain over 10,000 spot welds [19]. As a result, joining has significant effects on the final quality and cost of products [20].

One reason is that it introduces additional complexity to products, development, and production, as well as other potential failure modes, quality problems, and costs. Choi et al. [21] estimated that joining activities account for more than 50% of the total production time and 20%–40% of the total production cost. The large influence on production time and costs may be due to the fact that almost a third of people working in manufacturing enterprises perform assembly or related activities [22]. These activities are labor intensive and thus necessitate the consideration of people's skills and capacities. Consequently, difficult designs of joining elements might cause quality issues in manufacturing, which would require rework on the product and possibly the design. Hence, improvements in joint design have a direct and large impact on the income of businesses [21]. As Fig. 1.2 indicated, early development decisions largely determine the total costs. This phase also includes the design of joining elements, implying that it also largely determines these costs. The economic importance of joining element design has led to much effort being invested into improving its efficiency and cost effectiveness [23].

Optimizing joint design entails various material and process improvements, such as additive manufacturing. Additive manufacturing may reduce the necessity for as well as number of joints in a product [24]. It enables the creation of complex and integrated parts, for which the functionality had previously only been achievable through combining multiple parts. However, joint-free products are unrealistic in most cases [15]. Therefore, joints remain key in manufacturing. Furthermore, joining is an enabler for heightened performance requirements and increased number of product features, which require multi-material and -component solutions [15].

Now that the necessity for joining has been clarified, it is critical to address the process of designing joining elements. Their design is not trivial [25] and requires the consideration of the entire product life cycle [14]. Moreover, the design of joining elements involves numerous stakeholders due to the broad diversity of requirements. However, as discussed under Fig. 1.1, the design process for joining elements is a multidisciplinary endeavor with constantly changing requirements.

Additionally, to create joining elements, the work of designers is increasingly moving toward increased administration, communication, and informing [11]. Decisions from downstream phases of the product life cycle have moved into the design and engineering processes. This trend also includes the designers in the planning, procurement, and production processes. Hence, designers have less time to focus on their core activities. Moreover, their workflows, tools, and processes have not been optimized to support this transition of their work activities.

Furthermore, the joining element design process has undergone a historical transition from 2D paper-based design to 3D computer-aided design (CAD) approaches. Legacy design methods relied on experience, thus limiting the quality of designs for complex use cases [26]. Therefore, joining elements are mainly the result of past designs, experience, and knowledge [7, 18]. Designers only have access to limited support from automation. This lack of tools forces them to manually search for the information that they require to create the designs. However, searching for and analyzing previous designs can consume up to 20% of a designer's time [27, 28]. Moreover, in industrial design offices, typically 70%–90% of

designers' time is spent on modifying, adapting, or in some other way redesigning already existing and proven designs [29]. Consequently, joining element design is a trial-and-error approach [18] and accessible methodologies are lacking, which makes designs highly practical [30]. Moreover, the designers are unable to find global optima, potentially due to the complexity of tasks and deadlines [18]. After creating joining element designs, they send them downstream to engineers for validation. Here, the term "downstream" refers to the product life cycle phases that can sequentially occur only after the design phase is finished.

Notably, joints that adhere to all structural and manufacturing requirements are vital for producing safe and quality products [18]. Any design that does not fulfill the requirements must undergo a new iteration. The reasons for new iterations include, but are not limited to, the following:

- Crash-worthiness – for example, a detailed finite element analysis may reveal a lack of structural performance;
- Tooling access – for example, partially built product may have components that interfere with the accessibility of tools for setting joints;
- Resource availability – for example, certain factories might not have a spot welding robot;
- Tact times – for example, certain joining processes might be too slow for the production rate.

Consequently, a lack of holistic process knowledge or experience may lead to design changes, longer product development cycles, and extra costs [28]. Furthermore, holistic designs that consider multiple phases in the product life cycle will become even more difficult due to increased product variety. Moreover, designing for multiple product variants in parallel requires increased collaboration between designers. As a result, high product variety also increases the complexity of joining element design and documentation [31]. For example, each joining element requires design manufacturability validations for each affected product variant. Each product variant is different and brings an additional set of conditions and requirements. Together with the sheer number of joining elements in each product, this makes their design a cumbersome and sometimes redundant endeavor.

Additionally, joining element design is not a stand-alone process as it goes hand in hand with component design. That is, joining element designs depend on components and vice versa. Therefore, it is an integrative task [32]. Moreover, the impact of joining elements surpasses that of the components they join. Their designs depend on neighboring joining elements as well as on similar implementations in other product variants. Thus, all of these parts mutually influence each other [33]. For example, sharing the same technology in an area of a product would enable the worker in production to use the same tools and equipment. Reducing the number of technologies in a product reduces the challenges induced by variety and uncertainty [34]. In short, the multidisciplinarity, lack of tools, product variety, and sub-system-level considerations make joining element design an opaque and complex process.

Consequently, the design of an ideal joint is relative to moments in time. Product development is a co-iterative process with continuously updating requirements [11], thus

creating new boundary conditions on joining element designs. Hence, there may be an expiration date on designs due to new insights or product decisions. Nevertheless, ideal joints exist, although they are relative to the available data at a given moment. Once product development matures and new data become available, designs might not fit the requirements anymore. However, an optimal joining element design can stand the test of time. Such designs are sustainable as they were designed in anticipation of an uncertain environment. This may, for example, promote the use of only a few basic joining technologies, which may suit many different materials, costs, or resource considerations. An ideal joint might not imply, for example, the best structural performance, but it might imply robustness to change. Design iterations are not preventable and contribute to successful product development [12]; however, bad joining element designs induce *unnecessary* design iterations.

The relevant literature has presented many methodologies, tools, and processes aimed at coping with the complexity in the joining element design process. The following paragraphs briefly introduce the strategies that researchers and companies have proposed.

Holistic product development aims to consider the requirements of the entire product life cycle [35] in terms of products' design, production, use, maintenance, and disposal phases. However, most joining and assembly-related design aspects focus on manufacturing through the approaches of Design for Manufacturing (DFM), Design for Assembly (DFA), or both. The DFA approaches systematically concentrate on reducing assembly costs or increasing the ease of assembly in the design phase [36]. Examples of these approaches have been provided in works such as [30, 37, 38]. These approaches define heuristic guidelines that are often not concrete [34]. Madrid et al. [34] listed the following examples: "design parts to give access to the joint," "distortion can be reduced by designing symmetry in parts," "design simple or straight contours," "avoid intersecting weld seams," and "avoid joints." Moreover, holistic approaches require formalization to enable optimization [30]. Formalization requires the modeling of all requirements, geometry, and use cases, which is highly complex.

Besides holistic overarching approaches, the literature has also presented more concrete methodologies that solve aspects of joining element design. For example, studies have described many methodologies for selecting materials, processes, and technologies. These include multidisciplinary decision-making algorithms (e.g., [39]), knowledge base query tools (e.g., [40]), or simple tabular overviews (e.g., [41]). Usually, most methodologies first screen infeasible technologies and then rank the remaining ones according to a given set of criteria (e.g., [42, 43]). However, a major drawback is that they pay little attention to product variety and available product data. Additionally, these methodologies do not output the locations of where to join and often do not consider geometries.

In addition to the technology, a joining element design also regards the locations of where to join components. Research has presented several approaches for automating these joining locations. Various methodologies have considerably different principles. However, they usually implement some finite element analysis to evaluate results [25]. Recently, methodologies have integrated component geometry design with joining locations (e.g., [44, 45]). They apply topology optimization to structures without predefining the joining locations. Furthermore, these methodologies can include the selection of joining technologies during optimization (e.g., [46]). However, these methodologies focus on single performance metrics,

such as noise, vibration, and harshness (e.g., [47]) or stiffness (e.g., [18, 48]). The main drawback of solutions optimized for one metric, such as strength, may not be successful for others, such as light-weight design. The results produced using optimization methodologies require balancing and validation before application to products. As previously mentioned, joining element design is multidisciplinary with diverse requirements. Not only a structural but also a business-optimal design must be found, such as one that considers ecological, economic, and manufacturability requirements [39]. Other drawbacks of such approaches are discrepancies between simulations and the real world. Such discrepancies require conservative interpretations concerning optimal designs. Lastly, optimization methodologies require completely defined products to determine the structural behavior of the product as a whole. This data-availability requirement prevents quick design studies in early product development.

A simpler way to create joining locations is to use predefined sets of rules. CAD systems (e.g., Siemens NX [49]) feature various tools for designing joining processes. Although these tools mainly assist designers, they also apply rule-based methods (e.g., [50]) to create joining locations. These methodologies operate within engineer-specified boundary conditions and parameters. Hence, they neglect any information from the product or production. The designer's experience remains the basis for designs.

In short, a limited number of solutions exist for automating joining element design. However, these are not satisfactory because, for example, they do not include available product data. This challenge intensifies for industries with high product variety. Here, companies may have thousands of product variants, and each of which may have thousands of joining elements. The increased product variety-induced complexity requires increased effort in joining element design, which makes the work even more complex and cumbersome. The following subsection will address product variety into more detail.

1.1.2 Product variety

The drive to satisfy customers forces companies to diversify their product portfolio [6]. Product variety refers to the growing trend of companies offering highly configurable products [6]. Schaffer and Schleich [51] defined **product variety** as "*versions of a technical product that differ from other versions of a technical product of the same type in at least one area of their technical specifications and are created by the combinations of different driver characteristics during the production process*." Even the simplest of products has variety [6], such as light bulbs. Although product variety occurs in all types of products, the number of variants in the automotive industry is immense. For example, in 2008, the estimated number of vehicle variants of the Mercedes-Benz E-Class was over 10^{24} [8], whereas BMW has claimed that nearly all of their vehicles are unique [52]. Fig. 1.4 presents the effects of product variety at each phase in the product life cycle.

Although product variety aims to fulfill customer needs, its effects in industry cause increases in costs and lead time. Costs increase as companies require increasingly more effort and overhead to manage the variety [54]. whereas lead time increases due to more overhead of parallel and sequential processes for coping with variety [55]. Some effects can be optimized under the same amount of product variety [53]. In product development, this includes the amount of effort required for engineering, testing, maintenance, and documentation [53].

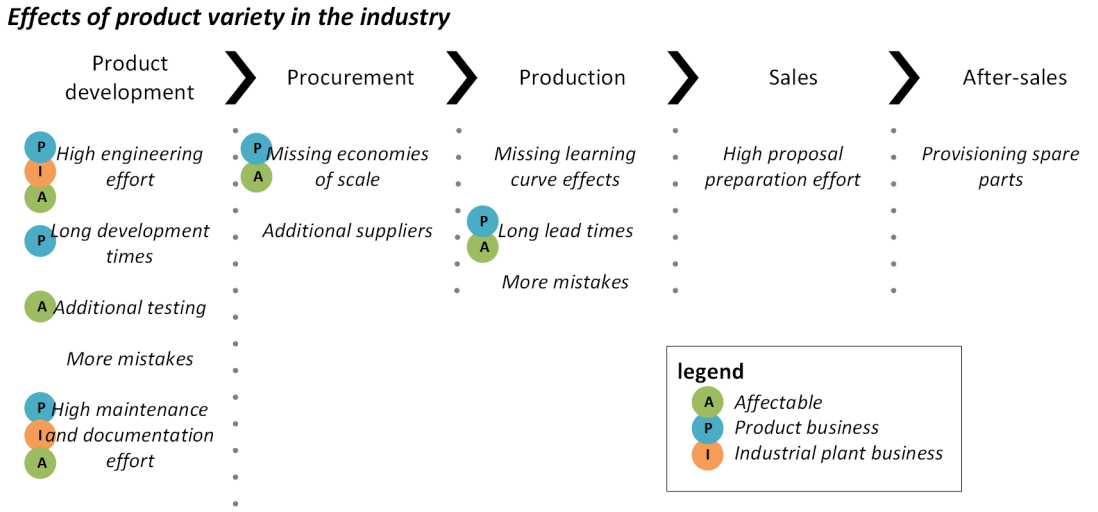


Fig. 1.4: Product variety's effects in industry, taken from Ripperda and Krause [53].

Processes in product development must consider product variety from early on, preferably through a concurrent strategy [38]. For example, DFA strategies may reduce the time required, difficulty, and resulting defect rates [56].

Variety decisions in product development affect downstream phases [11], such as production. For example, product variety causes a significant increase in the number of process variations, including changes to machines, tools, fixtures, set-ups, cycle times, and labor [57]. These variations induce planning, control, and logistic difficulties in production [58]. Poor development decisions affect the tasks of operators who, consequently, may reach their cognitive and physical limits [59]. As an illustration, every manufacturing system must be aware that any product variant can come at any time [60]. The effects cause the unit cost to increase by more than 20% when the variety of manufactured items is doubled [61]. Reported costs caused by product variety-induced complexity range from 20% to 40% of the total costs of a product [62]. Variety decisions must consider the added value compared with the costs that new product variants create. Hence, a trade-off occurs between the revenue from differentiation to meet customer demands and the resulting costs. Fig. 1.5 depicts the cost-, revenue-, and profit effects of product variety.

A wider variety corresponds to higher costs, lower margins, and lower quantities. The point of maximum profitability is often overestimated due to the difficulty of quantifying the complexity costs using traditional accounting methods [64]. Increasing product variety also increases the prices of standard products and prevents their competitive placement in the market [6].

Optimal product variety refers to the right set of product variants with the right feature combinations for precisely targeting customer needs and demands [6]. Limiting variety might interfere with a company's ability to serve customer needs [65]. Nevertheless, companies should not offer product variants that are not possible or desirable [66]. Furthermore, adding many arbitrary variants will create little additional value [6]. Product development includes

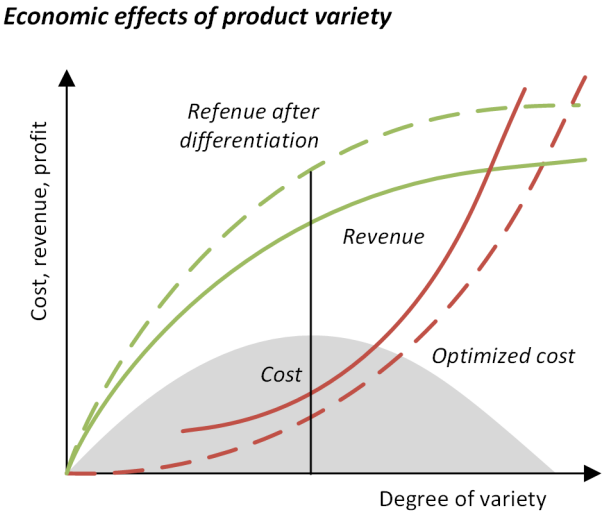


Fig. 1.5: The effects of product variety on the costs, revenues, and profits of companies; taken from Schuh et al. [63].

product variety considerations during the design phase to strategically address customer requirements [67]. Schmidt et al. [68] argued that three main variant management strategies exist, which are detailed as follows:

- **Reduction** – this is a short-term approach that strategically removes external variants or contains associated work. It includes methods such as portfolio downsizing and minor re-engineering of parts [69].
- **Handling** – this approach works mid-term and attempts to manage variety efficiently while still offering as many variants as possible, for example, through standardization and modular approaches [69].
- **Prevention** – this is the most potent and long-term strategy for reducing product variety. It considers the added value of each offered variant, only to offer possible product variants [8]. Prevention includes mainstream variety management approaches, such as modularization, scalable product platforms, and variety control [69].

Product variety can be divided into two types [8], namely external and internal variety. These are described as follows:

- **External product variety** refers to the number of product variants made available to the market. External factors that influence the creation of product variants can include customer demands, legislation, geographical regions, and market segmentation [6].
- **Internal product variety** refers to the resources and means available to companies to offer external variety. It involves making the external product variants available.

Industries seek to offer as many external product variants as possible while minimizing the induced internal variety [8]. However, companies can only affect the internal product variety

while preserving the external variety [70]. These factors influence the effects presented in Fig. 1.4. The architecture and technology of the product family determine the internal–external product variety ratio [71].

The organization of internal product variety is vital for companies. For example, the effective management of Volkswagen's product architecture allegedly saves the company 1.7 billion dollars in development and production costs annually [72]. This is mainly the result of component commonality [73], which refers to the reuse of the same components in various product variants. It is a handling strategy that enables companies to offer variants while containing costs [69]. Another effective strategy for product architecture management is modularization [74]. Modularization decomposes products into chunks (i.e., modules), which can be combined to create new product variants [75]. This can generate a competitive advantage and has significant importance in product development [76]. Together with commonality, modularity can affect the ratio of internal and external product variants positively. These approaches also enable companies to offer a wide variety of external products without requiring a large number of components [77].

Modularity and commonality were first implemented in 1914 [78], when an automotive engineer sought to reduce costs by mixing and matching components [78]. The engineer demanded standardized automobile subassemblies, such as axles, wheels, and fuel feeding mechanisms [78]. Modular product design includes the benchmark commonality and modularity [79], as well as other approaches such as standardization [69]. Modular product design increases manufacturing efficiency and effectiveness [75] and reduces inventory costs and lead time. These benefits align with the "affectable" effects in Fig. 1.4.

However, as previously mentioned, product development is a dynamic environment. Changes in design also affect modular design [80]. Hence, modules may change, which in turn may affect others [81]. The interfaces between modules also require commonalization to sustainably enable changes to modules [82]. Joining elements can reside on these interfaces. Without commonalization of interfaces, changes to modules – or to any components – may induce changes to joining elements. However, no relevant approaches can be found in the literature.

Nevertheless, the literature on modular product design has described many modularization methodologies (e.g., [71, 83, 84]) as well as commonality methodologies (e.g., [85–87]). These methodologies have various focal points, such as assembly complexity (e.g., [83]), product architectures (e.g., [88, 89]), standardization (e.g., [84]), and assembly stations (e.g., [90]). This variation led Gauss et al. [91] to synthesize a meta-process for developing modular products. Regardless, modular product design largely neglects optimization for module interfaces, which contain the joining elements that connect modules.

Noteworthily, AI includes optimization techniques that can create modular designs. The following subsection introduces AI and addresses current trends.

1.1.3 Artificial intelligence

Automating designs increases speed and quality while reducing costs [29]. Furthermore, it offers opportunities to utilize the characteristics of product variant design [29]. Software can efficiently take over cumbersome, repetitive tasks [29]. In this regard, AI aims to solve

problems that were traditionally solved by humans. According to Lu et al. [92], **AI** is defined as "*[a]ny theory, method, or technique to analyze, simulate, exploit, and explore human thinking process and behavior*." AI concerns computation in an intelligent manner. Lu et al. [92] provided the following four main properties of AI:

- It involves studying the features of human activities;
- It involves constructing a certain intelligent system;
- It involves computers performing tasks that only humans could do in the past;
- It involves simulating the underlying theories, approaches, and techniques of human behavior.

AI encompasses the takeover of *any* cognitive function. This study views AI as the capability to solve problems regardless of their type, origin, or difficulty. Theoretically, it is possible to view a calculator as a form of AI. Although most humans understand the processes of addition, subtraction, multiplication, and division, they are often unable to reconstruct a calculator or to specify how the internal processes work. In some sense, there is a kind of magic in this device. As such, computer systems that support decision-making or automation may be regarded as intelligent [29]. This broad interpretation of AI includes any possible technique for automating human tasks, and thus, it includes joining element design.

AI is on course to disrupt the economic and labor landscape [93]. It is considered one of the most predominant innovation drivers and technology trends in industrial product development and is continuously increasing in complexity. Recent breakthroughs have propelled the popularity of AI for all types of applications [94]. This technology enables the automation of complex tasks and can enhance process efficiency [95]. It aims to mimic humans' cognitive functions and understanding of the world [95]. The goals of most AI applications are to decrease error rates, assist in decision-making, and increase computational efficiency [95].

Most applications are considered in mainstream research fields, such as natural language processing, computer vision, and robotics [96]. Exploring the applicability of AI in other complex areas, such as product development and the manufacturing industry, is difficult. These areas require extensive amounts of expert knowledge and research to find feasible solutions [97]. Moreover, these areas have complex environmental conditions with ever-changing specifications, such as shortening product development cycles and increasing market competition [7]. Furthermore, most applications require enormous datasets, expertise, and resources for process optimization, monitoring, and control of applications at an unparalleled level of accuracy [98]. Therefore, manufacturing projects involving AI applications are limited to sub-processes instead of the whole process in this industry [99].

Today, the manufacturing industry has seen multiple successful applications of AI [100]. Four AI fields are popular in the manufacturing industry [101]: rule-based reasoning, case-based reasoning, search and optimization, and machine learning (ML). All four fields entails different levels of cognitive capability and model complexity. For example, programmers explicitly define processes and decisions in rule-based reasoning, applying all of their knowledge to the algorithms prior to usage. By contrast, ML finds patterns in data by

itself and only requires programmers to define boundary conditions. AI methods vary widely; thus, use cases determine which method should be used for which application.

Recently, ML has received much attention, which has led to the acceleration of AI and resulted in research breakthroughs for many applications [94]. In particular, since computational speeds have increased significantly, the popularity of deep learning has increased [95]. This refers to large neural networks (NNs) that can learn complicated concepts and identify features at higher levels of abstraction. These networks can find complex patterns that are not recognizable to humans. Most ML applications focus on image recognition, natural language processing, and the analysis of customer data for business-intelligence purposes [102].

Virtually no AI applications have been implemented for joining element design, although applications in adjacent fields have indicated promising results. Applications of AI in the manufacturing industry include finding the best geometry under static loads (e.g., [103]), selecting manufacturing processes (e.g., [104]), and predicting the number of spot welds (e.g., [105]). AI is implemented in various engineering applications such as automotive Body-in-White (BIW) structures (e.g., [47]), buildings and civil engineering (e.g., [95]), and rim design (e.g., [106]). AI models can predict structural topologies and display an understanding of geometric spatial structures [103]. Joining element design is a field with similar boundary conditions. The overlap between the fields invites the exploration of the application of AI methods, especially as appropriate solutions for the automated design of joining elements are lacking.

The following subsection summarizes the topic and context section.

1.1.4 Summary

Much research has examined joining element design, including holistic and knowledge-based development approaches, joining process selection, and joining location generation. However, these approaches often depend on input, preferences, and the experience of engineers. Suitable automation approaches are lacking, which results in time-consuming practical solutions that may require costly rework and unnecessary design iterations.

Moreover, in modular product design, many methodologies exist for optimizing product architectures. Their aim is to offer as many product variants to customers while containing the internal product variety-induced overhead and complexity. However, popular approaches, including commonalization and modularization, do not regard joining elements as objects of interest. This lack of consideration results in additional complexity and effort in the design and handling of joining elements throughout the entire product life cycle.

Moreover, AI is evolving rapidly with consistent breakthroughs in both research and business. In particular, the field of ML is growing and enables automation in virtually any field of product development. ML can do this due to its ability to capture patterns, extract knowledge, and apply them to new problems. Recently, multiple works have identified both the necessity and potential of AI in the manufacturing industry. Moreover, fields similar to joining element design have seen successful applications of AI. This raises the question of whether and how AI can be used to automate joining element design, thus reducing the time-consuming practical solutions for designers as well as the effects of product variety.

These considerations are taken into account in the following section, which presents the content and objectives of this study.

1.2 Content and objectives

This section discusses the purpose of this dissertation study as well as the environment in which it was conducted. Subsection 1.2.1 addresses the research problem. Then, subsection 1.2.2 presents the research aims, objectives, and questions. Next, subsection 1.2.3 discusses the scope, subsection 1.2.4 the significance, and subsection 1.2.5 the formatting of this dissertation study.

1.2.1 Research problem

Joining element design is a central part of the product design and detailing phases [43]. Here, holistic approaches advocate for the integrative design of components and joints to ensure effectively manufacturable products (e.g., [30, 107]). More concretely, the literature has presented various methodologies for partial aspects of joining elements, including joining technology selection (e.g., [40, 108, 109]) and the generation of joining locations (e.g., [18, 46, 50]). Moreover, modular product design can optimize product architectures, considering indices such as assembly complexity (e.g., [83]). Modularization makes it possible to manage variety-induced complexity and ensures profitable production.

However, these methodologies have not considered the availability of successfully marketed products. These products represent designs that have proven quality, regardless of the process for their design. Many state-of-the-art methodologies do not integrate the designs of successfully marketed products for new designs. Although, designers create joining elements using their experience as well as by analyzing past designs, but this information does not flow back into the methodologies. Moreover, literature lacks modularization and commonalization methodologies that consider product variety in joining element design.

As a result, the existing research is inadequate for industries with high product variety. Product variety and its induced manufacturing complexity will continue to increase. However, no proper methodologies exist for adequately handling the consequences for joining elements. Furthermore, new and ever-changing requirements in product development are making joining element design increasingly difficult and complex. This makes the joining element design process increasingly more time-consuming by creating only practical solutions. Hence, product development risks unnecessary design iterations and costly rework.

1.2.2 Research aims, objectives and questions

Based on the identified research problem, the main research aim is described as follows:

> *Given the lack of research on joining element design that considers successfully marketed product data and product variety, this study aimed to create a process with supplementary tools for automating joining element design in industries with high product variety.*

The following research objectives were developed to assist in achieving the research aim:

- To identify and evaluate methodologies for designing joining elements;
- To propose a methodology that implements machine learning for automating joining element design;
- To propose a methodology for considering product variety in joining element design;
- To define a generic framework for automating joining element design for high-variety products.

Based on these research objectives, the following central research question was developed:

How can joining element design be automated for high-variety products?

To assist in answering the central research question, the following research subquestions were developed:

- *How are products developed and how does this impact joining element design?*
  Through this question, this study attempted to determine the nature and processes of joining element design. It reviewed practice and the literature to evaluate available solutions and their advantages, drawbacks, and requirements.
- *How can joining element design be automated?*
  This question was specifically aimed at the application of AI in design engineering. AI technology mimics human thought process and behavior; thus, AI represents a collection of methods and tools that could possibly support the automation of joining element design. Designs have implicit knowledge and experience, which must, together with partial solutions in the literature and practice, be analyzed and integrated. Through this question, this study aimed to determine the applicability of various AI fields to the design of joining elements.
- *How should product variety be considered in joining element design?*
  Answering this question required the analysis of issues, challenges, and methodologies for designing for product variety. Product variety techniques have been used extensively, although mainly on functional levels in product architectures. Through this question, this study searched for the possibility of transferring modular design approaches to joining element design.

1.2.3 Scope

The research questions indicate a broad field of research for a dissertation. The research scope sets boundaries in order to investigate the topic profoundly and thoroughly.

The methodology for automating joining element design was validated using multiple concepts. These concepts worked within a confined use case in the automobile industry, specifically resistance spot weld designs of many BIW structures. A BIW structure refers to a car's frame once it has been joined together but before it has been painted and the motor or

any subassemblies have been integrated. As a result, this use case contains high product variety, and the designs are market validated.

This study focused on joining processes in the categories of forming, welding, soldering, and gluing according to DIN 8593 [16] (see Fig. 1.3). These processes do not rely mainly on the geometry of components to create joints. Hence, this study limited itself to locally applied shapeless substances; see DIN 8593[16].

The concepts validated the applicability of automation for joining element design. The aim was to determine whether automation can create feasible designs to support designers. The research objectives were not aimed at optimizing nor finding the optimal implementation of each concept. Thus, joining designs were not validated from a mechanical perspective that, for example, would evaluate the crash-worthiness of the design. However, the results did need to be plausible and useful. In short, this study was experimental and the concepts are not developed for a productive implementation.

The automation of joining element design is focused on early design phases in product development. These phases contain barely any tools for designers, as joining element design comes mainly from their experience and past designs. Automation can support designers through quick initial joining element designs and liberate them to focus on their core holistic and creative problem-solving skills.

1.2.4 Significance

This research contributes to design engineering by proposing a framework for automating joining element design in high-variety products. The framework supports designers as their jobs are becoming increasingly difficult and complex. In addition, it supports academia with a new perspective on design automation and modular design for joining elements. Moreover, this study evaluates and identifies the applicability of AI for automated joining element design. Consequently, this research proposes and validates several novel methodologies for automating joining element design and their modular design. Furthermore, this study addresses the current shortage of research in this area and provides real-world value to industries with high product variety.

1.2.5 Formatting

In this dissertation, different formatting is used to denote special text as follows:

- Definitions of terms are in **boldface** and their explanation are in *italics*;
- Quotes have “double quotation marks” and are formatted in *italics* with the corresponding source.
- Literature references are embedded in [square brackets];
- Sections are in **boldface** and the following sentence starts on the next line.

1.3 Structure of the dissertation

The remainder of this dissertation is organized as follows. *Chapter 2* (State of the art) addresses benchmarks in and surveys the literature on joining element design, modular product development, and AI. The chapter concludes with an overview of the literature and motivation for using ML in joining element design.

Chapter 3 (Methodology) first introduces an overarching framework for automating joining element design for high-variety manufacturing industries – also called VICTOR. Then, it evaluates the applicability of AI for each process step. It continues by presenting several concepts that use different AI techniques to predict the locations of joining elements. Lastly, it presents two processes to apply modular product design, namely modularization and commonalization, in joining element design.

Chapter 4 (Validation) presents implementations of selected concepts presented in the methodology chapter along with their results, and then discusses those results. Finally, *Chapter 5* (Conclusion) answers the research questions. It presents the results of the study as a whole and proposes recommendations for further research and implementation. Fig. 1.6 visualizes the structure of this dissertation:

The following chapter describes the state of the art of methodologies, practices, and processes. This enables defining the boundary conditions and requirements for developing a framework.

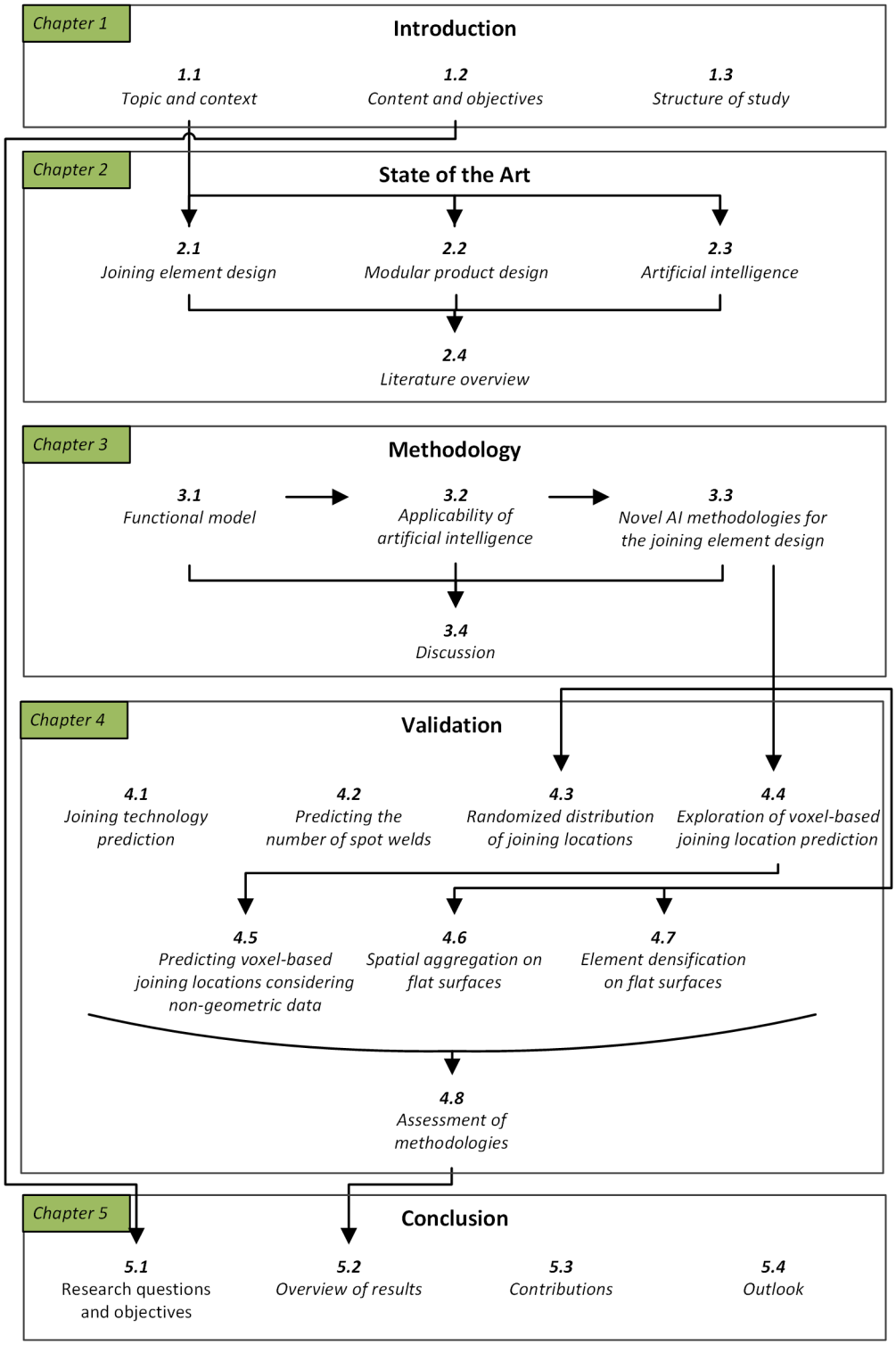


Fig. 1.6: Structure of the dissertation.

Chapter 2

State of the art

The aim of this chapter is to discuss benchmark approaches in the design of joining elements, modular product design, and artificial intelligence. The evaluation of these methodologies addresses a gap in the literature and presents the need for sophisticated automation of joining element design.

This chapter first addresses the state-of-the-art in joining element design in Section 2.1 by discussing the design process in the industry, DFA methods, joining technology selection, and the design of joining locations. It continues with modular product design in Section 2.2, explaining the complexity and its influence on assembly, and later product families, product platforms, and sustainable modularization. Next, Section 2.3 presents the state of the art of AI before going into depth about ML approaches, their considerations, and their applications in the manufacturing industry. Furthermore, Section 2.4 presents a literature overview, a discussion of the research gap, and the case for using ML in joining element design. Lastly, Section 2.5 summarizes and concludes this chapter.

2.1 Joining element design

This section addresses the practical and theoretical processes and methodologies used in joining element design. The following subsections address various vantage points on the design of joining elements. Firstly, subsection 2.1.1 addresses the practical process for designing joining elements. Then, subsection 2.1.2 discusses holistic Design for Assembly and Manufacturing approaches related to joining element design. Next, subsection 2.1.3 presents benchmark approaches of joining technology selection methodologies. Furthermore, subsection 2.1.4 discusses the design of the locations for joining using rule-based approaches. To that extent, subsection 2.1.5 presents optimization approaches to determine the locations for joining elements. Lastly, subsection 2.1.6 summarizes the findings in the state of the art in the design of joining elements.

Many definitions of joining concern the bringing together of individual parts to form a whole, such as the following: "*Joining is to put or bring together so as to form a unit*" [110]; "*[t]he process used to bring separate parts of components together to produce a unified whole assembly or structural entity*" [111]; or "*[a] large number of processes used to assemble*

individual parts into a larger, more complex component or assembly" [112]. The European Union Manufacture Technology platform defines joining as "*[c]reating a bond of some description between materials or components to achieve a specific physical performance*." These bonds can be mechanical, chemical, or thermal processes [15].

Other terms related to joining include assembly, fastening, or montage, and they partially overlap in meaning with joining. Fastening uses physical parts – fasteners such as screws, bolts, and rivets – to create joints specifically to attach, especially by pinning, tying, or nailing [110]. Joining is broader and includes processes such as welding and adhesive bonding. Assembly is "*[t]he fitting together of manufactured parts into a complete machine, structure, or unit of a machine*" [110]. Joining focuses on durable connections [16], whereas assembly more broadly concerns connections as in putting things together [110]], thus including temporal, reversible processes, such as screwing. Still, the terms joining and assembly in the manufacturing context largely overlap and can be used synonymously.

Joining is a critical process in manufacturing that provides the function to a product as a whole and increases its manufacturability [15]. Joining elementsare individual entities created for joining processes. This study acknowledges the representative digital counterparts of joining elements with the same terminology. Joining elements are an interface between products and processes. They are part of product design yet specify manufacturing tasks and operations.

To structure this information for early design phases, Eggink et al. [113] distinguished three aspects: technology, locations, and parameters (see Fig. 2.1). For example, these aspects enable simulation programs to analyze designs as well as production planners to create assembly sequences.

1. **Joining technology** as, for example, listed in DIN [17], refers to the processes of joining, such as resistance spot welding or adhesive bonding. Each technology brings its own set of requirements to manufacturing (e.g., tooling and material supply) and parts (e.g., materials and CRs). ISO categorizes it into various layers of detail [114].
2. **Joining locations** refer to the shape and position of joining and differ depending on the technology. For example, joining locations for riveting are point-shaped, 0-dimensional, and can be described by Cartesian coordinates. Another example is laser beam welding, where joining locations are curve-shaped and 1D, representing the line a tool traverses along workpieces to create a joint. Usually, the normal vectors – depending on the joining parts – describe the angle of tools for processing joints.
3. **Joining parameters** refer to information such as the diameter, material, or object type of the joining element. They describe the information required to pinpoint the type of elements, such as the type of rivet, the diameter of the clinching point, and the specific glue to use for adhesive bonding.

Fig. 2.1 illustrates the joining aspects using the examples of riveting, adhesive bonding, and laser beam welding.

Fig. 2.1 illustrates two U-shaped profiles that each have two flanges. The coinciding flanges create two CRs (green). A normal view of this CR provides much insight into the geometry

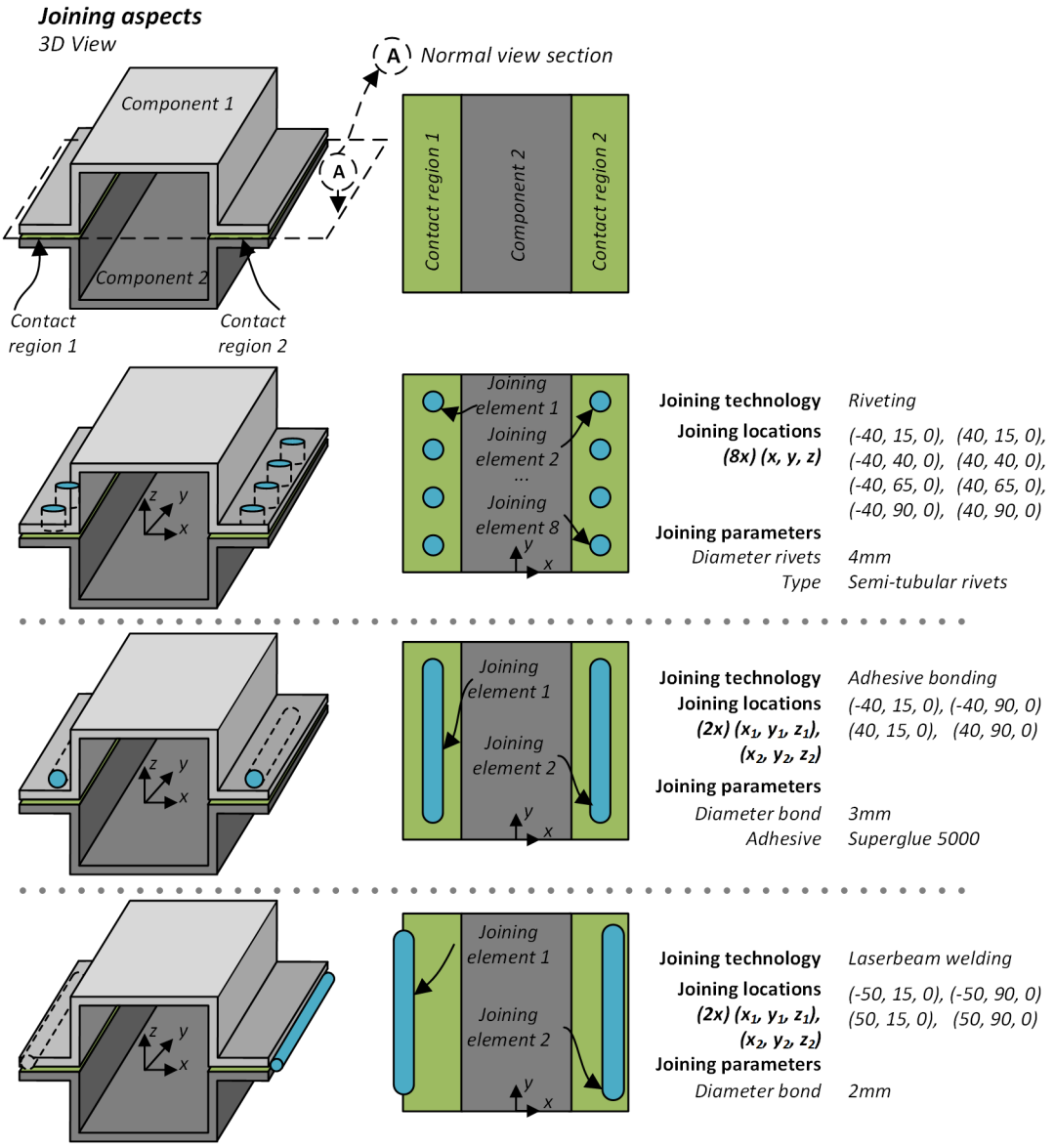


Fig. 2.1: Visualization of exemplary joining aspects: joining technology, locations, and parameters.

of joints. The right-hand side presents information on all three joining aspects. The structure and formatting of this information is exemplary and may differ between software packages, companies, and other applications.

Components in joints have CRs, which are geometrically coinciding areas due to individual part surfaces touching each other. The geometry of CRs may differ drastically, ranging from large overlapping surface areas to small coinciding edges. CRs influence applicable joining technologies to create a geometry-dependent design process [32].

Furthermore, joining locations describe where to set the tool for joining. Hence, they do not necessarily describe the geometry of the joining elements. The joining geometry also depends on the joining parameters, such as the diameter, to enable the joining element to be represented in 3D space.

Studies have argued that component geometry, materials, and joining technologies depend on each other and require iterative design processes [15, 32]. For example, CRs limit the available areas for setting joining elements by considering minimal edge distance requirements. Moreover, component design should consider potential joining technologies beforehand. International standards describe the organization of manufacturability requirements for many joining technologies (e.g., resistance spot welding for steel sheet metals [115]). They set boundary conditions on designs for both component and joining elements. Kaspar et al. [116] presented a selection of interrelations between component and joint section design. This selection visualized the many interdependent relations and considerations within joining element design, including weight, strength, materials, and economic and ecological requirements. For further references to state-of-the-art joining methods and their newest developments, this study referred to the work of Kim et al. [117] and Martinsen et al. [15].

Regardless of the many stakeholders, dependencies, and properties of joining elements, their main function is often to provide structural performance. They assist in holding a product together. Furthermore, joining elements create structural performance between components, which implies the ability to cope with the loads and forces that the product will undergo during use. The mechanics of materials is a field with many methods for determining the stresses and strains in structures [118], including beams and columns. Various types of stresses act on joints, such as tension, compression, bending, torsion, and buckling [118] (see Fig. 2.2).

Besides the raw geometrical considerations presented in Fig. 2.2,there are many other performance related parameters in joints, such as joining technologies, materials, and surface preparation [116, 119]. However, the design of joining locations is one of the most complex tasks for designers due, for example, to their high solution space. This design problem is further explained in Sections 2.1.5 and 3.1.3. The following list briefly addresses some rudimentary geometric issues for joining elements in relation to structural performance (see Fig. 2.2):

- **Static tension and compression** stresses are mainly correlated with the number of joining elements and the size of the CR [119]. Together, these parameters determine the load-bearing capacity of a joint. By increasing the number of joining elements, or the length, for example, of an adhesive bond, the ability to cope with tension and compression increases.

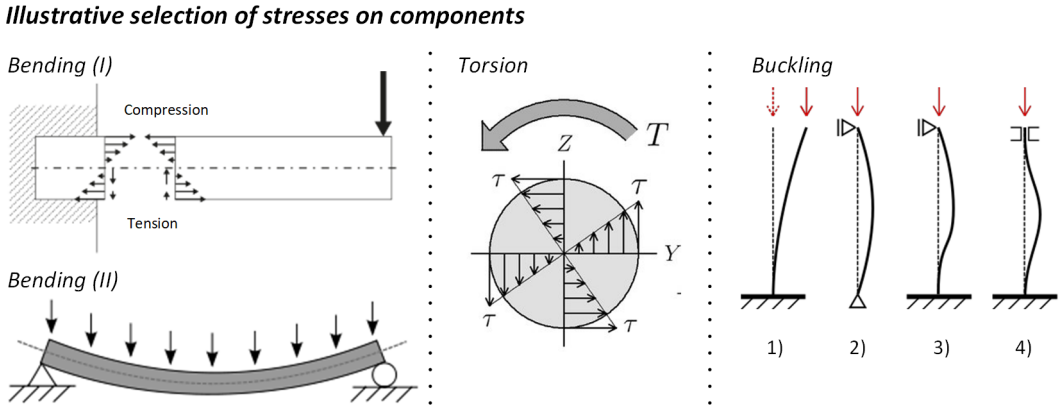


Fig. 2.2: Simple examples of stresses on typical joined components, taken from Soetens [119].

- **Bending and torsion** stresses (illustrated in Fig. 2.2) are mainly related to the section modulus [120]. The section modulus refers to the geometrical cross-section of the load-carrying components. The performance of the joint then depends on the moments of inertia of the load-carrying cross-section [118]. Bending has a neutral surface or axis where deformation is negligible. The strains and stresses are proportional to the distance from the neutral surface. Similarly, torsion has a neutral axis where deformation is lowest, which proportionally increases with the radius. Hence, placing joining elements furthest from the neutral surface optimizes them to endure stresses from bending and torsion (see Fig. 2.3). Here, it is the combination of the number and location of joining elements that determines the load-bearing capacity of the joint. Moreover, the shape of the CR must enable the design of the required joining locations.
- **Buckling and denting** (illustrated in Fig. 2.2) are, as previously mentioned, highly challenging failure modes to control [120]. These failure modes may occur when a thin component is compressed and deforms sideways. The geometrical cross-section and length of the components have a large influence [118]. Similar to bending and torsion, joining elements need to be located as far as possible from the neutral surface to more effectively handle stresses.

Buckling is often a dominant failure mode over the other stresses [118]. As such, the study elaborates on the cause and variables that affect buckling behavior. Consequently, and to the extent of buckling, the study discusses the relationship between joining locations and their capacity to cope with moments of inertia. Euler defined four meaningful cases of buckling to model beams in skeleton constructions [120] (see the right-hand side of Fig. 2.2). For example, the automotive industry implements these as longitudinal beams in the chassis of vehicles, and therefore, they are vital in crashes. Each case has a maximum vertical downward force that makes the beam buckle. Equation 2.1 describes the required force F_k for the second Euler case:

$$F_k = \frac{\pi^2 \cdot EI}{L^2} \tag{2.1}$$

where E is Young's modulus (modulus of elasticity, a material property); I is the area moment of inertia (also known as the second moment of area [118]); and L is the length of the beam. Both the length and second moment of area describe the geometry, although the latter represents the cross-sectional design of the beam. As this might include the joining of beams, the second moment of area describes the performance requirements of joining elements. To illustrate the calculation of the second moment of inertia, Fig. 2.3 presents a rectangular beam, which has a width b and height h.

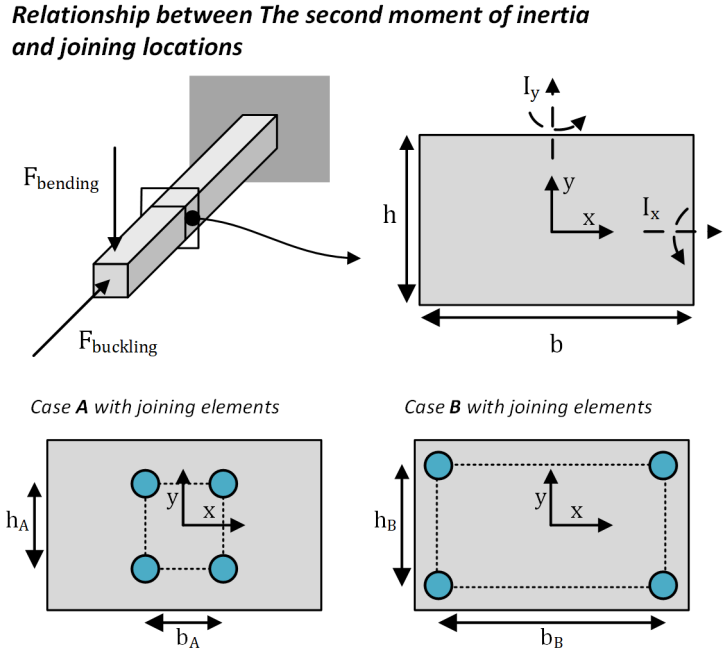


Fig. 2.3: Simple example of the relationship between spot weld locations on the second moment of area.

The centroid of the beam is located at the origin. The second moment of area with respect to the x-axis is I_x, while I_y represents this with respect to the y-axis. They measure the ability of cross-sectional shapes to resist bending or buckling caused by loading; see Eq. 2.2.

$$I_x = \frac{bh^3}{12}; \quad I_y = \frac{hb^3}{12} \tag{2.2}$$

This equation indicates that the section moment of area concerns the base and height up to the third power. Considering Eq. 2.1, this implies that beams with larger cross-sections require a higher force to make them buckle. Hence, a joined cross-section across the length of such a beam must transfer the same loads.

Let us assume a cross-section with spot welds that join two rectangular components. Moreover, as Fig. 2.3 illustrates for spot welds, the second moment of area increases when placing these components apart. Placing them away from the neutral center increases the substitute rectangle I. Hence, joining locations must be positioned on the outer edges of CRs to cope with buckling. Moreover, calculating bending, torsion, or denting stresses also relies

on cross-sections, and thus, on the second moment of area I [120]. For example, see the following stiffness equations for bending and torsion:

$$M = EI\kappa \tag{2.3}$$

$$T = \frac{J_T}{r}\tau_T = \frac{I_z}{r}\tau_T \tag{2.4}$$

$$I_z^{\text{rod}} = \frac{\pi}{2}r^4 \tag{2.5}$$

In Eq. 2.3, M is the applied bending moment and κ is the resulting curvature of the beam. Again, E and I appear. Their combination EI represents the bending stiffness of the beam, also known as the flexural stiffness [120]. In Eq. 2.4, T is the applied moment of torsion and J_T is the torsion constant for the cross-section. For example, this constant is equal to the polar moment of inertia of the section for a rod I_z (see Eq. 2.5). Next, r is the largest perpendicular distance from the rotational axis. Lastly, τ_T is the maximum shear stress at the outer surface.

Moreover, an infinite number of ways exist to apply stress to components, each of which affects the structural performance differently. For most purposes, it is sufficient to distinguish loading types into short-time static, long-time static, repeated, dynamic, or sudden loads [118]. Naturally, joining elements need to hold under normal conditions, but their performance is most critical in extreme cases. For example, in the aviation industry, turbulence causes unexpected and uncertain forces on the wings that fatigue the material [121], while in the automotive industry, joining elements must hold during crashes [18].

During crashes, buckling is a dominant failure mode and tends to overpower the other stresses [118]. Because of buckling's nonlinearity and difficulty to control, designing while considering buckling is a complex task. Moreover, there are certain areas in automotive structures (crumple zones) that even promote buckling to make vehicles more safe, thus further increasing the design complexity. Together, these sudden, unpredictable instability problems are highly complex [118]. To cope with these failure modes, designers often apply large safety factors [122].

The list above contains some rudimentary design consideration for the locations of joining elements. However, books such as Issler et al. [120] and Young et al. [118] have discussed many more loads and stresses in great depth. However, this study was unable to find reference material on the transfer of loads through cross-sections or equivalent representations of joining elements.

Still, the list provides some basic design rules for joining elements. For example, positioning joining elements at the outer regions of the CR optimizes the structural performance of joints. Additionally, increasing the number (or length) of joining elements enables the structure to cope with stresses, such as tension and compression. Moreover, over-dimensioning creates necessary safety factors for uncertain instability problems [122]. The inclusion of considerations for joining elements as well as the numerous other design parameters required by various stakeholders necessitates a systematic joining element design process.

The following subsection describes the design process of joining elements.

2.1.1 Design process

The joining element design process can be theorized from classical development models. For example, Fig. 2.4 depicts the W-shaped design model of Kaspar et al. [32] based on the product development process of Pahl and Beitz [123].

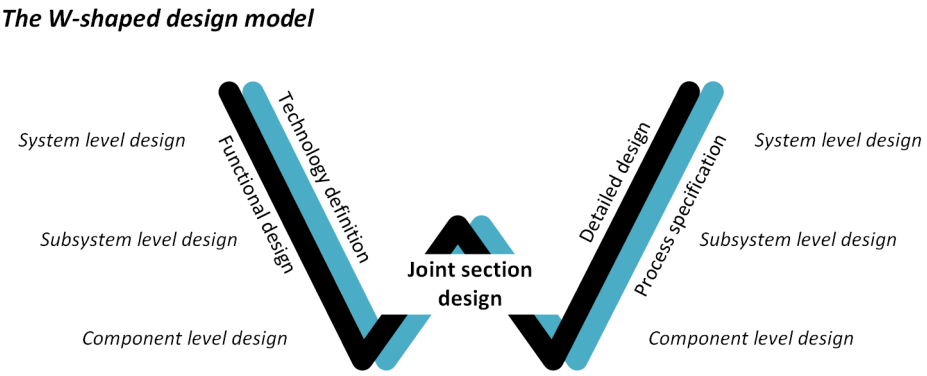


Fig. 2.4: Schematic and reduced view of the W-model, adapted from Kaspar et al. [116].

The W-model extends the classical V-model [124], which hierarchically decomposes from a top-down system design toward a detailed component design. Then, it integrates detailed component design from the bottom up by testing and integrating components, subsystems, and systems. The joint-section design is an additional phase of the V-model that details component designs while considering their neighboring parts. It regards joints as integrating subsystems. The W-model is a holistic approach for reducing the focus on optimal *individual* component design and increasing the focus on optimal *subsystem* design. Moreover, it emphasizes crucial cross-component aspects in engineering [32].

Fig. 2.5 presents a generic joining element design process as observed in the industry, which visualizes the steps and roles involved. The process requires the analysis of current joining element designs as well as the validation of multidiscipline requirements. Four prominent roles (coordinator, designer, engineer, and planner) contribute to the success of joining element design in the manufacturing industry. The same person or group may occupy multiple roles, and furthermore, multiple processes may run asynchronously. Every time the process runs through a feedback loop after requirements are not met at a quality gate, a new design iteration starts. Moreover, the process is a model, and as joining element design is multidisciplinary, roles and process steps might not follow the depicted flow strictly. The following list elaborates on the colored steps in the joining element design process in Fig. 2.5:

1. The **coordinator** creates a design request that constitutes the requirements, expectations, and objectives of the design. For example, some joining elements may require minor changes, or a new scenario may require a new joining element design that preferably uses adhesive bonds instead of spot welds. The request defines the boundary limits, general strategy, and possible solution directions. The coordinator also communicates with stakeholders, validates designs, and documents them accordingly.

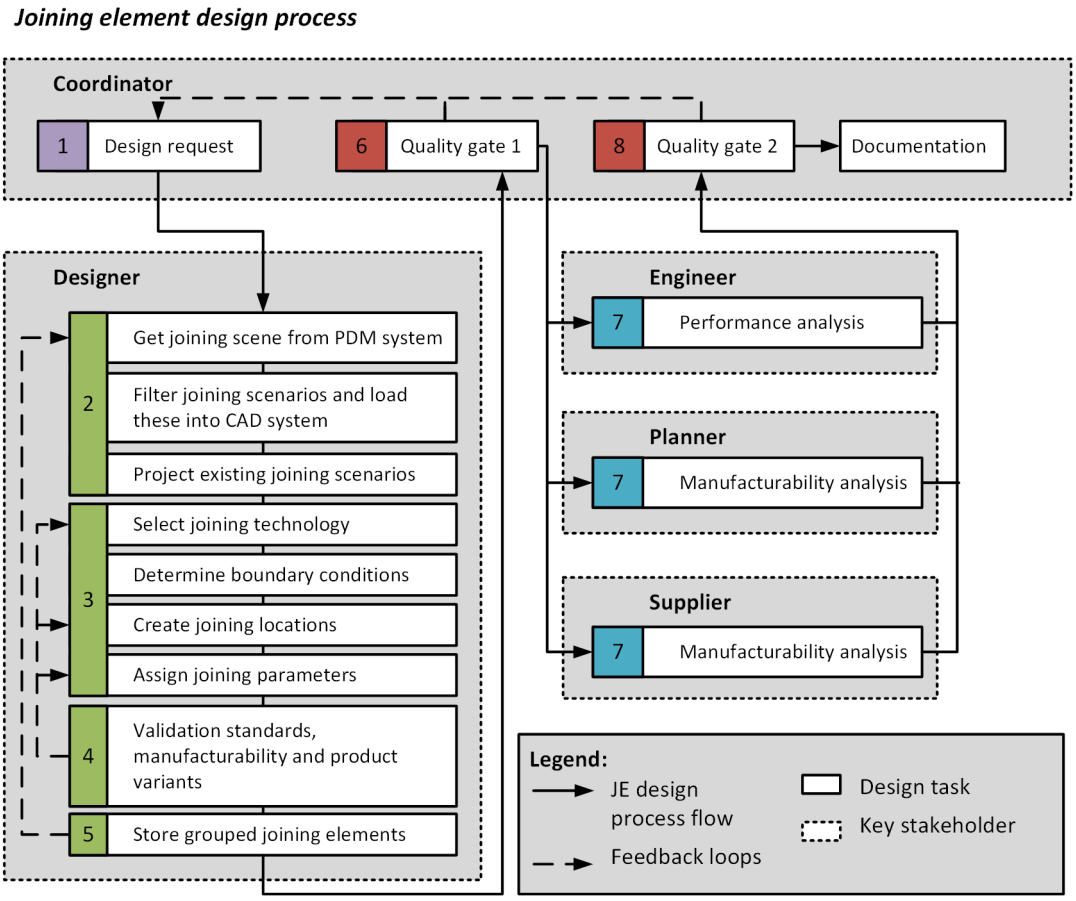


Fig. 2.5: Design process of joining element design in the manufacturing industry [1]. The starting point is the design request of the coordinator.

2. The **designer**, who are often outsourced [8], create joining elements in CAD software. They load data from product data management (PDM) systems that contain repositories for design files to enable the systematic, modular, and practical design of products as well as to archive data [14]. Depending on the availability of information from previous successful products and variants, designers must filter and select the appropriate data.

3. 3. After loading all resources, the designers can author joining elements. First, they select a joining technology using feasibility checks and preferences (see Section 2.1.3). Then, CAD systems are used to generate joining locations as curves or points by hand or using rule-based algorithms (see Section 2.1.4). Lastly, product manufacturing information (PMI) links nongeometric data to geometry (joining locations), which describes joining technologies and parameters. For example, PMI may describe the material of components, the type of glue for an adhesive bond, or the joining technology for a point-shaped joining location.

   Designers initially tend to implement one of the following three approaches when

designing a new joining aspect [29, 113]:

(a) **Use case consultancy** [27]: This involves interpreting previous designs and applying them to a new problem. It also includes consulting handbooks, making this a traditional approach [29]. Even for experienced designers, applying successful designs to new use cases is difficult. Knowledge management and the documentation of design choices often fail, requiring backtracking or even re-experimentation with a similar design [14].

(b) **Experience-based design** [7, 18]: The designer acts freely based on their experience without explicitly consulting previous designs or knowledge. For example, Andersson et al. [125] observed welding in production and found that the same repeated weld designs would often fail. They concluded that the designers did not reuse previous knowledge.

(c) **Minimal viable design** [25]: Regardless of the designers' experience or successful solutions, this approach uses the cheapest technology with as few joining elements as possible. Designers follow a set of rules to create an elementary design [29]. However, this bears a risk, namely that it may require the arbitrary addition of extra joining elements to regions with insufficient structural performance. Such actions can result in increased manufacturing and material costs [18].

4. Once the joining elements are authored, the designer validates their compliance to standards and company requirements to ensure their manufacturability. These norms and guidelines may originate from associations (e.g., ISO 18595 [114]), but also from company experiences and goals. Every theoretically possible product variant requires validation for the potential use of these joining elements.

5. Lastly, the designers save the joining elements in CAD files and store them in the PDM system. From here, downstream processes use the files for the simulation, planning, and validation of designs.

6. The first quality gate occurs here. The coordinator validates the designs based on the requirements specified in the design request. If the designs are adequate, the coordinator forwards them to simulation and planning engineers.

7. The **engineer** performs numerical analyses and validates designs, often by using finite element analysis (FEA), based on properties such as crash-worthiness [126, 127], noise-vibration-harshness [47, 128], and stiffness [18, 129, 130]. Simulation engineers consider joining elements in the product as a whole, evaluating joining designs for at least one complete product variant.

   The production **planner** is responsible for the manufacturability and work planning of, for example, assembly robots, which include topics such as assembly sequencing, tool accessibility, and resource capacities. Production planners, **suppliers**, or third-party stakeholders also analyze the manufacturability of the joining elements and may make suggestions for design changes.

8. The coordinator then receives the evaluations from the downstream stakeholders. The design requires a positive evaluation to pass the second quality gate. Unsuccessful designs may invoke a design iteration (i.e., a new design request is initiated). Although redesigning is not trivial, it is not a creative task [29]. Moreover, it requires new and existing product knowledge to be applied [29].

   The coordinator documents successful designs, which implies creating unique identifiers, storing data in the PDM system, and storing information regarding which product variants the joining elements belong to. Documentation is vital for preventing development delays [29]. For example, designers in the aircraft industry reported that the lack of documented design knowledge causes over 50% of delays in development [131].

Original equipment manufacturers (OEMs) compartmentalize work packages and tend to increase the outsourcing of processes and disciplines, such as design [8], engineering, and manufacturing [132]. This enables OEMs to reduce investment and complexity costs while potentially increasing the quality. However, a drawback of outsourcing is that companies may lose knowledge and weaken their control over activities [8] and variety [133]. Moreover, designers cannot be expected to have complete and high-level knowledge of the entire product life cycle [14]. Hence, stakeholders in disciplines such as simulation, planning, and supply must validate joining element designs independently and in parallel.

In addition, the design process has continuously changing requirements. Later in product development, designs become fixed, which increases the costs of rework and iterations (see Fig. 1.1). In addition, advanced product maturity limits design freedom and optimization potential, and it also imposes increasingly complex requirements. Joints interact with parts and may locally change their properties or performance. Safety margins help to overcome these problems and uncertainties and ensure assembly performance. However, increasing margins also increase processing times and costs. Moreover, economic and ecological trends enlarge the scope to include recyclability, disassembly, and environmental impact. Therefore, optimal designs are relative. That is, they change over time, although consider for a given moment all relevant stakeholders and fulfill all of their requirements.

Increasing product maturity raises product variety through the subsequent design of variants [134]. Newly designed product variants often contain many overlapping parts and components, enabling the reuse of joining elements in multiple variants [133]. The sequential design also causes unnecessary variety in product variants by potentially missing shareable parts. In practice, this results in lower commonality between product variants compared with simultaneous development [135]. However, simultaneous development might produce partially redundant designs. Development strategies, such as outsourcing, can hinder collaboration, creating new variants of otherwise shareable parts. Hence, joining element design becomes highly iterative due to the sequential design of new product variants [136] and concurrent design methods [30]. In a worst-case scenario, the complexity of documentation doubles for every new option or component added [137].

Moreover, product variety can be huge. Hence, it is often not viable to test and measure all product variants individually [138]. Instead, companies select a limited number of product variants to validate designs. These variants are considered representative of the performance

of most parts and variations of the product family [138]. Kreis et al. [19] argued that manually validating all joining elements for one product can be highly time-intensive. Additionally, it is often a manual task and thus prone to errors [19]. Automation software has also seen limited usage for such tasks [19]. Furthermore, the interaction between designers and engineers, typically between CAD and simulation environments, does not have general and uniform standards [19]. This lack of standardization results in company and supplier-specific software, making the design process prone to irregularities, errors, and delays.

Moreover, high product variety and mature products increase the potential for design automation [29]. Most development processes suffer from increased data and tasks. However, aligning the documentation and inner-workings of companies provides an opportunity for design automation. The identified issues in joining element design in the manufacturing industry are summarized as follows:

- **Time consumption**: Much time is spent on analyzing and verifying past and new joining element designs, especially when considering high variety. The actual authoring is a somewhat repetitive task. Unaccounted for product life cycle requirements, human errors, and adaptation due to new variants lead to unnecessary design iterations. These additional iterations leave little time for challenging tasks that require holistic and creative thinking.

- **Practical solutions**: Designers cannot find global (business) optima. They design in local space increasingly on an outsourced basis, making it difficult for them to consider holistic requirements and design consequences. Design tools solve only some aspects of joining element design, and thus, do not support proper modularization.

The current issues in joining element design had attention in the literature. The following subsection describes traditional DFA approaches for the design of joining elements.

2.1.2 Design for Assembly and Manufacturing

The concept of DFA refers to designing products for ease of assembly [36], often by systematically concentrating on reducing assembly costs in the design phase. DFA includes qualitative guidelines (e.g., [36]) and quantitative methodologies (e.g., [30]) to evaluate cost and manufacturability, enabling cost efficiencies to be increased from 20% to 70% [139]. Hence, Boothroyd argued that DFA is the prime consideration when simplifying products [140]. Similarly, DFM refers to designing products for the ease of manufacturing the parts [36], and Design for Manufacturing and Assembly (DFMA) or similar approaches integrate both design philosophies.

The DFA and DFM principles of Andreasen et al. [141] and Boothroyd and Dewhurst [142] originated in the 1980s. The methodologies aimed to simplify product designs to consist of fewer components, thus reducing their assembly time. For example, the DFA approach includes the following traditional methodologies [34]:

- The **assembly-oriented design process** of Warnecke and Babler [143], which considers four product development aspects [140]: product architecture, subassemblies,

components, and joining techniques. Design rules and assembly appropriateness reduce iterations and increase design effectiveness.

- The **assembly evaluation method** of Hitatchi et al. [144], who argued that each part may have only one assigned motion, which is evaluated by measuring the assembly difficulty and product architecture assembly cost. Today, studies are implementing assembly evaluation methods to assess assembly time, such as in [145, 146].
- The **Lucas method** of Chan et al. [147], which evaluates designs using complexity indices that consider the fitting and handling of components in production [147].

Moreover, Pahl and Beitz [123] defined graphical guidelines for practices and stressed simplification, standardization, automation, and quality. They provided a classification of assembly steps and requirements together with design steps for applying DFA systematically. Favi and Germani [148] optimized the ease of assembly in early product design by considering the product architecture and assembly sequence.

As previously mentioned, the DFMA approach integrates DFA and DFM. Both individual approaches advocate for reducing material, overhead, and labor costs by utilizing standards to reduce product development time and costs [36]. Originally, the methodologies that have applied DFMA to products with complex geometries and large quantities of components [34] with the aim of reducing costs and lead time and increasing reliability [14, 139]. Design influences approximately 70%–80% of the price. Usually, DFA and DFM are applied alternately, starting with DFA with the aim of reducing component quantity and variety, and continuing with DFM to simplify components [34].

Still, by definition, DFMA imposes design iterations, as product-specific production knowledge is necessary to design holistically [14, 30]. Design iterations are not necessarily bad as they enable one to optimize solutions. Only unnecessary iterations induced by limited information, bad decisions, and mistakes need to be prevented. Tasalloti et al. [14] argued that DFMA practices might induce tremendous front-loading into the design phase. For example, welding processes and their parameters can reduce the need for rework in the design phase. However, metallurgical effects require designers to be experts in material science and welding technologies, which are not always possible due to high complexity in manufacturing. Kwon et al. [28] listed significant welding decisions for engineers, indicating the complexity of design, which included accessibility and fixture interference, penetration depth and the number of passes, and levels of distortion within accepted ranges.

Besides the holistic approaches within DFMA for describing and aiding the design process, much simpler solutions such as handbooks and guidelines have long been used to support designers. Handbooks for construction (e.g., [107, 149, 150]) describe guidelines that address general construction methods and processes. Design requirements generally contain boundary limitations on the locations of joining elements in relation to the associated components. These requirements ensure that production can manufacture the joint, regardless of performance or business requirements. They also represent the bare minimum requirements that joints must comply with. Through standardizing component design requirements, such as minimum overlapping dimensions, joints can be freely designed. Companies can extend standards and guidelines that conform to their needs.

Discussion

The available methodologies in DFA are numerous and range from vague guidelines to data-driven assessment models. Although some recent work has integrated data-driven knowledge-based engineering and expert systems [151], many recent studies have still incorporated the original DFA and DFM philosophies (e.g., [75, 83]). Furthermore, most evaluations of manufacturability by specific technologies only consider limitations and requirements [30]. For example, standards impose design, production, and quality requirements that only limit the solution space, such as ISO 14373 [115]. The design of joining elements requires as much information as possible about both product and production. However, this information is often not available in early design phases [14, 34]. Design strategies also barely address component reuse, product evolution, and company knowledge.

Guidelines and practices are a necessary basis for designing joining elements, but they often ignore varying product geometry [34]. DFA aims to minimize variety and assembly complexity [14], but tends to ignore assembly performance [34] and product variety [6]. DFMA approaches can manage complexity [14, 34], although they tend not to be concrete enough for making appropriate decisions during product design [34]. Design problems are often not explicit and require formalization to enable optimization [30]. Furthermore, the industry struggles to have quality and consistent data for enabling this [101]. Moreover, guidelines and best practices are difficult to automate due to the high fluidity in designers' problem-solving process.

Concrete approaches for the design of joining elements includes the selection of processes. The following subsection describes the state of the art in joining technology selection.

2.1.3 Technology and parameter selection

In the early phases of product development, designers make decisions that have significant consequences for the resulting product. Designs need to consider the entire product life cycle. Hence, joints are subject to requirements that originate from production, usage, reparation, and disposal [8]. Traditionally, joining technology selection (JTS) was a manual process with incomplete supplementary information and relied heavily on experience and company practices [42]. Selection procedures were prone to inefficient solutions, misjudged parameters [152], and considerable time investments. Today, DFA has evolved and includes systematic JTS, where the methodologies aim to find the optimal technology given a set of design requirements [30]. However, standardized selection procedures are still vastly underused [108].

JTS is similar to manufacturing process and material selection methodologies. These problems correlate and require concurrent selection [153]. Although differences exist in the search space and input parameters, the methodologies are generic and, thus, are applicable to various industries. Martinsen et al. [15] described seven criteria for selection: (1) the design of the joint; (2) material selection and galvanic corrosion consideration; (3) joining process conditions; (4) health, environment, and safety; (5) flexible automation and design for X; (6) sustainability and life cycle engineering; and (7) profitability. Moreover, the products consist of more varied and complicated joining technologies [10], which increases the variability in joining, costs, and complexity. JTS requires the consideration of the entire product and moves away from decision-making for individual joining element designs.

Eglise et al. [152] argued that the literature has described roughly four types of systematic

approach since the 1960s:

- **Case-based selection**: This refers to the documentation of use cases, books, and habits. Designers pick solutions closest to the problem at hand. Recent work includes descriptions in handbooks such as that of Haberhauer [107].
- **Question-based selection**: This guides designers through a questionnaire to converge to a solution proposal.
- **Table-based approaches**: These arrange technologies using several properties and their quantities, although they leave the decision to the user. Recent work includes studies by Bond et al. [41] and Pruß et al. [154].
- **Task-based selection**: This uses a knowledge base to store relevant selection attributes and implements a procedure for determining the feasibility and ranking of solutions. Recent work includes studies by Chien et al. [155], Mesa et al. [156], Choudry et al. [39], Ghazilla et al. [42], Geda et al. [157], and Kaspar et al. [158].

  This quantitative approach enables the objective optimization of possible solutions and is regarded as the most effective methodology for selecting joining technologies [159]. The method is also generic as Kadkhoda et al. [160] demonstrated its applicability in the field of additive manufacturing.

Quantitative optimization algorithms can evaluate criteria and give designers objective recommendations for technology selection. A database contains all available joining technologies and their attributes. An attribute describes a value or range for which the technology is applicable. For example, resistance spot welding has an eight-millimeter maximum stack thickness of joined components [114]. Component combinations that create thicker stacks violate the requirement and cannot be selected. Attributes of joining element designs have specific types (Boolean, categorical, numerical, or unstructured), enabling filtering and ranking. The algorithms match design requirements to attributes of materials, processes, and components. Multicriteria decision making (MCDM) methodologies roughly comprise three steps, which are derived from the material selection approach of Ashby [161]: (1) screening, (2) ranking, and (3) selection. Fig. 2.6 depicts these three steps, which are detailed as follows:

1. The **screening** step filters joining technologies that are not applicable for a given design. For example, a requirement for a Boolean property, such as waterproofness, can directly filter technologies such as resistance spot welding or riveting, which are not waterproof. Numerical attributes might indicate a burning temperature for adhesives, making them inapplicable in high-temperature environments.

2. The **ranking** step ranks technologies according to their criteria fulfillment. For example, the methodology might consider the applicability of resistance spot welding on aluminum and the ability of adhesive bonds in high-temperature environments. Numerical properties optimize mathematically; for example, technologies with higher stiffness receive higher scores depending on the optimization objective.

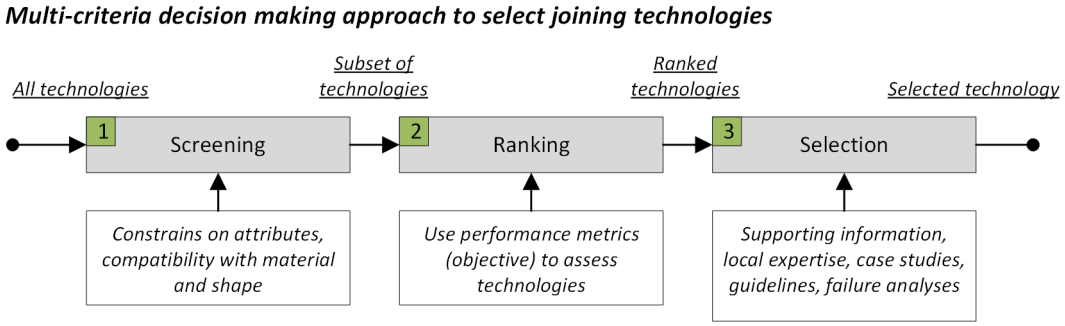


Fig. 2.6: The generic steps for joining technology selection in multi-criteria decision making methodologies. Adapted from Ashby et al. [161].

3. The **selection** step outputs the most appropriate joining technology from the rankings. The weighing of technologies and criteria enables designers to quantify preferences, enabling MCDM models to balance various aspects, such as design, cost, and function [42]. This step may also include supplementary information for designs that are difficult to represent in MCDM models, such as company strategies, resource availability, or results from detailed failure analyses.

Recent JTS methodologies have modeled the properties and characteristics of joining element designs in detail. Kaspar et al. [32] presented a holistic approach to support the selection of both part and joint designs. This concurrent approach implements three incrementally detailing design cycles, namely functional, fundamental, and specific. The cycles contain three substeps for detailing, namely screening, joinability, and detailing. These cycles integrate product and process design. Regarding joining technology assessment, Choudry et al. [108] incorporated a product life cycle assessment, calculating life cycle costs, and other technological values. Marini and Corney [162] identified seven general approaches for selection methodologies, and Renzi et al. [163] discussed the optimization methodologies of MCDM methodologies and mapped them to automotive problems. Instead of the popular MCDM methodologies, Geda and Kwong [109] implemented a genetic algorithm to find optimal fasteners while considering assembly and disassembly costs. Jeandin and Mascle [164] implemented a pairwise decision-making method using the importance of parameters on (dis-)assembly complexity. Das and Swain [40] proposed a knowledge-based framework to find similar joining element designs using ontologies, similar to the work of Kim et al. [117]. However, they still needed an MCDM module to analyze the feasibility and fitness of solutions. Chaimae et al. [165] also presented an ontology-based approach that uses case- and rule-based reasoning (see Section 2.3). On a more abstract methodology selection level, Hoefer and Frank [166] used ML to select manufacturing processes.

Joining parameters

Joining parameters are specific to each joining technology. They include the additive filler materials required for technologies such as MIG welding or adhesive bonding, as well as object types for technologies such as riveting and stud welding [28]. However, joining parameter

selection is the same type of problem as technology selection, but with more company and resource-specific solutions. The solution space depends on the product's maturity and is often relevant once detailed production information is available. Parameters tend to be considered at the manufacturing stage after finishing a product's detailed design [14]. Designers consider JTS earlier as it more frequently impacts part design [15].

The industry standardizes joining parameters, reducing product complexity [71] through using the same rivets, adhesives, and machine settings. CAD systems often incorporate catalogs to retrieve company standardized components [149]. For example, rivets are defined according to ISO 14588 [167]. Studies have presented the following methodologies that address parts of joining parameter selection: weld feasibility screening by Kwon et al. [28], screw parameter optimization by Friedrich et al. [168], and product recovery fastener selection by Ghazilla et al. [42]. Similarly, NNs can predict production parameters, as has been seen in studies on nugget diameter prediction by Sim and Kim [169] and Kim and Ahmed [170], on laser welding parameter selection by Yuguang et al. [171], and on spot weld distance prediction by Pillai et al. [105].

Discussion

Most JTS methodologies are data-driven and obtain an optimal result concerning one optimized metric; however, they require a mature stage in product development. Much remains unknown in the early stages of development, while decisions on products and processes highly depend on one another. Kaspar et al. [32] aimed to solve this problem using a concurrent, incremental detailing methodology. However, implementing a life cycle cost analysis and a life cycle assessment requires great modeling effort, which limits the rapid evaluation of variant alternatives. Moreover, life cycle assessments and cost analyses weigh heavily on, for example, adhesive bonding due to manufacturing investment and ecological costs [108]. However, such information is rarely present during product design [34]. Furthermore, Bond et al. [41] stated that the complexity and generality of many technology selection methods overshadow the required utility of having a simple tool that considers the product's functional requirements.

Benchmark JTS methodologies use MCDM techniques to screen infeasible technologies first and then optimize them accordingly [32, 42, 108]. All techniques require engineers to go through an interface to enter requirements and criteria. This process entails the detailed modeling of the properties of joining element designs to fit technology-selection tools, which is time-consuming and reduces efficiency. However, it also makes methodologies dynamic and enables flexibility due to changes in environments and requirements. For example, companies that implement new materials and technologies change the variables in JTS. Designers can tweak outcomes using preferences and weighing [108, 152, 155, 172]; however, such actions invoke unwanted bias in the system and reduce the quality of the objective optimal solution [14, 15], which remains prone to errors [52, 173].

MCDM methods all start from a clean slate and optimize for a single isolated joining element design. A significant drawback of these methods is their lack of consideration of stored product data. For example, if all joining elements in the geometrical vicinity are spot welds, then selecting adhesive bonding as the new technology would increase the

technological variability and manufacturing complexity. Furthermore, none of the methodologies consider product maturity. They treat joints individually without simultaneously considering designed joints or the adaptability of early designs. Due to product variety, components of one joint can also occur with different components in others. For example, part A joins with part B in product variant 1, and part A joins with part C in product variant 2. The consistency of part A in multiple variants might enable the standardization of other processes. The knowledge-based methodologies of Das and Swain [40] and Chaimae et al. [165] have considered similar joining requirements for new designs, but they lack concrete product variety considerations.

In addition to the selection of joining technologies, the design of joining locations is also an aspect of joining elements. The following subsection presents the rule-based approach for the prediction of joining locations.

2.1.4 Rule-based location prediction

CAD systems support designers in creating digital models and technical drawings (e.g., [49, 174, 175]). These systems contain various tools that implement rule-based methods to create joining locations (e.g., [50]). Such tools systematically guide designers in fully designing joining elements, including the creation of geometry and the documentation of joined parts and PMI. Rule-based location prediction is a pragmatic benchmark methodology for creating joining elements in the manufacturing industry, yet the literature has rarely addressed it. Fig. 2.7 presents a simple flowchart of rule-based design along with its inputs and results.

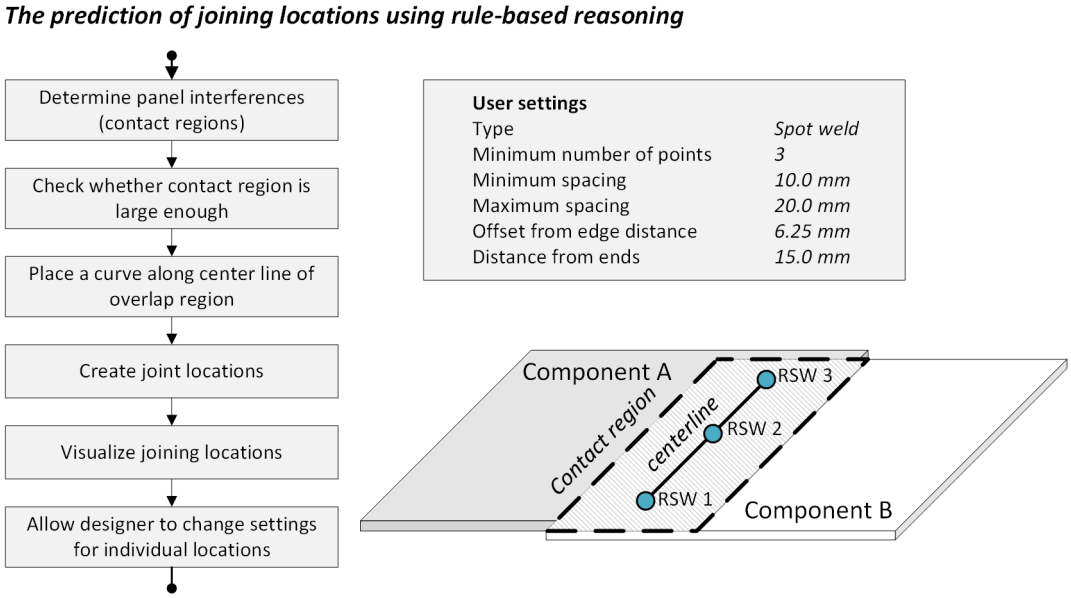


Fig. 2.7: Overview of rule-based design for developing joining elements in CAD systems, adapted from Thompson and Salerno [50].

Parts have CRs where their geometry coincides. First, the algorithm finds their

overlapping surfaces and determines their boundary. Then, it draws a centerline (guideline) along the longest side of the boundary. This line considers the edge distances specified by the designer. Next, it equally distributes the joining elements according to the specified number or mutual distance over the guideline. For curved joining locations, such as laser beam welds, the algorithm positions a tube element on the guideline. Designers can edit the individual joining elements and assign further PMI, such as joining parameters and component variants. CAD-based approaches enable front-loading by the designer, allowing them to finish the design directly after creation without other programs.

Parametrization is a flexible approach that makes components depend on a set of variables [29]. Here, components are frozen states of models based on sets of parameters that determine their features. Parametrization enables the automatic creation of part families, variations, and versions. By applying parametrization to joining element design, the number or length of joining elements may increase as the CR becomes larger due to changes in components' length [176]. The design of joints depends on CRs, and they change accordingly. Designers define rule sets for variables and use their expertise to automate designs. Parametrization also enables shape optimization techniques (discussed further in Section 2.1.5) and supports reuse by increasing the commonality between product variants [6]. CAD systems can often parameterize any design parameters (e.g., [174, 175]). There are approximately three types parametric geometry in parts [29], and larger parts often use a combination of these types [29].

- **Dimension-driven** parts only vary in dimensions [29]. Changes to the topology require the definition of new parts.
- **Generic** parts handle changes in topology through the activation and deactivation of part features [29], such as by adding or subtracting a chamfer along the edge of a part.
- **Modular** parts are constructed from multiple small parts [29]. Combining small parts creates a variety of topologies.

The design of joining elements depends on the geometry type of parts. For example, dimension-driven parts align with the parametric design method for spot welds by Pakalapati et al. [176]. Once the parts becomes larger for a longer product variant, the joining elements on the interfaces of the part may increase as well.

Discussion

Determining joining locations using rule-based algorithms enables the rapid creation of joining elements. These elements are directly editable as CAD systems often integrate such tools. Rule-based design supports designers in automating their workflow in CAD systems. It works side by side with traditional manual approaches performed in CAD systems. As joining element design remains a highly experience-based endeavor [18], this approach is an industry standard due to its fast results. Rough geometry suffices for the functioning of rule-based design and makes fast designs possible.

The experience-based workflow and tool's simplicity represent the first drawback of this approach and lead to suboptimal, highly practical results. The method automates the workflow of designers, not the tasks. Furthermore, it only considers component geometry, neglecting

utilization in products and the functional requirements of joints. Rule-based design considers parameters entered by designers that are biased and subjective. Although designers use their expertise for joint design, they often have little concern for or access to the product as a whole due, for example, to outsourcing, complexity, and time limitations. Weak spots are added arbitrarily to regions that lack performance. At the same time, strong sections of the product do not undergo the required reduction of joining elements [18], resulting in arbitrarily defined numbers and mutual distances of joining elements.

Rule-based methods predict from clean slates, meaning that all created joining elements are new. In industries with high product variety, rule-based methods lack an interface to PDM systems, reuse information, and other designs. This method leaves such considerations and necessary work in the hands of designers. Parametric design enables the integration of product variety. Furthermore, regular component reuse in product variety considers the entire specification of parts, including geometry, materials, and other PMI. Similarly, on a deeper level, parametric design embeds design rules to reuse features over multiple components [177].

Lastly, CAD systems require significant computational resources. They can display only small product variety portions due to file sizes and visualization limitations, making it difficult to determine optimal solutions.

However, the literature has more sophisticated methodologies to predict joining locations. The following subsection discusses optimization approaches for prediction of joining locations.

2.1.5 Location optimization

To achieve high joining performance in the overall product, joining locations should be systematically derived rather than experience-based [178]. Finite element-based optimization approaches aim to find joining locations that comply with a set of performance criteria as well as to minimize the number of joining elements. The number and location of joining elements greatly influence performance characteristics, such as static, dynamic, and crash behavior [130]. Additionally, joining elements affect cost and production time, making it necessary to minimize their number [25]. The optimal number of joining elements depends on their contributions to the entire structure's performance [130]. Notably, the optimal joining distribution is coupled tightly with the optimal structure itself [179].

Generally, the literature tends to optimize completely defined product variants for singular performance metrics, such as crash-worthiness (e.g., [47, 126, 127]), noise-vibration-harshness (e.g., [47, 128]), and stiffness (e.g., [18, 25, 129, 130]). Such methodologies use various types of FEA to calculate detailed loads and forces on joints during simulation. FEA uses objective functions that quantify the performance of a joining element according to its position by considering the peak loads and forces. A redistribution of joining elements would yield a different simulated performance. Furthermore, it would enable the reduction of the number of joining elements [18, 180] by removing those that do not carry loads. Having fewer joining elements reduces costs and production time [25]. Reduction methodologies start with a performance baseline. The difference between the baseline and the structural performance after calculation determines the quality of the joining locations. The difference might imply removing joining elements that contribute little to the structural

performance. It also can indicate that joining locations need improvement or that more joining elements are required. Certain methodologies consider both structural layout and reduction methods simultaneously [44, 45].

The solution space in structures can be enormous. For example, it implies distributing thousands of elements in the automobile industry [18]. The movement of one element can make the entire structure fail under specific loading conditions due, for example, to tearing behaviors that are similar to opening a zipper. Sensitivity research considers local failure by modeling real-world imperfections as failing joining elements. Methodologies model sensitivity by arbitrarily removing joining locations to model local failure before calculation.

Today, research focuses mainly on the topology optimization of spot welded structures, such as automobile BIW structures [18, 180]. Spot welding is highly common in manufacturing, and its modeling is relatively simple [25]. The finite element-based methodologies can model spot welds as beam objects with component material, enabling linear approximations during calculation, in contrast to other technologies with highly nonlinear behavior. For riveting or adhesive bonding, for example, methodologies require different materials to be inserted into the structure, thus requiring the modeling of varying material behaviors and multimaterial interactions. Joining technologies requiring the deformation of components, such as riveting, clinching, or folding, create additional tension inside component materials that influences joining quality. For example, Evren and Ozkol [181] optimized rivet locations for a fuselage frame that still required calculations by hand. However, they highly simplified the geometry of rivets as plain holes with beams.

Optimal spot weld layouts should derive models that are over-populated with spot welds and fulfill all performance requirements beforehand [25]. Furthermore, optimization should be run on a subselection of spot welds, not including those at, for example, the ends of flanges [176]. Ryberg et al. [25] distinguished three types of spot weld optimization techniques: (1) binary variable optimization, (2) parameterized size optimization, and (3) topology optimization. Due to recent advances in the field, structural optimization must be added an additional technique. These four spot weld optimization techniques are described in detail as follows:

- **Binary variable optimization**: This technique assigns a binary variable to spot weld locations indicating their presence in a product. Initially, the method requires an engineer to determine a rough viable spot weld distribution. The approach overpopulates this nominal structure with spot welds by filling up space in between. Next, it assesses each element's importance with various load cases. Then, spot welds are ranked by their carrying loads, which enables their systematic reduction while considering, for example, feasibility and proximity constraints. Generally, the results are dependent on the initial distribution of spot welds. Therefore, radically different paths to converge are not attempted and searching remains close to the suggested design space of the engineer. Examples of this approach can be found in the studies of Savic and Xu [182], Hasegawa et al. [183], Ouisse and Coghan [129], Bhatti et al. [130], and Ertas and Sonmez [184].

- **Parameterized size optimization**: This technique models the number of spot welds per

CR as a design variable for optimization, in contrast to the discrete spot weld basis seen in the binary variable approach. Practically, this approach automates parametric rule-based design in CAD systems. A designer defines guidelines along which spot welds are set. The approach determines the number of spot welds on a flange and distributes them equally. This process enables meta-models and simulation approximations to be utilized to reduce computational costs [176]. Compared with the binary variable approach, size optimization has fewer variables. Examples of this approach can be found in the studies of Eom et al. [185], Pakalapati et al. [176], and Geißler and Hahn [186].

- **Topology optimization**: This is the most widely used approach for optimizing structures [25] and was introduced in 1997 by Chirehdast and Jiang [187]. Today, the approach defines artificial densities of 3D material to represent spot welds, as proposed by Long et al. [180]. They are design variables that the optimization process updates depending on loads and forces in simulations after each iteration. Low values (empty) correspond to spot welds to be removed, whereas high values (material) correspond to spot welds to be retained. The objective is to minimize mass or volume fractions while adhering to performance metrics. Topology optimization automatically finds the optimal number of joining elements through the reduction of welding material. Examples of this approach can be found in the studies of Long et al. [180], Guirguis and Aly [179], Yang et al. [18], Chavare et al. [188], and Saito et al. [189]. Topology optimization also enables the consideration of joining technologies other than spot welding, such as bolts by Oinonen et al. [190] and Hou et al. [191].

- **Structural optimization**: Recently, research attention has shifted to optimizing both structure and joints simultaneously using the topology approach. Joining regions and quantities are derived from interim structural optimization results, not from previous designs. Additionally, these approaches consider the joining technology simultaneously [45]. Recent examples can be found in the studies of Florea et al. [46], Woischwill and Kim [45], and Ambrozkiewicz and Kriegesmann [192].

Discussion

Optimization techniques for joining locations enable critical structural performance criteria to be satisfied, thereby creating a baseline for further design [193]. Optimization is a systematic design approach that reduces designers' biases and, in contrast to rule-based design, considers the entire product.

The input for optimization problems requires the definition and positioning of all components of a product. This implies developed designs with determined materials, thicknesses, and geometries of all components, as optimization functions require high data quality. Preprocessing is often manual work and requires explicit engineering knowledge [18, 44].

Topology optimization creates joining location distributions for entire products, but all start from clean slates. There is no consideration of previously or parallel developed products. Hence, relying on their results creates significant additional complexity [194]. Topology optimization creates unique joining element distributions for every product variant. Joints are

unique for each product variant, even though they may have essentially the same components. Although topology optimization methodologies can consider fixed geometry (e.g., [195]), such as previously designed joining elements, this would imply the integration of some initial designs, which would consequently limit the optimization results.

Optimization techniques that involve FEA have a high computational cost. Furthermore, they require much effort in terms of data preparation, meshing, and postprocessing to obtain quality results. The techniques must optimize entire structures and cannot predict for individual joints.

Moreover, many optimization methodologies focus on single performance metrics, resulting in suboptimal overall results. Moreover, final joining element distributions leave room for interpretation. The highly nonlinear behavior of deformation and loads in simulations requires large safety factors. Models must consider the uncertainties of real-world products, such as buckling and crash behavior. For example, buckling sometimes requires safety factors to add several times the required strength to a result. However, this is true for every automated design as no model is a perfect reflection of the real world. Therefore, Ouisse et al. [129] argued that results are not necessarily informative considering, for example, the design robustness required due to manufacturing uncertainties as well as performance degradation caused by fatigue.

The following subsection summarizes the section regarding the design of joining elements.

2.1.6 Summary

Many product development approaches advocate for integrated multidisciplinary design (e.g., [30, 35, 123]). These overarching methods rightfully state that holistic design and a systemic view lead to better products compared with narrowly scoped counterparts. Unfortunately, they also result in a slower design process in early product development, which naturally contains many unknowns. Hence, the creation and evaluation of designs needs to be accelerated, thus enabling designers and engineers to perform their core competencies, namely creative and holistic thinking.

Current research tackles partial problems of joining element design. DFA is a general approach for aiding designers with guidelines, standards, and assessments. However, DFA processes are often not concrete [30] and require formalization and quality data to enable optimization [42].

More concretely, JTS methodologies aim to find an optimal joining technology considering design, production, and other product life cycle requirements [52, 108, 109]. However, none of these methodologies actively reuse designs or consider product variety. Moreover, they are data-driven and become highly complex assessments of individual joints, such that the literature advocates for more straightforward tools, especially in early product design phases [41].

CAD systems contain various tools that support joining location design [50] and can also assess it [30]. Rule-based and parametric design approaches automate designers' workflows and do not consider the product as a whole, thereby promoting trial-and-error approaches. This approach is highly reliant on the designer's experience, which may affect quality. Kahneman and Klein [196] found that specialists do not increase their expertise with experience in tasks that have unclear, incomplete information or nonrepetitive patterns. For

tasks that are not obvious with delayed or inaccurate feedback, due to the vast solution space, successful joining element designs require a systematic knowledge-based approach.

Recent studies have aimed to find optimal joining locations, considering performance metrics such as crash-worthiness, noise-vibration-harshness, stiffness [18, 188], and structural layout [197]. However, mathematical optimization functions require high data quality. Furthermore, the methodologies require models of completely defined products. These approaches are computationally expensive, require explicit engineering knowledge in development, and leave room for interpretation.

The former section discussed the design of joining elements. However, the section did not discuss any specific methodologies to cope with modular product design. Currently, these methodologies are lacking in the design of joining elements. Consequently, the following section presents a generic state of the art on modular product design, which is later reflected on joining element design.

2.2 Modular product design

This section addresses modular product design as an approach for coping with product variety. Firstly, it describes the activities in modular design, its uses, and design strategies. Subsection 2.2.1 addresses commonalization, which refers to finding parameters, components, and technologies to share between product variants. Subsection 2.2.2 discusses modularization, which refers to creating interchangeable parts. Then, subsection 2.2.3 addresses sustainable considerations for modular design. Lastly, subsection 2.2.4 summarizes the findings of modular product design.

According to Ma and Kremer [198], **modular product design** involves "*[s]ubdividing complicated products into components and considering them individually instead of as an amalgamated whole*." It reduces a product's complexity through decomposition, splitting harder, larger systems into easier, smaller subsystems [198]. These smaller subsystems are building blocks. Next, combining these building blocks makes more complex subassemblies. In turn, these are eventually combined to create products [198]. The building blocks are also called modules. Bonvoisin et al. [199] defined a **module** as "*[a] group of functional carriers (such as components, parts, or physical elements) in a product that contribute to a given function or a set of functions*." Preferably each functional component aligns with one module, with only a few interactions between them [200]. Thus, Bonvoisin et al. [199] considered modular product design as "*[a]n activity of designing a product that is made up of modules*."

Modular product design is a popular coping strategy for product variety in companies. It enables them to offer affordable products for customers [8] and manufacturing with nearly mass production efficiency [75, 201]. Modular product design enlarges the external customer-experienced product variety while shrinking the internal product variety [6]. Modular product design can reduce the in-house product and manufacturing complexity, thus positively affecting the modularity in production [38]. Modular products tend to have fewer components, enable pre-assembly processes, and use common interfaces [202]. Although modular product design initially requires a large investment [203], all requirements of the resulting product variants must be aligned for each module [203]. However, in return, the recurrent costs are significantly

lower [203].

To achieve these benefits of modular products, approximately three types of activities are required for designing them [204]:

- **Design with modules**: This activity regards designing products using existing modules [199]. It describes combining modules to create product variants. For example, using Lego™ bricks to build a house.
- **Identification of modules**: This activity determines new models from existing products. It requires the evaluation of component groupings, their clustering, and potentially their redesign [199]. Then, these outcomes need to be integrated into modules and corresponding interfaces to create products [199]. An example of module identification is to group components that have the same function, such as metal plates in fitness studios to load a barbell for training. Module identification is an activity performed *after* design processes [199], such as when evaluating designs and possibly considering new design iterations.
- **Design of modules**: This activity entails the design of new modules for (new) products. The design requires the grouping of functional carriers into modules as well as the design of their interfaces [199]. For example, a new keyboard could be designed such that it can be attached to computers using a USB port. The design of modules is an activity *within* the design process.

Components have intrinsic properties that determine their ability to be grouped or separated into modules [199]. However, module generation must be systematic and objective, which is crucial for comparing alternatives [205]. Furthermore, the designed modules require an arrangement to create products, that is, a scheme that allocates modules. Ulrich et al. [206] defined a **product architecture** as "*[s]chemes by which the function of a product is allocated to physical components*." The product architecture arranges components and functions into modules [199]. Besides arranging functional elements, it also maps them to physical components and specifies interfaces between them [38].

Hence, product architectures enable one to organize product variants. This affects product development and the entire product life cycle [199]. Moreover, optimizing product architectures is a benchmark strategy for reducing product variety-induced complexity costs [207]. The results are simplified and standardized interactions between modules [208]. Preferably, each module represents one functional element, with few interactions occurring between modules [200].

The mixing and matching of modules enables the creation of different product variants [91]. These are similar products that each possess specific functionalities for meeting customer demands [209]. A product family refers to the resulting group of product variants [210]. Simpson et al. [79] define a **product family** as "*[a] set of products that share one or more common 'elements' (e.g., components, modules, subsystems, fabrication processes, assembly operations) yet target a variety of different market segments*."

Product families can be completely modular without any common elements between any of their variants. Each product variant consists of a unique set of components or modules.

Hence, modularity is a core technique in designing product families [211]. However, optimizing product families also benefits significantly from increasing commonality [58].

Synthesized from their systematic review, Gauss et al. [91] described a meta-process for designing module-based product families. The meta-process included processes such as the planning and positioning of company strategies [200], translation of customer needs into functional requirements [79], and configuration of the optimal products out of a family [79]. Moreover, a part of the meta-process entails product family modeling, which generates the modules, platforms, and configuration structures [79]. Fig. 2.8 depicts this process, where each step in the process is detailed as follows, as defined by Gauss et al. [91]:

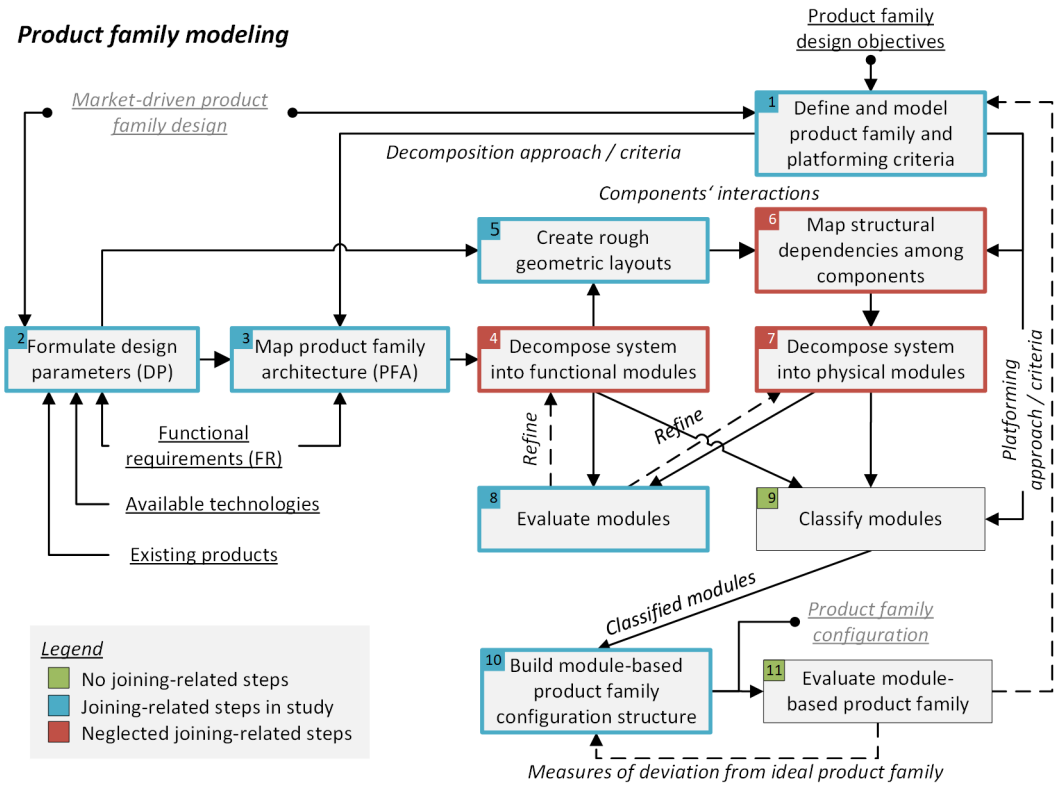


Fig. 2.8: Product family modeling meta-process, taken from Gauss et al. [91].

1. The process starts with defining and modeling the product family and platforming criteria. These steps can consider the variability of functional requirements [212] and redesign effort for multiple generations [213]. Low variety and redesign effort become platforms, as the components and modules that fulfill these functions can be shared between multiple product variants. Furthermore, high variety and effort enable specific customer demands to be met. [214]. These functional requirements create the necessary variety for differentiating product variants without negatively affecting the internal product variety.

2. The process continues with the formulation of design parameters, which map physical elements to fulfill functional requirements [91]. This formulation of design parameters involves the use of available technologies and existing products [215]. Together with the functional requirements, the design parameters set boundary conditions for the product architecture [91].
3. The product architecture concerns decoupled interfaces and mapping between functional requirements and design parameters [216]. In this step, the mapping process is performed, often using a design matrix [91].
4. Next, the product architecture is decomposed into functional modules [91]. These modules must physically match the working structure [123].
5. The fifth step identifies interactions among physical components [215], using the geometric layout for identification. Geometry enables the subsystems, subassemblies, and coupling of components to be determined [217]. The interface describes interactions between components, such as through CRs. Hence, this step sets boundary conditions for joining element designs.
6. Then, structural dependencies between components are modeled [87] by, for example, using a design structure matrix.
7. Next, the structural dependencies are decomposed into physical modules. The components in the modules require a high degree of interconnection between them, but a low degree to components outside of the modules [200].
8. Subsequently, the created functional and physical modules are evaluated according to criteria [91], such as modularity, commonality, or cost. The results may necessitate the definition of these modules being refined.
9. Next, the modules are classified into platforms and differentiated modules [191]. Platform modules contain less variety, enabling them to be shared in series of products [218]. Differentiated modules enable the instantiation of product family variants [91].
10. Then, a structure for product configuration describes the construction of products from modules [91]. Moreover, it defines the combination of modules through interconnections for various assembly levels [219].
11. All models in the configuration structure need to add value to the product [91]. This step evaluates the family and enables refinement.

Not all methodologies that involve modular product design use all of the steps in Fig. 2.8. Depending on the use cases, focus points, or relevance, the meta-process permits the omission of steps.

Furthermore, the blue boxes in Fig. 2.8 indicate the steps that involve joining element design considerations. Specifically, the figure highlights considerations of the assembly-related methodologies in the study (the works of Emmaty and Sarmah [220] and Jiao and Tseng [219]). However, most methodologies consider assembly and joining as criteria, such as the number and level of assembly steps between components in the work of Jiao and Tseng [219]. Furthermore, Emmaty and Sarmah [220] presented a modular design

framework that integrates DFMA (see Section 2.1.2). Their framework systematically urges designers to improve designs using DFA considerations that affect the product architecture. Neither of the methodologies actively create modules for joining element design (steps 4 and 7 in Fig. 2.8). Moreover, no mapping of structural dependencies occurs among components (step 6). Hence, the physical interactions have not attracted interest. Fig. 2.8 highlights the lack of methodologies to consider joining elements during both module creation and mapping of structural dependencies with red boxes. The literature review of this dissertation study (see Section 2.4) revealed that no modularization methodology has actively considered joining (or assembly) elements.

However, there are methodologies that have considered the interfaces between modules (i.e., [221, 222]). However, these modules are not found in the review of Gauss et al. [91] as they do not support the design of a module-based product family. The interfaces relate to step 5 in Fig. 2.8. They create component interactions as a result of creating geometric layouts. These interactions describe the structural dependencies between modules, after which the physical modules are created. However, describing interfaces is much broader than merely the joining of components. Interfaces also include topics such as energy, signals, materials, connections, and interactions [223]. An example is information regarding the movement of a mouse to a computer using a USB interface.

First, corresponding to step 5 in Fig. 2.8, Wang [221] presented a methodology that uses geometries of interfaces as constraints in modeling assemblies. For example, interfaces may have the modeling constraints of axis–axis and plane–plane interfaces. These interfaces have one degree of freedom (i.e., in the direction of the axis). This would enable screws or rivets to make a joint. Using these geometric constraints, any joining technology that fulfills them is feasible. The geometrical parameters also create scalability for the components. As a result, this would enable assembly modules to be defined that could meet those constraints. However, the methodology only creates a theoretical option for assembly modules. It does not aim to decompose the system into physical modules.

Tseng et al. [222] presented an algorithm for clustering joints (aligned with step 7 in Fig. 2.8). Each joint is classified by its type, disassembly, and joining direction. The values are ordered such that a distance exists in the values between each joint. The algorithm then searches for the most similarly defined joints, and similar joints become a module. The classical aim is to have highly similar joints within modules that distinguish themselves significantly from other modules. However, there are no considerations of geometry, nor of the creation of a product family. Hence, the modules do not relate to modules with components.

Tseng et al. [222] also clustered joints based on their similarity. Hence, they created groups of similar components. Referring to product families, they implemented common elements to share them between product variants [91]. Simpson [79] defined "*[a] set of common elements (parts, components, processes, sequences, etc.) sharing the underlying core technology based on which a stream of derivative products can be efficiently developed and launched*" as a **product platform**. Product platforms create a structure for developing product variants [224]. Moreover, product platforms enable commonality to be maximized and individual performance deviations to be minimized between product variants [224]. The design of product platforms involves "*[d]etermining the variables to be shared, as well as*

optimal values, for both shared and unique variables among variants" [79].

Schuh et al. [225] argued that there are three generic objectives of modular platforms: reducing the time to market, reducing the costs, and managing the customer demand for variety. Cerqueira et al. [226] analyzed the trends and opportunities of product platforms and identified three main problems, which are described as follows:

- **Configuration problems** with product platforms risk additional costs and worsen maintainability. Additionally, the authors argued for increasing effort in the allocation and arrangement of assembly modules [226] and avoid usage of different modules with the same function [82].
- **Inefficient product platform development** requires a streamlined information flow between all stakeholders to harvest the benefits of mass customization [226].
- **Manufacturing and assembly issues** for platform-based products require production lines to optimize the use of platforms and be more flexible and adaptable [226].

These identified problems align with the takeaways from the meta-process of Gauss et al. [91] in Fig. 2.8. Product platforms also lack interest in joining-related topics. The approaches seem highly concentrated on components and module creation, without considering their interfaces and assembly or joining technologies. Optimizing designs and platforms, such as by reducing variety through commonalization, can reduce resulting problems in production caused by design [132].

Product platforms apply various commonalization approaches to product variants, such as scalability through shrinkage or the extension of variables [79], but they also consider requirements for design changes in variation for the next generations of products [57]. However, with respect to modularization, modular platforms enable the addition, removal, and substitution of modules to create differentiated products [79]. Hence, platform-based design can maximize the sharing of common modules without threatening functional requirements [220]. Two approaches exist for designing product families using platforms [220]:

- **Top-down** approaches, which enable companies to strategically develop platforms for groups of modules; and
- **Bottom-up** approaches, which develop platforms using technologies and components to group them.

Referring back to Fig. 2.8, the meta-process depicts a top-down definition, for example, considering design objectives, parameters, the architecture, and most crucially decomposition. However, for large product families, the development process does not necessarily permit such a sequential process. It may have many parallel activities that are subject to iterations [12]. Moreover, top-down designs increase the number of constraints in detailed design phases significantly [221]. Hence, bottom-up approaches can reduce the created variety by defining new modules. Consequently, Fig. 2.8 also illustrates the refinement of feedback for evaluating created modular designs.

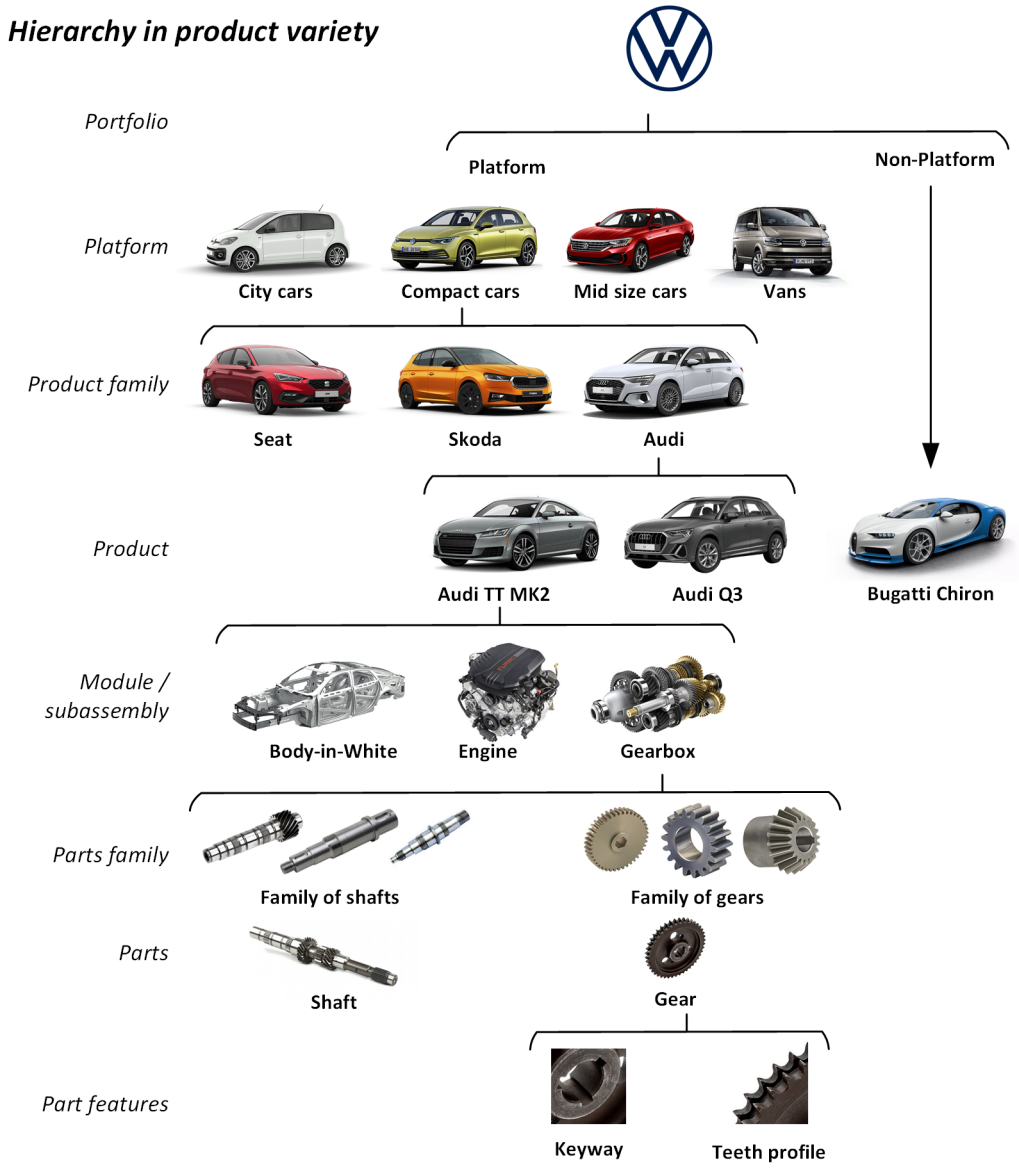


Fig. 2.9: Product variety hierarchy describing various terms and concepts, adapted from ElMaraghy [227]

This section has presented many terms for describing modular product design. Fig. 2.9 visualizes these terms for an enhanced understanding, using the example of Volkswagen by ElMaraghy [227].

The following list briefly addresses eight levels of variety in the hierarchy. Furthermore, Simpson et al. [79] explained these fundamental concepts in-depth.

- **Product portfolio** – the range of different products offered by a company [227]. It may consist of multiple product platforms and families.

- **Product platforms** – the sets of common elements for creating a number of product variants, such as the shared grills between the variants of A-compact cars in Fig. 2.9.
- **Product family** – the related product variants that share some common elements, such as parts, technologies, or processes. An example is all of the product variants of a Skoda (Fig. 2.9).
- **Products** – these emerge through combining subassemblies and modules [227]. The variety in the combination of these elements creates different product variants. One product variant relates to one specific configuration, such as an Audi TT MK2 (Fig. 2.9).
- **Subassembly** – this term describes several connected components that behave as a single entity [227]. They have a manufacturing perspective as they are rigid once assembled, such as the chassis subassembly in Fig. 2.9. Moreover, product **modules** are independent building blocks that consist of one or more components that can perform one or more functions [199]. For example, the interchangeable engine with variations of cubic capacity (Fig. 2.9).
- **Parts family** – this is equivalent to a product family and groups components according to geometrical or functional similarities. For example, the parts families in Fig. 2.9 are the various variations of shafts and gears.
- **Parts and components** – these cannot be decomposed without losing their ability to perform a function. They are single entities, such as a shaft or gear.
- **Part features** – these include both geometric features (e.g., flat, cylindrical, and thickness) and functional features (e.g., holes, flanges, and chamfers) [227].

Many of these terms occur in the literature, where many reviews can be found of state-of-the-art approaches; examples are the works of Gauss et al. [91], Bonvoisin et al. [199], Otto et al. [211], Simpson et al. [79], and Jiao et al. [57]. These examples indicate the interest of research and corporate firms in modular design. However, these reviews have had different focus points, such as the identification of structures, logical sequences, or performance effects. Regardless, they have all aimed to define an overarching process. They include all aspects of product development, from the identification of market needs to product configuration. However, as identified in Fig. 2.8, a need exists for modularization with regard to joining, which is a part of modular product design.

Moreover, the latter two activities in modular product design (i.e., module identification and design) *generate* modules (modularization) [199]. Bonvoisin et al. [199] regarded **product modularization** as "*[d]efining modules into the product architecture*". The following subsections discuss commonalization (subsection 2.2.1) and modularization (subsection 2.2.2). Commonalization aims to find parameters, components, and technologies to share and may aid bottom-up designs of product platforms. It supports the improved identification of modules. Modularization can be combined with commonalized results to create modules.

2.2.1 Commonalization

Commonalization unifies similar parts and modules to reduce unnecessary variability in similar product variants using modules [57]. AlGeddawy andElMaraghy [85] defined **commonality** as the "*[d]egree of similarity between products*". They emphasized that it is a part of the solution for reducing the complexity of product variants' design and development in a dynamic, uncertain environment. Commonalization is also associated with the standardization of product interfaces, modules, and components [132]. Standardization enables reuse where variability is not required [6]. Standardization applies to both components and interfaces [228]. Moreover, the standardization of interfaces is vital in modular product platforms for minimizing information flows [229]. As such, commonalization can create sets of common elements to share between product variants. Thus, commonalization supports the creation of product platforms.

The aim of commonality is to use as few different components in as many products and their variations as possible, and for as long as economically possible [135]. High degrees of commonality imply more significant economies of scale and scope [230]. However, complete commonality prevents necessary differentiation in product variants to fulfill customer needs [135]. As a solution, product architectures capture the overarching relationships between products through, for example, functional requirements [91].

In addition to *component* commonality, *process* commonality refers to the degree to which processes share the same building blocks [132]. It enables manufacturing with fewer processes [132]. Process commonality can cluster product components with similar design features or manufacturing processes into families, thereby reducing manufacturing costs and lead time while increasing flexibility [75]. Component and process commonality are related, as component commonality increases commonality in processes. However, this does not hold in the opposite direction as product components may require several different manufacturing processes [231]. This relationship emphasizes the importance of commonality in DFA approaches, which are (as mentioned in Section 2.1.2) often only captured in vague guidelines [34].

Systematic commonalizationis a crucial strategy in creating product platforms. Three effective approaches exist for creating appropriate platforms using a commonalization approach [87]:

- **Quantitative functional models** help to compute product families and rate modules according to their fulfillment of customer demands [87]. The approaches are broadly applicable and only give designers several platform selection heuristics rather than a physical solution [87]. An example can be found in the study of Stone et al. [67].
- **Design optimization** aims to maximize technical product performance and commonality or to minimize cost. Optimal design only regards specified evaluation metrics and often neglects modularity, complexity, adaptability, visual appeal, and ergonomics [87]. The problem is similar to the topology optimization methodologies for joining locations. Recent relevant works include those of Moon et al. [232] and Eichstetter et al. [194].
- **Exhaustive enumeration** refers to the charting of all component-sharing possibilities

for qualitative evaluation [233]. However, this is only suitable for product families with a small number of components, as the number of possible partitions of a set scales exponentially [87].

Commonality indices can measure the effectiveness of product platform design approaches. These indices can also measure the similarity between product variants. Moreover, they act as guides for designers to introduce new products by evaluating alternatives [85]. Commonality indices may serve as a proxy for predicting the activity costs of design, manufacturing, and assembly of new product variants [234]. Wong and Wynn [135] even argued that reducing the variety between successively designed variants improves product platforms significantly. Hence, it is vital to constantly measure and track the commonality between sequentially designed product variants

To assess commonality, Baylis et al. [87] summarized nine indices between product variants, all of which have a different focus point and limitations [235]. An example is the Percent Commonality Index ($\%CI$) of Siddique et al. [236], which can express commonality for individual product variants. This property enables one to evaluate design alternatives early in product design. Other commonality indices usually consider the entire product family [79], which often relies on unavailable information in product design. The Percent Commonality Index has three viewpoints: component, connections, and assembly, and takes the weighted sum. Eq. 2.6 presents an example of the component–component connection viewpoint:

$$\text{Commonality of connections} = \frac{100 \times \text{common connections}}{\text{common} + \text{unique connections}} \tag{2.6}$$

Methodologies for automating the creation of product platforms or for commonalization are found mainly in a few studies, including those of Baylis et al. [87], AlGeddawy and ElMaraghy [85], and Galizia et al. [86]. These solutions all solve the problem differently; see the three effective approaches [87] mentioned earlier in this section. Baylis et al. [87] used exhaustive enumeration to balance modular design. Their method decreases commonality, which in turn enables modularity to be increased. AlGeddawy and ElMaraghy [85] implemented design optimization by using hierarchical clustering and liaison graphs. This methodology combines common parts into integral parts and evaluates them. Galizia et al. [86] balanced variety in product platforms and the number of (dis)-assembly tasks. They optimized product platforms to determine the final product variants. Jung and Simpson [213] redesigned product families and platforms to prioritize parts and interfaces for redesign. Ukala and Sunmola [237] employed a simple yet sophisticated approach in the form of a rule-based methodology for redesign while considering DFA to reduce complexity. Based on *IF-ELSE* conditions, designers can obtain new systematic design suggestions that consider preferred design solutions. Therefore, they proposed an assessment method for deciding between reusing or newly designing components. Commonalization in product design can even be as trivial as considering the commonalization of shapes [238]. In short, many different approaches and perspectives exist for commonalization. Together with the methodology, it is the time and place in product development that make commonalization an enabler for managing product variety.

Discussion
Product platforms with a high focus on shared units might restrict potential product variability [239]. Variable costs might increase due to the availability dependency over multiple products [240]. As commonality increases the ratio of shared components, it may also prevent the differentiation of products. It is challenging to allocate additional development and coordination costs for sharing components in multiple product variants to individual designs [241].

Product platforms tend to affect DFA approaches [85]. Both modularization and commonalization opt for reuse and flexibility, which may conflict with the integrative and static production-oriented design of DFA approaches [220]. This is backed up by the trends and opportunities of platform design identified by Cerqueira et al. [226]. They argued that the consideration of joining elements is lacking and advocated for the creation assembly modules. These considerations could reduce the problems in manufacturing and assembly that result from modular design.

However, the literature does not provide specific methodologies for commonalizing joining elements. Some studies have included assembly considerations (e.g., [85]), or reduced technological variety (e.g., [237]), but such studies are often either highly complex (e.g., [86]) or they oversimplify (e.g., [237]) joining element design. No systemic methodology has aimed to reduce the variety in joining element design to enable the creation of adequate modules and product platforms.

Contrary to commonalization, modularization is an approach that not only aims to find shareable parts between product variants, but defines functional blocks as placeholders for interchangeable parts to create product variants. The following subsection discusses modularization into detail.

2.2.2 Modularization

Modularity deconstructs products into independent functional blocks, where interchangeable parts fulfill these functions. Ulrich et al. [206] defined **modularity** as "*[t]he standardization of components and processes in an organization that can be configured into a wide range of end products to meet specific customer demands*." Therefore, modular product design includes the act of reducing costs through creating reusable groups of components. Jiao and Tseng [57] defined modules as "*[p]hysical or conceptual groupings of components*." Modules can be treated as logical units in a system or product architecture [242].

Relating modularization as a design approach to management, Campagnolo and Camuffo [243] performed a literature review. They opted for fundamental research on both modularization economics and the relationship between modularity and performance. Modules are simple sections of a product to outsource. However, whether companies make modules to outsource or whether outsourcing requires the development of modules is still debated [243]. Nevertheless, the relationship between the product architecture and the continuously shifting boundaries of firms requires consideration in modular product design [243].

Besides managerial considerations, Kubota et al. [132] analyzed the relationships between modularity in design and production, which are not one-way relationships. They have

conceptual elements that affect both product and organizational structure [132]. This is another example of the nonsequential process of product development, as was observed in the joining element design process in Section 2.1.1. The benefits of modularity are reduced if one neglects to consider the implications on manufacturing processes [132]. However, modularization may conflict with DFMA [199]. For example, it may increase the number of parts and thus the number of assembly errors [85].

As such, Simpson et al. [79] classified modularization objectives into the following two categories:

- The **strategic** approach, which aims to develop modules that can be shared by many products [84] (e.g., [244, 245]); this aids in gaining benefits from economies of scale and scope [216]; and
- The **technical** approach, which develops modules from highly connected components (e.g., [83, 246]). This approach often implements matrices and aims to reduce the overall lead time [216]. Some studies have combined both (e.g., [216]).

Components and modules can have a multitude of functions that follow from product requirements [247]. Similar to the physical product, methods deconstruct requirements and functionalities into elements. Then, mapping between elements identifies inter-component interfaces and interactions.

Architectures and schemes handle the coordination of modules [248]. Product architectures simplify interactions by reducing component variation as well as unifying component, product, and process specifications [208]. Standardizing the interfaces between modules prevents module linkage and product assembly problems [132]. When interfaces remain constant, designers find it easier to design a quality part or module for all product variants. Furthermore, a low interface complexity enables the outsourcing of design tasks, reduces communication between designers, and facilitates faster design changes [199]. However, changes to interfaces might require changes to modules or even the design of multiple modules [33]. These considerations require attention early in product design to prevent costly redesigns [249].

Ma et al. [198] roughly divided modularization methodologies into the following two groups:

- The first group is the **matrix-based** approaches, which focus on the similarities and differences in the relationships between components [75]. However, they neglect functional relationships [75]. Matrix-based approaches enable modularization while considering the product architecture [75]. AlGeddawy et al. [83] incorporated assembly complexity and DFA into modular product design. Li et al. [250] implemented agglomerative hierarchical clustering using the modularity measure of Sinha et al. [251]. Daie and Li used hierarchical clustering with design structure matrices (DSMs) on product architectures, where similar matrices described overlapping relations between shared components. Asaga and Nishigaki [252] modularized using hierarchical clustering on a DSM only after a topology optimization and before a NN predicted materials.

- The second group is the **function-based** approaches, which focus on functional relations while mostly ignoring component-level properties [75]. For example, Stocker et al. [84] first determined module alternatives using multidomain matrices and then selected modules based on available packaging spaces. Alternatively, Ma et al. [75] used heuristic clustering to modularize while considering assembly cost using key components. Chaimae et al. [253] implemented a function-based approach from Stone et al. [67] that used several heuristics to find the product liaison and interfaces among modules. Ren et al. [90] applied a fuzzy clustering heuristic on similarity matrices to create assembly system modules. These often rule-based algorithms are less popular in the literature, possibly due to the increased amounts of effort and knowledge required to model them.

Salonitis [202] argued that DSMs – and their extension through multidomain matrices – are highly suitable for dealing with the complexity of product modularity due to their systematic step-by-step design of final modules. However, the results of DSM-based modularization can be highly arbitrary due to different rating schemes [254]. Furthermore, DSMs rely on data of high quality, which is not always a given, resulting in inadequate definitions of modules [254].

Modularization requires metrics to evaluate results and optimize accordingly. Metrics evaluate design states to identify modules. Relevant studies have listed many metrics that all take different viewpoints, including interaction strengths and density [255], product architecture [88], and customer preferences [65], among others [256–258]. Sinha and Suh [246] defined two types of modularity metrics: the degree of coupling and the similarity of modules. The coupling degree measures a module's independence, examples of which are found in [217, 259] as well as many DSM-based approaches. Module similarity mainly considers relationships between elements within the module. Regardless of the modularity metrics, roughly two design principles exist [85], which are described as follows:

- **Similarity maximization** aims to create one-to-one mappings between the functional and physical design in product families [200]. It enables one to standardize interfaces. Furthermore, it prevents "change-waves" into other modules when modules have architectural changes [33].
- The **minimization of dependency** between physical components reduces their interaction and enables variety through distinct module combinations [240]. Module independence encompasses the degree to which modules can function independently of one another and follows from functionality and standardization decisions [260]. Increasing module independence enables more flexibility in combining modules. Almost independent modules enable concurrent development in separate organizational domains [85].

Consequently, modularization involves the clustering of functional elements into modules that maximize similarity within the modules as well as the independence of modules from others [199]. However, Bonvoisin et al. [199] argued that the measurement of modularity is prone to low levels of standardization. As such, they argued that these metrics are hardly

usable [199]. Consequently, the authors stated that it is not possible to measure the promised advantages of modularity beforehand [199].

The optimization methods used for modularization are diverse. Jose and Tollenaere distinguished the following five categories [203]: clustering methods, graph and matrix partitioning methods, mathematical programming methods, AI methods, and genetic algorithms and heuristics. Practically speaking, any clustering method is applicable, but the implementation and use case will determine its effectiveness. However, achieving optimal modularity is not always beneficial. For example, commonalization and modularization practices intertwine. Fig. 2.10 demonstrates how a reduction in commonality may improve modularization.

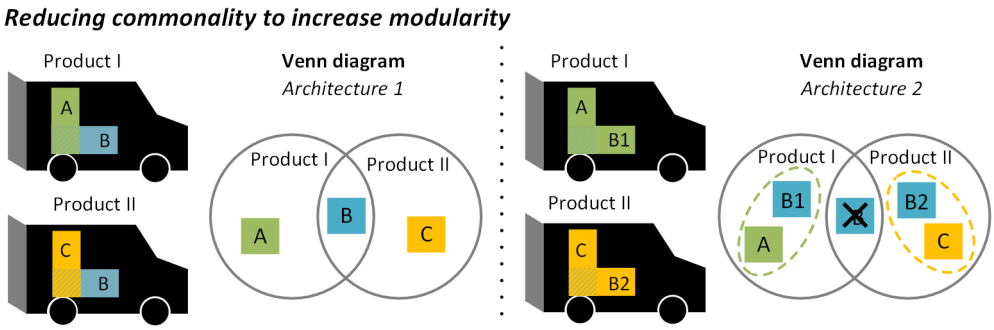


Fig. 2.10: Venn diagram of the shared groups of a family and the splitting of component B to enable modularization in both product variants, adapted from Baylis et al. [87].

Fig. 2.10 presents a Venn diagram [261] that visualizes overlapping items between sets, such as product variants, architectures, or joining element designs. Each section of the circles represents a unique set of product variants that share a particular set of components. The left-hand side of Fig. 2.10 indicates that component A only occurs in product I and component C only occurs in product II. Products I and II both share component B. Any shared component between multiple products may be 'copied' by neglecting the commonality; see the right-hand side of Fig. 2.10. Instances of the same parts in different product variants enable the modularization of that part into separate modules. This process works for each part until no common component is left, resulting in the modularization of every product independently.

However, maximizing commonality may create numerous small modules for many different unique sets of product variants [87]. It optimizes the reuse of every component, reducing the development effort for individual components. However, every individual component requires cumbersome management for determining which product variant it will be built into. This induces high complexity in managing the combination of all components. On the other hand, modularity considers the use combinations per product variant. The shared parts are larger and not all subsets of product variants share a unique set of components [87]. The variety of components is lower, but this is at the cost of parts with redundant functions increasing the development effort.

Discussion

Modularization can reduce complexity costs by creating independent groups of components that act as singular units and can be combined to create new product variants. Much research has been conducted into modularization, commonalization, and the design of product families and platforms [91]. Although the literature has presented many varied methodologies, Gauss et al. [91] argued that their practical application lags behind due to the limited knowledge of available methods. Additionally, locating these methods in product development processes is difficult [199].

DFA conflicts with modularity as minimizing the number of components creates highly integrated designs, thus reducing the number of configurable products [83]. Therefore, AlGeddawy et al. [83] stated that DFA and modular design need to be balanced. However, it is essential not to overlook other design metrics of a product family, such as cost, technical performance, complexity, sustainability, and adaptability [87]. In sum, modularization is a complex task that requires the balancing of many aspects of the process.

The literature often ignores geometrical boundaries in modular design [84]. Modules have physical connections to other modules. Methodologies that only consider functional levels may not create geometrically feasible results. Only Stocker et al. [84] were found to use a bin-packing algorithm to select appropriate modules instead of finding clusters with the highest interactions as in matrix-based approaches.

However, no modular design methodology was found that explicitly addresses assembly and joining elements. Methodologies attempt to reduce the number of relationships between modules and sometimes express assembly complexity or costs in relationships [83]. The literature lacks the modularization of component relationships and the reuse of the assembly information to join other modules. Interfaces between modules are the CRs for the design of joining elements. Module interfaces encounter great variety due to interchangeable modules. Changes to module interfaces affect other modules and their interfaces [81]. Hence, component-based clustering alone may potentially result in frequent changes of module interfaces and consequently of modules. This uncertainty of changes is the reason that sustainable modular design should be considered, which is detailed in the following subsection.

2.2.3 Sustainable modular design

Long-term maintenance is one of the foremost challenges in modularity [262]. Han et al. [80] performed a systematic literature review on product platform design. They argued that traditional product platforms could not adapt to dynamic market changes [80]. Additionally, the authors stated that "*[c]ommonality is limited to common modules defined by fixed physical standards*" [80]. Uncertainty categorizes modularization problems as being either inside or outside of the system [263]. Endogenous uncertainty concerns the risks, costs, and propagation of module changes [80], whereas exogenous uncertainty concerns the market, requirement, technology, legal, and regulation drivers [80]. Hence, for modular product platforms, it is vital to standardize interfaces, which can minimize information flows [229].

Sustainable modular product families manage the maintenance of modular products and the inclusion of newly designed variants [262]. Reusing implemented modules entails variety-

induced complexity. The arbitrary design of components and modules may not fit existing structures, thus reducing the controlled internal complexity [81]. Module reuse is a trade-off between profit and complexity costs [264]. Pakkanen et al. [215] reviewed the literature and determined key product development tactics for managing reuse and variety. They argued that design reuse also requires sustainable solutions often created by modularization methods [215]. As a result, modularization and design reuse affect each other in both ways [215].

Furthermore, product architectures need to have defined interfaces between modules [206]. These interfaces invoke design requirements for new modules. When changes to models are made, the interfaces create a trade-off between flexibility and costs. Changing the product architecture's interfaces affects other modules and interfaces as well [81]. There are three routes for new demands for products or features [264], which are described as follows:

- **Creating a new modular structure**: This is only done when the demand can only be implemented with high effort as it may create new product platforms [265].
- **Changing the modular structure**: Refinement of the modular structure involves the creation, editing, or deletion of modules or sections in the product architecture [70].
- **Reducing the demand**: This prevents unwanted development effort in the modular design, but one must consider the consequences of a lack of product offerings.

Many product family evaluation methodologies assume that all product variants come from the same platform [135]. This has led to the idea of maximizing the commonality between all of the variants of a single optimal product platform. However, as Fig. 2.11 indicates, product variants are developed sequentially and the platform constantly updates during development. As a result, the sequential design limits commonality maximization. Moreover, the benefits of commonalization prevent improvements to the flexibility of platform design.

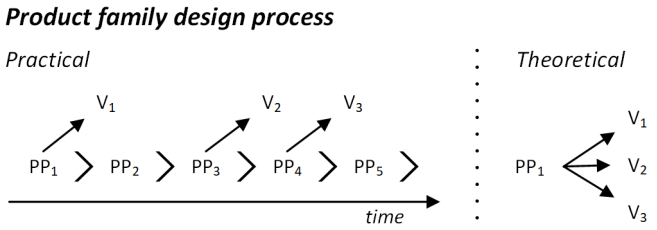


Fig. 2.11: A comparison between practical and theoretical product family design processes. PP_i represents a product platform while v_i represents a product variant; taken from Wong et al. [135].

Furthermore, design processes have not adapted adequately to the growing product variety, reducing transparency in the implications of variants [133]. Therefore, companies often rely on a few core products covering the majority of customer-specific requirements [58]. Here, production quality is inversely proportional to variability [266]. Designers need to select the adequate shareable components among other product variants from these core products, which is often performed by independent design teams [87].

Additionally, the sequential design process requires both commonalization and modularization to deliver sustainable results. These approaches bring robustness to

uncertainties, preventing rework in design. Sequential design can imply that designs for individual product variants may conflict with the profitability of overall product design [18]. As the overall design only gradually changes between sequentially released individual variants, the overall product design might lose quality [135].

Integrated documentation approaches (e.g., [31, 33, 35]) have extensively handled the combination of product and process data with product variety. They describe the boundary conditions under which to design joining elements through variety management throughout the entire product life cycle. Integrated documentation approaches are critical as, in a worst-case scenario, the documentation complexity may double for every new option or component [137].

Discussion

Continuous product development through the sequential release of product variants based on the differentiation of product platforms may affect the quality of modular designs. In uncertain environments, the modules and product architecture become prone to changes. Consequently, the interfaces between modules may change, resulting in additional uncertainties in joining element design.

Therefore, the standardization of interfaces is necessary for enabling joining element design in industries with high product variety. However, standardization is only a framework with which joining element design must comply. Together with modularization and commonalization, the reuse of designs, features, and modules supports the maintenance of the product architecture.

After discussing sustainable modular design, the following subsection summarizes the state of the art on modular product design.

2.2.4 Summary

Modular product design aims to manage product variety while creating affordable products for customers [8]. Various strategies can reduce variety-driven complexity at both the product and process levels [58], such as modularization, commonality, product configurations, and delayed differentiation. Mainly the former two strategies apply to joining element design in early product design [79].

Commonalization techniques increase the number of shared components between product variants [85–87]. Modular product design must be sustainable as uncontrolled module generation will increase complexity [81]. This is crucial for module interfaces and thus for joining element design due to the sequential design processes and component interchangeability. Commonalization on joining elements is not available in the literature, although Ukala and Sunmola [237] reduced the technological variety using condition-based statements. Still, this method requires much engineering knowledge and ignores crucial aspects of joining element design, such as geometry.

Today, researchers are highly interested in finding optimal modularity [91], such as product architectures [88] or product family design [87], by using DSMs. However,DSMs are incapable of representing individual connections and commonalities. They require multistep approaches to consider geometry (e.g., [84]).

Modularization methodologies can consider production requirements, such as assembly system design [90] and complexity [83], but they are more appropriate for early product design. These methodologies need to consider structural and interconnections, as in [88, 89, 267]. Current modularization methodologies only cluster components and do not recognize the potential of modularizing joining elements.

The results of commonalization and modularization must be sustainable and usable throughout product development. This is because product platforms update after releasing new variants [135]. Moreover, development leads to new insights, and modules will change [80]. Product development promotes standardized interfaces between modules, such that the modules can remain independent. This standardization can create an environment equipped for uncertainties in product development as well as a basis for reusing parts and modules.

Systematic commonalization and modularization require sophisticated algorithms. The following section discusses AI as a collection of techniques to solve design problems.

2.3 Artificial intelligence

This section addresses the state of the art in various AI fields. Subsection 2.3.1 briefly addresses evolutionary algorithms that – inspired by nature – can approximate optimal solutions. Next, subsection 2.3.2 introduces the various subfields within ML. Then, subsections 2.3.3 and 2.3.4 discuss state-of-the-art techniques in SML and UML, respectively. Furthermore, subsection 2.3.5 addresses data representations and their relationship with ML approaches. In addition, subsection 2.3.6 introduces the multimodality of data and their influence on ML. Lastly, subsection 2.3.7 provides an overview of reviews and state-of-the-art methodologies related to the design of joining elements.

AI refers to any method that aims to mimic human cognition [92]. These methods in the manufacturing industry can be divided into the following four fields [101]: rule-based reasoning, case-based reasoning, search and optimization, and ML. Sunnersjo [29] determined that the latter three fields enable the representation and processing of implicit knowledge in design automation. Hagemann and Stark [101] listed the following characteristics of these fields:

- **Rule-based reasoning** (RBR) is a method for creating designs that follow a strict path of predefined constraints. Data input is a step-by-step process and results in an output. Hence, RBR primarily handles knowledge in geometrical and mathematical (e.g., formulas) form [29].

  RBR has a low computational cost and is deterministic as it always outputs the same results when using the same input. Predictions are correct as long as the algorithm understands the input, which requires complete and error-free data. The development of the algorithm requires every variant of data input and all of the stakeholders' requirements to be considered for each use case. RBR is not generic, implying that changes to either data, requirements, or rules demand manual updates of the algorithm by design and programming experts.

RBR has been implemented for predicting joining locations (e.g., [50]) and screening for feasible joining technologies (e.g., [42]). Furthermore, RBR is typically used to optimize product architectures and as a basis for modularization (e.g., [89]) and commonalization (e.g., [85]).

- **Case-based reasoning** (CBR) finds similar previous problems and applies solutions to new use cases. CBR is equivalent to designers analyzing similar use cases and reapplying successful implementations. It is mainly suitable for comparative knowledge, such as in [29]. Additionally, CBR can handle tacit (e.g., intuition and skills), experimental (e.g., facts and relations), and heuristic (e.g., guidelines based on experience) knowledge.

  Furthermore, CBR requires an extensive database with knowledge representations of joints, including geometry, PMI, and design requirements. Hence, a large amount of memory is required to support the search. This method is a bottom-up approach and inductive, where the data lead to nondeterministic results. Once developed, CBR updates the database after each new design.

  Recent works using CBR include the knowledge-based frameworks in JTS methodologies (e.g., [40, 165]).

- **Search and optimization** (S&O) methods aim to find the best solution to a problem objectively. Search algorithms include permutation approaches that evaluate all possible solutions to find the best one. They cope best with knowledge in mathematical (e.g., expressions and relations between properties) or heuristic (e.g., guidelines and standards) form [29].

  Moreover, S&O methods require handcrafted boundary conditions and constraints to ensure convergence when solution spaces are large. The boundaries limit the high computational costs of these methods. More sophisticated algorithms, including heuristics or evolutionary algorithms, may approximate an optimal result. S&O algorithms require data that are complete and available to enable optimization with mathematical functions. The algorithms tend not to be deterministic, and approximated results are use-case-specific. Furthermore, S&O algorithms are complex and require much development effort to address the required detail and all potential use cases. The input data must also be complete and error-free.

  S&O methodologies can predict joining technologies (e.g., [42]), locations (e.g., [18]), and parameters (e.g., [168]).

- **ML** is a set of methods that aim to find patterns in datasets for predicting outcomes on unseen data. Deep learning, a subfield of ML, is currently the benchmark technique due to its better performance over vast numbers of applications and research studies [100]. ML best exploits tacit knowledge [29]. This information is undefined and embedded in the minds of designers as common sense and experience [29].

  ML methods require vast amounts of data to handle complex tasks with quality. The training process is computationally expensive, especially when datasets have high dimensionality, such as geometry. These methods attempt to generalize data and can adapt continuously to trends from new data samples. The developmental effort required

is considerable, and training may be sensitive to small changes in the settings of parameters. Furthermore, ML results are not deterministic. Each trained model can be different due to random initialization (RI). The methods are also inductive as the data define the parameters of the model. These data-driven methods create generic results in cases where the training data cover all use cases.

Lastly, ML has seen implementations for joining element predictions (e.g., [105, 169]). It is more prevalent in the manufacturing industry, such as in process prediction (e.g., [166]), sheet metal forming (e.g., [104]), or topology optimization (e.g., [47]). It includes predicting the nugget diameter of spot welds (e.g., [169]) and the mutual spot weld distance (e.g., [105]).

The following subsections address several AI fields into more detail, starting with evolutionary algorithms.

2.3.1 Evolutionary algorithms

The S&O field includes evolutionary algorithms that, inspired by nature, enable one to approximate optimal solutions using randomized searching. Many optimization problems are convex, implying that there exists a computable global optimum. However, many tasks may have numerous optima, with no, a non-unique, or an unstable solution, considering noisy or missing data [268]. Evolutionary and swarm algorithms can search for solutions in vast solution spaces without being trapped in local optima, an issue typically experienced in ML. Nakane et al. [268] identified three main challenges during the implementation of evolutionary algorithms: (1) the selection of appropriate algorithms with hyperparameter tuning, (2) a time-consuming optimization process, and (3) finding solutions on a Pareto front due to multiobjective optimization.

Contrary to, for example, Swarm or Monte Carlo and Simulated Annealing algorithms, evolutionary algorithms take the most successful solutions, update them, and then evaluate whether the solutions have improved. In the next iteration, the algorithm again takes the best solutions, makes mutations, and evaluates whether the performance has improved. Slowik et al. [269] distinguished the following five types of evolutionary algorithms: genetic algorithms, genetic programming, differential evolution, evolution strategies, and evolutionary programming. These algorithms have many different varieties and enable further customization for a wide variety of applications, such as hyperparameter tuning of NNs (e.g., [270]), machining (e.g., [271]), land-use allocation (e.g., [272, 273]), JTS (e.g., [157]), noise-vibration-harshness (e.g., [47]), and topology optimization (e.g., [179]).

Slowik and Kwasnicka [269] discussed evolutionary algorithms in state-of-the-art applications. Moreover, Nakane et al. [268] surveyed the application of evolutionary and swarm optimization in computer vision. Recently, Atali et al. [274] reviewed meta-heuristics through swarm intelligence algorithms and the lens of chaos theory.

Another AI field besides evolutionary algorithms is ML, which is discussed in the following subsection.

2.3.2 Machine Learning

This section discusses the state-of-the-art in ML. This dissertation presumes basic knowledge of ML. Therefore, this section briefly introduces relevant nomenclature and concepts before discussing recent methodologies and challenges. The literature has inexhaustively described fundamentals, expressions, and in-depth explanations of terms and concepts. As a further reference material, this study recommends the book of Goodfellow et al. [275].

Contrary to algorithms developed by software engineers, ML acquires knowledge through its ability to extract patterns from historical data [275]. Software developers only set the boundary conditions. The algorithm itself needs to learn how to perform a given task. One can roughly distinguish the following four types of ML [275]: supervised, unsupervised, semi-supervised, and reinforcement learning.

- **Supervised ML** (SML) approximates an unknown function between many input and target samples. It analyzes training data, creates an approximation function, and applies it to new unseen samples. Every training sample has an input with a correctly labeled target.
- **Unsupervised ML** (UML) determines properties and structures within datasets that do not contain output values. Similarities are identified between samples and the data are organized accordingly.
- **Semi-supervised ML** combines the abovementioned approaches to create better results than either alone. Often it uses small datasets with target values (SML) and large datasets without (UML).
- **Reinforcement learning** (RL) uses feedback loops with a reward system that evaluates the actions of an agent in an environment. It does not have a fixed dataset to learn but instead applies an unlimited number of trial-and-error experiments to extract knowledge and then apply it.

ML algorithms cannot extrapolate knowledge, and hence, depend on the supplied information. Missing, incomplete, and false data create biases in the system, potentially leading to severe problems [276]. Features are single pieces of information within each data sample. It is crucial to create a suitable feature set for a specific task [275]. The selection, creation, and extraction of features are the most difficult steps for particular learning tasks. The quality of ML methods directly correlates with the dataset size and dimensionality. It makes the training of models computationally costly. However, ML can find generic solutions with much less expert engineering of models. This work will address SML and UML as concepts in the implementations. The following subsections provide background information on the workings of these methods.

2.3.3 Supervised machine learning

SML extracts a mapping between input and output states [166] using one of two tasks, namely classification or regression:

- **Classification** categorizes target variables into binary (only two classes present), multiclass (more than two classes present), or multilabel (more than one correct label per sample) [275]. For example, classification would predict whether an image contains either a cat or a dog. Models create conditional probabilities for each class given the input and model parameters. The final output prediction is the class with the highest probability. Zhang et al. [277] reviewed the state-of-the-art of classification algorithms and found that decision trees typically yield the highest accuracy.
- **Regression** considers continuous target variables and returns a conditional expectation given the input and model parameters [275]. Hence, the task is similar to classification, only with a different output format, such as house price prediction based on the number of rooms and surface size. Supervised tasks have little influence on the required model architectures. Well-known SML methods are decision trees and NNs.

Various methods within SML can perform these tasks. The following paragraphs will discuss decision trees and neural networks as popular and relevant methods in this study.

Decision trees This SML technique refers to acyclic graphs that represent branches for attributes at all internal nodes [278]. Each node holds a decision of which answers represents branches to the next nodes. Every branch holds a feature value, and every leaf node describes an output class. Fig. 2.12 presents an example of a decision tree on the left-hand side. The right-hand side depicts the corresponding distribution of labels from the decision tree.

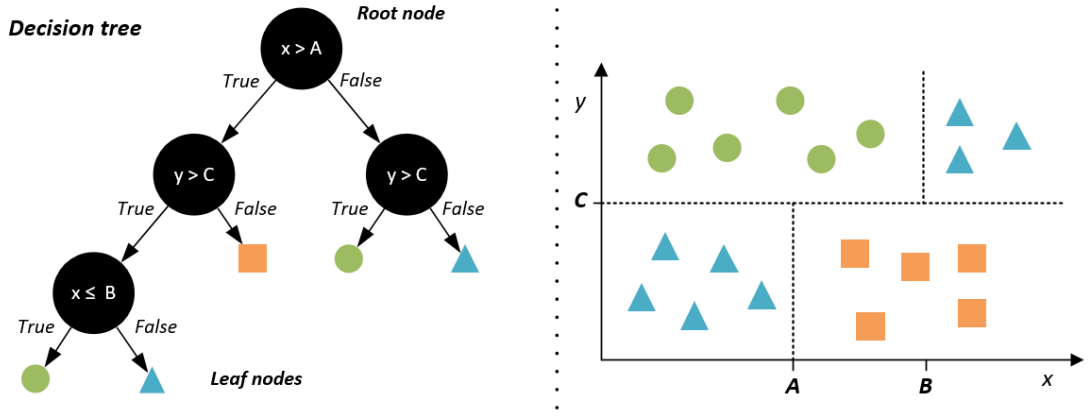


Fig. 2.12: Example of a decision tree and a distribution of labels in 2D space.

Decision trees can perform regression and classification tasks. The path from the root to the leaf represents a sequence of decisions [279]. Decision trees are simple to understand, require little preprocessing, and can handle discrete and continuous variables [279]. They automatically select variables, are insensitive to the monotone transformation of inputs, are robust to outliers, and scale well into large datasets [279]. However, their models lack general robustness and tend to overfit data [279]. Small changes to the training dataset may result in completely different optimal trees. Changes in the root node (or upper internal nodes) affect the subsequent tree

due to their hierarchical structure. Tree ensembles aggregate predictions of multiple models to make up for individual disadvantages [280]. Benchmark implementations are bagging and boosting [281].

- **Bagging** generates multiple versions of the same prediction model to create one aggregated model [282]. Every version trains on a different subset of the training data. **Random forest** methodologies combine many decision trees using bagging and train on subsets of features [283]. Slight changes to training data will only affect individual trees and not the overall tree ensemble, thus reducing the robustness problem [280]. Bagging reduces the variance for all trees by aggregating many estimators with individual higher variances into one.
- **Boosting** iteratively adds and trains weak learners that all aim to learn something new and make up for their predecessor's mistakes [279]. Parameters and coefficients are static during the sequential addition of models [284]. **XGBoost** is a popular algorithm that uses a specific modification of gradient boosting trees to learn from weak learners' residuals [285].

Neural networks

NNs are models that derive their name and concept from human brains. Neurons take several inputs, aggregate them, and fire their output to the subsequent neurons. The learning task updates parameters (weights) to improve prediction on the same inputs. An activation function determines the fire intensity (output values) of neurons. Historically Sigmoid functions [275] and currently Rectified Linear Units (ReLU) [286] are highly successful. NNs organize neurons in parallel in multiple layers. Fig. 2.13 presents an exemplary overview of a neural network and neurons.

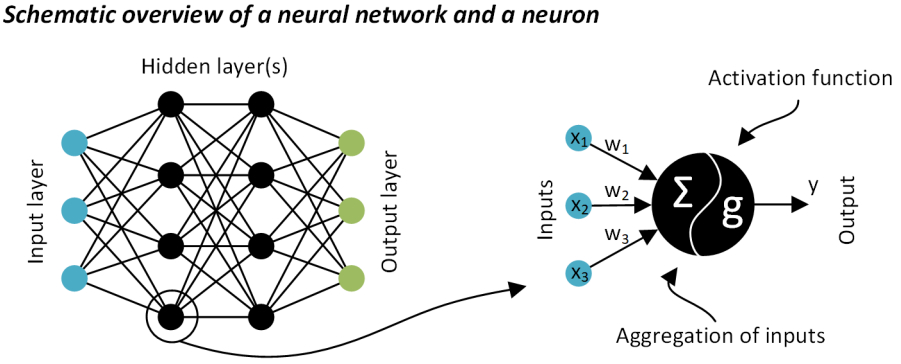


Fig. 2.13: Overview and nomenclature of a neural network and a neuron.

Feedforward NNs process from the input layer sequentially through each layer to the output layer. Hidden layers are those neurons that undergo learning and have states that are unknown until evaluation. Deep learning involves huge NN architectures that enable the learning of complex tasks but at the cost of training and quality difficulties.

Autoencoders are a specific type of NN. They have a purposefully implemented bottleneck and are designed for tasks such as dimensionality reduction (DR) and feature discovery [279, 287]. Autoencoders aim to mimic the input in the output layer, but through an intermediary lower-dimensional representation [288]. First, the encoder creates latent factors from samples, and then, the decoder reconstructs samples from encoded data. The initial layers have a declining number of neurons and force the model to extract essential information. Fig. 2.14 presents a small example of an autoencoder that reconstructs the image of the number two from a lower dimensional representation in the bottleneck of the architecture.

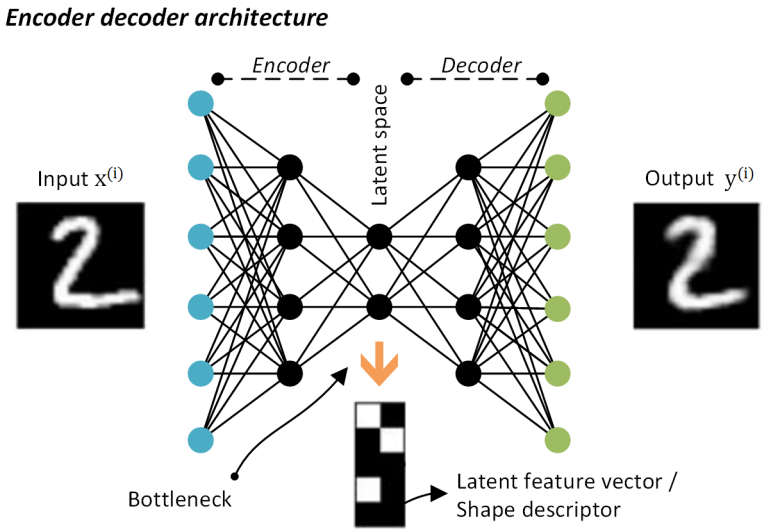


Fig. 2.14: Overview and nomenclature of an autoencoder.

Autoencoders primarily focus on maximizing the variance of the data in the latent space. A drawback is that this results in a suboptimal mapping of the local data structure. Furthermore, it requires a NN to be trained, which has a high computational cost for high-dimensional data. Interestingly, instead of only mimicking input data to create latent factors, the architecture of autoencoders suits generative tasks, such as object detection or segmentation. Image segmentation aims to extract meaningful information by identifying conceptually similar sets of pixels [289]. Oliveria [290] evaluated the efficiency and robustness of encoder-decoder (EncDec) methods for segmentation tasks. Complex deep learning approaches are successful through EncDec architectures, as they enable the number of parameters to be contained [291].

Moreover, convolutional NNs (CNNs), introduced by LeCun et al. [292], reduce the number of trainable parameters by reusing weights between layers for datasets with grid-like topologies. Fig. 2.15 depicts the structure and behavior of a CNN using the example of classifying a traffic sign.

The structure of CNNs is invariant to translational and rotational transformations [279]. Moreover, it also prevents a rapid increase in computational cost for large networks [279]. Instead of the multiple layers of neurons in regular NNs, CNNs implement convolutional layers

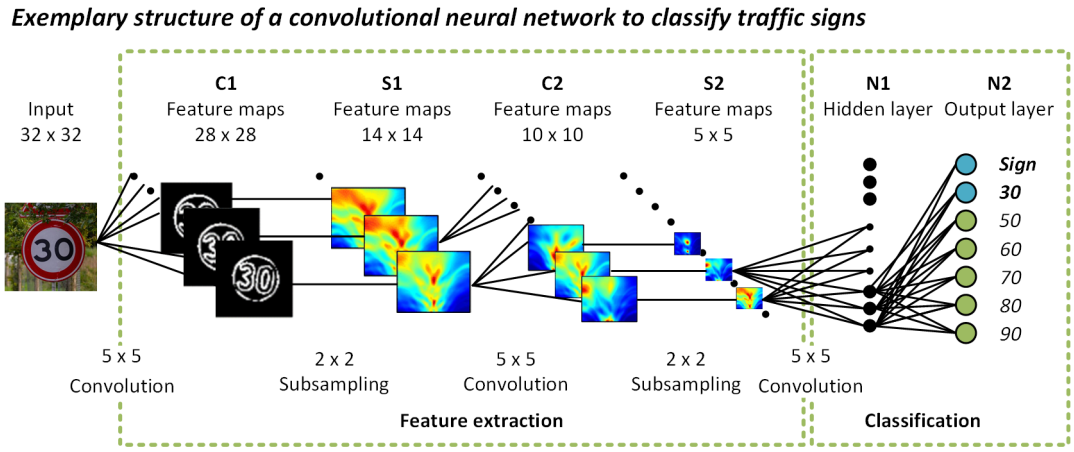


Fig. 2.15: The traffic-sign input image is convoluted by four 5×5 kernels into the first set of feature maps (C1). Then, they are subsampled using max pooling (S1). Another sequence of 5×5 kernels (C2) and max pooling (S2) extract the features for a final fully connected layer to classify the sign; not all connections were drawn for the sake of readability; adapted from Peemen [293].

that perform matrix multiplications and output feature maps [275]. Kernels are the matrices of trainable parameters within the convolutions of each layer.

Gradient descent is a conventional method for training NNs [275]. The NNs optimize an objective function. In each iteration, it updates the model's parameters, but in the opposite direction of the objective function's gradient [294]. The learning rate tunes the magnitude during parameter updates. Mini-batch gradient descent uses a small set of samples to calculate the loss (i.e., cost or model error), balancing the computational cost and optimal direction. This optimal direction varies as a local one-step update might not be in the direction of the global minimum. The gradient descent optimizer Adaptive moment estimation (Adam) [295] is a benchmark and balances the learning rate and direction automatically [275].

Typically, confusion matrices evaluate a model's performance on classification tasks. They count the occurrences of correctly predicted presences or absences of each class. These counts enable calculate popular metrics, such as the accuracy and F1-score. These are intuitive and commonly used metrics [296]. Perfect scores have a value of 1, whereas the worst scores have a value of 0. However, these metrics do not consider the distribution of false classifications nor the marginal distribution [297]. Cohen's kappa indicates a model's superiority compared with a random classifier based on class frequencies [297]. Cohen's kappa (κ) considers the chance of accidentally correct predictions. However, the performance metric may exhibit undesirable behavior and its validity has been debated in the literature [297].

Furthermore, the Matthews correlation coefficient (MCC) describes the relationship between observed and predicted binary classifications [298]. MCC evaluates binary classifications as more meaningful than the F1-score and accuracy as it considers their frequencies in the confusion matrix [298]. A perfect prediction has a value of 1, whereas a random prediction has a value of 0. A value of -1 describes a contradiction between the prediction and target. Furthermore, the study of Chicco and Jurman [298] details the

aforementioned performance metrics, presents their equations, and addresses their relationship to the confusion matrix.

As previously mentioned, there are several types of ML. After discussing SML, the following subsection addresses UML.

2.3.4 Unsupervised learning

UML methods can discover knowledge and, in contrast to SML, have no target values [279]. Roughly, they consist of three main applications: (1) clustering, which aims to distinguish groups in a dataset; (2) DR, which aims to densify information from high-dimensional data; and (3) generative adversarial networks (GANs), which can generate new data samples from noise.

Clustering

Clustering aims to distinguish groups in a dataset based on similarity. This approach includes many families of algorithms [299]. In supervised clustering, the number of clusters is known before the execution of the algorithm, whereas it is not known in unsupervised clustering. The dataset determines the appropriateness of clustering methods by considering noise, incompleteness, and samples [300]. Cluster analysis has been used in optimization to identify promising design regions [301] and eliminate near-duplicate designs [302, 303]. Popular clustering algorithms include the following [300]:

- **K-means** [304]: This is a widely used partitional approach [305] with a low computational cost aimed at tasks such as data segmentation [306] and anomaly detection [307]. K-means algorithms iterate between assigning data points to the closest cluster center and recalculating cluster centers based on assigned data points. The main limitation is the definition of the number of clusters before clustering, which requires iterative optimization. Furthermore, k-means has difficulties with nonconvex distributions [308], highly different cluster sizes [309], and high-dimensional data [300].
- **Expectation-maximization** (EM): This is a well-known model-based algorithm that clusters by modeling multivariate normal distributions of every class. The maximum likelihood method finds the parameters of every distribution. EM is applicable for incomplete datasets [310, 311]; however, it relies strongly on initial clusters [312] and might fail to recognize small clusters [313].

The literature has described many similar measurements for determining the performance of clustering with known labels [314], including the Jaccard [315], Adjusted Rand [316], and Fowlkes Mallows [317] indices. The Jaccard index (also known as intersection over union [IoU]) is the ratio between the intersection $A \cap B$ and union $A \cup B$ of two sample sets (A and B); see Eq. 2.7. In addition to clustering, the Jaccard index is often used to evaluate object detection tasks by taking the overlapping area of the ground truth and predicted bounding box of objects (e.g., [318, 319]). Notably, the Jaccard index is also the current standard metric for

segmentation approaches as it considers both false alarms (FP) and missed values (FN) class-wise [290]. A Jaccard index J of 1 expresses perfect similarity with precisely overlapping sets and 0 expresses complete dissimilarity.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}; \qquad J = \frac{TP}{FP + TP + FN} \tag{2.7}$$

Clustering with unknown labels has a few popular performance metrics, including the Silhouette method [320] and the Elbow method [321]. The Silhouette method [320] validates consistency within clusters by measuring the similarity between its cluster and dissimilarity to others. It aggregates the scores of all individual objects ranging from -1 and $+1$. High scores indicate high-quality clustering. The popular Elbow method [322] enables one to determine the optimal number of clusters [321]. It relates the percentage of explained variance in clusters to the number of clusters. The Elbow method states that the optimal number of clusters is where added information drops significantly after an extra cluster is added. Plotting the amount of variance over the number of clusters will indicate this point as an angle in an otherwise gradually reducing curve.

Dimensionality reduction

This method creates compact, information-dense representations of high-dimensional data. Ideally, it transforms data to contain its internal dimensionality, namely the minimal number of required parameters to consider the observed properties of data [323]. Latent factors (also observed in autoencoder architectures) are the number of dimensions in the low dimensional representations that describe the main characteristics of the data. The curse of dimensionality refers to an unrestricted rise in space compared with the available data. It creates statistical significance and performance problems [323]. DR enables the classification, visualization, and compression of high-dimensional data.

Ayesha et al. [324] conducted an extensive overview and comparative study on DR methods. Cunningham and Ghahramani [325] surveyed linear DR and Ting and Jordan [326] discussed nonlinear DR in detail. Linear techniques have lower computational costs but more restricted data modeling quality compared with nonlinear techniques [324]. Ayesha et al. identified four main challenges in applying DR [324]: (1) method selection, which depends highly on the available data; (2) identification of redundant features without affecting performance; (3) selection of proper dimensions for visualization; and (4) interrelatedness of most high-dimensional features. Consequently, Nguyen and Holmes [323] argued that DR methods are often misused or misinterpreted. In response, they listed 10 guidelines for effective data reduction to support the proper implementation of DR, interpretation of outputs, and communication of results. Ayesha et al. [324] found that the following DR techniques support structured data: principal component analysis (PCA), Singular Value Decomposition, and t-Distributed Stochastic Neighbor Embedding. The properties of PCA are addressed further herein due to their relevance to the present study's implementation.

PCA is a linear transformation method for finding lower-dimensional space to maximize the variance of a dataset [327]. The low-dimensional spaces should thus hold the most information [328]. PCA is nonparametric, implying that the number of parameters to estimate may change

during execution and tends to grow proportionally to the amount of training data [279]. PCA has uncorrelated dimensions; however, they are challenging to interpret, and information loss occurs through a lack of complex polynomial relations between features [329]. The objective of PCA is to preserve variance.

Generative adversarial networks

GANs are a very different approach to clustering and DR in UML. Recently, GANs have been used in computer vision tasks to generate new images. GANs, developed by Goodfellow et al. [330], have high potential as generative models due to their visual quality and diversity [331]. They combine NNs, typically a generator and a discriminator, which challenge one another during training [330]. However, GANs are difficult to train, and only a few architectures have proven to be successful. Fig. 2.16 displays the general architecture of a GAN.

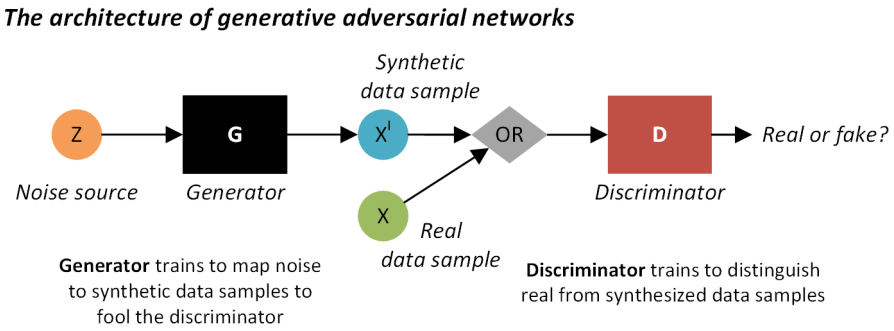


Fig. 2.16: Visualizes the generative adversarial network with two models: the discriminator (D) and the generator (G); taken from Creswell et al. [332].

During the training phase, a GAN creates images from noise. However, during testing and usage, GANs receive "regular" input images instead of noise. A typical GAN has the following two parts:

- The **generator** (G in Fig. 2.16), which transforms an input into an output image; the input image may be noise or a ground-truth training image; and
- The **discriminator** (D in Fig. 2.16), which assesses whether the generated image was synthesized from noise.

The generator creates images from noise, and the discriminator judges their validity by using ground truth images as a reference; see Fig. 2.16. This is an unsupervised technique as the data do not have labels, nor does training generalize data samples into a model. However, it uses a supervised loss in training. The discriminator (adversary) calculates distances between targets and generated images [330]. GANs often implement EncDec architectures (e.g., Stargan_v2 [331]) to create latent space representations of data. By interchanging decoders, for example, one can reconstruct a feature vector from an unseen encoding and create a new image. However, the generator and discriminator can be implemented by any differentiable learning model that maps data from one space to another [332].

For further reference, Creswell et al. [332] presented an overview and discussed training methods, network architectures, and applications, including classification, regression, image synthesis, image-to-image, coloring translation, and super-resolution [332]. Furthermore, Kusiak [333] summarized applications of GANs in manufacturing and observed their use in image synthesis, engineering design, surface inspection, condition monitoring, fault diagnosis, service robotics, energy, business, and security.

A highly researched field of GANs is image-to-image translation, which creates mappings between visual domains [334]. Domains are sets of images that belong to a visually distinctive category, such as cats, dogs, or faces. A style refers to the unique appearance of an image. For example, the dog domain may include three different images of an Australian shepherd, of which all three have different styles. Wang et al. [335] studied the state of the art and found CycleGAN [336], UNIT [337], MUNIT [338], DRIT [339], TransGaGa [340], and RelGAN [341] to be the most important networks for image-to-image translation. Choi et al. [331] proposed StarGAN_v2, which combines the benefits of its predecessors and uses a single generator to predict for multiple domains. The authors distinguished between two image synthesis approaches, namely reference- and latent-guided synthesis, which are visualized in Fig. 2.17:

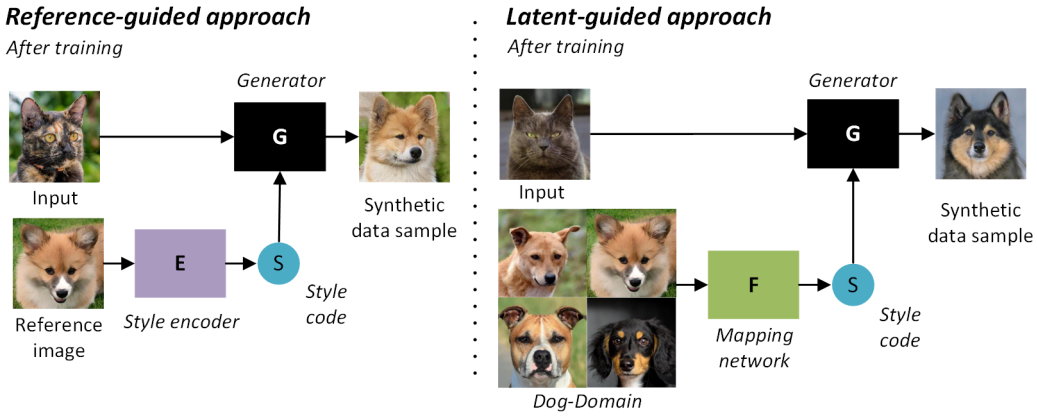


Fig. 2.17: The use of GANs for synthesizing new images. The figure visualizes the reference- and latent-guided approaches from Choi et al. [331].

Based on Fig. 2.17, the following list describes both approaches:

- **Reference-guided synthesis** requires a chosen style, namely the reference, for image generation to be provided along with an input image. The generator modifies the input image according to the style of the reference image [331], which describes a typical image-to-image translation process [334, 342]. Further examples of works that have used reference-guided synthesis include the studies of Chang et al. [343], Cho et al. [344], Ma et al. [345], and Park et al. [346].
- **Latent-guided synthesis** uses styles of domains [331] as latent feature vectors and applies them to input images. Hence, a GAN does not apply the style of one specific image onto an input image but rather the features of a domain of styles. Furthermore,

latent-guided synthesis can use random sampling to select a style within the domain to generate an image [331]. Examples include the studies of Liu et al. [347] and Yu et al. [348].

The framework of StarGAN_v2 has, besides the generator and discriminator, two additional modules [331]:

- **Style encoder** (E in Fig. 2.17): This module extracts the style code from an image and its corresponding domain [331]. It can create different style codes using different reference images. It enables the generator to synthesize an output image with the reference image's style.
- **Mapping network** (F in Fig. 2.17): This module generates a style code for the latent-guided approach. It requires a latent code and a domain [331]. The mapping network can create multiple style codes as it samples the latent vector and domain randomly [331].

The previous subsections discussed various subfields of AI. However, their use and application depend on the data samples they get. Hence, the following subsection discusses the properties of various data representations for geometry and PMI.

2.3.5 Data representation

Joining element data may come in a wide variety of formats [19], including native CAD system formats as well as neutral data exchange formats. Neutral data formats include lists (e.g., Microsoft Excel) or XML files (e.g., xMCF [349]). Joining element information has a geometric component that describes joining locations as well as a nongeometric component that includes PMI. The latter is structured data generally represented in tabular form. Here, features correspond to columns, and each data sample represents a row entry. Nongeometric data describes PMI assigned to parts, such as the weight, material, or joining technology. Moreover, geometry is less trivial to represent for ML. Data representations can be grouped into Euclidean and non-Euclidean structures [350], which are described as follows:

- **Euclidean** structures have grid-like properties, such as descriptors, voxels, and multiviews [350]. They consistently describe objects using the same parameters (global parametrization) [350]. Additionally, Euclidean structures have common systems of coordinates [350].
- **Non-Euclidean** data representations contain randomness and lack structure [350], such as point clouds, meshes, and graphs.

Fig. 2.18 depicts representations with Euclidean structures and applicable deep learning processes.

Recently, several research groups have surveyed work on 3D geometry and deep learning including Ahmed et al. [350] (which was continued by Gezawa et al. [351] and Xiao et al. [352]). They have discussed representations, datasets, architectures, and limitations in great depth. The survey by Xiao et al. [352] found that research on geometry learning examines

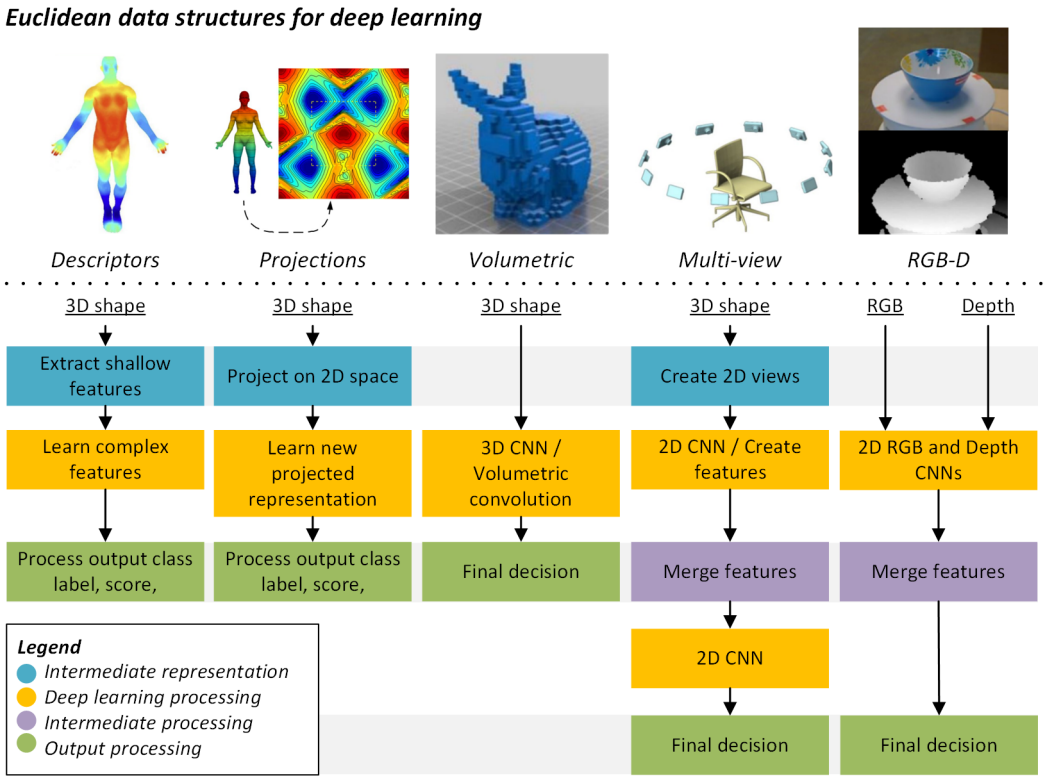


Fig. 2.18: Various popular representations of 3D data and their deep learning models, adapted from Ahmed et al. [350].

either reducing computation and memory demands or increasing detail and structure. They also looked into shape analysis, where methods usually extract latent factors for applications such as the classification, retrieval, and segmentation of shapes [353]. Representations are highly particular to applications but rely on their ability to retrieve the essence of shapes. Shape classification determines the object class based on a 3D shape, for which many benchmarks are available [353]. Fig. 2.18 presents the popularity and performance of voxel-, point-cloud-, and multiview-based methods. Shape retrieval aims to find the most similar shapes to an input 3D shape in a database, while shape segmentation aims to identify sections from 3D objects and requires a spatial understanding of the shape [353]. Garcia-Garcia et al. [289] reviewed semantic segmentation using deep learning techniques. Furthermore, Shen et al. [354] surveyed object classification on 2D and 3D datasets and argued that multiview and volumetric representations fit CNN-based architectures better compared with nonstructured point clouds. The popular data representations are described in more detail as follows:

- **Shape descriptors** express key geometric or topological properties at abstract levels [355]. They include familiar mechanical engineering terms, such as volume, Zernike-moments (or moments of inertia), latent feature vectors, and diffusion-based approaches. Simple shape signatures enable processing with low computational cost as well as fast similarity evaluations and shape retrieval. The used shape descriptor and its

implementation affect the nature and meaning of its shape signatures [350]. However, descriptors lack specific geometrical information, which prevents applications in classical computer-vision tasks, such as image segmentation and object detection. Kazmi et al. [356] published a comprehensive survey on 2D and 3D shape descriptors. Recent work often combines descriptors with learning-based models to extract features on various hierarchical levels [350].

- **Multiview** representations describe 3D geometry using multiple 2D snapshots obtained from different point views [357]. They function similarly to the 3D perception of humans, which is a probable reason for its success [358]. This method can take advantage of well-researched 2D computer vision approaches to classify or detect 3D objects [354]. Multiview approaches aim to learn functions that model each view separately before aggregating all views together into 3D shapes. Balancing the number of views is essential; having too few views might not capture all properties, and having too many may result in unnecessary computational cost [350]. Moreover, models with multiview data produce better results for some applications than for voxel-based datasets [358]. Furthermore, multiview representations are uniform and have a Euclidean structure. However, recreating coordinates in a global coordinate system is problematic. Seeland et al. [359] researched combining strategies of multiview images for classification.
- **Voxels** are volumetric representations that describe the distribution of geometry in regular grids with defined sizes and shapes [350]. Voxels have recently gained attention in the ML field in classification and segmentation tasks, such as in the work of Maturana et al. [360] and Meng et al. [361]. They are the 3D equivalents of pixels in 2D images. The representation is straightforward, creates high semantic descriptions of geometry, and can carry additional arbitrary data. However, voxels increase the dimensionality of the data cubically. Hence, detailed representations have high computational cost [362]. Voxels also store information on empty grid cells and create sparse occupation matrices. Data representation becomes inefficient when large sections of grids are empty [350]. Hence, Hane et al. [363] proposed considering only surface voxels, reducing the number of parameters, and enabling smaller voxel sizes. The architectures of various successful 2D NNs convert easily into voxel form [352].

Discussion

The latest works on geometrical deep learning methods have aimed to reduce computational and memory costs as well as to increase detail and structure [352]. Specific tasks require tailored 3D representations of geometry [352]. Gezawa et al. [351] created a summary that listed the mode of acquisition, key features, advantages, and limitations of each 3D data representation. Concerning the aforementioned Euclidean representations, they argued that

- Shape descriptors are ideal for shape analysis tasks; however, they may not learn enough discriminating features;
- Multiviews are effective and simple but cannot fully represent 3D geometry; 2D approaches may occlude important features that are only available in 3D; and

- Voxels have a structured volumetric description of a shape but are less suited for complicated tasks due to information loss bounded by the level of detail.

Initially, shape descriptors enable the analysis of 3D geometry using traditional 2D approaches. They are easy to implement but cannot discriminate unique 3D features of shapes [350]. However, shape descriptors can transform data into 2D formats using benchmark image processing approaches. Alternatively, Sinha et al. [364] presented a projection-based method for transforming 3D features into a 2D structure. Still, 2D formats lack detailed geometry descriptions. Hence, adding a depth layer to images can improve performance greatly [365].

Recently, complete 3D geometry methods have gained more attention. Volumetric representations are appropriate for analyzing rigid data with minimal deformations [350]. Volumetric methods use CNNs with 3D kernels and perform significantly better than other approaches [350]. However, their main drawbacks are a high computational cost and inefficient data representation [362]. Multiview approaches perform better on object recognition tasks than full 3D-based methods, possibly due to the multitude of views [350] and the benchmark in 2D architectures. This is also a catch as performance relies on the number of views. Models may fail to recognize intrinsic geometrical properties. One study was unclear on which the better representation is and concluded that it depends on the task and resources [350]. The availability of high calculation capacity enables the use of complex models and thus volumetric approaches.

However, many problems can not be described within one representation of data. For example, not all problems are solely geometric. Hence, models need to cope with a multitude of representations. The following subsection discusses various methodologies for their integration.

2.3.6 Multimodality

Data that describe joining elements is multimodal; that is, it consists of multiple representation types. For example, geometry describes the location to join, while PMI describes the materials, thicknesses, and joining technology. Hence, ML needs to consider various types of data simultaneously to make informed predictions. Multimodal ML (MMML) integrates information from multiple representations to increase prediction performance compared with unimodal approaches [366]. Although the field has become more significant and has high potential, it faces the following five main challenges [367]: representation, translation, alignment, fusion, and co-learning. The challenges that are relevant to joining element design are described as follows:

- The **representation** of data can be distinguished into joint or coordinated representation. Joint representations combine unimodal data in a representation space [367]. Coordinated representations process unimodal data separately but under a set of similarity considerations. Generally, the choice between either depends on the availability of all modalities in the testing phase.
- **Fusion** integrates information from multiple modalities to increase the prediction performance. Two distinctive categories distinguish the explicit dependency of ML

methods for fusion. Model-agnostic approaches may use early, late, or hybrid integration of modalities [368], whereas model-based approaches use specific methods for MMML.

Guo et al. [353] presented an overview of multimodal representation learning that focused on early fusion. This is the first and most researched subfield in MMML and enables more robust predictions, captures complementary information, and remains functioning when a modality is absent [367]. However, this subfield suffers from the integration of multimodalities, supplementary information, and temporal dependency [367]. In contrast to early fusion, late fusion aggregates results of multiple model outputs using ensemble ML with averaging or voting schemes, such as in the work of Morvant et al. [369]. Guo et al. [353] introduced three categories in early fusion, which are also depicted in Fig. 2.19:

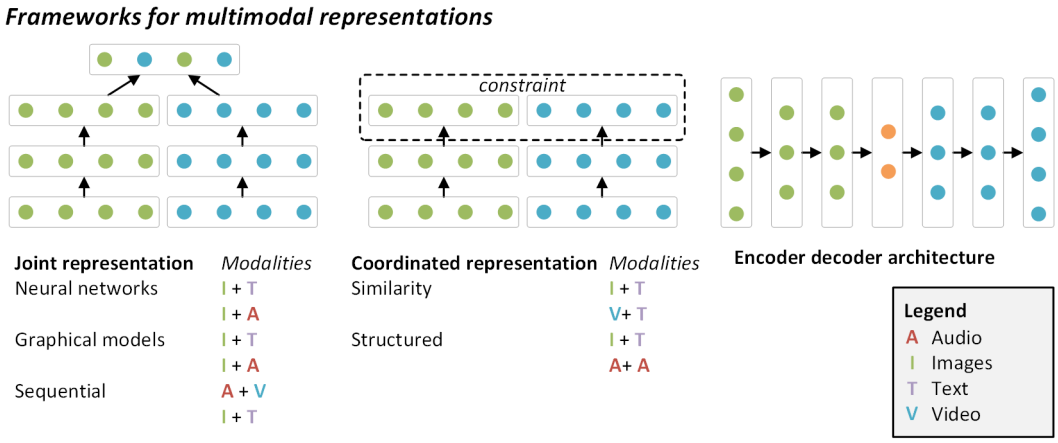


Fig. 2.19: Three types of frameworks about deep multimodal representation: Joint representations, coordinated representations, and encoder-decoders, adapted from Guo et al. [353]. The multimodal methodologies for joint and coordinated representations are taken from Baltrusaitis et al. [367].

- **Joint representations**, which aim to learn a shared semantic subspace;
- **Coordinated representations**, which have separated architectures but shared constraints between each modality; and
- **EncDec frameworks**, which translate one modality into another, keeping their semantics consistent.

Studies may use the terms joint representation and early fusion interchangeably [353]. Both terms bring multiple unimodal datasets to a shared representational space. Concatenation is the most popular method for performing early fusion, and is also known as additive methods [353]. Examples include the studies of Liu et al. [370], Walsman et al. [371], Petscharnig et al. [372], and Sindagi et al. [373]. A multiplication approach takes the product of the properties, as in the work of Zadeh et al. [374] for fusing audio and video, or in the study of Fukui et al. [375] for fusing visual and text properties through bilinear pooling.

Pelka et al. [376] presented branding, which is an approach for fusing encoded text properties and other data onto images. Fig. 2.20 depicts the branding approach using the example of radiograph data samples.

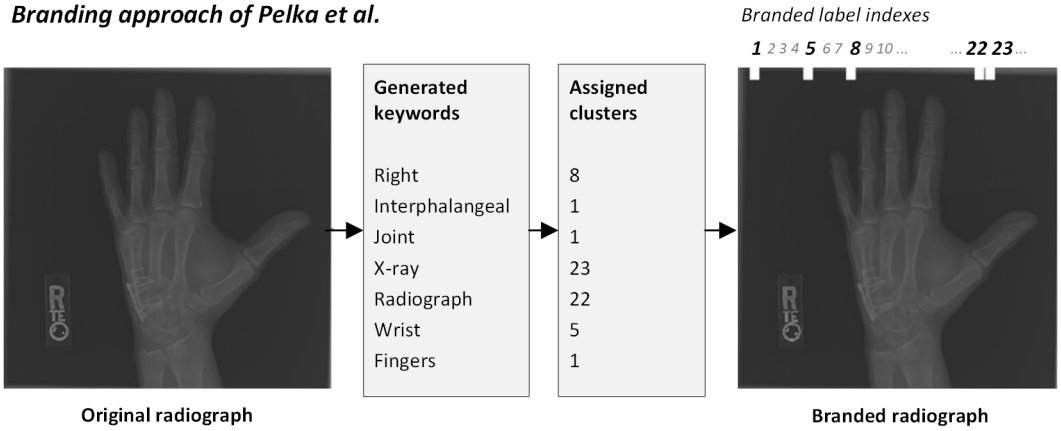


Fig. 2.20: An overview of the procedure for adding branding to radiograph data samples; taken from Pelka et al. [376].

Here, a second modality (keywords) is placed as blocks in images, expanding the graphic information content. The keywords that belong to an image are clustered. The branded radiograph has a white box on the index of each cluster identifier. This method retains the original dimensionality but at the cost of overwriting informational content in the image. Pelka et al. [377] later presented branding with gray-scaling, which could potentially fit a regression approach.

MMML based on 3D data was not found in the surveys of Guo et al. [353] and Baltrusaitis et al. [367]. However, some work has addressed multimodal voxels and multimodal segmentation. Sindagi et al. [373] combined image and point cloud modalities to detect 3D objects. They used the VoxelNet architecture and projected the features of images to regions of interest in the voxel grid. Soltaninejad et al. [378] detected tumors from multimodal brain scans using segmentation. The combination of MRT modalities enabled the correction of weak borders in images that only occured in one modality.

Discussion

Although multimodality is gaining interest in the literature, implementing it correctly is a complex endeavor. Much research has examined the creation of optimal unimodal studies with assumed boundaries. Multimodality combines different datasets and often increases the complexity of ML tasks. Baltrusaitis et al. [367] argued that it is difficult for models to learn supplementary but not only complementary information. Information has a mutual influence that models must consider as a whole and not perform two separate tasks at once. Furthermore, noise is not constant between the two modalities and may affect samples in both ways [367].

Model-agnostic fusion combines data with different structures and representations. Early

fusion integrates features immediately after extraction [367] and before training, such as concatenating representations [353]. By contrast, late fusion integrates data after decisions on individual modalities [367]. Joining elements contain both geometry and PMI (i.e., materials), which affect each other simultaneously. Hybrid and late fusion tend to ignore low-level interactions between modalities, and hence, are less appropriate to implement for joining element prediction.

Lastly, 3D geometry-based MMML is a little researched domain with a few exceptions in health and medical imaging. Soltaninejad et al. [378] and Sindagi et al. [373] have combined Euclidean representations of geometry that are supplementary. The combination of structured and unstructured data, such as in the study of Pelka et al. [376] on images, was not found for 3D geometries.

This section already discussed many benchmark techniques and methodologies. However, many are theoretical and lack a link to the research problem. As such, the following subsection discusses applications of AI that relate to joining element design.

2.3.7 Applications of AI in the manufacturing industry

This subsection presents a dense overview of the application of AI in the manufacturing industry. The aim is to demonstrate the variety of implementations in each field and applications. To rate the complexity of the models and the mimicked cognition, Ullman [379] categorized design tasks according to the degree of creativity in ascending order as follows:

- **Selection** is the choice of individual components, technologies, or parameters;
- **Parametric design** concerns dimension-driven parts that scale along predefined axes;
- **Configuration** refers to choosing multiple components for a system with specified properties; it is possible to combine this with parametric design;
- **Redesign** describes work for adapting, modifying, and optimizing an existing solution to meet a new set of requirements;
- **Original design** has boundary conditions from requirements; however, the principles and details are free.

Hence, noncreative tasks are the simplest to automate [29]. Moreover, they create low-cost and -maintenance models that can create large benefits. As such, most methodologies in the literature start here.

Various reviews have been conducted on state-of-the-art methodologies that implement AI. First, Dekhitar et al. [100] described various areas and applications of ML in manufacturing, such as CAD, product life cycle management, generative engineering, and recommender systems. Next, La Rocca [151] defined knowledge-based engineering as the product of AI and CAD systems. In addition, Salehi and Burgueño [95] extensively analyzed the applications, capabilities, restrictions, advantages, and potentials of AI in structural engineering. Then, Burggräf et al. [380] reviewed the state of the art in knowledge-based problem solving for physical product development. They synthesized CBR methodologies

into an overall architecture. Next, Dogan and Birant [381] provided an overview of ML and data mining in manufacturing. Lastly, Kusiak [333] reviewed applications of CNNs and GANs in the manufacturing industry.

Research has not yet studied ML methodologies for predicting joining elements. Studies have addressed the prediction of production processes (e.g., [104, 166]) and welding parameters (e.g., [105, 169]). Other work had focused on the optimization of mechanical structures (e.g., [47, 103]) and touched on location prediction.

AI also has applications that perform similar tasks to JTS. Hamouche and Loukaides [104] implemented NNs to classify sheet metal forming processes. Hoefer et al. [166] classified CAD models to machine or cast-then-machine using self-defined shape descriptors by evaluating methods as k-nearest-neighbors, decision trees, and random forests. Machalica and Matyjewski [382] classified CAD models with various benchmark ML algorithms on a set of predefined features. Asaga and Nishigaki [252] used a small NN to predict material combinations for modules for new layout design. These studies have demonstrated that NNs are applicable for understanding CAD models for various classification tasks.

Besides technology selection, the literature contains studies on joining parameter prediction. Sim et al. [169] predicted the weld nugget diameter based on parameter settings of machinery, whereas Kim et al. [170] used ontologies and decision trees. Pillai et al. [105] presented a methodology for predicting the mutual spot welding distances of spot welds from part combinations. It integrated PointNet [383] to create part labels and prototype a classification and regression approach. These experimental studies have demonstrated the prediction of joining parameters and indicated the particularity of the design problem for automation.

AI fields are also applicable in topology optimization and other geometrical tasks. Kiani and Yildiz [47] applied genetic algorithms to evaluate crash-worthiness and optimize the noise-vibration-harshness behavior of automobiles. Guirguis et al. [179] applied a similar method to optimize frameworks for welded structures. Banga et al. [103] used CNNs to accelerate topology optimization and predict structural topology using voxel representations. Oh et al. [106] presented a combination of topology optimization and GANs. They used NNs to increase the creativity of designs and topology optimization to increase their quality. Cao et al. [384] implemented NNs with a graph algorithm to 3D-model assemblies from topological wireframes. These studies demonstrate that AI can support structural designs and understand geometry.

The following section presents an overview of the applications of AI in manufacturing industry as well as methodologies mentioned earlier in this chapter.

2.4 Literature overview

This section presents an overview of the referenced literature in the previous three sections. It first discusses relevant properties of methodologies in joining element design. Then, it presents a table that evaluates all of the methodologies.

This study integrated three different fields: joining element design, modular product design, and AI. The discussions in the respective sections have indicated that the fields have

little to no overlap in the literature. Labeling methodologies using properties create an overview of interfaces and relevant work for joining element design and AI applications in the manufacturing industry. Table 2.1 contains all relevant state-of-the-art methodologies that were addressed earlier in this chapter.

The properties used to evaluate the methodologies originated from the literature as well as from the requirements for joining element design. For example, the properties from the literature include those from the study of Hagemann et al. [101], who evaluated various algorithms in the manufacturing industry. They described the properties of algorithms, such as computational cost and determinism, and how these affect the results. Furthermore, the requirements of joining element design referred to the identified advantages and disadvantages of methodologies (see the summaries on joining element design [Section 2.1.6], modular product design [Section 2.2.4], and applications of AI in the manufacturing industry [Section 2.3.7]). The properties are described in more detail as follows:

- **Design problem** $[JT,\ JL,\ JP,\ C,\ PP,\ PF,\ or\ O]$: This represents the design problem that the methodology solves. It includes the joining aspects and modularization related perspectives: joining technology JT, joining locations JL, joining parameters JT, complexity C, product portfolio PP, product family PF, and others O.
- **AI Field** $[RBR,\ CBR,\ S\&O,\ or\ ML]$: This describes the main implemented AI field used in the methodology; see Section 2.3.
- **Deterministic** $[Y\ or\ N]$: This describes whether models output the same values for the same given input. Deterministic processes enable engineers to verify results while the output remains consistent. They provide transparency and insight into how algorithms come to their conclusions. S&O, as well as ML approaches, may create different outputs depending on a changing dataset or model due to built-in approximations and randomness.
- **Deductive** $[D\ or\ I]$ or inductive reasoning: This describes whether algorithms create results from the top down or bottom up. Deductive reasoning D follows from several steps of arguments with general rules to reach a logical conclusion. Inductive reasoning I generalizes cases and examples into generic rules such that results have some form of uncertainty.
- **Optimality** $[1-5]$ of results: This describes how close the methodology reaches to the optimal solution. Optimal designs require little to no rework, especially when considering the scope of the entire product family over its life cycle. Mere feasibility, suboptimal solutions, is the ability to reach a solution, not the solution itself. Values from 1 to 5 represent feasible to optimal results, respectively.
- **Scope** $[D,\ P,\ or\ F]$: This represents the size of product variety considered when applying the methodology to a problem: domain D, product P, or product family F. Domain refers to a local solution applied on a small section of products, whereas product family refers to variety and its consequences.
- **Genericity** $[P,\ I,\ or\ G]$: This represents the ability to use the methodology outside of its domain. Genericity enables the use of methodologies outside of the specified and

limited experiments. The values are product P, industry I, or global G. Product describes methodologies that are applicable only for the same product. Global implies the applicability for any product, even outside of the industry or field of study.

- **Holicity** $[1-5]$: This describes the consideration of requirements and knowledge from multiple phases of product development in the methodology. A value of 1 describes a precise solution with a narrow scope that does not translate well into applicability over the product's life cycle. A value of 5 describes the full consideration and implementation of life cycle requirements.
- **Computational cost** $[1-5]$: This represents the required calculations and resources for acquiring a solution. Computational costs express the idle time of designers in using these algorithms. Values 1 to 5 represent low to very high computational costs, respectively.
- **Automation degree** $[1-5]$: This represents the ability to automate the methodology and the independence of experts and human interference. The degree of automation expresses the developmental and maintenance effort required to implement the methodology in practice. A value of 1 implies significant influence and input from designers to achieve results. A value of 5 implies the ability to create almost no human-in-the-loop processes.
- **Geometry** $[Y\ or\ N]$: This describes the consideration of 3D shapes, features, or models. Considering geometry enables a complete picture of the joint as well as the prediction of joining locations.
- **Knowledge-based** $[Y\ or\ N]$: This represents the ability to take successful benchmark solutions and transform them into the current solution. Knowledge-based methodologies contain a database for utilizing the available data to optimize results and reduce unnecessary design iterations.

Reference	CL	AI	DE	DI	O	S	G	H	CC	AD	3D	K
AlGeddawy et al. [85]	PP	RBR	Y	D	4	F	G	3	2	4	N	N
AlGeddawy et al. [83]	PF	RBR	Y	D	4	F	I	3	2	2	N	N
Alkan et al. [385]	C	RBR	Y	D	4	P	I	4	5	3	N	N
Ambrozkiewicz et al. [192]	JL	S&O	N	I	5	D	G	2	5	3	Y	N
Asaga et al. [252]	PF	SML	N	I	4	P	I	4	5	4	Y	Y
Banga et al. [103]	JL	SML	N	I	5	P	I	3	5	3	Y	Y
Baylis et al. [87]	PF;PP	S&O	Y	I	4	F	G	2	4	2	N	N
Bednar et al. [136]	C	RBR	Y	D	3	F	G	2	2	2	Y	N
Bhatti et al. [130]	JL	S&O	N	I	3	D	P	2	5	3	Y	N
Bond et al. [41]	JT	RBR	Y	D	4	D	I	3	2	2	Y	N
Cao et al. [384]	JL	SML	N	I	2	D	P	1	4	2	Y	Y
Chaimae et al. [253]	JT	CBR	Y	I	4	F	G	5	3	4	N	Y
Chan et al. [147]	C	RBR	Y	D	1	F	G	2	1	1	N	N
Chavare et al. [188]	JL	S&O	N	I	3	P	I	2	5	2	Y	N
Chien et al. [155]	JT	S&O	Y	D	3	P	I	3	2	2	N	N
Choudry et al. [39]	JT	RBR	Y	D	4	D	G	4	2	1	N	N

Continues on the next page ...

... continued from the previous page.

Reference	CL	AI	DE	DI	O	S	G	H	CO	AD	3D	K
Daie et al. [89]	PF	S&O	Y	D	2	P	G	2	2	2	N	N
Das et al. [40]	JT	CBR	N	D	2	F	G	4	2	2	N	Y
Desai [386]	C	RBR	Y	D	3	D	G	2	2	2	Y	N
Eilmus et al. [387]	CC	RBR	Y	D	3	G	C	5	1	2	N	N
Eom et al. [185]	JL	S&O	N	I	2	P	I	2	3	2	Y	N
Ertas et al. [184]	JL	S&O	N	I	3	D	P	2	5	2	Y	N
Etienne et al. [388]	CC	S&O	N	I	4	D	I	3	5	2	N	Y
Favi et al. [148]	PF	RBR	N	D	2	P	G	3	2	1	N	N
Florea et al. [44]	JL	S&O	N	D	5	P	G	4	5	4	Y	N
Florea et al. [46]	JL	S&O	N	I	5	D	G	3	5	5	Y	N
Friedrich et al. [168]	JP	S&O	N	D	4	D	G	5	3	3	Y	N
Galizia et al. [86]	PP	RBR	Y	D	3	F	G	4	3	4	N	N
Gauss et al. [91]	PF	S&O	N	I	5	F	G	5	4	3	N	Y
Geda et al. [109]	JT;JP	S&O	N	I	4	D	I	4	5	4	N	N
Geda et al. [157]	JP	S&O	N	I	4	P	G	3	5	3	Y	N
Ghazilla et al. [42]	JP	S&O	Y	I	5	P	G	5	3	2	Y	N
Guirguis et al. [179]	JL	S&O	N	I	4	P	I	4	5	4	Y	N
Haberhauer [107]	JT	RBR	Y	D	1	G	G	5	1	1	Y	N
Hamouche et al. [104]	JL	SML	N	I	3	D	I	1	4	2	Y	Y
Hasan et al. [389]	C	RBR	Y	D	3	F	T	2	2	2	N	N
Hasegawa et al. [183]	JL	S&O	N	I	3	D	P	2	5	3	Y	N
Hoefer et al. [166]	JT	SML	N	I	3	P	G	3	4	5	Y	Y
Jeandin et al. [164]	JP	RBR	D	D	3	D	G	3	2	3	N	N
Jung et al. [213]	PF	S&O	Y	D	4	F	G	2	3	2	N	N
Kadkhoda et al. [160]	JT	RBR	N	D	4	P	G	4	3	3	Y	N
Kaspar et al. [158]	JT	S&O	Y	D	5	P	G	5	3	4	Y	N
Kim et al. [390]	C	S&O	Y	D	3	F	G	5	3	3	N	N
Kim et al. [170]	JP	RBR	Y	D	2	P	I	4	2	3	N	Y
Kim et al. [117]	JT	RBR	D	D	3	D	G	4	2	4	Y	N
Krus [391]	C	RBR	Y	D	3	F	G	2	2	2	Y	N
Kwon et al. [28]	JP	RBR	Y	D	1	D	G	3	2	1	Y	N
Li et al. [250]	PF	RBR	Y	D	3	P	I	3	2	2	N	N
Long et al. [180]	JL	S&O	N	I	3	P	G	2	5	3	Y	N
Ma et al. [75]	PF	RBR	Y	D	3	P	G	2	2	2	N	N
Madrid et al. [34]	JP	RBR	Y	D	1	P	I	3	1	1	Y	N
Marini et al. [162]	JT	S&O	N	D	1	D	I	2	1	1	N	N
Mesa et al. [156]	C	RBR	Y	D	2	D	G	2	1	2	Y	N
Modrak et al. [392]	C	RBR	Y	D	2	F	G	2	1	2	N	N
Oh et al. [393]	C	SML	N	I	4	D	I	3	5	2	Y	Y
Ouisse et al. [129]	JL	S&O	Y	I	4	P	P	2	4	3	Y	N
Pakalapati et al. [176]	JL	S&O	N	I	2	P	I	2	4	2	Y	N
Pillai et al. [105]	JP	SML	N	I	2	D	P	1	4	3	Y	Y
Prüß et al. [154]	JT	RBR	Y	D	2	D	G	1	1	1	N	N
Ren et al. [90]	PF	S&O	D	I	4	P	4	3	2	3	N	Y
Samy et al. [22]	C	RBR	Y	D	3	P	G	3	2	1	Y	N
Savic et al. [182]	JL	S&O	Y	I	3	D	I	2	4	3	Y	N

Continues on the next page ...

... continued from the previous page.

Reference	CL	AI	DE	DI	O	S	G	H	CO	AD	3D	K
Schmidt et al. [68]	C	RBR	N	D	2	F	P	1	1	1	N	Y
Schuh et al. [394]	PP	RBR	Y	D	1	F	G	2	1	1	N	Y
Schuh et al. [395]	C	RBR	Y	D	2	F	G	4	1	1	N	Y
Sim et al. [169]	JP	SML	N	I	3	D	P	1	4	3	N	Y
Sinha et al. [246]	PF	S&O	Y	D	4	F	G	4	5	3	N	N
Stocker et al. [84]	PF	S&O	Y	D	3	D	C	3	3	2	Y	N
Thompson et al. [50]	JL	RBR	Y	D	1	D	G	1	1	3	Y	N
Ukala et al. [237]	C	RBR	Y	D	1	D	I	1	1	1	N	N
Weiser et al. [396]	CC	RBR	Y	I	2	P	T	4	2	1	N	Y
Woischwill et al. [45]	JL	S&O	N	I	5	P	G	5	5	5	Y	N
Wong et al. [135]	PF	RBR	N	Y	1	F	I	5	1	1	N	Y
Yang et al. [18]	JL	S&O	N	I	3	P	I	2	5	3	Y	N

Table 2.1: Literature overview. Column abbreviations: reference (R), class (CL), main AI field (AI), deterministic (DE), deductive / inductive (DI), optimality (O), scope (S), genericity (G), holicity (H), computational cost (CO), automation degree (AD), geometry (3D), and knowledge-based (K).

In addition to this literature overview, the following section summarizes this chapter.

2.5 Summary

This chapter has reviewed the state of the art in joining element design (Section 2.1), modular product design (Section 2.2), and artificial intelligence (Section 2.3). The literature overview (Section 2.4) summarizes and evaluates relative literature. This section summarizes the findings of this chapter.

Large manufacturing industries generally spread out the design of their products and variants. Holistic approaches in joint and component design require great collaboration effort. Joining element design is a central part of product design and detailing phases and includes many design dependencies [43]. Design processes are time-consuming due to work compartmentalization and the many stakeholders involved. The literature contains various solutions for designing joining element aspects, such as JTS (e.g., [40, 108, 109]), rule-based design (e.g., [50], topology optimization [18]), and joining parameter determination (e.g., [169]). However, they barely consider product variety or successfully marketed products.

Moreover, modular product design is often used to manage product variety [90], but such approaches do not support joining element design. A gap exists between joining element design, modular product design, and knowledge reuse. This gap prevents designers finding global optima and causes joining element design to remain a highly experience-based endeavor [18], resulting in error-prone processes and costly rework [14].

Today, human-engineered AI methods support the design of joining element aspects [1], such as RBR (e.g., [50]), S&O (e.g., [18, 168]), and CBR (e.g., [40]). However, these methodologies are stand-alone, focusing on specific properties of joints. They do not consider the holistic requirements of joining element design, nor is this always possible in early product design. Studies have not explored ML as a solution path, although the methods have been proven to automate expert-driven processes and opt for cost reductions [100].

Manufacturing industries with large product variety possess much data on successful and verified products. As product variety will continue to grow, the industry is developing new products and variants, thereby increasing the amount of data. AI could use these data to predict new joining element designs. Thus, successfully marketed products are ground truths, representing some global (business) optimality in their designs. Hence, AI can harvest the knowledge of these products. The structuring and automation of expert knowledge is a necessity for creating design quality early in the development process [34]. This requires the extraction of patterns and engineering rules from technical information about use cases, know-how, inspection, and simulation data [34]. AI methods can extract implicit knowledge of cross-company requirements, and some methods can generalize this knowledge to support automation. Many joining elements and manufacturability requirements create the necessary engineering data for exploring ML techniques and automating their design. Successful joining element designs contain considerations of properly implemented standards, guidelines, performance, modularity, complexity, and design experience. However, these considerations require several challenges to be solved, such as the dimensionality, quality, heterogeneity, and structure of data [100, 101].

Therefore, AI enables the automation of repetitive ambiguous tasks in joining element design while reducing failures and lead time. In addition, AI methods can learn particularities to predict joining elements closer to global optima as well as increase the effectiveness of design processes. Fast and consistent predictions of joining elements are an enabler of evaluating design alternatives and kickstarting design and integration processes. This ability would enable designers to concentrate on their core competencies and to work on creative, holistic problems.

After concluding the state of the art chapter, the next chapter presents a framework for automating joining element design in high-variety products.

Chapter 3

Framework

This chapter introduces a framework that organizes the main steps for automating joining element design for high-variety manufacturing industries in Section 3.1. The section also maps the state-of-the-art approaches within each step. Then, Section 3.2 addresses the applicability and considerations of popular AI fields in manufacturing at each step. Based on the framework, Section 3.3 describes novel concepts for predicting locations using an evolutionary algorithm, SML, and a GAN. Furthermore, it presents commonalization concepts for joining technologies, locations, parameters, and complete joints. Moreover, this section presents a concept to modularize joining elements. Lastly, Section 3.4 summarizes and concludes the resulting framework with respect to the research gap.

3.1 Framework for automating joining element design

For the sake of readability, "automated joining element design for manufacturing industries with high product variety" is shortened to *automated joining element design* hereinafter. This section presents a framework for automating joining element design called *VICTOR* (Variety Integrated Connection Technology OptimizeR).

The development of a framework can provide structure and direction for automating joining element design. In contrast to processes, frameworks can guide designers without being too detailed or rigid. For example, as the state of the art indicates (see Section 2.1), there are many concurrent, ambiguous processes in product development that affect joining element design. A framework can respect these dynamics and uncertainties in development, giving structure to applicable automation methodologies in various moments and circumstances of product development.

Moreover, a framework can provide guidance. Steps within the design process tend to be performed sequentially. Hence, VICTOR must align with the joining element design process (presented in Fig. 2.5). To distinguish between the joining element design process of designers and the framework, the former process shall be further addressed as the *user journey*. Sequencing tasks give structure and meaning to the organization of automation methodologies. For example, designers require a joining technology before they create joining locations. The framework can take over such considerations. Fig. 3.1 visualizes the

framework.

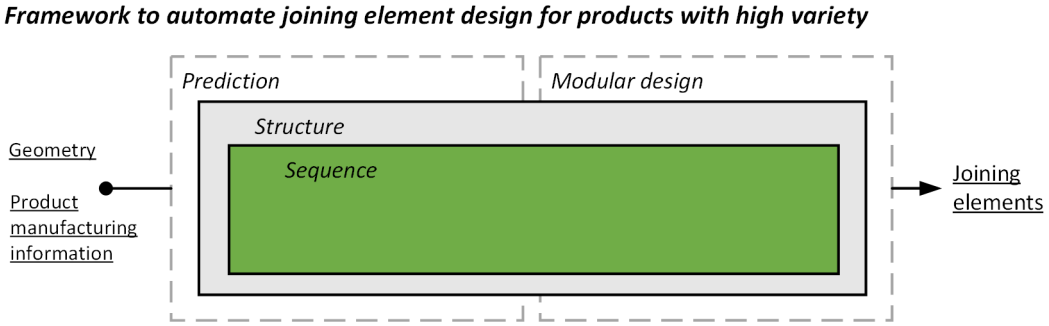


Fig. 3.1: Proposed framework for automating joining element design, which has four blocks: prediction, modular design, structure, and sequence.

The framework for automating joining element design consists of four blocks: prediction, modular design, structure, and sequence. The following list introduces these blocks, which are explained in more detail in the upcoming sections:

- The **prediction** block addresses the automation of design in the user journey. It describes how designers can integrate various methodologies to predict joining technologies, locations, and parameters. These methodologies may implement an AI technique.
- The **modular design** block describes the context of modular design while considering joining elements. Thus, it focuses on the boundary conditions and requirements necessary for fusing joining element design and modular product design. The modular design block addresses the environment for commonalization and modularization methodologies to create meaningful results. Together, the fundamental building blocks set the necessary groundwork for structuring the state-of-the-art methodologies.
- The **structure** lies on top of the prediction and modular design blocks. It breaks automated joining element design down into multiple individual design problems. Each design problem represents a small part of automating the design of a joint. Design problems include the prediction and commonalization of each joining aspect (technologies, locations, and parameters) as well as the modularization of joining elements. Design problems can implement any methodology as long as it fits the circumstances of the user journey. For example, using ML requires a dataset; thus, a methodology that implements ML in an early design stage with no available data is not applicable. However, the same methodology might become applicable once data become available, such as in later design stages.

  Moreover, the breaking down of automated joining element design into design problems enables the organization of applicable methodologies from the state of the art. Each methodology implements certain techniques to solve the design problem. The structuring of these techniques indicates how the literature has tackled the problems. Additionally,

it shines a light on underused techniques, identifying potential new fields for solving the problems.

- The **sequencing** block takes the design problems of the structure block and aligns them to the user journey. By traversing the design problems sequentially, they become steps in the automated joining element design process. The sequencing block organizes the structure of design problems and guides designers through their user journey. For example, the sequencing block first describes the selection of a joining technology.

The next subsections address each block of VICTOR in more detail.

3.1.1 Prediction block

Before discussing the implementation of AI in automated joining element design, it is important to clarify its purpose and use. The state of the art describes the many methodologies, approaches, and perspectives on joining element design. However, many of these remain theoretical. In practice, designers often do not have the time, capability, or resources to fully incorporate these methodologies [12]. Moreover, many of these methodologies also have limitations that, once used, still require in-depth interpretation, analysis, and possible manual rework. Furthermore, joining element design is a holistic multidisciplinary problem [15]. Many designers employ an experience-based approach [18]. Therefore, much information is tacit, for which ML approaches are suitable [29]. Additionally, companies with high variety tend to have many parallel and sequential development processes, including joining element design. Moreover, the potential for automating design is positively proportional to increasing product variety and product maturity [29]. Hence, there are many diverse stakeholders and requirements to consider. These circumstances affect the applicability of current methodologies.

Consequently, the theoretical state of the art does not represent the applied state of the art. Automating joining element design needs to be low-level, simple, and fast [29]. These properties help designers to create joining elements without needing to learn unnecessary programs, software, or processes. Simply put, a designer should click a button, and recommendations for designs should appear. These requirements create a need to explore AI, which can take over cognitive functions while considering numerous dependencies. Instead of outsourcing design requests to other companies, joining element design can be outsourced to a service consisting of an intelligent model. The sophistication of AI is that processing occurs behind the scenes. This processing may be of any level of complexity. However, higher complexity implies an increased capability to balance more parameters; thus, it may lead to potentially higher-quality designs. However, designers must experience AI models as inherent to their work [12]. Their functioning may not draw unnecessary attention. The tools or techniques need to be easily graspable and may not intrude on the designer's craftsmanship [12]. In short, designers should receive recommendations for a given set of inputs. Additionally, AI can be integrated into the user journey, enabling designers to retain their workflows. Fig. 3.2 presents an exemplary process of implementing a prediction model into the user journey.

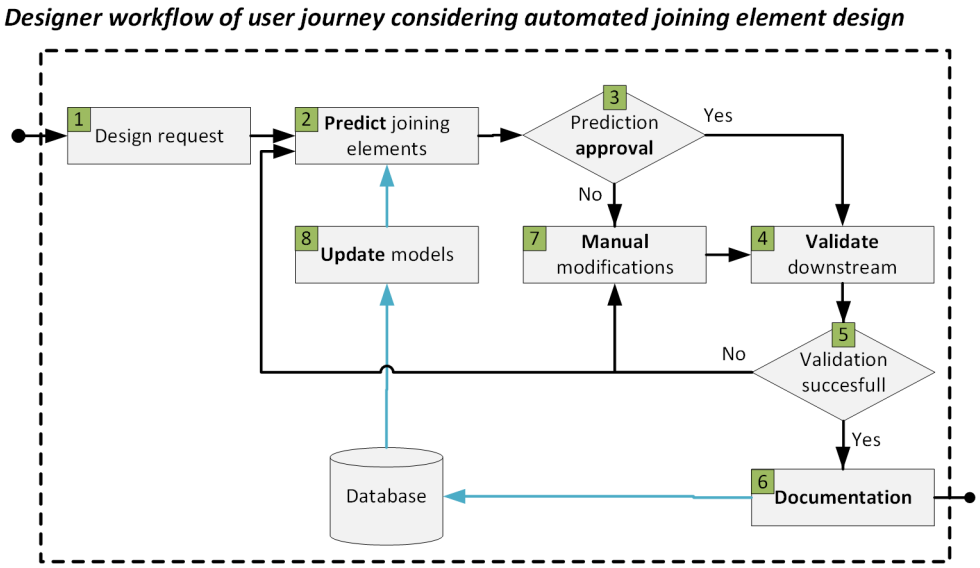


Fig. 3.2: High-level process of implementing artificial intelligence in the joining element design process (Fig. 2.5).

For readability, all implementations of methodologies that generate (aspects of) joining elements are referred to as *prediction models* hereinafter. This definition does not differentiate between ML prediction and multicriteria-based decision algorithms. Ultimately, they all aim to generate data based on a set of inputs, while a designer remains responsible for accepting their outputs.

Now, a walkthrough of the process depicted in Fig. 3.2 is provided. After receiving a design request to generate joining elements, a recommender system suggests joining elements. The designer evaluates a prediction and can optionally modify it using a CAD system. As depicted in the user journey, the designer performs an initial validation of manufacturability; see Fig. 2.5. Unsatisfactory designs require new predictions or manual design effort to adapt them to the requirements. Next, storage, formatting, and documentation are performed in the database for downstream processes. The coordinator compiles the results for further validation. Lastly, the prediction models may need updates over time, such as when requirements change or new technologies become available.

The prediction block substitutes steps in the user journey, as depicted in Fig. 2.5. Designers still need to filter the joining scenario and select the proper boundary conditions. The design part itself is taken over. The process is valid for any of the three prediction tasks, joining technology, locations, and parameters, with the properties described in Section 2.1. For example, in very early design phases, only predicting the joining technology might be sufficient to continue development. However, once development matures, designers may also need to predict joining locations and their parameters. In such cases, the process may run multiple times sequentially, or the prediction step (step 2 in Fig. 3.2) may predict multiple aspects after one another.

Designers may also opt to let the model suggest all joining aspects at once, such as in the topology optimization methodology of Florea et al. [46]. The division into joining aspects

is not necessary. However, it reduces the complexity of prediction models significantly. The prediction step allows any methodology to be implemented.

Moreover, designers may request multiple predictions to choose from by, for example, letting the model predict joining designs from the three best-suited joining technologies. Such recommendations may enable designers to take holistic decisions, such as whether to create three spot welds, create two rivets, or use adhesive bonding. Consequently, recommender systems in combination with designers could determine the optimal design for creative and holistic design problems.

After approval by the designer, the joining designs require detailed analysis for validation. The primary function of joining elements often concerns structural performance. The prediction models may or may not explicitly integrate structural knowledge (in step 2 of Fig. 3.2). For example, when joining performance primarily relies on the ability to absorb structural forces and loads, prediction models could actively consider them. This consideration might involve the use of certain joining technologies or combinations, such as combining spot welds with adhesive bonding, which is often seen in crash-relevant structures in automobiles [397]. Moreover, by placing joining elements in outer sections of CRs, their ability to handle bending, torsion, or buckling increases.

On the other hand, ML approaches exploit patterns in data to predict unseen cases. The application of ML would enable learning from successful joining elements and applying them to new joining designs. Here, the structural considerations are implicit. The prediction models rely on designs that have already been validated. Their outputs assume that for similar cases, similar patterns would result in similar structural performance.

In any case, the validation step acts as an adversary that evaluates the designs. The separated adversary enables the prediction models to generate joining element designs. In this sense, this setup functions equivalently to a GAN with freely creating and discriminating agents working together. However, as depicted in Fig. 3.2, the designer remains in the loop to oversee and verify results and processes. As such, the designer remains responsible for the joining designs in case of changes or other issues arising.

3.1.2 Modular design block

After the prediction steps in the framework, the objective is to reduce variety by implementing a modular design. The aim of the modular design steps in the framework is ultimately to minimize the number of joining elements and the variety in their design. Furthermore, they aim to create modules that retain the flexibility to create new product variants. Practically, modular design converges the diversity in designs, bringing them to a necessary level.

The most applicable modular design approaches are commonalization and modularization. Commonalization aims to unify joining element designs between product variants to reduce unnecessary variety. Fewer joining elements imply lower complexity, lower variety, and lower costs. Modularization aims to define shareable groups of parts (and/or joining elements) to interchange between product variants. The following subsections explain the background and strategy for commonalizing and modularizing joining elements.

3.1.2.1 Commonalization

Commonalization is an approach for creating product platforms. Product platforms are the groups of components shared by multiple product variants. Utilizing common components adequately enables the efficient development of multiple product variants [398]. The same strategy of sharing components between product variants can be implemented for joining elements. Even reusing aspects of joining elements can increase development efficiency.

Large product platforms imply more shared components between all product variants and lower overall costs. However, they also imply a reduction in flexibility and potentially the offered variety. However, joints often remain the same, while components interchange due to different product variants. As the joining element design remains constant, their variety does as well.

As with the prediction of joining elements, the commonalization of joining element designs might be simpler when considering the individual joining aspects. Commonalizing a single joining aspect already reduces variety. For example, the reduction of the number of joining technologies can reduce the change time and complexity [201]. Moreover, after commonalizing individual joining aspects, the resulting joining elements also become more alike and their commonalization is simplified. This bottom-up approach can even continue by considering entire joints.

A study found that product family design and platform-based product development enable the trade-off between product variety and the induced cost-effectiveness to be managed [91]. This balancing act can be the result of implementing both commonalization and modularization. Here, commonalization increases the sharing of components over product variants, whereas modularization combines shared components to create new product variants.

3.1.2.2 Modularization

Modularization is an enabler for product families. Product families are collections of product variants that share one or more components [210]. These shared components can also be modules. Each product variant satisfies specific customer needs through the selection of appropriate modules. Preferably, modules are interchangeable without the need to change other parts. Hence, to generate a new product variant, the goal can be to merely interchange one module for an equivalent one. For readability, components are often also referred to as modules hereinafter. Notably, modules can consist of one component

This one-to-one interchangeability of modules only works if the interfaces between modules remain constant. Interfaces are the connection between a module and the rest of the product. Hence, the standardization of interfaces enables the design of modules that can interchange freely. Standardization helps to create robust designs and increases module sustainability [82]. To do so, it induces a set of common requirements for the design of parts and modules.

However, standardizing interfaces creates a conflict in modular design. Modularization requires a set of commonalized requirements in design phases to prepare for the interchangeability of parts. Hence, it limits the design freedom of components. Additionally, modularization needs to determine the best interfaces for optimizing the interchangeability of

modules, which limits the design freedom to create modular products. These conflicts are vital to consider during modular design, especially when considering joining elements.

Designers create joining elements on the interfaces of parts (or modules). When the interchanging of modules is enabled, each module still needs to be attached to the rest of the product. Standardization of the interface would require the connecting components in a module to remain equal in their properties for joining. Hence, the joining elements that exist on the interface can remain the same. However, completely static, standardized interfaces are not sufficiently flexible to meet the new requirements caused by continually changing customer demands. Therefore, the interface and the assigned joining elements need to cope with a certain variety.

Furthermore, the joining elements must comply with the changing conditions of interchangeable modules. The properties, such as the materials, that a module brings to an interface may change. The joints need to consider the combinations of changing properties of modules for that interface. The same joining elements may connect different modules as long as their properties on the interface remain constant. This is true regardless of the product variant. Moreover, to workers or robots in the factory, the joint is the same. The joining operations remain constant regardless of the modules they need to join. Consequently, joining element designs are preferably the same for a given interface regardless of the modules they join.

Consequently, this study refers to sets of joining elements to be reused in products as **joining modules**. They may consist, for example, of three spot welds or five rivets. This enables the selection of joining modules for module interfaces. Then, specific joining modules may join specific combinations of modules. As a result, module interfaces need some commonalization requirements only for selected combinations of modules. The interface similarly requires the selection of an applicable joining module.

Optimally, the properties of modules on the interfaces are constant, implying that the joining elements and their boundary conditions are the same. Constant properties on interfaces induce one joining module being necessary to join all combinations of modules. However, increasing variety in properties would increasingly require more and different joining modules. Fig. 3.3 visualizes the concept of joining modules.

The product platform presents shared modules in product 1 and product 2 of a product family; see Fig. 3.3. Design spaces represent functional requirements that adopt component modules to fulfill their function. As a side note, these functional requirements are known as positions in PDM methodologies [31]. The example presents four design spaces in each product variant. Module interfaces (white dotted areas) indicate design spaces. Each interface can take up a joining module (e.g., D or E). The design space of the interface is not necessarily bounded by, for example, the geometry of the modules. As the combination of modules can change, so too can their CRs. For an example, see the switch of module A and B in design space A. This must also be considered in the selection of joining modules. Hence, joining modules are free to be interchanged on interfaces as long as they respect manufacturability requirements. Fig. 3.4 illustrates different uses of joining modules: intra- and inter-modular.

Product A in Fig. 3.4 uses the same module twice. Each module consists of three parts: one joining module and a blue and a gray component. The joining module is used

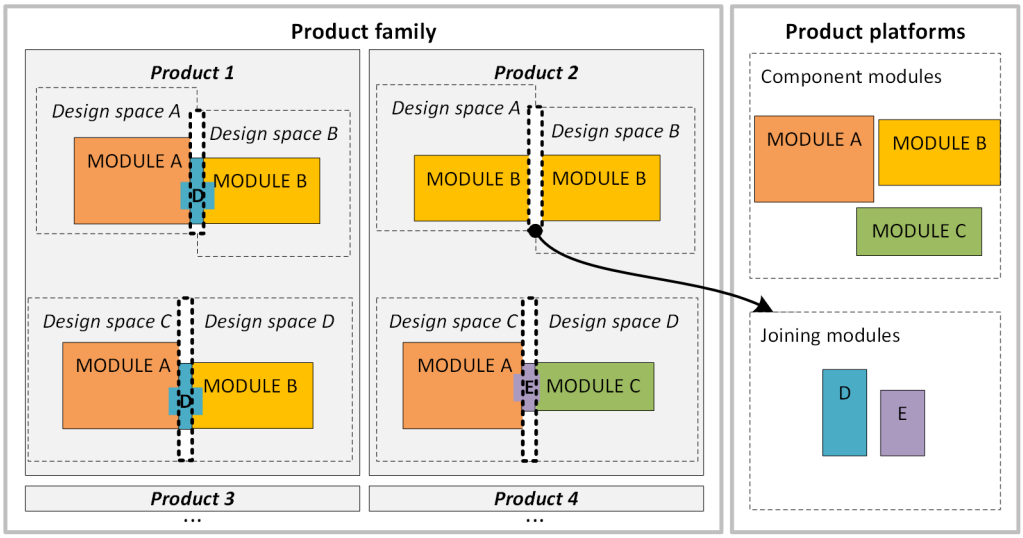


Fig. 3.3: Overview of product family and platforms that convey the ideas of modularization and commonalization. Module interfaces are highlighted to indicate the design space for joining modules.

intramodularly; that is, it exists within another module. The created module can also be viewed as a subassembly. A potential use could be the outsourcing of their manufacturing. Moreover, the module can be configured multiple times into the product. However, these joining modules are bounded in terms of variety by components of the module they create.

For a more flexible use of joining modules, they should be used inter-modularly. For example, product B (Fig. 3.4) does not have predefined modules. However, it integrates four components and two joining modules directly in the vehicle. Again, note that modules may consist of one component. Hence, product B instantiates the joining modules on the module interfaces. The advantage manifests when one also considers product C. Here, the purple joining module in product C connects different modules as in product B. It is also re-instantiated twice, and both times to join different modules. Inter-modular joining modules demonstrate the flexibility of considering joining elements in modular design.

Joining element modularization is the process of creating joining modules. Fig. 3.5 presents exemplary states before and after modularization.

Fig. 3.5 visualizes three scenarios (A, B, and C) that all consist of two U-profile metal sheets. The only difference between these components is that they have different lengths. All joining scenarios have two CRs. However, scenarios A, B, and C have 3, 5, and 7 joining elements, respectively. The stacked variety column represents the joining elements and modules considered over the entire product family.

To visualize joining element modularization, the row *without modularization* presents joining element designs directly after prediction. Each scenario generates one joining module and, without modularization, results in 30 joining elements. The *modularized* row presents the

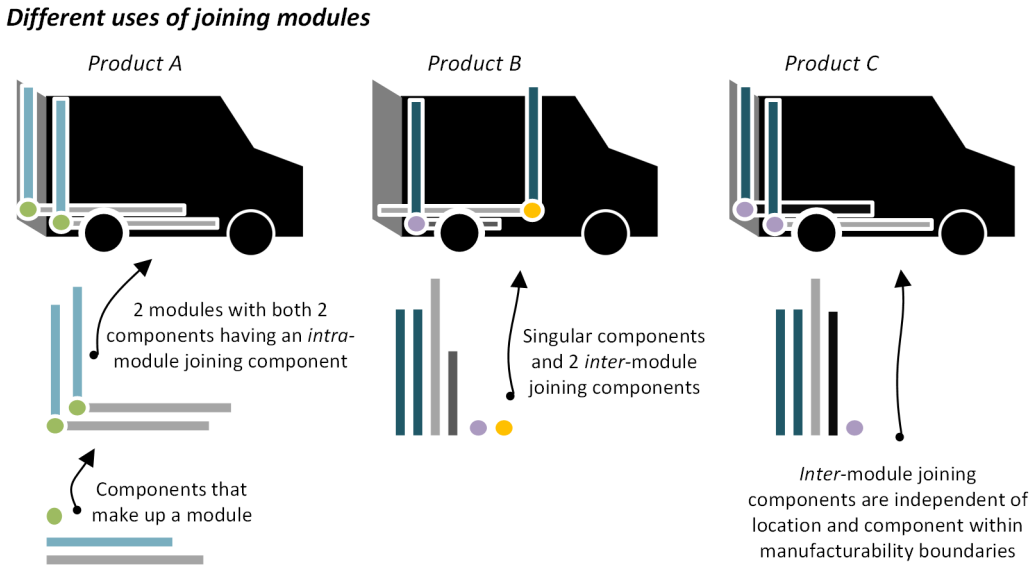


Fig. 3.4: Three products that utilize joining modules differently. Product A has intra-module joining elements to create subassemblies; Product B utilizes inter-module joining elements on interfaces of interchangeable modules; and Product C indicates that the purple joining module can be combined differently with other modules.

results after the predicted joining elements are commonalized and modularized. It reuses joining elements where they overlap and determines joining modules to reuse over the scenarios. As a result, the joining modules contain fewer joining elements. Furthermore, combining joining modules can create the joints for scenarios B and C. The stacked variety columns indicate that a total of 14 joining elements remain. Moreover, scenario C has two joining modules that both have four joining elements. Theoretically, re-instantiating the green joining module twice could also make the brown joining module redundant.

To summarize, a **joining module** contains a fixed set of joining elements. It can be configured multiple times into product variants, each time joining different modules. Joining modules are not bounded by CRs but are linked to commonalized requirements on module interfaces. In this respect, they function as any other module. For this, commonalization and modularization approaches consider joining elements as individual components. Joints can consist of multiple joining modules located on one or more CRs.

3.1.3 Structure block

The structure block defines the joining element and modular product design problems. Fig 3.6 indicates that it lies on top of the fundamental blocks (prediction and modular design). To simplify automated design, the prediction block breaks the joining element design into smaller design problems. There are three prediction design problems, one for each joining aspect. Then, the modular design block reduces the predicted variety by employing a bottom-up approach. There are two modular design problems to commonalize the joining aspects. Furthermore, there is one commonalization approach for joints. Lastly, there is one modular design problem

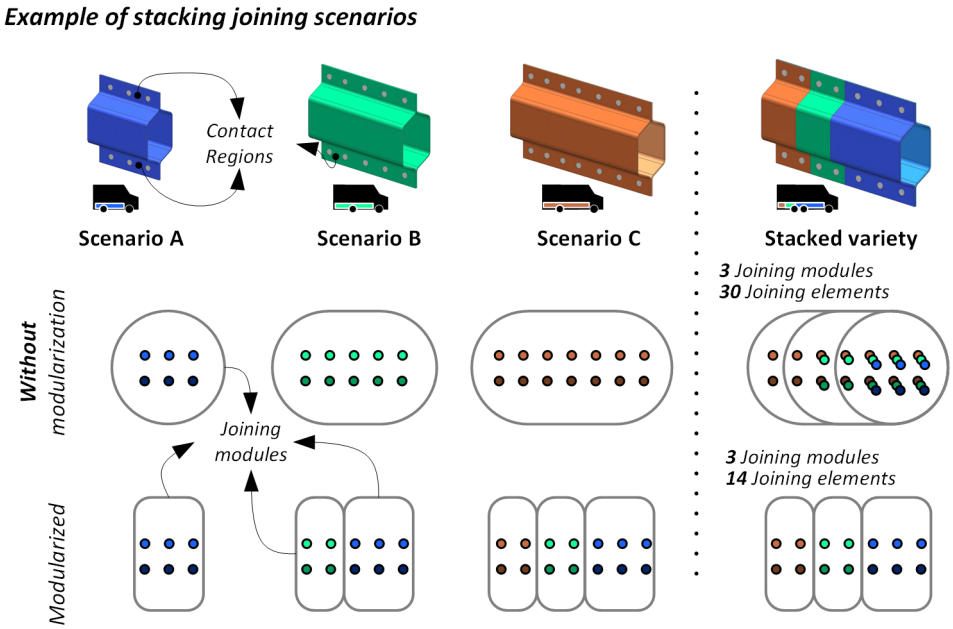


Fig. 3.5: An example of three scenarios and the virtual stacked result depicting a significant reduction in the number of joining elements through commonalization.

that modularizes joining elements. Fig. 3.6 depicts the seven design problems for automating joining element design, which is described in the following list:

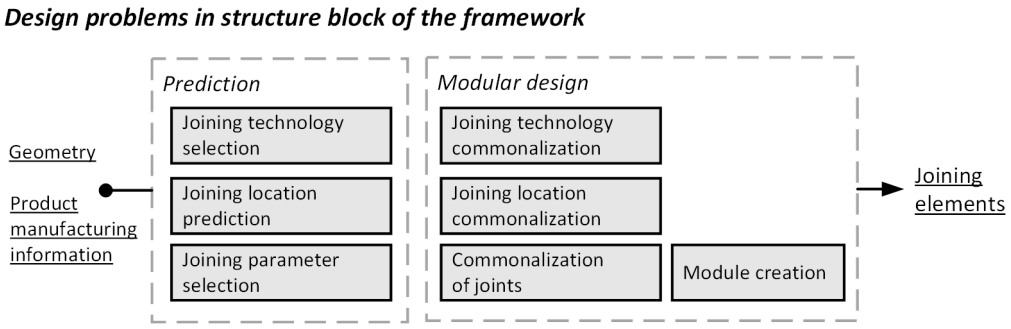


Fig. 3.6: Identified design problems in the structure block of the framework.

- **JTS** regards the task of finding the best process for a set of to-be-joined components as listed, for example, in DIN 4063 [17].
- **Joining location prediction** creates locations for joining objects (often points or curves) that represent joining technologies.
- **Joining parameter selection** predicts the detailed information to describe the PMI of the joining elements.

- **Joining technology commonalization** reduces the variety of joining processes by looking at multiple product variants at once. It can also reduce the variety in joining parameters, with it being a similar design problem.
- **Joining location commonalization** determines substitutions for joining locations that are shareable between product variants. Thus, it reduces the geometric variability between overlapping joining elements.
- The **commonalization of joints** unifies them by substituting joining elements between product variants. It reduces the number of joining elements in documentation.
- **Module creation** collects the joining elements of product variants into modules, enabling the creation of, for example, subassemblies and joining modules.

The structure block in the framework enables the mapping of the methodologies to the design problems. Various and multiple methodologies may solve each design problem. The design problem has no defined input or processing requirements. They are relative to each methodology and the stage of product development. For example, a topology optimization approach for predicting joining elements requires a fully defined product variant (see Section 2.1.5). Moreover, rule-based prediction algorithms only require the CR between two components. Although in both cases they use completely different techniques and data, they still predict joining locations.

By organizing the methodologies in each design problem, research and company interests and trends become clear in terms of how they intend to solve the problem. For example, there is a great focus on multicriteria decision-making methodologies for predicting the joining technology and a lesser focus on RBR or CBR. Each technique has its advantages and disadvantages. For example, multicriteria decision-making methodologies tend to optimize for one joining element design, often neglecting the product as a whole. Consequently, one can identify gaps in unused techniques to apply their properties for solving the design problem (as discussed later in Section 3.2).

In short, integrating the set of all design problems enables one to automate joining element design. Each design problem may have various methods for joining element prediction as well as commonalization. Combining these methods with modularization approaches enables the reuse of joining elements in modules within the same product or product variants. Together, these approaches determine the joining modules to reuse in the same product and other product variants.

3.1.4 Sequence block

The design problems from the structure block have relationships between them. Identifying these relationships will guide the designers. VICTOR enables the selection of processes for the specific job at hand for any product maturity. Hence, it does not enforce selecting them all sequentially. However, by aligning with the user journey (Fig. 2.5), designers tend to have a theoretical workflow. Fig. 3.7 represents this workflow with black arrows, indicating a sequential, theoretically sound, and smooth process.

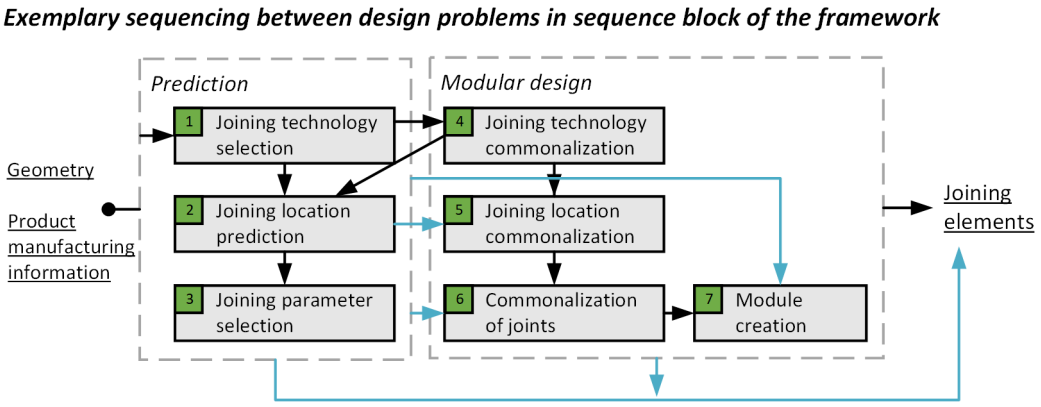


Fig. 3.7: The addition of the sequence block to the framework with exemplary process flows.

Similar to the user journey presented in Section 2.1.1, designers first need to determine the joining technology. From that, they can derive the geometry, after which they attach PMI to the objects. Each of these prediction steps has a commonalization counterpart. Joining *technology* commonalization reduces the variety in joining processes. Joining *location* commonalization aligns the geometric variance of joining locations. To commonalize the joining *parameters*, the same methodologies can be used as for joining technology commonalization due to the similarity in the design problems. Furthermore, the commonalization of joints unifies them by adding joining elements from other product variants. Lastly, the modularization step contains module grouping, which creates joining modules.

However, as the black arrows in Fig. 3.7 indicate, designers do not have to perform commonalization approaches after all predictions. For example, in early product design, designers may choose to unify joining technologies before predicting joining locations. This sequence would prevent the creation of many joining locations that potentially need to be removed as the technology would change due to unification. The blue arrows in Fig. 3.7 visualize these other sequences with higher probability.

In another example of performing nonsequential steps, after designing a few product variants for a large product family, modularization practices may not be relevant. The modules might require changes due to the continual development of product variants. Moreover, modular approaches are often impractical. For example, the (outsourced) designers may not have access to that information [8]. Hence, the sequential design of product variants in evolving product platforms may require modular approaches to strategic moments in development [135]. These strategic moments may also apply to commonalization approaches. For example, unifying the technologies after every prediction may theoretically create a unitechnology product.

3.1.5 Discussion

Joining element design is an integrative task; for example, it should also consider component design. Product development is a creative process. Not all processes in product design

co-occur or flow in perfect succession. Company strategies, such as outsourcing and work compartmentalization, may cause unnecessary design iterations and variants. As seen in the state of the art, methodologies do not always rely perfectly and synchronously on one another. Hence, the starting point may not always be joining technology. Referring back to the W-model of Kaspar et al. [43], joint section design is an intermediary state in component-level design that integrates material and technology selection (Fig. 2.4). Joint section design is a subsystem-level design methodology. It requires the simultaneous, integral design of components and joints. However, VICTOR focuses specifically on the design of joining elements, but this does not mean that the methodology conflicts with state-of-the-art beliefs:

- None of the joining element or assembly design methodologies consider product variety in the authoring of joining elements – not in technology, locations, nor parameters. Methodologies that, for example, determine an optimal joining technology are often multicriteria decision-making methods that do not consider uncertainty or varying characteristics of different components.
- Methodologies do not consider joining elements on the same hierarchical level as components, resulting in a skewed dependency. Joining elements are designed afterwards and are dependent on product design, although assembly account for approximately 50% of the production costs. The joining elements are derivatives of joined components when thinking purely in terms of subsystems and subassemblies. This thinking causes problems for selection algorithms due to a lack of consideration of the whole. Various product documentation methods have already highlighted the significance of adequately documenting joining elements in a network instead of hierarchical structures (e.g., [31, 33]).
- The prediction of joining elements after component design does not ignore the integral component in joint section design. Functional requirements can already incorporate assembly considerations in early design phases. Moreover, some topology optimization methodologies integrate component and joining element design (e.g., [46]). Furthermore, automating joining element design shortens the time of design iterations. For example, ML predictions are almost instant. As a result, designers can have head starts for new product variants and directly start collaborating with other stakeholders. This increased speed of design supports the rapid evaluation of design choices for both assembly and component design.

After presenting the framework for automating joining element design, the following section evaluates AI fields on their applicability for solving each design problem.

3.2 Applicability of artificial intelligence fields

Based on the literature review, benchmark methodologies for every design problem in VICTOR (Fig. 3.7) are mapped in the structure block. The methodologies are clustered according to the main techniques they use, as illustrated in Fig. 3.8. The figure builds upon Fig. 3.7. It positions design problems the same as it does for sequencing. The main difference is that Fig.

3.8 depicts a wide range of approaches that solve each design problem. Each approach may concern one or more AI fields (listed in Section 2.3). They also all involve an evaluation of their appropriateness for the task. The structure block is the evolution of the automated design processes initially published by Eggink et al. [2]. Fig. 3.8 presents applicable approaches for each AI field for each design problem and distinguishes the methodologies from state of the art and VICTOR.

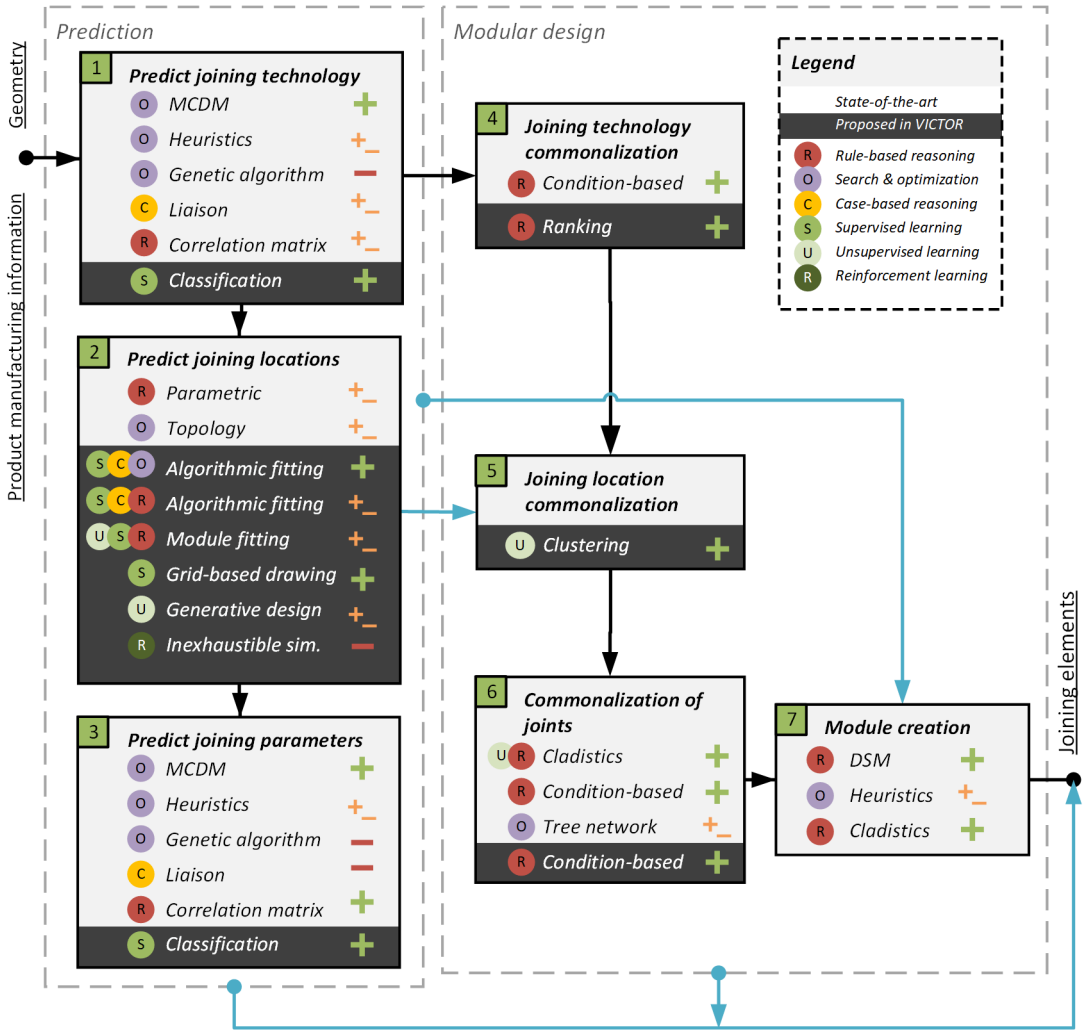


Fig. 3.8: The applicability of artificial intelligence for predicting joining elements. The image is an extension of the structure block in the framework shown in Fig. 3.7. It visualizes the state of the art and proposes new methodologies from VICTOR; adapted from Eggink et al. [2].

The four main AI fields in the manufacturing industry (RBR in red, CBR in orange, S&O in purple, and ML variants in shades of green) indicate for each approach the type of technique used to solve a design problem. This section explains the novel approaches in more detail and briefly addresses the benchmarks. The evaluation and grading of approaches follows from

their ability to solve the joining element design problems from Fig. 3.7. They also address the discussions in the state-of-the-art chapter (Chapter 2). The appropriateness of new techniques was determined through the literature (e.g., [2, 29]), experimental results, and discussions with experts.

Fig. 3.8 also visualizes the heavily researched domain of process and material selection with the many different approaches. Joining location design also focuses on topology optimization or similar methodologies. However, the commonalization of product families has received little attention, and modularization approaches increasingly focus on matrix-based methodologies and heuristics to derive modules. The following sections discuss the novel approaches for joining technology and location prediction, commonalization, and modularization in more detail.

This study defined **joining scenarios** (JSs) as *the information that describes a joint using geometries, PMI, product architecture, and assembly information* [2]. A joining scenario describes the physical and systemic environment, requirements, and components for the design of joining elements.

The following subsections address the individual steps in the framework. They elaborate on the evaluation of the applicability of each AI field from Fig. 3.8. For readability, each subsection has a figure that is excerpt taken from Fig. 3.8. Additionally, the steps indicate opportunities for new AI fields to solve these design problems.

3.2.1 Joining technology selection

Fig. 3.9 is an excerpt from Fig. 3.8 that indicates that the selection of joining technologies has many different approaches.

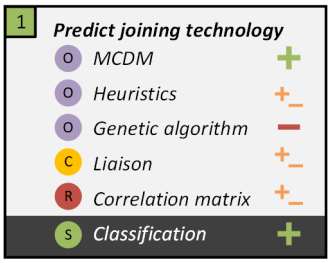


Fig. 3.9: The applicability of artificial intelligence for predicting the joining technology. The image is an excerpt from Fig. 3.8.

Multidisciplinary optimization includes methodologies (e.g., [108, 109]) that used heuristics with a genetic algorithm. Hence, S&O approaches can solve this problem well. However, they rely on consistent data, development effort, and designer inputs, and neglect product variety. Moreover, RBR can screen for feasible technologies. However, it requires extensive development and maintenance effort, mainly due to the vast solution space and many distinct joining scenarios. Furthermore, liaison-based methodologies include the CBR approach of Das and Swain [40] and the ontology approach of Chaimae et al. [165].

Selection is a classification task and can be solved using SML. However, it is crucial to consider the quality and completeness of available training data. Joining scenarios become

more detailed as a product matures. Hence, over time, the predictions of models will become more accurate. SML considers historic design decisions implicitly, which is an advantage of an inductive approach over a deductive one. The complexity of use cases prevents human-engineered deductive rule-based approaches. Moreover, the complexity requires data-driven approaches to automatically find correlations in data [399]. Furthermore, JTS for high-variety industries with continuous development does not necessarily require a true optimal result for individual scenarios. Steps that include commonalization and modularization techniques will remove unnecessary variability from multiple predictions on multiple joining scenarios to reach a global optimum.

3.2.2 Joining location prediction

The structure block reveals that current literature provides two main approaches for creating joining locations (Fig. 3.10): RBR (e.g., [50]) and topology optimization-based design (S&O; e.g., [18, 46]).

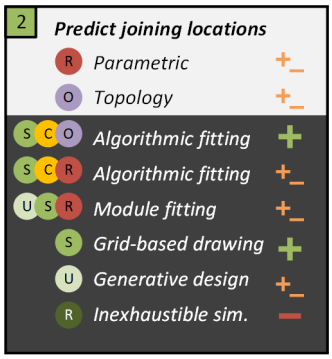


Fig. 3.10: The applicability of artificial intelligence for predicting joining locations. The image is an excerpt from Fig. 3.8.

Joining locations derive partly from joining technologies. The technologies set requirements for component geometry, such as a minimum CR size (flange overlap) for enabling resistance spot welding [115]. Thus, technologies create boundary conditions for the prediction of joining locations. Examples include whether to predict coordinates for curve-based technologies such as clinching and riveting, or a set of coordinates for curve-based technologies such as adhesive bonding and laser beam welding.

Joining location prediction requires knowledge of the technology to create the proper output format, for creating either a set of individual coordinates that all represent individual joining elements or sets of coordinates that represent curved joining elements. In any case, individual methods may use different types of data formatting to represent these coordinates through, for example, the use of images (e.g., Section 3.3.3), voxels (e.g., Section 3.3.2), or points (e.g., Section 3.3.1). However, these types of data formatting require the reconstruction of Cartesian coordinates.

Three AI techniques can assist in predicting joining locations on different levels of cognitive automation and draw inspiration from manual design processes (see Section 2.1.1).

First, **algorithmic fitting** is the automation of the designer's variables to set rule-based joining locations using use-cases (CBR) or history (SML). Second, **grid-based drawing** discusses a fully SML approach that represents the visual aspect of analyzing geometry through experience. Lastly, **inexhaustible simulation** uses RL to mimic the infamous trial-and-error approach of designers and obtains joining design knowledge through many numerical simulations.

The following section, Section 3.3, presents specific methodologies for algorithmic fitting (Section 3.3.1) and SML (Section 3.3.2 and 3.3.3). The present section continues by discussing and evaluating the properties, advantages, and disadvantages of using various AI fields to predict joining locations. Addressing the approaches in detail will assist in imagining the benefits of the new AI techniques:

- **Algorithmic fitting**

  This approach utilizes human knowledge and engineers it into algorithms. The idea is that sets of joining locations or joining-related features get reused without detailed consideration of geometry, function, or structural performance. Hence, algorithmic fitting may use different and straightforward AI techniques to predict joining locations. Fig. 3.11 depicts several algorithmic fitting techniques. Each technique is briefly addressed as follows:

  – *Joining feature prediction*

    First, algorithmic fitting may utilize SML to predict relevant features and positions on CRs; see Fig. 3.11. The features may be a combination of the number and distance of joining locations. Predicting such joining features can support other algorithms in creating joining locations. For example, by using the number of joining locations, an algorithm could distribute them on a CR according to a specified rule set or optimization objective.

    Moreover, the prediction of the number of joining elements correlates with the structural performance. As mentioned in Section 2.1, static pull and push forces in particular are directly related to the number of joining elements. For readability, the length of curved joining elements is also referred to as the number of joining elements hereinafter. This is because higher numbers of joining elements tend to be placed on larger components that are required to handle greater forces.

    Pillai et al. [105] demonstrated that NNs can predict mutual distances between spot welds. Combining their results with the parametric RBR methodology of Pakalapati et al. [176] creates a method for automatically predicting joining locations on centerlines of CRs that consider successful designs.

    Interestingly, predicting the mutual distance or number of joining locations is an equivalent task in cases where the prediction model considers the CR. Equally distributing a number of joining locations within a given space is similar to determining how many joining locations with a given mutual distance would fit in the same space. However, it is equivalent to point-based joining locations. For curve-based joining locations, it is more interesting to predict the length as a substitute for the number of discrete joining locations.

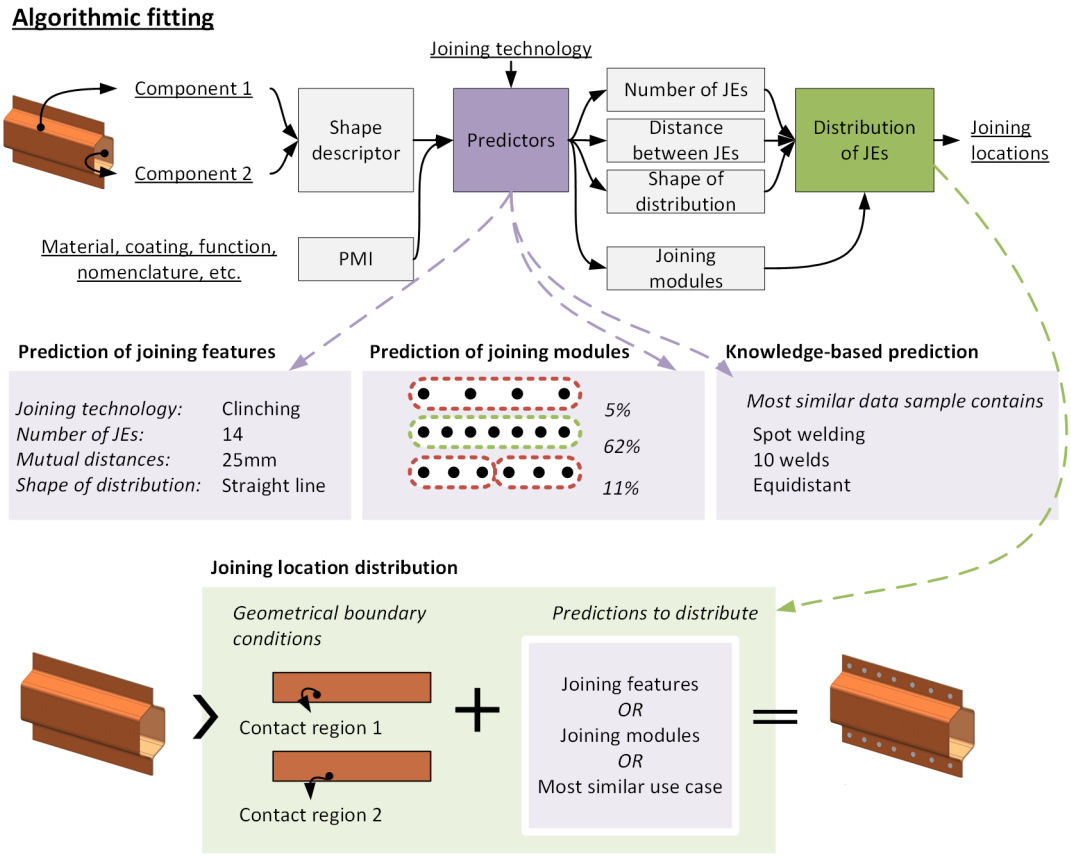


Fig. 3.11: Overview of algorithmic fitting approaches.

– *Joining location distribution*

Joining elements populate CRs and are bounded by manufacturability requirements, such as mutual joining element and edge distances. Joining location distribution takes predicted joining features, such as five spot welds, and places them in CRs. Where the joining locations end up depends on the distributing algorithm. For example, the algorithm might use joining locations from databases or aim to maximize mutual distances.

IInterestingly, optimizing for mutual distance or edge distance influences does not necessarily optimize for structural performance, although the algorithms may have an incentive to move joining elements out of the center (e.g., see Fig. 2.3). Still, stresses induced by bending, torsion, or buckling benefit from joining locations positioned specifically outside of CRs. The algorithm might need additional terms to push joining locations further outward.

Defining the boundary conditions to limit the solution space is a typical problem for RBR (e.g., [28, 50]). The number of joining locations depends on the performance requirements of joints and firmly on component design [43],

implicitly considering function, loads, and structural performance [188]. Prediction of the number of joining elements enables, for example, an RBR algorithm to distribute joining locations equally on guidelines. In genetic and evolutionary algorithms, S&O entails iterative processes that can create locations within the boundary of the CR. Similar problems include evolutionary land-use optimization algorithms that can distribute points and regions in geometry (e.g., [272, 273]). The implementation of this approach using an evolutionary algorithm is described in Section 3.3.1.

– *Joining module prediction*

  In addition, joining modules can consider modularization results directly. Joining modules can be imagined as the positioning of LEGO® blocks on surfaces. For example, a joining module may contain three spot welds or two laser beam welds picked and placed on a CR. An algorithm would retrieve joining modules from a database and attempt to placing them on CRs. Applicable joining modules may also be selected by filtering with joining features.

  Reusing joining modules reduces complexity and prevents unnecessary development effort [81]. The modules can prevent redundant and ambiguous joining element development. For example, reusing joining modules prevents the design of new joining elements for other products, versions, or variants. The selection of joining modules is a task for all AI fields, including RBR, CBR, SML, and S&O. SML can predict specific modules or features of models to use as the foundation for fitting joining elements. RBR would require high development effort due to many variations and use cases. It can increase design complexity and negate the advantages of automation. The approach requires the explicit programming of all use cases, exceptions, and particularities that, through changing environments, require constant updating.

– *Knowledge-based prediction*

  Furthermore, CBR requires a large development effort. It also requires knowledge representations of joining scenarios that are coherent and complete. These are common structured vocabularies (i.e., ontologies) that define a fixed format for information of joining scenarios. Knowledge representations structure joining information, which then enables one to find relations between them. These relationships (e.g., similarities) enable the retrieval of similar use cases. CBR can then apply the designs of similar use cases to new design problems.

  However, CBR also needs to retrieve the geometries of components. Shape retrieval concerns similarities between components [400]. However, they may be difficult to manage in large databases, where, for example, small features can be crucial for distinguishing shapes. CBR with feature recognition would require additional development effort. S&O algorithms can evaluate various module combinations and position them accordingly.

  The prediction of joining features or modules contains the solution space and leaves much control in the hands of designers. The approach is transparent, so designers can comprehend the model's decisions. Furthermore, models train and

predict rapidly, thereby limiting the required developmental effort. However, optimal designs are not possible due to the fitting of predictions and the limited design freedom of algorithms.

- **Grid-based drawing**

  The second approach directly uses geometry to predict joining locations instead of intermediate features that require positioning. This approach, namely grid-based drawing, refers to the data representation and the task that the models attempt to perform. The term "drawing" comes from the cells in the grid, which the NNs need to "color." That is, the models need to *draw* joining locations on a given sample. The term "grid-based" refers to the required discretized data formatting for NNs. Euclidean data representations enable consistent mappings between input and target samples, regardless of whether 2D or 3D data are used. Implementing NNs to draw joining locations is an SML technique. The successful use of SML for similar structural design tasks was demonstrated by Banga et al. [103]. They successfully implement voxels and a CNN to optimize 3D topologies. Fig. 3.12 depicts an overview of grid-based drawing approaches using the example of image segmentation and probability estimation.

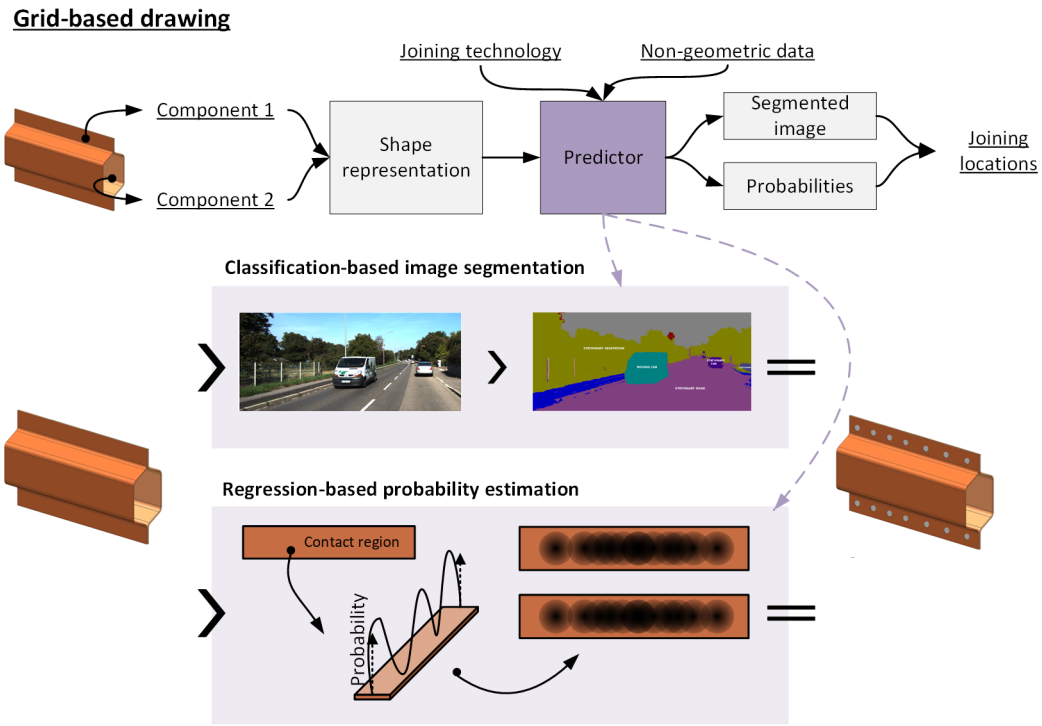


Fig. 3.12: Overview of grid-based drawing approaches.

The grid-based drawing approach relies on patterns in successful joining designs. Similar to algorithmic fitting, it has no objective to generate designs with high structural performance. The approach merely bases its predictions on designs that originate from

similar input data. This reliance enables the implicit consideration of structural performance. The data samples contain these considerations. Hence, an assumption exists that the prediction model extracts this knowledge from the data samples in training and applies them in new designs.

SML has two learning tasks, namely classification and regression. A typical classification task is image segmentation. Image processing uses segmentation to extract meaningful information by predicting sets of pixels [289], for example, to extract a road, sky, or forest from an image. Similarly, a model could segment shapes, as Wang and Lu [401] illustrated. A segmentation task can identify geometrical areas to classify regions of geometry as joining locations. Moreover, Nibali et al. [402] illustrated a regression approach to indicate points of interest in images. Typically, these points are at the centerlines of CRs. Similarly, a regression task can indicate joining locations by increasing probabilities. The model creates a heat map with increased values at locations with a high chance of joining locations. Moreover, Choi et al. [331] demonstrated a UML approach for applying the information of one image to others. Models can learn patterns of joining element designs and apply them to unseen joining scenarios to create new designs. Oh et al. [106] applied this concept to design new vehicle rims. GANs could use the same data formatting but a different training setup to perform predictions.

Euclidean data only describe geometry, but joining elements also include other technological information and joining parameters. A pure geometrical concept overlooks important supplementary information, such as PMI. This SML approach requires additional processes to include this information. UML can create feature clusters to add nongeometric data to datasets, as seen in the studies of Pelka et al. [376], Soltaninejad et al. [378], and Sindagi et al. [373]. Such methods are not nondeterministic and incomplete, but their results are generic as long training data cover new joining scenarios.

Furthermore, 2D and 3D Euclidean data representations have the advantage of transformable architectures of benchmark computer vision methodologies. Shape-descriptor approaches are not applicable as their geometric abstraction prevents the reconstruction of joining location coordinates, such as moments of inertia, Zernike moments, or other feature-based properties [403]. The loss of geometrical cohesion in encoded input samples prevents the construction of joining locations directly.

The SML approach also has complex models that need to understand the concept of geometry and its influence on joining locations. This approach potentially creates more accurate predictions due to the increased information, but at the cost of higher development effort and model complexity. The implementation of grid-based drawing using an EncDec architecture and a GAN are described in Sections 3.3.2 and 3.3.3, respectively.

- **Inexhaustible simulation**

  A simulation environment enables one to freely try joining location distributions and to measure their performance. By inexhaustibly trying many distributions, the chance of

achieving a good result increases. RL is an AI field that can learn the relation between the joining location distributions and their performance. Fig. 3.13 visualizes a simulated version of this trial-and-error approach.

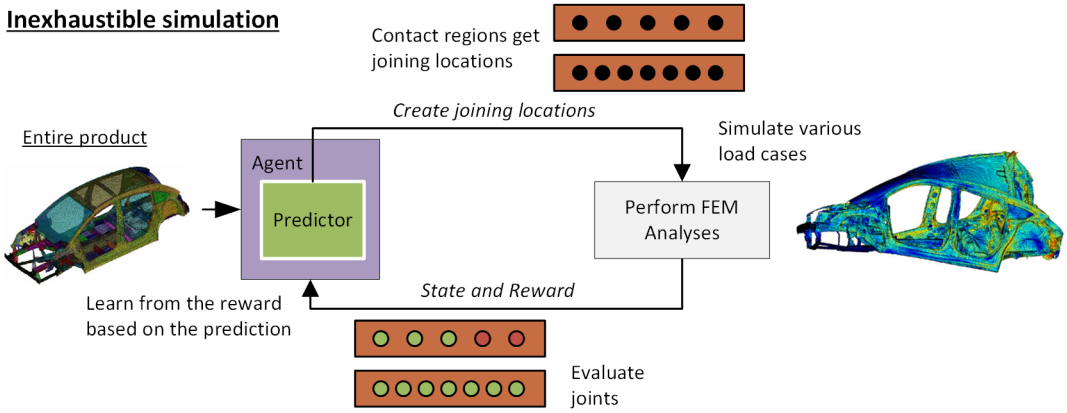


Fig. 3.13: Overview of inexhaustible simulation.

RL first creates a joining element distribution and puts it into an environment (e.g., a finite-element model). Then, an adversary (RBR) evaluates the design by performing FEA with predefined load cases. The results of the FEA reward the model for its action. Hence, good distributions will provide greater rewards. This feedback loop enables the model to learn the relationship between joining distributions and performance. By learning on many different product variants and load cases, the model can generalize joining element design.

FEA simulates various individual performance metrics. Theoretically, RL models can learn these simulations [404]. FEA is evaluated, for example, as seen in binary and size optimization methodologies. The results aggregate into a reward for the RL. The reward function explains the model's performance, enabling it to learn. Better joining locations comply more with performance requirements, and therefore, return greater rewards for the RL model. Nevertheless, both RL and FEA have very high computational costs as many FEAs must be run to enable learning. Furthermore, considerable development effort is required for meshing, modeling, and simulation.

RL is most similar to designers' learning and can generalize the results for product variants and load cases. It is the only approach that may become more effective than human capability and beat the Bayes error [284]. However, this is at the cost of enormous computational resources. Furthermore, to achieve a global optimum design, all factory and working processes require digital twins. Digital twins are virtual models that describe and are linked to real-world products [405]. Hence, they can enable one to perform simulations and optimize designs accordingly.

3.2.3 Joining parameter selection

The joining parameters include dimensions and supplementary information of joining elements, such as length, diameter, type of object, or additive material. These parameters are standardized to optimize manufacturing according to DFMA considerations. Joining parameter selection is a similar problem to JTS and can use the same methodologies. Fig. 3.14 is an excerpt from Fig. 3.8 presenting applicable AI techniques for predicting joining parameters.

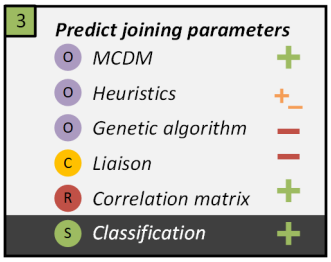


Fig. 3.14: The applicability of artificial intelligence for predicting joining parameters. The image is an excerpt from Fig. 3.8.

Joining parameter prediction is a classification task that suits SML. However, developing separate SML models for each technology's joining parameters is an exhaustive and ambiguous task [1]. Some exotic technologies may occur a few times in products but with very different parameters. Others, such as spot welding, are bread-and-butter technologies in, for example, the automobile industry [25]. One SML model created for all technologies would be more difficult to train through addressing specialist technologies.

Moreover, joining parameters and their locations are intertwined. Parameters influence performance in both design and production. Joining parameters have discrete and overseeable solution spaces. Due to their high degree of standardization, RBR and CBR are viable approaches. Furthermore, S&O can find an optimal set of parameters [168], but it maximizes some objective functions and may consequently favor exotic parameters. Such results would increase the variety in parameters with possible negative effects. Geda et al. [109] optimized technology and parameters simultaneously, thus reducing the need to create separate models but ignoring geometric boundary conditions.

3.2.4 Joining technology and parameter commonalization

To reduce product complexity, it is necessary to reuse joining elements over multiple product variants. Product variants often share components using product platforms, which is a transferable concept to joining elements. Commonalization redesigns joining elements to reduce their total variety. Hence, these joining elements need to be managed as components. Although, commonalization can apply to various levels in the product variety hierarchy (Fig. 2.9). With respect to joining elements, the framework identifies four commonalization design problems; the three joining aspects (technology, locations, and parameters) as well as joints themselves. This subsection addresses commonalizing joining technologies and parameters,

as visualized in Fig. 3.15.

Fig. 3.15: The applicability of artificial intelligence for commonalizing joining technologies. The image is an excerpt from Fig. 3.8.

This can reduce variability in manufacturing, practically automating the standardization guidelines of DFA. Here, RBR is a typical AI field for managing and controlling selection. Its effectiveness by was proven AlGeddawy et al. [85]. Other fields are less appropriate as RBR provides stability to the product architecture through its determinism, and deductive character [101, 237]. Rules can define actions or conclusions for simple IF-conditions, as Ukala and Sunmola [237] proposed. For example, IF there are rivets of different diameters, AND there is a sufficient area for replacing small-diameter rivets with larger ones, then DO recommend using larger rivets for all holes [237]. This approach is further addressed and detailed in Section 3.3.6.

3.2.5 Joining location commonalization

Joining locations describe continuous geometry, in contrast, to discrete classifications for joining technologies. Regardless of interchangeable components, local geometry remains the same for joining locations, which enables their reuse. However, predictions may have small perturbations in locations. As depicted in Fig. 3.16, clustering (UML) the joining locations of individual variants creates shared locations and reduces geometric variability.

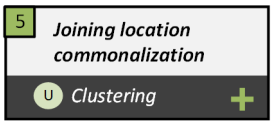


Fig. 3.16: The applicability of artificial intelligence for reducing variability in joining locations. The image is an excerpt from Fig. 3.8.

Cluster algorithms can consider constraints (e.g., [406]) to comply with manufacturability requirements. This approach is further addressed and detailed in Section 3.3.7. Furthermore, S&O and RBR are appropriate to lesser degrees and require unnecessarily high development effort to consider all use cases.

3.2.6 Commonalization of joints

The state of the art provides multiple applicable approaches for the commonalization of joining elements; however, they were not designed specifically for this design problem. Fig. 3.17 presents the applicable AI techniques for commonalizing joints.

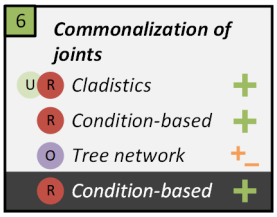


Fig. 3.17: The applicability of artificial intelligence to reduce variety in entire joints. The image is an excerpt of Fig. 3.8.

RBR typically reduces the differences in designs between two product variants of similar parts. It requires the joints to be viewed as individual modules. Condition-based approaches (e.g., [237]) can reduce process variety on each joint, similar to joining technology and parameter commonalization. This variety reduction includes taking over the joining elements of one joint and using them for others. However, a mere condition-based methodology would require much developmental effort to create the RBR algorithm and might not be able to cope with the required complexity. The task requires a more holistic view toward other product variants to adapt adequately.

Moreover, a UML with RBR approach searches functional overlap between product variants [85]. Joining elements also require the physical domain and explicit consideration in determining relationships between components. This approach is further addressed and detailed in Section 3.3.8. Furthermore, S&O approaches are also viable through using a tree network optimization [86], but they are a rather cumbersome methodology for this relatively simple task.

3.2.7 Joining module creation

The last design problem in the framework creates subassemblies and modules to manage product variety. Modularization is a trade-off between larger cost-saving modules and smaller flexible modules. Defining joining elements as components within modules enables them to be standardized, but potentially at the cost of commonality and DFA [87]. Fig. 3.18) presents applicable AI techniques for creating joining modules.

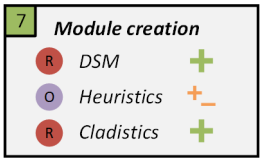


Fig. 3.18: The applicability of artificial intelligence for creating joining modules. The image is an excerpt from Fig. 3.8.

S&O algorithms can balance these requirements and evaluate every candidate module. This approach is further addressed and detailed in Section 3.3.9. RBR is feasible and its applicability has been demonstrated in the state of the art (e.g., [75]) despite high development and maintenance effort; however, due to its deterministic nature, it can stabilize results. UML

could work as well; however, it requires many constraints and its results lack transparency. Additionally, UML has not been implemented in modularization.

The previous section evaluated AI fields for their ability to solve the design problems in the VICTOR framework. Based on the evaluation and promising AI fields, the following section presents new AI methodologies for automating joining element design.

3.3 Novel AI methodologies for automating joining element design

This section presents new methodologies to fill the current research gaps. These include new methodologies for adding to the structure block of the framework (Section 3.1.3). Furthermore, they include methods derived from the identification of unused AI fields for various design problems (discussed in Section 3.2). The first four subsections concern the prediction of joining locations, which is the most challenging task (compared with the other joining aspects) to perform with ML. This section presents methodologies that implement an evolutionary algorithm (Section 3.3.1), SML (Section 3.3.2), and GANs (Section 3.3.3). Then, Section 3.3.4 presents an approach for evaluating predictions of joining locations. After presenting new joining location prediction methodologies, this section continues with five subsections that regard the modular design for joining elements. Firstly, Section 3.3.5 presents an overview of novel modular design methodologies for joining elements as well as discusses the required preprocessing considerations. Then, Sections 3.3.6, 3.3.7, and 3.3.8 present methodologies for commonalizing joining technologies and parameters, locations, as well as entire joints, respectively. Lastly, Section 3.4 discusses the modularization of joining elements.

3.3.1 Joining location prediction using randomized optimization

This methodology is based on the algorithmic fitting approaches presented in Section 3.2.2. Specifically, it implements an evolutionary algorithm. Algorithmic fitting distributes a number of joining elements evenly over a CR. This blunt geometrical distribution approach neglects requirements of the product as a whole; however, an even distribution generally creates a strong static performance of spot welds on normal directions. Furthermore, standardized mutual distances of joining locations support commonalization as the designs are not specific to particular product variants. It is a simple approach for benchmarking complex prediction methodologies. Additionally, it may make designers' involvement in creating guidelines for RBR approaches obsolete. The approach requires the number of joining elements, implying their technology, as the input. This number determines the complexity of the distribution and may originate from designers, but also from prediction algorithms using any field of AI. Gerlach [407] implemented and prototyped this algorithmic fitting approach. The approach assumes that the optimal distribution of joining locations implies maximizing the mutual distances and edge distances, as illustrated in Fig. 3.19.

This enables one to define a function that sums the distances of all joining locations to the nearest other joining location s_{nn} and nearest edge s_e; see Eq. 3.1, where n_{je} is the number of joining locations. Fig. 3.20 visualizes an example to calculate the objective function of a

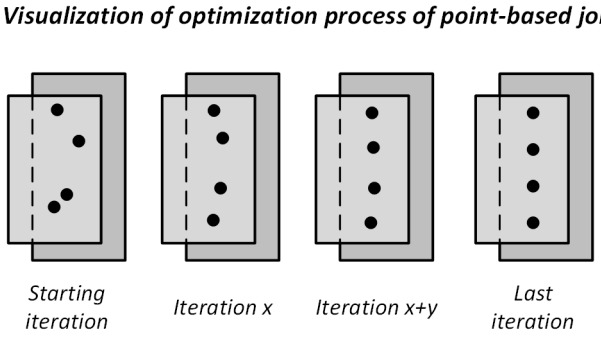


Fig. 3.19: An exemplary visualization of an optimization process for distributing joining locations.

sample image.

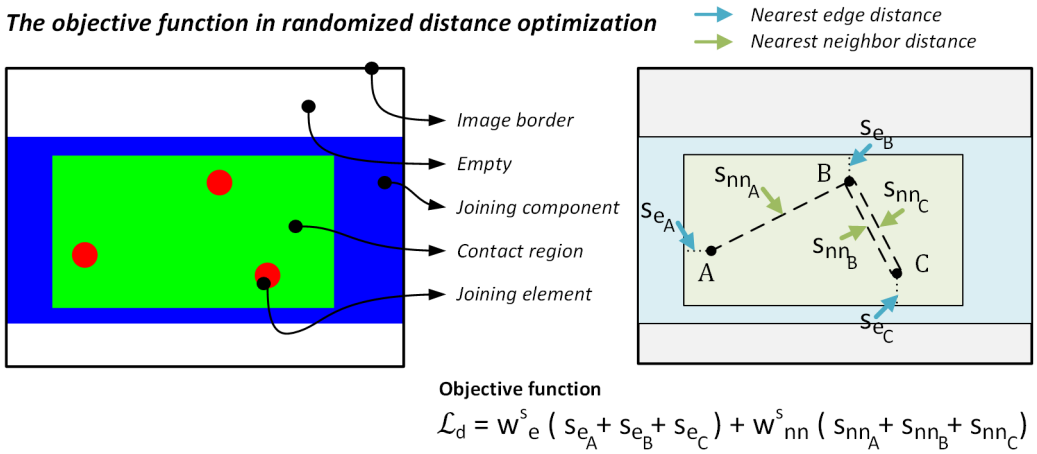


Fig. 3.20: An example of calculating the objective function $\mathcal{L}_d$.

Using the variables depicted in Fig. 3.20, Eq. 3.2 presents the Euclidean implementation for distance measurements. The coordinates p_{ex} and p_{ey} describe the closest edge point to joining location j^l; the coordinates p_x and p_y describe the i-th joining location; and lastly, p_{nnx} and p_{nny} are the coordinates of the closest (nearest neighbor) joining location p_{nn}. All coordinates are real numbers: $p_{ex}, p_{ey}, p_x, p_y, p_{nnx}, p_{nny} \in \mathbb{R}$. Weights for edge w^s_e and nearest neighbors w^s_{nn} enable the contribution of each term to be tuned.

$$\mathcal{L}_d = \sum_{i=0}^{n_{je}} \left(w^s_e s_{ei} + w^s_{nn} s_{nni} \right) \tag{3.1}$$

$$\mathcal{L}_d = \sum_{i=0}^{n_{je}} \left(w^s_e \sqrt{(p_x - p_{ex})^2 + (p_y - p_{ey})^2} + w^s_{nn} \sqrt{(p_x - p_{nnx})^2 + (p_y - p_{nny})^2} \right) \tag{3.2}$$

Reaching the global maximum $\mathcal{L}_d$ implies that locations are the furthest from other points and edges for a given sample. Hence, the algorithm converges and theoretically outputs an even distribution. The distance function is similar to those defined in land-use optimization problems by Mohammadi et al. [272] and Schwaab et al. [273].

Suppakitpaisarn et al. [408] used a gradient-based optimization methodology. However, Gerlach [407] demonstrated that the distance-based function of Eq. 3.1 is not differentiable. The change of distance values jumps when a point is reassigned from one closest point to another. The change rate of distances is not continuous; hence, the function is not differentiable, preventing a conventional gradient-based optimization. Gerlach [407] proposed a basic evolutionary algorithm that makes iteratively random pixel-wise steps of joining locations and evaluates whether $\mathcal{L}_d$ increases. An optimization of $\mathcal{L}_d$ makes the algorithm use the new solution for the next iteration. In case the values did not increase, the algorithm uses the old solution. Hence, the methodology rewards successful generations with further optimization and removes unsuccessful ones, equivalent to evolution in nature.

Table 4.5 presents the evolutionary algorithm for point-based joining elements. It also includes the properties of RI of locations and learning rate decay, which are defined as follows:

- **Uniform RI** on the CR enables the joining locations to initialize relatively close to the target joining locations without considering the input geometry, especially compared with the initialization of all locations in the middle of the CR. Gerlach demonstrated that the initialization of locations in the center of CRs leads to slow optimization progress [407].
- A high **learning rate** (using large steps) boosts the first set of iterations to take larger steps, thus increasing the objective value more rapidly. However, this is at the cost of overshooting, as the algorithm may position new points outside of the CR. Lower learning rates enable little tweaks to locations in the latter optimization iterations. Learning rate decay slowly reduces the step size. After a specified number of iterations, the learning rate will drop by, for example, half its previous value.

An exemplary process for distributing joining locations using randomized optimization is depicted in Fig. 3.21. The following list elaborates each step:

1. One starts with a dataset that contains input and target samples with 2D snapshots of CRs and spot welds. The process first retrieves the number of spot welds from the target image, so the algorithm knows how many spot welds it needs to distribute.
2. The target samples are only of use during evaluation. The input samples continue into the prediction process.
3. RI creates random points on the CRs. The initial objective value is calculated using Eq. 3.1 and the weights of the specific model.
4. The joining locations and objective value are stored as an intermediate result. This storage helps to calculate whether the following iteration improves.

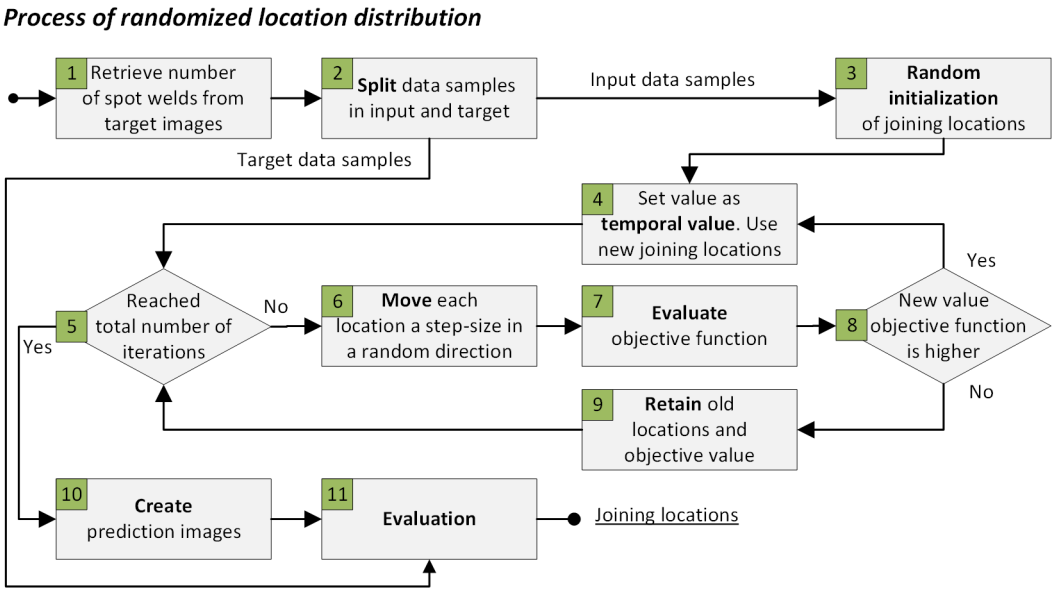


Fig. 3.21: A flowchart that visualizes the process of randomized joining location distribution.

5. As long as the total number of iterations is not reached, the algorithm moves each joining location independently in a direction (up, down, left, or right) with a step size equal to the learning rate. The step size depends on the number of passed iterations.
6. The new objective value of the new locations is calculated.
7. Once the new objective value is better than the temporal one, the algorithm uses the new joining locations and the objective value for the next iteration. If the new locations create a worse objective value, the algorithm uses the old locations and objective value for the next iteration.
8. Once the total number of iterations is reached, the algorithm creates the final output images. The evaluation step compares the predicted images with the target images to determine the performance.

The complexity of the problem scales exponentially. Every joining location has nine potential new locations after the iteration, one for every adjacent grid cell plus the original cell. Hence, every iteration the algorithm for five joining locations randomly picks one of $5^9 = 1.953.125$ possible solutions. The movement of one joining location creates a more effective solution for many distributions. Simultaneously, other joining locations may reduce the objective function, cutting out the one better joining location. This effect causes CRs to have many joining locations, which may significantly slow the optimization process. This breaking effect leads to more iterations for larger problems. Additionally, it requires the joining locations on the outside to first move to create room in the middle for redistribution.

The algorithm can only include manufacturability requirements at the end of optimization. RI and optimization may include many invalid solutions. These non-manufacturable joining locations may be necessary for achieving more effective solutions. Including a

manufacturability check during optimization could prevent the creation of any new solutions at all. Hence, one check at the end of optimization would either result in a correct result or a requirement to restart the distribution of joining locations on that CR. Compliance to, for example, the edge distance can also be considered at the start of the algorithm by offsetting the outline of the CR by the minimum edge distance.

Curves require additional constraints for optimization. Discretized curves through points quickly have too many points for the evolutionary algorithm due to its scaling complexity. The discretization of corner and edge points reduces the calculation load tremendously. Constraints enable one to move the points simultaneously, for example, to respect the length of each curve. The algorithmic fitting of curves is performed on the edge points, although for nonstraight lines, curves need middle points.

Instead of using an evolutionary algorithm, the following subsection presents a methodology that implements SML to predict joining locations.

3.3.2 Joining location prediction using encoder-decoder architectures

This section addresses concepts that implement SML. One approach was presented in Section 3.2.2, namely grid-based drawing. This section also presents both a segmentation and regression approach for predicting joining locations. Next, it discusses data formatting and computational cost requirements as well as the similarity of the model between 2D and 3D data representations. Lastly, it presents a methodology for implementing multimodality into the models.

The most critical aspect for data representations of geometry for predicting joining locations using ML is the reconstruction of coordinates, which is explained in more detail in subsection 3.3.4.1. The discrete output of an ML model must be transformed into a continuous coordinate system. Unstructured 3D representations may create different data samples for the same joining scenario as the techniques include randomness. Then, input data mapping becomes inconsistent with the fixed inputs of a model. For example, point clouds are inconsistent when the same 3D surface is sampled multiple times. The SML approach uses Euclidean structures to model data, which enables joining locations to be reconstructed from Cartesian coordinates.

The postprocessing of a model's output forms the joining locations. The model requires an EncDec architecture to limit the computational cost and construct joining locations. An autoencoder only aims to reconstruct the input from an information-dense latent vector. However, it can generate data in the decoder by using different targets, as the autoencoder typically does. Then, the latent space will contain the essential information of an input, from which the decoder reconstructs coordinates. EncDec architectures have been implemented not only in SML models for segmentation tasks (e.g., [409–411]) but also in coordinate regression (e.g., [402]). The underlying idea is to provide the model with a semantic understanding of the joining scenarios by giving geometry as the input and both geometry and joining locations as the output. Hence, the encoder creates a latent vector of the input data, and the decoder takes this and reconstructs the input together with joining locations.

The SML concepts in the following subsections were theorized by Eggink et al. [2], detailed by Perez-Ramirez [412], and subsequently published by Eggink et al. [3]. Both the

classification and regression concepts theoretically work regardless of Euclidean data representations. The concepts define pixels or voxels as grid cells and images or voxel-grids as grids. Moreover, the concepts use both a width W and height H dimension, and the volumetric implementation includes a depth D dimension. The equations include all three dimensions for completeness and readability. After the voxel concept, the subsection discusses concept specifications for the multiview approaches.

$$x^{(i)} \in \mathbb{R}^{W_x \times H_x \times D_x},\ \forall \{x^{(i)}\}_0^n; \qquad y^{(i)},\ \hat{y}^{(i)} \in \mathbb{R}^{W_y \times H_y \times D_y},\ \forall \{y^{(i)}\}_0^n \tag{3.3}$$

The generalized matrix $x^{(i)}$ represents the formatted input geometry (Eq. 3.3) for the data sample (i) and similarly for the target $y^{(i)}$ and prediction $\hat{y}^{(i)}$ (Eq. 3.3). The input, target, and prediction grids depend on the geometry's level of detail and sizes. The resolution r expresses the size of each grid cell (i.e., in $mm/pixel$ or $mm/voxel$). The dimensions of geometry must consider thickness. For example, the minimum mutual joining distance for spot welding on steel sheets is $20\ mm$ [115]. This would force a maximum cell size of $10\ mm$ to prevent sampling issues considering the Nyquist–Shannon sampling frequency requirement [413]. The grid sizes W, H, and D depend on the size of the geometry in the data samples.

The following sub-subsections give structure to the presentation of the methodology for predicting joining locations using EncDec architectures. Firstly, they present concepts for classification and regression in sub-subsections 3.3.2.1 and 3.3.2.2, respectively. Then, sub-subsection 3.3.2.3 discusses the formatting of data samples, which require filtering to remove similar data samples, as sub-subsection 3.3.2.4 addresses. Furthermore, sub-subsection 3.3.2.5 argues that 2D and 3D approaches using this methodology are equivalent to one another. Lastly, sub-subsection 3.3.2.6 presents the implementation of PMI into geometric data samples using a branding approach.

3.3.2.1 Segmentation concept

Segmentation is a multiclass classification approach and aims to classify grid cells that belong to joining locations. It uses the number of geometry occurrences in each grid cell to distinguish between classes. The following list presents the four classes of geometry occurrences:

- **Class 0** describes empty voxels that represent the absence of geometry;
- **Class 1** represents the geometry of a component;
- **Class 2** describes CRs as two components that coincide in the same grid cell counting two geometry occurrences;
- **Class 3** represents joining locations as they add a third geometry into the same grid cell.

The definition of the four classes enables to determine a data formatting concept. Fig. 3.22 presents exemplary data formatting for the segmentation concept.

It depicts a joining scenario with two components and joining elements. Each grid cell gets the class for the number of geometries with which it coincides. The figure provides examples of input and target grids for a volumetric and image-based approach. The model needs to segment class 3 from the input grids. Target grids are one-hot encoded along with the classes. One-hot

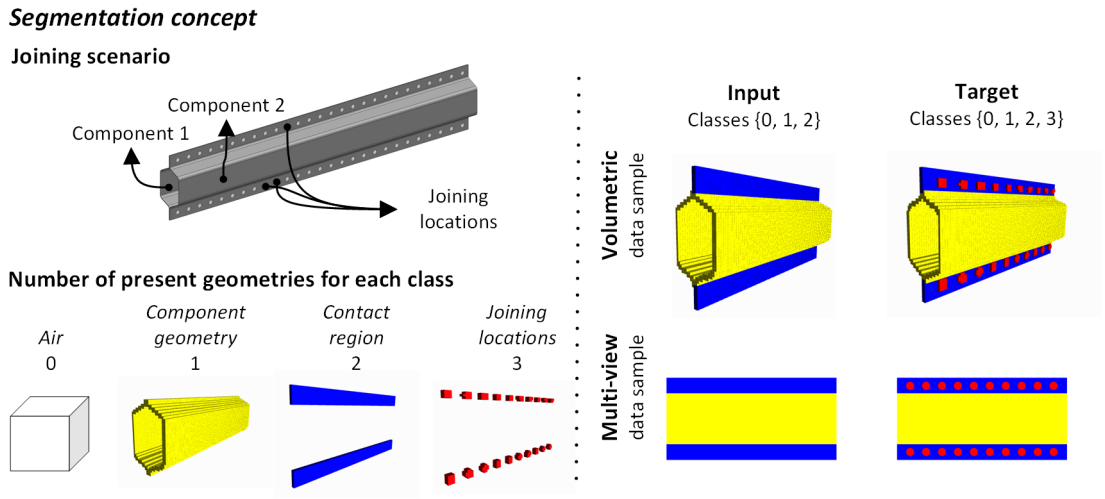


Fig. 3.22: Visualization of data formatting for segmentation concepts. The top of the image depicts a joining scenario where two CAD components are spot welded at top and bottom flanges. Each case encodes classes according to the number of geometries that occupy a voxel. The bottom part of the image presents representations of input and target grids.

encoding discriminates one from others over groups of bits [275]. Each group has one positive bit ($\equiv 1$) and the rest are zero ($\equiv 0$). Each bit represents one label. Thus, each grid cell in the target has four binary values, one for every class.

$$\mathcal{L}^{seg} = \sum^{W}\sum^{H}\sum^{D}\sum^{C} \alpha_c y_{w,h,d,c} \ln \hat{y}_{w,h,d,c} \tag{3.4}$$

The loss function $\mathcal{L}^{seg}$ is the sum of all grid cell-wise weighted cross-entropy losses across dimensions W, D, and H as well as classes C. Equation 3.4 expresses the loss function for one data sample, where $\hat{y}_{w,h,d,k}$ is the prediction of grid cell position w, d, h for class c and $y_{w,h,d,k}$ is its target value. A normalized factor a_c weighs grid cell classifications for class $c \in 0, 1, 2, 3$ with $a_0 < a_1 < a_2 < a_3$. Weights a_c consider the occurrence and priority of classes over the entire dataset; see Eq. 3.5.

Skewed class distributions reduce the ability to identify minority classes, yet class 3 grid cells are most crucial to predict correctly. Weighing is vital for ensuring that the loss of nonrelevant regions (i.e., class 0 or 1) has a lower impact on the learning than highly relevant joining regions (i.e., class 2 or 3); see Eq. 3.5, where n^M is the total number of data samples, and $n_{c,i}$ represents the number of voxels with class c for data sample i. Furthermore, a scaling factor β_c enables the arbitrary tuning of each class c. Additionally, the weighting could suffice by taking the inverse class frequency [414].

$$a_0 = \frac{\beta_0}{n^M} \sum^{n^M} \frac{n_0}{\sum_{c \in C} n_{c,i}} \qquad \forall c \in \{0\} \tag{3.5}$$

$$a_c = \frac{\beta_c}{n^M} \sum^{n^M} \frac{n_{c,i}}{n_{1,i} + n_{2,i} + n_{3,i}} \qquad \forall c \in \{1, 2, 3\} \tag{3.6}$$

3.3.2.2 Regression concept

The probability mapping concept implements a grid cell-wise regression task instead of classification. Each grid cell has one continuous value in the range of $[0, 1]$. A cell with zero probability represents empty space, whereas a 100% probability value represents joining locations. One component geometry (class 1) has very low probabilities, whereas two-component geometry (CRs, class 2) assigns probabilities depending on distances to joining locations using a distance dependency function [415]; see Eq. 3.11. Fig. 3.22 presents exemplary data formatting for the regression concept.

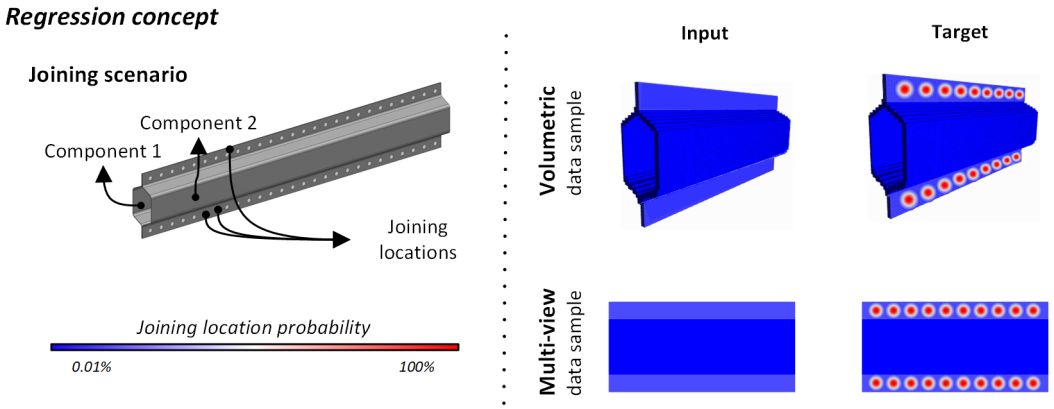


Fig. 3.23: Visualization of data formatting for regression concepts. The top of the image presents a joining scenario where two CAD components are spot welded at top and bottom flanges. The colors indicate the probability of a joining location being present at that pixel or voxel. The bottom part of the image presents representations of input and target grids.

Similar to segmentation, the regression methodology distinguishes four spatial regions: empty, geometry, CR, and joining locations. These regions enable a spatially dependent probability matrix $\tilde{T}$ to assign probabilities to grid cells. Additionally, this Boolean matrix aids in tuning the loss function as it weighs the attributions of individual regions for each prediction. The size of the matrix is equal to the target grids $\tilde{T}^{(i)} \in \mathbb{R}^{W_y \times H_y \times D_y}$.

- Zero-probability regions (i.e., class 0) correspond to nongeometry-occupying voxels. For a given sample, the matrix $\tilde{T}$ for class 0 has a value of 1 for those cells where the index w, h, d has a value of 0 in the target.

$$\tilde{T}_0 = (y_{w,h,d} == 0) \tag{3.7}$$

- Low-probability regions (i.e., class 1) correspond to geometry-occupying voxels that have no direct contact with joining locations (i.e., sheet metal). The threshold τ_1 creates a mask to limit the geometry-describing probabilities. For a given sample, the matrix $\tilde{T}$ for class 1 has a value of 1 for those cells where the index w, h, d has a value strictly larger than 0, but strictly lower than τ_1 in the target.

$$\tilde{T_1} = (y_{w,h,d} > 0) \ \wedge \ (y_{w,h,d} < \tau_1) \tag{3.8}$$

- Medium-probability regions (i.e., class 2) correspond to geometry occupied by both components and describe CRs. The thresholds τ_1 and τ_2 create a mask to define this region. For a given sample, the matrix $\tilde{T}$ for class 1 has a value of 1 for those cells where the index w, h, d has values larger than or equal to τ_1, but strictly lower than τ_2 in the target.

$$\tilde{T_2} = (y_{w,h,d} \geq \tau_1) \ \wedge \ (y_{w,h,d} < \tau_2) \tag{3.9}$$

- High-probability regions (i.e., class 3) correspond to joining locations. The threshold τ_2 describes the boundary of the probabilities that differentiate between CR and joining location. The center of a joining location has a 100% probability. For a given sample, the matrix $\tilde{T}$ for class 1 has a value of 1 for those cells where the index w, h, d has a value larger than or equal to τ_2 in the target.

$$\tilde{T_3} = (y_{w,h,d} \geq \tau_2) \tag{3.10}$$

The probabilities in the target grid rise from CR values to a joining location value. The distance-dependency function [415] creates a bell shape of increased probabilities around the joining location. At a greater distance from the nearest joining location, the probability of the corresponding cell reduces. The formula determines the target probability value of grid cell $y_{w,d,h}$ if its center point $\bar{p}_{w,d,h}$ is within Euclidean distance s_{md} of the nearest joining location p_{near}; see Eq. 3.11. The factor τ_d describes the width of the bell shape. Higher values of τ_d create smoother and larger probability curves but potentially reduces the model's ability to discriminate joining locations on CRs. Grid cells on a CR that are further away than distance s_{md} of the nearest joining location p_{near} receive a regular CR probability $P(c = 2)$ (class 2).

$$y_{w,h,d} = \begin{cases} \frac{1}{1+\frac{d}{\tau_d}} & \|\bar{p}_{w,h,d} - p_{near}\|_2 < s_{md} \\ P(c=2) & \text{otherwise} \end{cases} \tag{3.11}$$

The target contains gradients of probability values in joining location neighborhoods. Cross-entropy-based loss functions are less appropriate as they tend to push predictions toward extremes (ones or zeros). The regression concept considers spatially dependent weights to emphasize grid cells. The loss function $\mathcal{L}_{reg}$ is the sum of weighted mean squared errors per probability range c using Boolean spatial matrices $\tilde{T_c}$. The matrix $\tilde{T_c}$ has true ($\equiv 1$) values on indexes w, d, h, where the probabilities fall within a specified probability range using the boundaries τ_1 and τ_2. It is false ($\equiv 0$) otherwise. Weights a_c can be calculated

similarly to those in the segmentation concept using Eq. 3.5. Prediction $\hat{y}_{w,d,h}$ and target $y_{w,d,h}$ represent joining location probabilities at position w, d, h. Equation 3.12 expresses the loss function for one data sample.

$$\mathcal{L}_{reg} = \sum^{W}\sum^{H}\sum^{D}\sum^{C} \alpha_c \left(\tilde{T}_{w,h,d,c}\left(\hat{y}_{w,h,d} - y_{w,h,d}\right)\right)^2 \tag{3.12}$$

3.3.2.3 Data formatting for encoder-decoder architectures

Component geometries vary greatly in size and complexity. This geometric variety causes mapping issues and high data dimensionality [2]. Components may have any size and shape. For example, there are large components with a few features (e.g., panels on airplane wings) as well as small and complex components (e.g., holders). Furthermore, ML models have predefined input sizes. Products may have joining scenarios of any size that must all fit the model's input size. Naturally, a large input size enables one to input entire joining scenarios, but this results in high dimensionality and massively increases the computational cost [2]. Moreover, small components have many empty grid cells, unnecessarily affecting training performance. Reducing the level of detail during the discretization of geometry may lead to information loss, losing the ability to represent small features.

Splitting joining scenarios into smaller chunks creates a trade-off between the amount of information in each sample and computational cost. Fig. 3.24 illustrates the splitting of joining scenarios into connection cases (CCs).

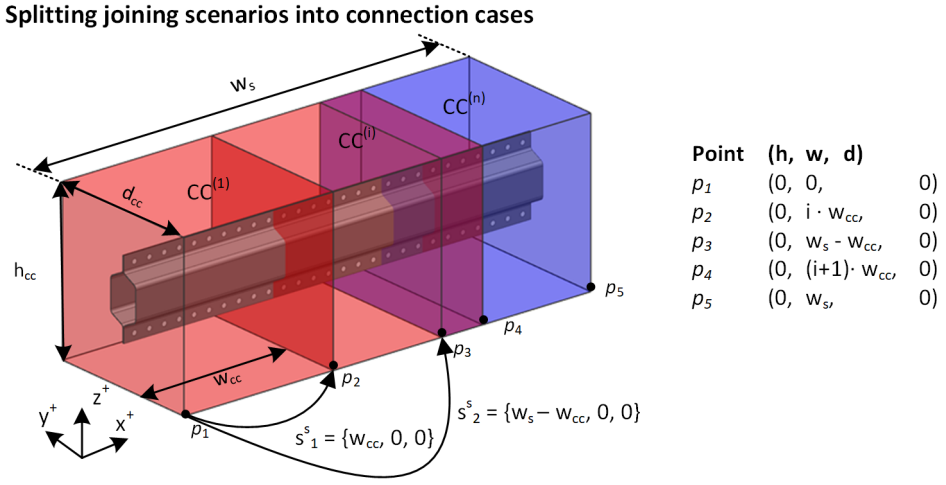


Fig. 3.24: Visualization of splitting joining scenarios into connection cases to balance between the level of detail and grid size.

The EncDec concept views each chunk as a data sample. The chunks need to include enough surrounding geometry to describe the form and function of components in a joining scenario. A grid is the Euclidean data representation of a chunk; hence, it can be 2D or 3D. A **connection case** is a data sample consisting of an input and a target grid.

Referring to Fig. 3.24, each CC has a size w_{cc}, h_{cc}, d_{cc} and is geometrically centered in the grid. A stride s^s describes the step size of CCs in all grids' axes $s_i^s = s_w^s, s_d^s, s_h^s$. As such, this stride is equivalent to the stride used in CNNs that describe the movement of a kernel over an image. The "strided" position represents the origin for the strides of each axis $p_i^s = j, k, l$.

The CCs in Fig. 3.24 have a stride of $s^s = w_{cc}$ that moves over the joining scenario. Small values and variations in stride distance can create more training data and be part of a data augmentation strategy. The last CC per axis is created in the opposite direction, spanning inwards from the maximum extent (see CC_n in Fig. 3.24). A filter for added geometry ensures the CCs at least bring in a fraction τ_s of new data of the joining scenario. Hence, the distance between points $p4$ and $p5$ must be larger than $\tau_s \times w_{cc}$.

3.3.2.4 Similarity filter for data samples

A drawback of the splitting process is that it might return similar CCs. Fig. 3.24 indicates that CC_1 and the other CC_i will contain similar geometry. Joining locations may differ, but the component geometries overlap. Including both cases in training may result in data leakage and noise. First, data leakage may occur as the unseen test set may contain similar samples as the training set, resulting in inaccurate performance measurements. Second, noise may occur as the same input grids would correspond to different target grids, lacking consistent input–output mapping.

The Jaccard index [315] helps to identify similarities between CCs. The Boolean matrix G is the sum of Boolean matrices of classes 1, 2, and 3; see Eq. 3.13. It is equal to the input grid with all values larger than or equal to one.

$$G = \tilde{T_1} + \tilde{T_2} + \tilde{T_3} = (x_{w,h,d} \geq 1) \tag{3.13}$$

The Jaccard index J must be less than the threshold value τ_J; see Eq. 2.7. It removes all CCs from the dataset that are too similar with Jaccard index values above τ_J. However, joining scenarios with thin components, such as sheet metals, may not have robust similarity filtering results. Small perturbations might cause similar CCs to have a low Jaccard index, such as when CCs shift by one grid cell. Morphological methods such as binary dilation [416] can buffer shapes and increase robustness in similarity analysis.

For example, matrix D is a $d \times d$ matrix with ones. Dilating matrix G by D offsets the surface of ones in G. An example is presented in Eq. 3.14.

$$D = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix} \quad G = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} \quad G \oplus D = \begin{vmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} \tag{3.14}$$

Eq. 3.15 presents the similarity criterion based on the Jaccard index and dilation of CCs A and B. It incorporates the dilation into Eq. 2.7 and removes one of the CCs from the dataset when the index surpasses τ_J.

$$J(G^{(A)}, G^{(B)}, D) = \frac{\left|(G^{(A)} \oplus D) \bigcap (G^{(B)} \oplus D)\right|}{\left|(G^{(A)} \oplus D) \bigcup (G^{(B)} \oplus D)\right|} < \tau_J \tag{3.15}$$

As a side note, various techniques exist for enlarging the dataset. Data augmentation methods such as mirroring and rotation are applicable. They perform transformations on both joining scenarios as CCs. Notably, newly created data samples add information to the dataset and should be sufficiently dissimilar to other CCs. For example, mirroring components on their symmetry axis may still create CCs with similar geometry. Hence, augmentation requires a subsequent similarity analysis as previously described, and also for the same reasons.

3.3.2.5 Equivalency between 2D and 3D data representations

Additionally, Euclidean data structures enable the implementation of benchmark architectures from computer vision. Both the multiview (2D) and volumetric (3D) approaches rely on Euclidean data structures and only differ in the third dimension. Both representations have a fixed grid-like structure with spatial dependencies that hold numerical values. Fig. 3.25 presents an example of the segmentation approach.

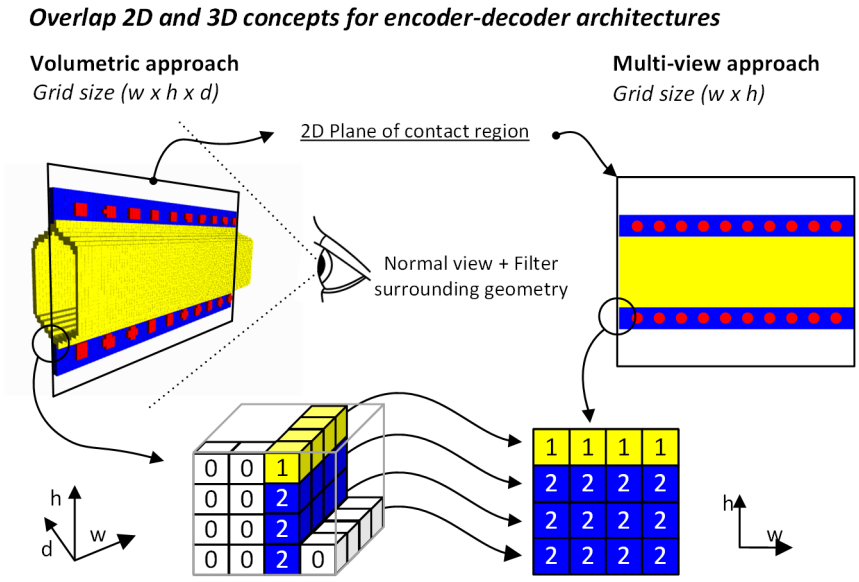


Fig. 3.25: Example of the similarity of training between a 2D and a 3D concept.

Su et al. [358] argued that single 2D images might reach a far higher accuracy than state-of-the-art 3D shape descriptors. Furthermore, within their context, they found that a single 2D view provided better results than a single view with a depth map. Multiview-based approaches often aim to recognize and detect entire objects. However, the prediction of joining locations focuses on specific areas of these objects. Hence, arbitrary rotations around an object, as performed using the benchmark classifier RotationNet of Kanezaki et al. [417], would work counterproductively for this study.

Furthermore, multiview methods with arbitrary cameras, for example, that rotate around an object, may occlude CRs and prevent the separation of components. Moreover, they introduce views with perspectives that require the handling of depth with a complex reconstruction of coordinates. Determining the Cartesian coordinates of pixels in an image with a perspective is complicated and requires approximation [418]. The same coordinate predicted in multiple images requires filtering to prevent the same coordinate being created twice.

Therefore, purposefully located 2D images prevent randomness and complex postprocessing; see Fig. 3.25. CRs are of interest in joining location prediction. A top-down camera view on the surface creates consistent and deterministic results. The 2D grids are specific slices taken from a 3D grid, but only the slices that contain joining elements. This remains a multiview approach as joining scenarios may have multiple CRs that may require multiple 2D images to represent them.

Ahmed et al. demonstrated that projecting curved surfaces to create flat 2D images enables their processing in deep learning [350]. Furthermore, CRs may be larger than one image can capture due to their resolution and size. Again, this requires a splitting process to create the CCs. Another drawback of the 2D approach is that data samples retain less geometric information of joining scenarios. For example, the 2D image in Fig. 3.25 cannot represent the joining of two U-profiles. Hence, it may require additional supplementary information through, for example, using a multimodality approach. Figs. 3.22 and 3.23 visualized exemplary input and target images.

3.3.2.6 Multimodality in encoder-decoder architectures

The presented SML concepts exclusively concern geometry. However, nongeometric data include PMI and provide descriptive information for designers when creating joining locations. For example, different materials and thicknesses of components require different mutual joining location distances. Expanding on existing concepts for multimodality requires this information to be processed using the same architectures. Early fusion approaches supply information to models before training, enabling them to find exploitable relationships between the modalities. CNNs consider spatial features, and hence, a separate classification or regression mapping between the input and output of the same data requires the same network to perform two separate tasks.

Pelka et al. [376] presented a branding methodology that proposed using early fusion through clustering text features and branding them as patches into images. These patches worked equivalently to one-hot encoding of labels. Pelka et al. used small patches on the image instead of singular bits. Similarly, each brand can represent one cluster, bringing the information of joining scenarios to CCs. The branded labels express information without creating additional classes for prediction or relevant, nonspatial regression tasks. Fig. 3.26 illustrates labels (black boxes) branded onto exemplary data samples of the segmentation concept.

The figure depicts branding for 2D and 3D segmentation tasks by presenting a separate fifth class to grids, indicated by the black boxes. Both the sample's input and target contain branded information. This forces the model to reconstruct these labels and take up the information into latent feature vectors. It is assumed that the lack of labels in the target would train the model to

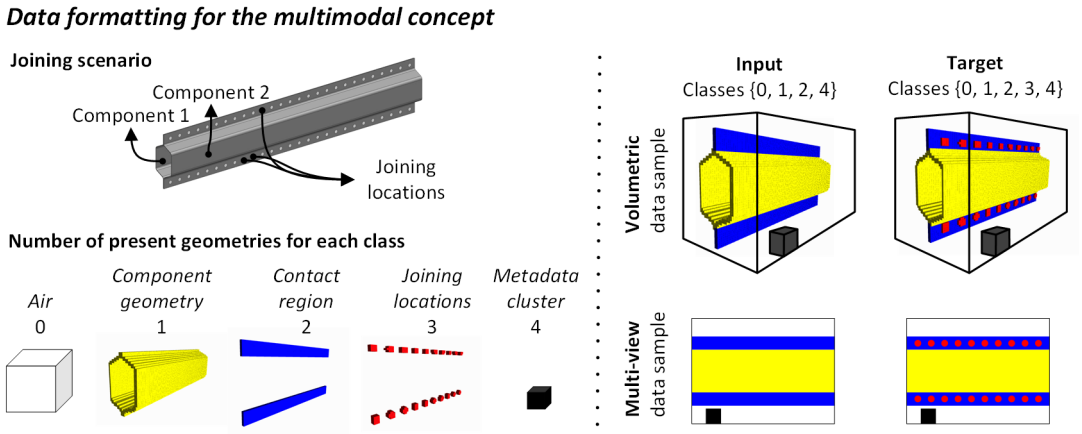


Fig. 3.26: Data formatting for multimodal machine learning using branded product manufacturing information on geometric data for a segmentation approach; based on Fig. 3.22.

remove the brand. This prevents these prediction results being considered in the loss function and the model being evaluated on them. The reconstruction of labels would train the model to find the relations between the input and target grids.

Nongeometric data for joining can be high-dimensional. It includes many heterogeneous types of materials and free text nomenclature that describes the functions of components. Linear clustering algorithms have difficulty with high-dimensional data. For example, k-means should not be used above a dimensionality of 10 [419] to prevent high-dimensional nonlinear behavior. Data reduction methods extract the essential information yet enable clustering algorithms to find clear and meaningful distinctions between sets of joining scenarios. After DR, clustering creates labels for branding. Each label position represents a cluster. Joining scenarios belonging to multiple clusters receive multiple labels [376], each of which has a fixed position on the edge of a grid spanning the entire width.

Linear methods easily find the nearest cluster for an unseen data point. Nonlinear data reduction methods, such as t-SNE, transform space to preserve local neighborhoods. New joining scenarios that do not fit this space do not obtain a consistent input–output mapping. These scenarios would transform space locally and change the clustering result. Van der Maaten [420] solved this problem by using an NN to reconstruct the functionality of t-SNE. Similarly, random forests can learn the mapping of input and output samples to transform unseen samples onto a nonlinear data reduction space [421].

Instead of using an EncDec architecture, the following subsection presents a methodology that implement a GAN to predict joining locations.

3.3.3 Joining location prediction using generative adversarial networks

In Section 3.2.2, the grid-based drawing approach was presented. However, this methodology does not use SML, but rather GANs. GANs are a benchmark approach for generative models in computer vision. Image synthesis methods generate new images using an input image and a style. Similarly, GANs can take input images with geometry (e.g., the left columns of Figs.

3.22 and 3.23 and apply a "joining location" style from a similar CC to predict joining locations. Gerlach and Eggink [5] presented this experimental methodology, which is based on the reference- and latent-guided synthesis as proposed by Choi et al. [331]. Firstly, reference-guided image synthesis enables style transfer through the selection of *one image*, as visualized in Fig. 3.27.

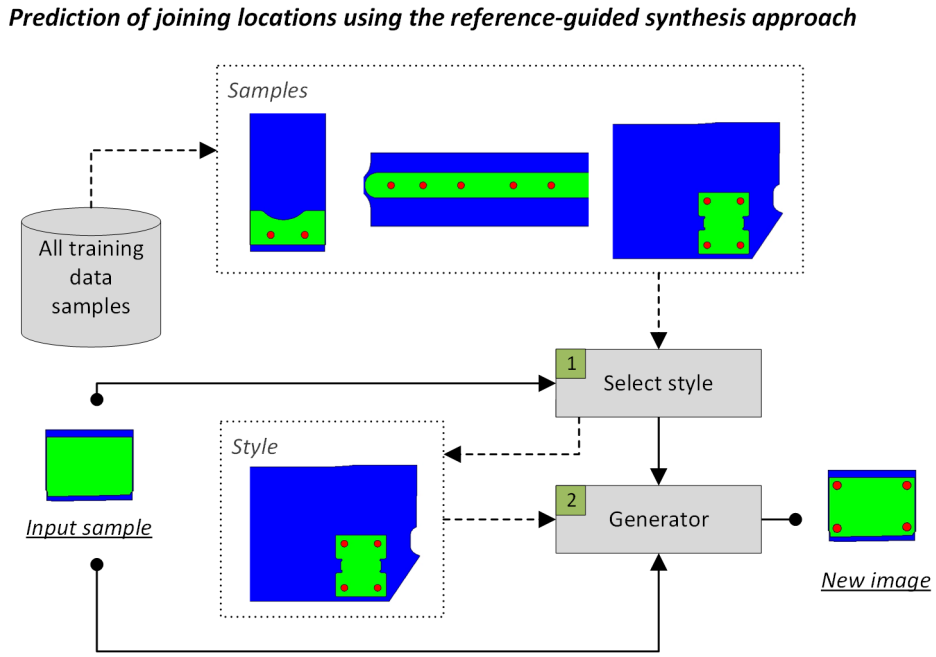


Fig. 3.27: Example of the process and results when implementing reference-guided image synthesis. The geometry of input samples apply the style information of similar data samples to synthesize new data samples.

The **reference-guided approach** applies preselected styles as joining location inspiration on an input image. The GAN trains on the entire training dataset. The selection of the right image as a reference is critical. The reference style should concern a joining scenario or CC similar to the input CC. This selection works under the assumption that similar joining scenarios receive similar joining element designs. Reference image selection is a use-case-based method, where the solution of a similar problem is applied to a new one. CBR and SML (classification) are appropriate methods for finding these styles. Much research has examined this topic, such as Rout et al.'s [422] review of image retrieval systems. The freedom of shapes makes it difficult for RBR to determine whether similarity limits results either to the nearest neighbor search of parameters (e.g., number of joining elements) or overlap (e.g., Jaccard index). In addition to the reference-guided approach, Fig. 3.28 depicts the latent-guided approach, which enables style transfer through the selection of an entire *domain*.

The **latent-guided approach** applies style codes, that is, latent vectors generated from a domain to predict joining locations. Here, an encoded style is the generalization of a set of joining scenarios. The domain has selected joining scenarios that are assigned through, for

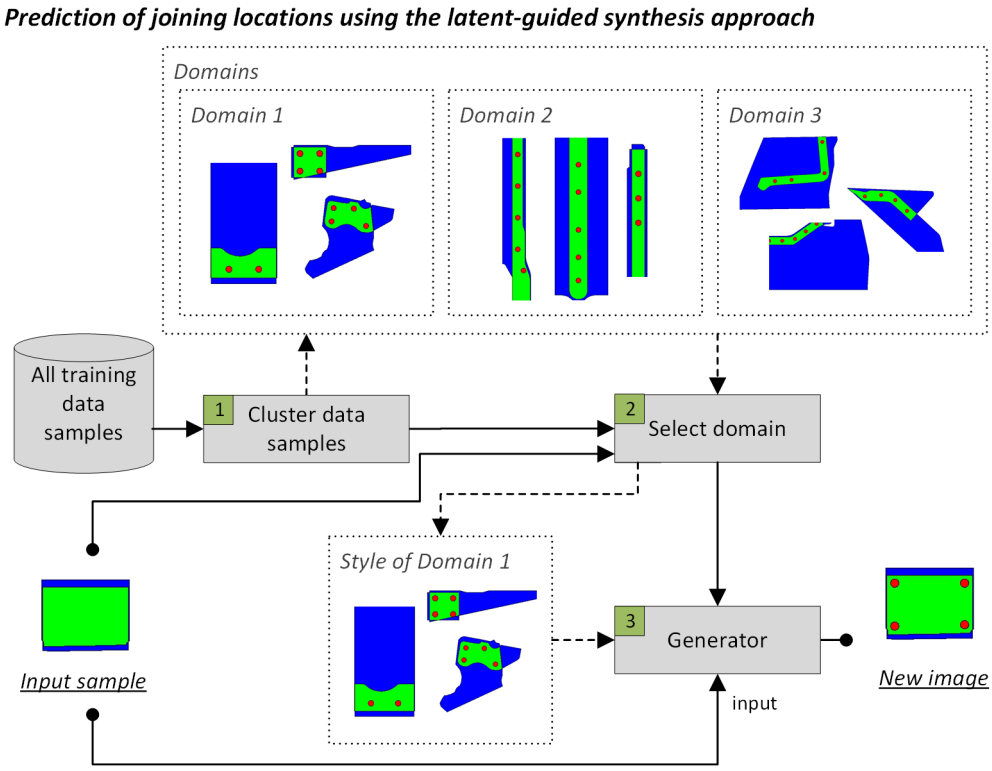


Fig. 3.28: Example of the process and results when implementing latent-guided image synthesis. The geometry of input samples apply the style information of a domain to synthesize new data samples.

example, clustering approaches. These can be the same clustering results created for MMML; see subsection 3.3.2.6. The clustering of shape descriptors can consider the entire joining scenario geometry, even when they are too large to fit in one CC. Furthermore, the clustering of nongeometric data for domains represents information as the materials, nomenclature, and function of the components.

After clustering, the latent-guided approach selects the most appropriate domain, which requires CCs to be classified to domains. Depending on linear or nonlinear clustering methods, this would require one to find the nearest cluster or to train an SML predictor. This selection is a similar problem to that in the reference-guided approach, although a direct comparison between data samples is not possible. First, these samples must be encoded and brought into latent space.

Regardless of the previously mentioned approaches, each class may take up a channel in an exemplary RGB image to differentiate between geometry frequencies. These classes are similar to the encoding of class in the segmentation approach (Section 3.3.2.1). A GAN has the same formatting considerations for the EncDec architecture, such as size and resolution. The fourth class that represents “emptiness” can be colored white, thus, uses fully the color channels in the RGB image. Segmentation tasks remain unchanged. However, data formatting directly corresponds to data working with computer vision algorithms. It can retrieve coordinates and

measure the performance of the models.

GANs apply styles to input images, but they also change them. Details of component geometries such as holes may disappear in the prediction, changing the geometrical layout. Bau et al. [423] observed similar behavior when experimenting with the generation of houses. They found that objects that occur randomly in training images, such as cars or people, are often discarded by GANs. As Gerlach and Eggink [5] suggested, letting the GAN predict only the red channel of RGB images solves this problem.

The experimental results of Gerlach and Eggink [5] demonstrated that GANs can learn the task to predict spot welds. The authors experimented with both the reference- and latent-guided approaches. The reference-guided approach created feasible predictions. Spot welds only appeared on CRs and were distributed logically. Interestingly, the number of spot welds on the reference images did not seem to influence the predicted number. The CR's border seemed to affect the predicted number of spot welds more.

Regarding the latent-guided approach, the authors' results indicated difficulty in synthesizing feasible images. To create the domains, they implemented both the clustering of nongeometric data (subsection 3.3.2.6) and images (using transfer learning [424] of a pretrained VGG16 network [425] on ImageNet [426]). Each input image had synthesized images from every domain. Interestingly, the model could only create feasible results from domains that contained the input image. Other synthesized images were completely deformed and did not resemble anything of the input image.

The reference-guided approach seems most promising due to its feasible results. The training of GANs is generally difficult [275], which may have played a role in the latent-guided approach, where clustering might have added unnecessary complexity into the system. The experimental results indicated that GANs are applicable for predicting joining locations; however, they are still unstable and require much additional research.

To measure the quality of prediction of the previous presented methodologies for predicting joining locations, the following subsection presents several performance metrics.

3.3.4 Measuring the performance of joining location predictions

This section does not address a separate concept for predicting joining locations; rather it presents an approach for evaluating prediction models regardless of their data formatting, architecture, learning task, or similar. First, it discusses the creation of coordinates from Euclidean data representations (subsection 3.3.4.1). Secondly, it proposes metrics for measuring the performance between the predicted coordinates (subsection 3.3.4.2).

3.3.4.1 Coordinate reconstruction

Predicted grid cells must be transformed back into Cartesian coordinates for use and evaluation. Each group of adjacent cells represents a joining location. Fig. 3.29 presents an overview to calculate joining locations from a grid-like Euclidean data structure.

The coordinate follows from the weighted center point of all grid cells' indexes. Each grid cell has a probability of belonging to class 3. A weighted average of the indexes can set locations between grid cells. The coordinate p of joining location j in a CC ($p_{j,cc}$) is the

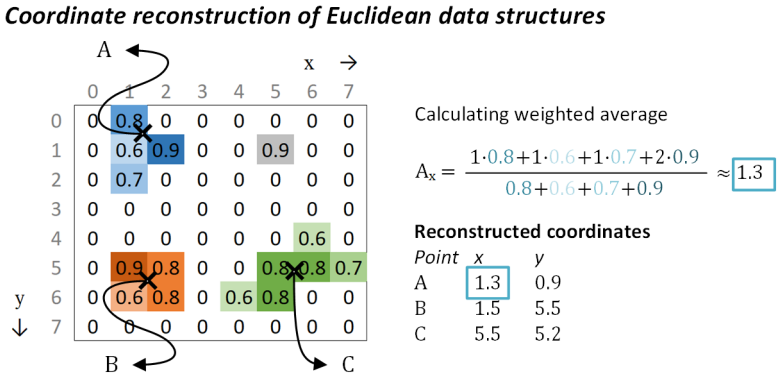


Fig. 3.29: A grid with probabilities of cells that belong to the joining location. 2D prediction generated coordinates are on the right. Coordinates follow from the weighted voxel indices for groups A, B, and C. Noise filtering prevents voxel groups that are too small becoming coordinates; see index $(5,\ 1)$

weighted sum of n indexes x_i with weights a_i divided by the sum of the weights a_i; see Eq. 3.16. A noise filter removes perturbations in the output by forcing each group to have a minimal number of grid cells τ_g. The average weight $\bar{a}^j_{cc}$ of location $p_{j,cc}$ expresses the confidence of that joining location; see Eq. 3.17.

$$p_{j,cc} = \frac{\sum^{n} a_i x_i}{\sum^{n} a_i} \qquad \forall n > \tau_g \tag{3.16}$$

$$\bar{w}^j_{cc} = \frac{1}{n}\sum^{n} a_i \qquad \forall n > \tau_g \tag{3.17}$$

The segmentation and regression concepts have slight differences when filtering the cells belonging to joining locations. Segmentation takes all class 3 values and uses their classification probabilities as weights, whereas regression has a continuous probability value filter, such as the spatially dependent spatial matrix $\tilde{T}_{\tau_l}$, with a threshold $\hat{y}_{w,d,h} \geq \tau_l$. This matrix gathers high probability grid cells and uses their values as weights.

The origin locations p_{cc} of CCs must be transformed back into the coordinate system of the joining scenario; see Eq. 3.18. This requires reverse engineering of the data formatting processes. The joining location p^l_i for $CC^{(i)}$ at the strided indexes $i^s = k, l, m$ with stride lengths $l^s = l_w, l_d, l_h$ compensate for the splitting. The grid resolution r_n transforms the coordinates between the data representation and product space. Unseen joining scenarios require a striding length equal to the CC size to be used to prevent overlapping or negligence of geometry in predictions, as visualized in Fig. 3.24.

$$p_s = r_n \times p_{cc} \times i^s \times l^s \tag{3.18}$$

Coordinate reconstruction for curves is more complex than for discrete joining locations. If a model predicts both points and curves, the latter require identification before the weighted

averages of grid cells are calculated. Curves have minimal lengths for the sake of manufacturability (e.g., [150]), which creates a lengthy distribution of adjacent cells. Vice versa, point-based joining locations may have a modeled cylinder or sphere representing a spot weld or rivet. The groups of cells that are significantly larger than point-based sizes are classified as curves. Theoretically, a joining location predictor might predict multiple joining technologies. For instance, the aerospace and automobile industries regularly use a combination of adhesive bonding and spot welding to create crash-worthy structures [46].

A weighted average of a grid cell group for curves returns its center of gravity. A topological skeleton, also known as the medial axis, is the thinnest version of a shape that is equidistant to its boundaries [427]. The skeleton of a circle is a point and that of a cylinder is a line over the rotation symmetrical axis. The discrete points of the medial axis can reconstruct curves. The centerline represents the curve line. Each point along this line describes a discretization point that a nearest neighbor algorithm may connect to for reconstructing the curve line. Various methods exist for finding the medial axis of shapes (e.g., [428]. They perform morphological erosion, which is the opposite of dilation; see subsection 3.3.2.4. The computational cost and complexity of postprocessing a skeleton-based approach are unnecessarily high for discrete joining locations compared with the aforementioned method of coordinate reconstruction, as there is no need to erode voxels or retrieve a centerline.

3.3.4.2 Performance metrics

The evaluation of coordinates measures the performance independent of data formats, learning tasks, or ML architectures. The task of models is to predict coordinates that align with target coordinates. First, this subsection addresses point-based, discrete joining locations, followed by curves. Performance evaluation requires a similarity measurement between the predicted and target coordinates, which are referred to as sets of points hereinafter. Models might predict too many or too few coordinates. The minimum number of points of the two sets of points defines the basis for similarity. Similarity can only measure equally large sets; for example, if the prediction on a CC has three joining locations, whereas the target contained four, the similarity evaluation is performed on three joining locations. Fig. 3.30 presents various examples for determining the performance when predicting joining locations.

Leftover coordinates in either set prevent meaningful similarity expression as a point in one set would refer to several in the other. Performance evaluation requires at least two measurements, namely the number of viable coordinates and the similarity. Let a valid coordinate pair $\tilde{p}_{ij}$ consist of a unique combination of a predicted i and target j coordinate within the distance limit τ_v. The accurate distance threshold at τ_v prevents the inclusion of outliers or random predictions in the similarity measurement. Similarity measures the distance between all valid pairs. Valid pairs prevent models that tend to predict too many joining locations to have accidentally high similarity values. The chance that a predicted joining location is closer to the target is higher when predicting more joining locations on the same CR.

Every point of both sets may occur in only one valid pair. The pairing of points between two sets is a cumbersome task. A brute-force pairing method (S&O) would determine valid pairs by minimizing the sum of distances between all possible pair combinations. The lowest sum finds

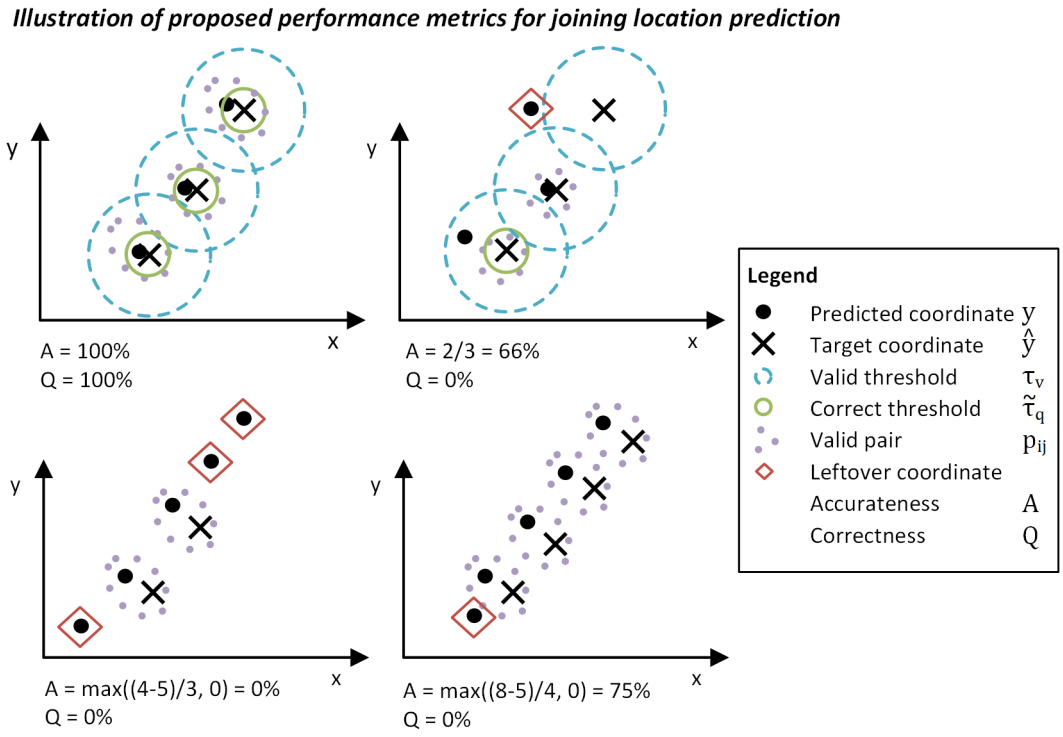


Fig. 3.30: Example of the performance measurement variables and use cases. The figure visualizes leftover coordinates, threshold distances, and the necessity of valid pairs.

the nearest valid pairs. However, calculating all possible pairing combinations is a factorial function that scales with the number of coordinates. By contrast, naive nearest neighbor-based combinatorics finds valid pairs in quadratic time but may result in incorrect measurements (see Fig. 3.31).

The Kuhn–Munkres or Hungarian matching algorithm proposed by Kuhn [429] and reviewed by Munkres [430] solves assignment problems in cubic time [431]. It is deterministic and aims to find a perfect matching in a bipartite graph, thereby optimizing the total cost. A typical use case of the algorithm is for assigning workers to jobs, where each worker has a different cost to perform each job. Thus, each worker must be allocated to a particular job to minimize the total cost. Similarly, workers, jobs, and costs are equivalent to predicted coordinates, target coordinates, and mutual distances, respectively. The algorithm considers predicted $\hat{Y}$ and target Y points of length $l_{\hat{Y}}$ and n_Y respectively, as a bipartite graph $G = (\hat{Y}, Y; E)$.

Each edge E in graph G represents the distance s_{ij} between point i from Y and j from Y. The perfect matching constraint in the Hungarian matching algorithm leaves leftover coordinates out of the results. The resulting assignment evaluates the similarity performance correctly. Four measurements determine a model's performance: the difference in the number of points, accurateness, similarity, and correctness. These measurements are described as follows:

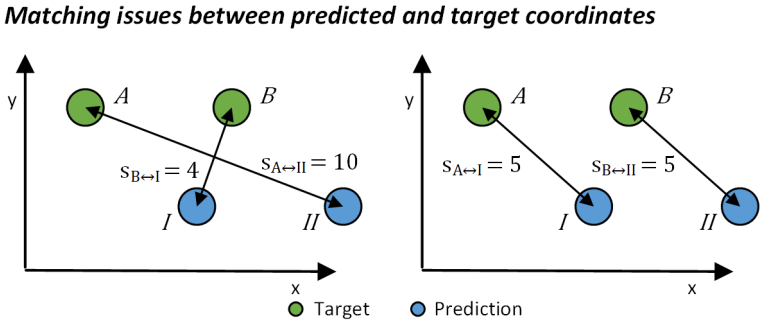


Fig. 3.31: Visualization of possible faulty measurements using nearest neighbor combinations to determine a model's performance. A and B are targets and I and II are predicted coordinates. The proposed performance algorithm finds valid pairs $\tilde{p}_{A,\mathrm{II}}$ and $\tilde{p}_{B,\mathrm{I}}$ (average Euclidean distance = 7) instead of the optimal pair combinations $\tilde{p}_{A,\mathrm{I}}$ and $\tilde{p}_{B,\mathrm{II}}$ (average Euclidean distance = 5).

- The **difference in the number of points** ΔN represents the difference between the number of points in the predicted $n_{\hat{y}}$ and target n_y set. Positive values of ΔN indicate that the model has predicted too many joining locations, whereas negative values indicate the opposite. The number of joining elements in a joining scenario is linked to the applied loads and forces. Joints that require a higher stiffness tend to have more joining elements (or, for example, longer adhesive bonds). Regardless of positioning, the number indicates the ability to hold static loads, which is an essential consideration in many location optimization methodologies.

$$\Delta N = n_{\hat{y}} - n_y \tag{3.19}$$

- **Accurateness** A considers the difference in the number of valid points between two coordinate sets. It provides insight into the usefulness of the predicted coordinates and measures the ratio between the number of valid pairs n_v and the number of targets coordinates n_y. It represents the deviation from n_y and expresses the structural contribution of every joining location. Thus, it assumes that predicting one additional joining location on a target with 20 joining locations is less harmful than doing so on a target with three joining locations.

  Notably, the term accurateness is purposefully chosen to express a form of accuracy on predictions. However, accuracy already has been defined in the state of the art (Section 2.3.3) as a metric for evaluating SML models. Hence, the change of the word accuracy into accurateness indicates a difference but retains the original connotations.

$$A = \begin{cases} \max\left(0, \frac{2n_v - n_{\hat{y}}}{n_y}\right) & n_{\hat{y}} > n_y \\ \frac{n_v}{n_y} & \text{otherwise} \end{cases} \tag{3.20}$$

- **Similarity** S measures the distance between coordinate sets bounded by the number of points in the smallest set. Similarity represents the mean distance of all valid pairs

V. Expressing the similarity using a Gaussian function (see Eq. 3.21) has a couple of advantages over the traditional Euclidean distance. A Gaussian function is nonlinear and decreases with an increasing distance – it ranges between 0 and 1, where a value of 1 represents complete similarity. It enables the modeling of stability and prevents a significant impact of outliers when averaging over all predictions. The value σ handles the steepness of the curve, which determines the rapidness of the similarity going to 0 with increasing distance.

$$S = \frac{r}{n_v} \sum^{\tilde{p}_{ij} \in V} exp\Big(- \frac{\|\tilde{p}_{ij}\|^2}{2\sigma^2} \Big) \tag{3.21}$$

A Euclidean distance expression is possible (see Eq. 3.22) but expresses dissimilarity. A distance of zero would be optimal, whereas high values of large distances measure *dis*similarity. Scaling this dissimilarity with the grid resolution r expresses it in input dimensions (e.g., mm).

$$S = \frac{r}{n_v} \sum^{\tilde{p}_{ij} \in V} \|\tilde{p}_{ij}\|_2 \tag{3.22}$$

- **Correctness** Q is a Boolean value $0\ or\ 1$ representing whether all coordinates in both sets map onto one another within the threshold distance τ_q. A prediction for a CC is correct when all predicted coordinates create valid pairs with all target coordinates, and all do this within τ_q distance without leftover coordinates.

The performance measurement of curves requires a small adaptation. Here, coordinate reconstruction creates centerlines. Curves consist of multiple discrete points that create a line after connection. Breaking a curve down into sets of points enables performance measurement to be conducted between the predicted and target curves. Similarly, as for discrete joining locations, the total sum of all absolute distances between each matched pair of points measures the similarity. Then, the Hungarian algorithm finds pairs of curves that correspond to one another. However, it uses a curve similarity metric instead of the Euclidean distance between two joining locations. Alternatively, a matching methodology can measure similarity between two curves [432].

The previous subsections regarded the prediction of joining locations. However, they did not regard modular design, thus, create a high variety in resulting joining elements. To cope with this, the following subsection presents an overview for the modular design of joining elements. In addition, it presents the required preprocessing steps.

3.3.5 Preprocessing for the modular design of joining elements

This section addresses boundary conditions for the modular design of joining elements. First, it provides an overview of the relevant design problems mentioned in the structure block of the framework (Fig. 3.7). Then, it discusses the use of overlapping CRs as the basis for modular design.

Overview

The methodologies within each design problem all have different input and output requirements. Their pre- and postprocessing steps are mostly methodology-dependent and are left out of the framework.

Joining aspects depend on one another. For example, a joining technology generally infers one of two geometrical shapes, namely points or curves. The technology sets the manufacturability, creating boundary conditions for designing joining locations. For example, joining parameters describe PMI and are assigned to joining locations in CAD systems [19]. The commonalization of joining parameters has no separate methodology. The same approach works for JTS due to the same nature and challenges of the problem.

The modular design steps follow the same sequence as the design of joining element aspects; see Section 2.5. They go from the bottom up, first commonalizing each joining aspect, then joining elements, and finally modularizing to create joining modules. Fig. 3.32 illustrates the process for the modular design of joining elements, which includes the following five-steps [4]:

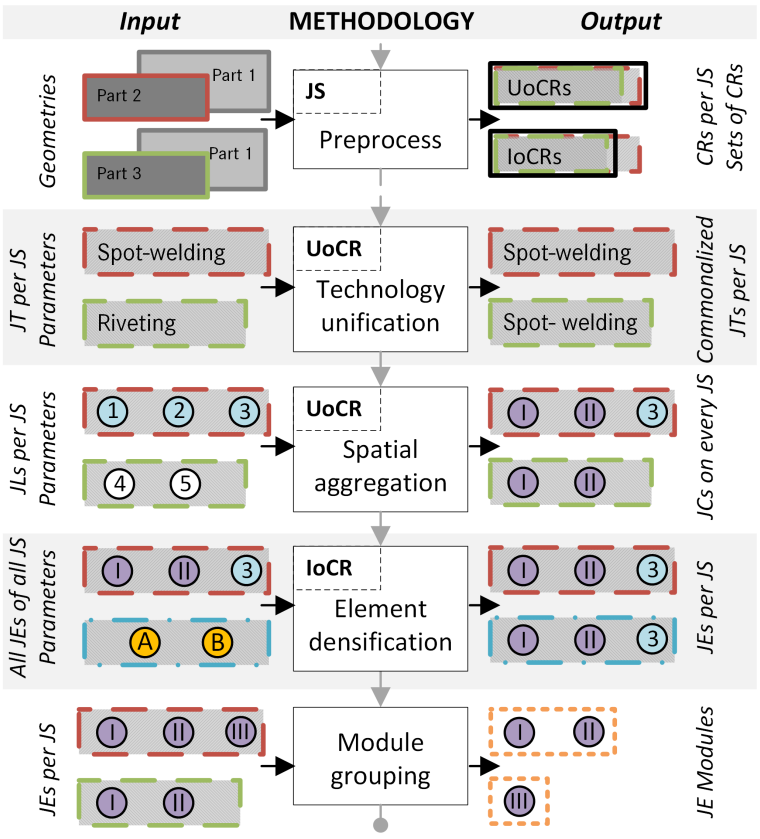


Fig. 3.32: Visualization of a methodology with input and output dependencies. Abbreviations: (JS) Joining scenario, (CR) Contact region, (UoCR) Union of contact regions, (IoCR) Intersection of contact regions, (JT) Joining technology, (JL) Joining location, (JC) Joining cluster, and (JE) Joining element.

- **Preprocessing** (described in the present section) determines the module interfaces through overlapping CRs between product variants. These originate from the virtual stacking of product variants (e.g., as illustrated in Fig. 3.5). Overlapping CRs consider both functional (i.e., joining scenarios) and geometrical boundaries (i.e., overlapping surface areas).
- **Joining technology commonalization** (Section 3.3.6) reduces the variety of processes on each overlapping CR. It evaluates joining technologies or parameters using a ranking method that considers all involved product variants.
- **Joining location commonalization** (Section 3.3.7) clusters joining locations on overlapping CRs. It determines a new location as a substitute to share between joining scenarios. Joining location commonalization can reduce the variability in predicted locations.
- The **commonalization of joints** (Section 3.3.8) aims to substitute the joining elements of one product variant into others. This methodology uses geometrical boundaries in overlapping CRs. The commonalization of joints reduces product platforms of joining elements at the cost of adding joining elements to joining scenarios.
- **Module creation** (Section 3.3.9) collects the shared joining elements between product variants. It balances commonality and modularity to create subassemblies and joining modules.

Preprocessing

Preprocessing determines the module interfaces through overlapping CRs between product variants. These originate from the virtual stacking of product variants (e.g., as seen in Fig. 3.5). Overlapping CRs consider both functional (i.e., joining scenarios) and geometrical boundaries (i.e., overlapping surface areas).

Commonalization uses CRs to identify module interfaces. CRs are geometrical interfaces between components. The positioning of components in a joining scenario determines a joining shape (e.g., lap or butt joint), thus reducing feasible technologies. Interchanging components between product variants tends to retain the joint shape, enabling the same joining technology to be applied. CRs may change in size but the joining shape remains the same; see Fig. 3.5. Hence, the joining elements defined on these CRs can also remain the same, although they remain subject to manufacturability requirements. The overlapping CRs can form the basis for the essential standardization of interfaces, which is in agreement with statements made by Ericsson et al. [229] about minimizing information flows.

A functional definition of an interface between two components ignores geometry. The decomposition of products into functional modules can include the definition of functional joining modules (e.g., see Gauss et al. [91] and Fig. 2.8). Such modules may describe preferred and available joining technologies based, for example, on resource availability in production, performance requirements, or aesthetics. Such joining modules receive functional requirements and design parameters in the joining element design process. Steps such as joining technology commonalization still work, even without overlapping CRs. However, they would require a basis for aggregating variants by, for example, design spaces or functional requirements (see

Fig. 3.3). Nevertheless, modular design of joining elements that considers geometry needs various considerations for determining overlapping CRs, as Fig. 3.33 depicts.

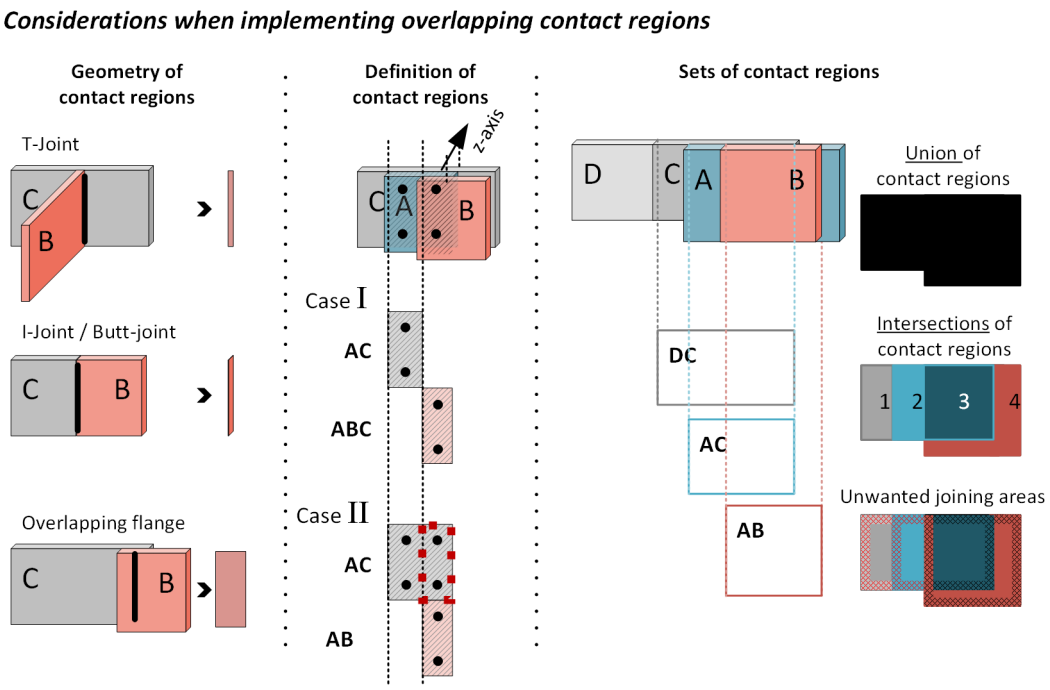


Fig. 3.33: Various considerations for overlapping contact regions in the preprocessing step for the modular design of joining elements. The left column presents examples of the geometry of contact regions. The middle column displays examples of overlapping contact regions and the required buffer for enabling joints that involve more than two components. The right column depicts the creation of the union and its intersection of contact regions from three joining scenarios.

Furthermore, the geometry of CRs may include not only curved lines but also complex curved surfaces (see the left column of Fig. 3.33). Simpler CR shapes decrease the complexity of design and modularization. Flat surfaces, such as flanges, enable the shifting of components. For example, the Lucas method [147] gives lower complexity scores for the simple fitting and handling attributes of components, such as symmetry in the (rotational) orientation of parts.

Overlapping CRs are the *shared* geometrical interfaces of multiple joining scenarios. The right-hand column in Fig. 3.33 presents CRs of three scenarios (with component combinations of $D\&C$, $A\&C$, and $A\&B$) created by four components (A, B, C, and D). They create two CR-based collections for commonalization (see Fig. 3.33): the union and intersection of CRs.

- The **union of CRs** (UoCR) refers to the merger of multiple CRs through the virtual stacking of product variants. The UoCR defines a collection of joining scenarios with overlapping CRs. It limits commonalization and modularization approaches and includes all joining scenarios relevant for module interfaces. The geometry of the UoCR determines the shape and size of the module interface.
- The **intersection of CRs** (IoCR) is a unique subset of the UoCR shared by a unique

combination of joining scenarios. IoCR geometrically defines potential joining element platforms. Within the IoCR, manufacturability conditions are constant. IoCRs with multiple joining modules or redundant joining elements would create unnecessary complexity. IoCRs from the same UoCR may have different mechanical properties due to varying component combinations. For example, the presence or absence of an additional metal sheet changes the thickness of the stack joint and thus also the production (and joining) parameters.

Joints might consist of more than two components. CRs are the result of components that touch each other. However, "touching" might be a harsh requirement in cases of three components. The "Definition of contact regions" column of Fig. 3.33 depicts a joint with three components (A, B, and C) consisting of two overlapping CRs (AB and AC). The thickness of component B creates a distance between CRs AB and AC.

The consideration of this distance results in different definitions of joining scenarios and IoCRs. The z-axis, or normal vector, needs a buffer during the assignment of CRs to the same UoCR; see Fig. 3.33. Ignoring the z-axis reduces design and modularization freedom. The buffer supports numerical irregularities and a more robust behavior during commonalization. An increase of the buffer length along the z-axis may assign more CRs to the same UoCR and include more joining elements into consideration for commonalization. Moreover, a buffer that is too large may include irrelevant CRs.

Preprocessing for the modular design of joining elements helps to create a basis for further algorithms. The use of CRs provides a geometric playing field for modifying joining element designs systematically. Overlapping CRs visualize the overlap of product variants. In this sense, the UoCR gathers all nearby product variants, while the IoCR identifies the joining elements that are uniquely shared for a certain set of product variants.

The following subsection presents a methodology that uses the overlapping CRs to commonalize joining technologies.

3.3.6 Commonalizing joining technology with technology unification

Process variety is one of the main factors in complexity and production cost [136]. Traditional JTS methodologies (e.g., [43, 161]) often implement MCDM approaches to find the optimal technology for a joining scenario. However, these approaches do not consider product variety and historical sources, potentially leading to a great variety of joining processes. It is not the individual optimal joining solution of a single component that leads to the most valuable cross-component integration [158]. Methodologies select joining technologies while considering a single joining scenario. Hence, technology unification (TU) is required to commonalize technologies on module interfaces to reduce process variety. For example, Fig. 3.34 illustrates exemplary TU between four joining scenarios.

Fig. 3.34 presents an example of joining technology prediction used to determine two feasible technologies for each joining scenario. The weights represent, for example, the confidence of each prediction, but they may include more parameters, such as sales volume or designer preferences. The figure demonstrates that when only the highest scoring prediction for each joining scenario is considered, spot welding has the highest sum of weights. Thus,

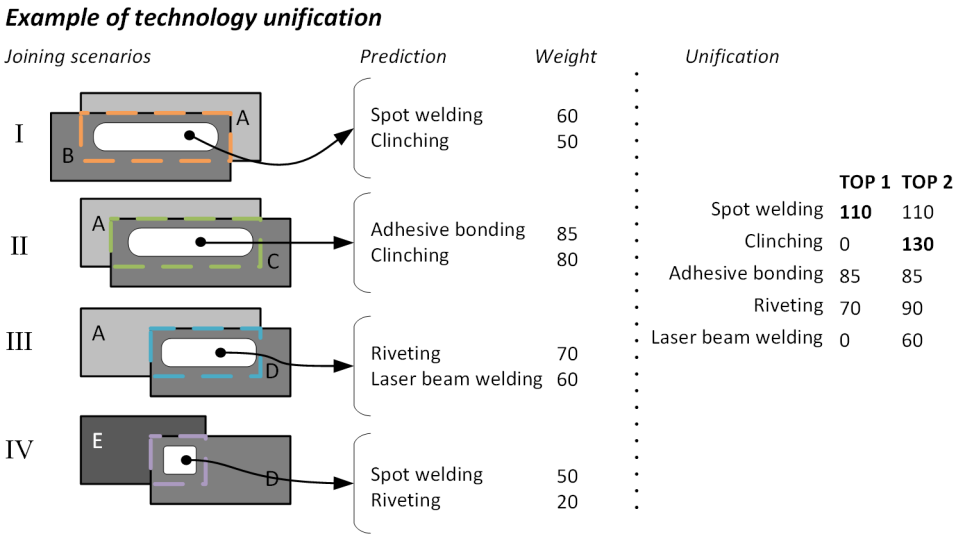


Fig. 3.34: Visualization of the choice between multiple unified technologies on a union of contact regions or the use of a suboptimal one.

spot welding becomes the unified technology. However, TU can also consider multiple predictions at once. Fig. 3.34 indicates that clinching scores higher when considering the second best predictions. TU can find the overall best technology by including the second best predictions.

TU is similar to a traditional MCDM approach as it adds an abstract layer to traditional technology selection methodologies. The methodology aligns with the traditional three-step process of Ashby et al. [161], namely screening, ranking, and selection, which is detailed further as follows:

1. **Screening** checks feasibility and filters technologies that do not fit all joining scenarios in the UoCR. Predicted technologies require validation against standards and guidelines to ensure manufacturability in terms of properties such as geometry, material, and function. Therefore, the screening step in TU functions similarly to the step in regular MCDM methodologies.

   However, it might be that none of the technologies applies to all joining scenarios. This would induce a trade-off between introducing an additional joining technology or choosing the second-best technology. The former requires determining technologies for subsets of joining scenarios in the UoCR. However, it also requires multiple valid technologies for each joining scenario. Splitting the UoCR into subsets introduces another challenge in finding the optimal selection of subsets. Multiple technologies are counterproductive in reducing process variety but retain better performance in each scenario. Selecting the second best technology creates a feasible and TU assigns it to all joining scenarios. This may result in more costly and complex joining element designs compared with introducing an additional joining technology.

2. **Ranking** takes the feasible processes and ranks them according to their score for every joining scenario. For each joining technology, ranking calculates a sum of scores weighted by variety-relevant parameters. This section later addresses these parameters, which include confidence, sales volume, CR size, implementation factor, and preferences. They weigh joining scenarios according to their significance to product variety. These unification weights may also include geometrical, functional, and manufacturing requirements, such as those listed by Kaspar et al. [32]. The benefit of ranking through weighting is that it can include popular criteria such as costs, complexity, modularity, or commonality, besides technical properties. It enables a holistic evaluation of relevant parameters that can be customized for each use case.

3. **Selection** outputs the unified technologies. As previously stated, a consistent second-best solution on every joining scenario may be preferable to a solution that works very well for only half of them. Including second- and third-best candidate technologies in TU can balance the results and increase the unification potential. Selection considers manufacturability and decides whether to pick a suboptimal joining technology for all scenarios or to pick several more optimal technologies for fewer scenarios.

TU employs an overall ranking for each technology. For this, it uses summing weighted joining scenarios; see Eq. 3.23, where u^u represents the unified technology resulting from the technology u with the largest sum; n_s includes all joining scenarios in a UoCR; and u_s is the joining technology u of scenario s. Mathematically, it is a binary vector with each value assigned to a specific technology. The vector u has only a 1 at the index of the joining technology u_s, and a zero for all other joining technologies. The weight a_s^u takes the importance of each joining scenario s into account and consists of several parameters; see Eq. 3.24. Consequently, Eq. 3.23 considers a unification formula that includes an additional weighted sum over the number of predictions per joining scenario $n_{\hat{y}}$:

$$u^u = \underset{u}{\operatorname{argmax}} \left\{ \sum_{s}^{n_s} \sum^{n_p} u_s a_s^u \right\} \tag{3.23}$$

The bottom section in Fig. 3.35 illustrates that riveting is the joining technology with the highest ranking in both the top 1 ($n_{\hat{y}} = 1$) and top 2 ($n_{\hat{y}} = 2$) of predictions in each scenario.

However, riveting is not feasible in scenario I, for example, due to thickness ratio or plate thickness violations [114]. Spot welding is possible on all variants and the second-best scoring technology. Strict commonalization would state that spot welding is the unified technology, even though its score is worse than riveting; see Fig. 3.35. Suboptimal solutions after unification might create other design problems, such as compliance with structural performance, aesthetics, or production time.

Moreover, when none of the joining technologies are feasible in all joining scenarios, the process outputs several technologies. It is incapable of fully unifying the joining technologies on the UoCR. The introduction of a threshold τ^u enables decisions between two unification strategies. It considers the percentage of the total score from weights for every technology. A threshold of at least $\tau^u = 30\%$ would require confident predictions and output a multitechnology result; see Fig. 3.35. A threshold value of $\tau^u = 20\%$ would enable full

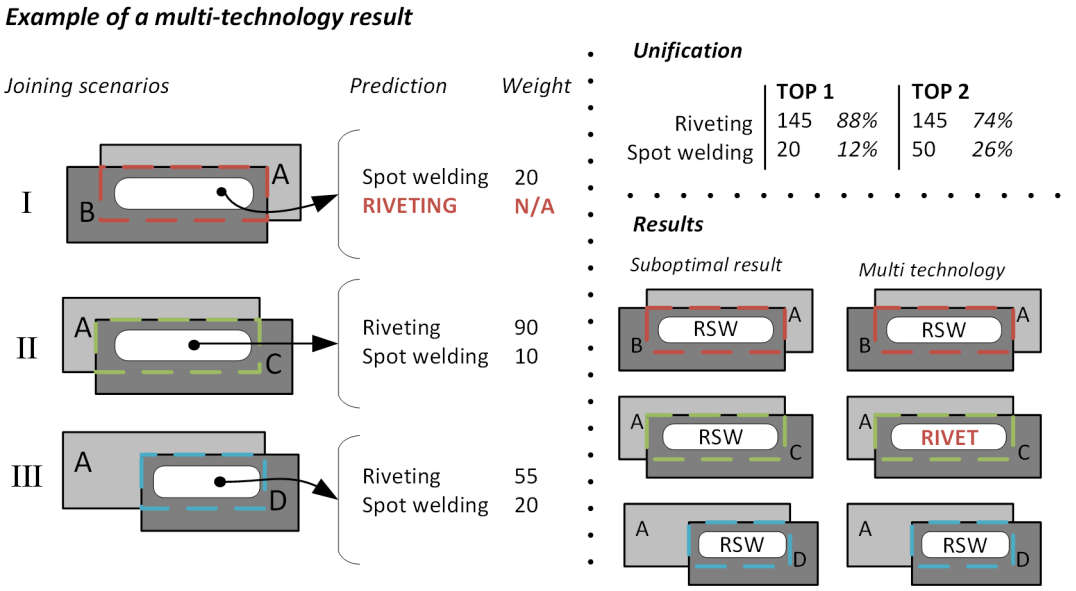


Fig. 3.35: Visualization of the choice between multiple unified technologies on a union of contact regions or the use of a suboptimal one.

unification for both cases when $n_{\hat{y}} = 1$ and $n_{\hat{y}} = 2$. In both cases, designers and engineers should analyze the design and possibly conclude that they should redesign the parts [433]. Fig. 3.36 briefly visualizes modular design of joining elements in case TU results in a multitechnology solution.

The modular design process must consider conflicting technologies, such as separate UoCRs. This separation permits the commonalization of locations, entire joints, and module creation to work properly without restrictions or additional complexity within their algorithms. Instead of one sequential process per overlapping CR, they become multiple simultaneous processes.

Contrary to unifying technologies, JTS methodologies may consider unification and variety requirements up front after predicting them, which might prevent the need for unification methodologies. Selection methodologies could integrate variety-related variables, use inductive approaches, or penalize variety in output. For example, SML tends to predict similar technologies for similar scenarios.

Variety-related parameters purposefully enable TU to treat joining technologies and scenarios unequally. This reduces randomness in the outcome and focuses on product variety impactful joining scenarios. The parameters can be extended with any relevant property for specific companies and industries to either favor or disfavor joining technologies. Weights contain various parameters for tuning unification. Eq. 3.24 presents a simple proposal of a generic linear function to weigh technologies a_s^u with a normalized vector x_i^u to balance, scale, and tune every i-th parameter.

$$a_s^u = E_t P_s^u (x_1^u I_s + x_2^u R_s + \dots) \tag{3.24}$$

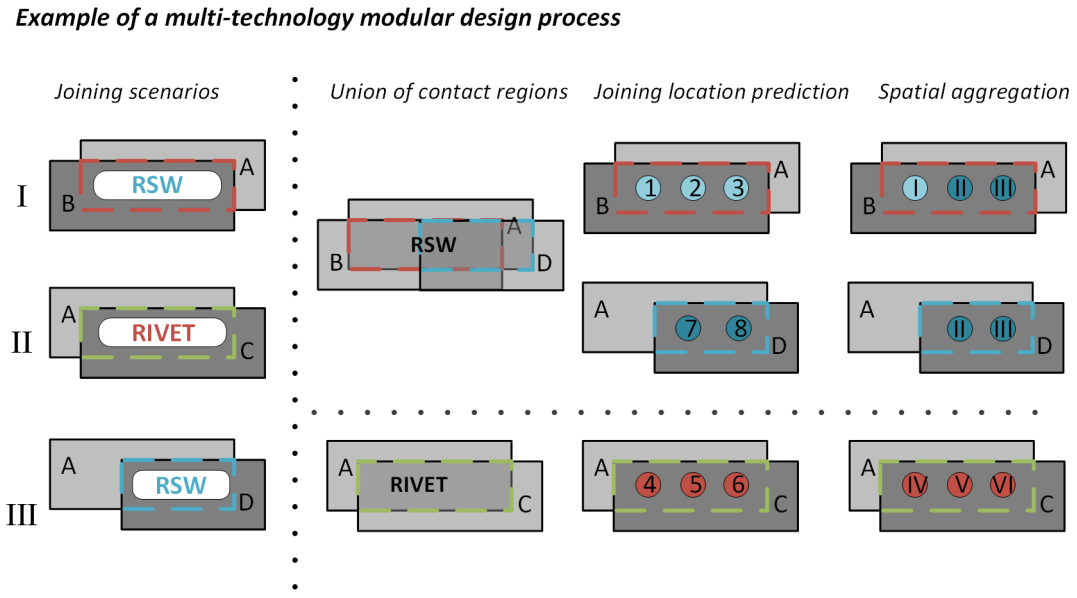


Fig. 3.36: Illustration of consequences in case of a multitechnology solution.

- **Confidence** (P_s^u) expresses certainty in prediction and reduces the importance of low scoring technologies. Algorithms that predict or select joining technologies (e.g., [43]) output scores to determine the optimal solution. Higher values indicate a more appropriate technology, and its weights and potential for unification should be increased. Values between joining scenarios in the UoCR must be in the same dimensions and ranges during all modularization steps. Normalization between values of equivalent methodologies sets joining technologies on the same scale. All parameters are of a dimensionless scale; hence, $P_s^u \in [0, 1]$.

- **Sales volume** (R_s) expresses the popularity of a set of product variants. A higher sales volume requires more optimal designs in terms of cost and quality, and therefore, they receive larger weights in commonalization. For example, joining elements configured in virtually all products have higher performance requirements and should receive a more cautious decision for changes compared with those that occur in a few exotic variants. The normalization of sales volume scales weights fairly between joining scenarios. Sales numbers may be difficult to estimate. A backup scoring function using, for example, the values $\{1, 0.5, \text{and } 0.1\}$ could describe high, medium, and low prospects, respectively.

- **Implementation factor** (I_s) considers different maturities of joining element designs due to continuous product development. Already in manufacturing, joining elements might need to be commonalized with joining elements of new variants. Joining elements in production have high rework costs for design changes (see Fig. 1.2) and require higher weighing. The rapid rise of change costs for joining elements increases interest in not commonalizing. Preferably, the implementation factor scales equally as fast as costs for a redesign. This requires an exponential relationship between steps.

Then, the factor could have the following values, again in a range of $[0, 1]$: 10^{-3}: concept / design phase, 10^{-2}: validation, 10^{-1} production ramp-up, and 10^{0}: product on market.

- **Preferences** (E_t) enable engineers to strategically tune for holistic requirements or company strategies. As mentioned by LeBacq et al. [159], local and company-specific parameters enable tuning as not all design considerations are easily expressed for optimization. Every technology has a specific preference value in the range $[0, 1]$.

In addition to commonalizing joining technologies, the following subsection presents a methodology to commonalize joining locations.

3.3.7 Commonalizing joining locations with spatial aggregation

Currently, joining location design methodologies use a topology optimization approach and only consider one product (e.g., [18, 46]). Multiple slightly different product variants may have different locations for the same joining scenarios. Joining element distributions are unique for every product, even though product variants share many components. Spatial aggregation (SA) finds shareable joining locations through searching for small groups of joining locations over product variety.

SA virtually stacks the joining locations of joining scenarios. The methodology clusters joining locations in close vicinity to one another. The UoCR holds the joining scenarios for each SA run. The UoCR ensures that clustering algorithms are bounded by local requirements and contain the search space. Fig. 3.37 visualizes SA creating joining locations I, II, and III from joining locations $\{1, \ldots, 6\}$.

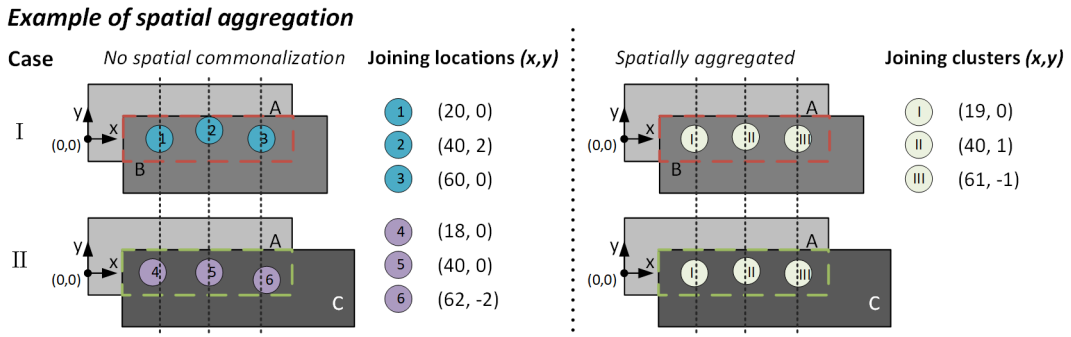


Fig. 3.37: Overview of spatial aggregation. The left-hand side shows two joining scenarios from the same union of contact regions. The right-hand side shows the created joining clusters that function as substitutes for the joining locations from the input.

Both joining scenarios ($A\&B$ and $A\&C$) have three joining locations in Fig. 3.37. Designers would create six joining elements without commonalizing joining locations as every location is unique. SA would take locations 1 and 4 and create a new joining location A as a surrogate to replace the two locations mentioned above. This joining cluster A is the cluster centroid calculated by taking the average of the two joining locations 1 and 4. The

figure depicts a similar process for locations 2&5 and 3&6, creating joining clusters B and C, respectively. Hence, SA creates three joining elements by reducing local variability.

SA can use K-means clustering with constraints (e.g., [406]) as its core algorithm to determine surrogate joining locations, also referred to as joining clusters for the sake of readability. K-means has several properties that fit this particular purpose:

- It creates circular clusters that solve the variability of joining locations. These circular clusters support compliance to minimal mutual distances of joining elements.
- Results are density-invariant such that cluster sizes and shapes do not change due to the distributions of joining locations.
- The maximum cluster size s^c ensures joining clusters for each outlier. Joining clusters must remain close to joining locations to prevent significant performance changes. Additionally, the maximum cluster size ensures minimum mutual and edge distances described in standards and controls the potency with which the algorithm clusters. Larger cluster sizes span over more locations of more product variants, creating higher degrees of commonalization.
- Constraints (referring to work of Wagstaff et al. [406]) prevent clustering joining locations of the same joining scenario.

Fig. 3.38 presents an example process of SA, which is explained as follows:

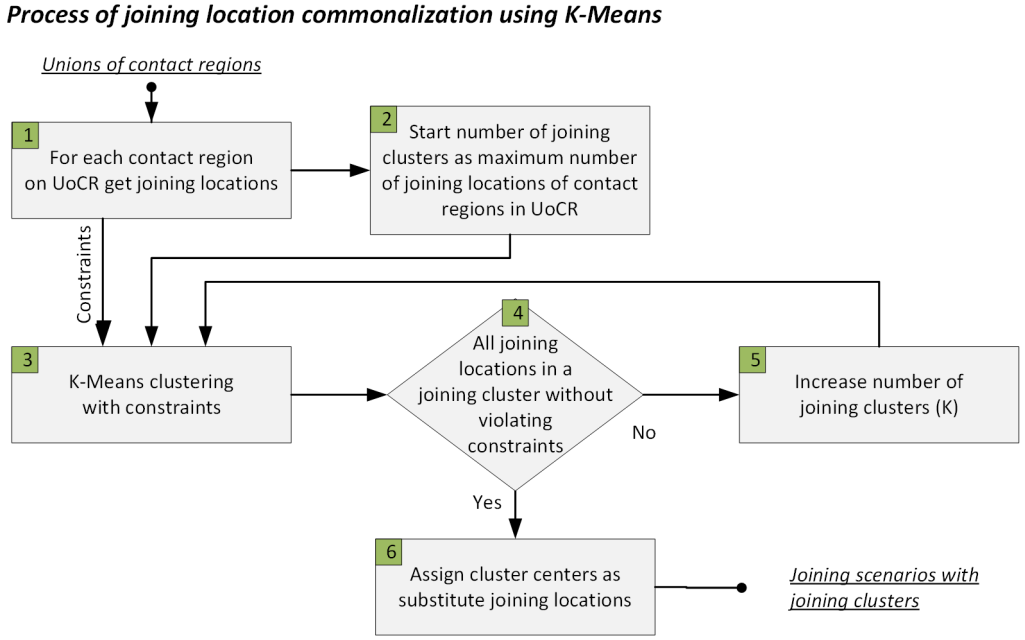


Fig. 3.38: Spatial aggregation clustering process flow.

1. The algorithm runs on each UoCR. These UoCRs are a set of CRs from different joining scenarios. Each CR has joining locations, such as the outcome of prediction models.

2. Every joining location is assigned uniquely to a cluster, ensuring transparency and simplicity. Joining scenarios must have an equal number of joining elements after SA. Hence, the minimum number of joining clusters in a UoCR is equal to the maximum number of joining locations on any CR in the UoCR.
3. The number of final clusters is unknown a priori. Hence, SA needs to perform an iterative search with k-means. K-means clustering's constraints concern, for example, the maximum clustering size and that joining locations from the same joining scenario may not end up in the same joining cluster.
4. The algorithm continues re-iterating as the number of joining clusters increases. It does this until all joining locations are assigned to joining clusters.
5. Once the requirement is met, the algorithm substitutes the original joining locations with the resulting centroids of joining clusters. These joining clusters become the new joining locations; however, they remain linked between the joining scenarios. This linkage induces the shareable locations. To its extent, it also induces the commonalization of joining locations.

The algorithm may stop once it finds the first solution that complies with requirements. However, additional optimization iterations might create better outcomes. More joining clusters move joining locations over smaller distances. These will remain closer to the initial locations but will create additional joining clusters, thus increasing variety. An evaluation approach (e.g., the elbow, average silhouette, or gap statistic method) could determine the optimal number of clusters.

SA also works for joining technologies with curved geometry, such as adhesive bonding or laser welding. Discretized curves have start and end points. These points create intersections with overlapping curves and divide them into shorter curves. SA can consider each of the points as discrete joining locations and then create joining clusters that simulate the start and end points of curves. Fig. 3.39 on the left-hand side depicts curves creating point–line intersections, which refers to where one joining element ends and another continues.

Using the point–line intersections, SA clusters the start (p_a^1, p_a^2) and end (p_b^1, p_b^2) points of curves, as shown in the center column of Fig. 3.39. This results in three joining clusters visualized with purple and yellow points (p_1^c, p_2^c, p_3^c). From these three joining clusters, the algorithm can reconstruct two curves, as seen in the right-hand side of Fig. 3.39.

The definition of IoCRs and the consideration of standards and guidelines create unwanted joining regions; see Fig. 3.33. These are sections on UoCRs that preferably should not contain joining elements. Joining elements that lie in unwanted regions cannot be fully commonalized, as at least one joining scenario prevents it through manufacturability requirements. Requirements from one of the CRs in the UoCR might also induce this (e.g., minimum edge distance). Fig. 3.33 visualizes the edge distances on each CR. The IoCRs are all bordered with a pattern to indicate the edge distance. Commonalization processes must recognize consequences for all affected products to prevent manufacturability problems.

Analogous to TU, SA can consider various parameters to control clustering results. These parameters expand on weight a_s^u for joining scenarios used in TU. The weighted average scales coordinates of joining locations to new cluster centroids, proposing a linear SA weight a_p^l for

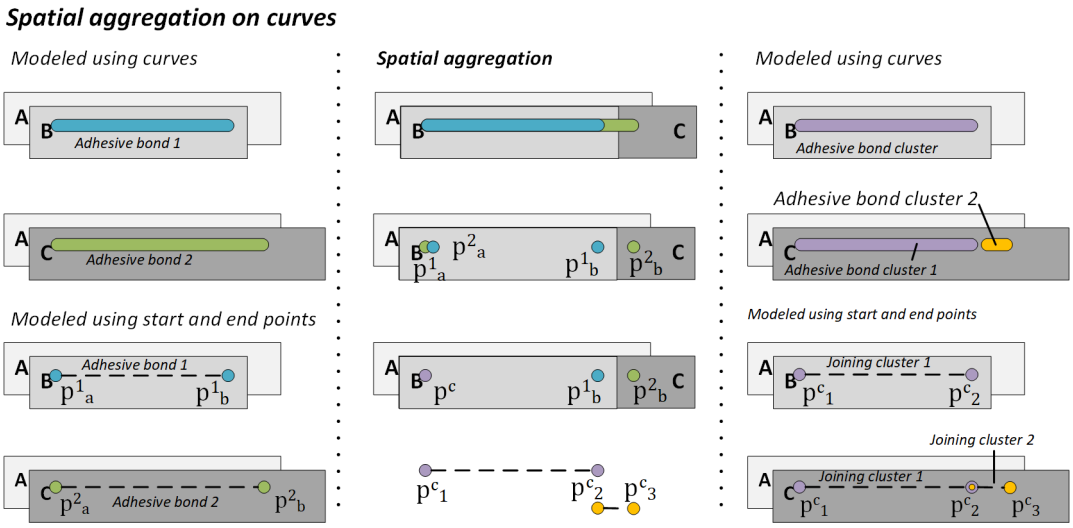


Fig. 3.39: The extension of spatial aggregation to curves by considering start and end points as surrogate locations for commonalization.

joining location p with normalized vector x^l to balance, scale, and tune every parameter; see Eq. 3.25

$$a_p^l = a_s^u \left(x_1^l P_p^l + x_2^l \left(1 - \frac{s_{p \to p^c}}{s_{min}^u} \right) \right) \tag{3.25}$$

- **Confidence** (P_p^l) considers a probability score P^l for joining location p. Less confident locations should affect the moving of cluster centroids less and vice versa.
- **TU weight** (a_s^u) brings all joining scenario dependent weights over to SA. The same variety of parameters apply to joining locations as to their technologies. Integrating the joining TU weight enables one to integrate parameters such as sales volume, implementation factor, and engineer preferences. Moreover, it applies these weights for each joining scenario. Uncertain predictions reduce the weights in TU, which can cascade through to joining locations in SA. Consequently, SA can also consider the product variety-relevant parameters relative to each joining location.
- **Distance-to-center** (s) measures the distance between the joining location and its joining cluster. This parameter helps to create joining clusters in the center of small dense groups and reduces the impact of outliers. $s_{p \to p^c}$ is the (Euclidean) distance between the joining location p and assigned cluster centroid p^c. Dividing the distance-to-center by the minimum joining distance s_{min}^u of joining technology u creates a dimensionless parameter. A negative fraction moves the parameter toward zero when a joining location moves toward s_{min}^t. Lastly, the weight of joining locations scales linearly with its distance-to-center.

Weighing may still induce the translation of joining elements, even for those with very high

weights. A solution for retaining implemented joining elements might be binary SA. Instead of taking a weighted average, the joining location with the highest weight becomes the joining cluster. All other joining locations in the cluster move accordingly. The term binary represents a weight of 1 for the most important joining location, whereas the other locations receive a weight of 0. Binary SA might be useful in product development where the first product variants have reached the market.

The effectiveness of SA follows from the ratio of reduced joining locations over their total number. Optimally, the algorithm achieves this by moving joining locations very little, as this may affect the structural performance of the joint [18]. Many performance metrics in the literature have considered the number of (sold) product variants to determine commonality (e.g., [213, 394]). Unfortunately, such information is unknown in early product design. The Percent Commonality Index ($\%CI$) of Siddique and Rosen [236] is a simple yet effective metric for measuring commonality; see Eq. 2.6. More specifically, one of the terms, the commonality of connections, specifically addresses joints. It enables one to measure commonality before and after SA. As SA specifically considers joining element commonalization, this study considered the Percent Commonality Index ($\%CI$) as the only relevant factor for expressing the commonality of connections.

The previous subsections addressed commonalization of individual joining aspects. However, entire joints also require commonalization to reduce unnecessary product variety. The following subsection presents a methodology for this.

3.3.8 Commonalizing joints with element densification

Creating product platforms implies the sharing of components in as many product variants as is beneficial. TU and SA reduce much of the process variety in joining aspects. The objective of element densification (ED) is to standardize sets of joining elements on module interfaces. Optimal commonalization would imply that the interface has one joining module for any module combination.

ED uses the IoCR as a geometric boundary. The IoCR is precisely the shared region over a unique set of joining scenarios. Commonalizing joining elements that lie on the IoCR enables them to be shared between all joining scenarios of the IoCR. ED considers IoCRs that have the same joining technologies over multiple product variants.

ED becomes a trade-off of joining element quantity when it is performed after TU. The previous step commonalizes technological variety, and every UoCR uses the same joining technology, as addressed in the multitechnology process (Fig. 3.36). Thus, the goal is to determine whether to substitute all joining elements of one scenario on the IoCR into all other joining scenarios. The same piece of geometry with similar boundary conditions will have different joining elements depending on the product variant. Increasing the number of joining elements in some joining scenarios can create more efficient joining element modules, which can be utilized in multiple variants. However, this may lead to increased manufacturing costs as it induces unnecessary joining processes. Therefore, ED must balance joining elements. Ultimately, ED may increase the number of joining elements in product variants, but it reduces the number of joining modules and as a result the total complexity.

ED may not reduce the joining performance. The approach only commonalizes through

the substitution (densification) of joining elements. Removing joining elements reduces structural performance requirements. The process assumes that increasing the number of joining elements does not drastically reduce structural performance. However, this may not always be the case. Yang et al. [18] argued that a proper distribution of joining elements is crucial in high-performance structures as the uncontrolled or arbitrary removal and addition of joining elements reduce performance. Fig. 3.40 depicts the process of ED, which is also explained in the following list:

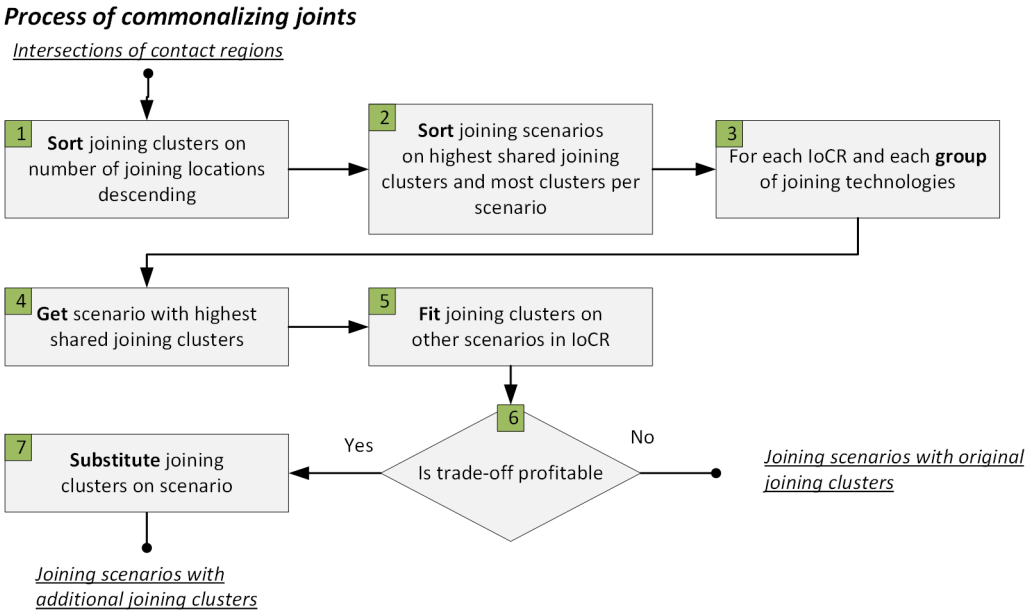


Fig. 3.40: Element densification process flow.

1. The algorithm starts by sorting joining clusters in descending order by the number of spatially aggregated joining locations. Sorting helps to increase the shareability of key joining locations.
2. Then, it sorts joining scenarios according to the number of joining locations in their joining clusters in descending order. Successively, it sorts the number of joining elements in the joining scenarios.
3. Next, the algorithm iterates through all IoCRs as well as iterates through all similar joining technologies for each IoCR.
4. The following nested iteration creates the retrieval of joining clusters. Densification is possible for joining clusters of both joining scenarios that lie on the same IoCR. The algorithm does not consider individual joining elements to densify, as these might require unavailable space and reintroduce variety.
5. Testing for standards and guidelines ensures that joining clusters that create new joining locations in other joining scenarios are manufacturable, considering edge distances, sheet thicknesses, or materials.

6. Lastly, the algorithm considers the trade-off to determine the benefit of densification.

ED is economically beneficial when the reduction in development costs (due to fewer joining modules) is greater than the increase in production costs (due to additional joining elements on some variants). From a performance point of view, each joining element that is added to a joining scenario and thus to a set of product variants will be manufactured unnecessarily. Once the sales volume of product variants with that joining scenario becomes greater than the reduction in the complexity of product documentation and design effort, densifying stops being advantageous. The catch here is that the benefits of modular design, especially for one joining element, are difficult to calculate as it is often too challenging to determine positive and negative effects in trade-offs [434].

Unfortunately, it is challenging to calculate investment and recurring costs in early product design, mainly due to product variety-induced complexity costs [435]. Therefore, studies have used indices, such as commonality metrics, as proxies for activity-based costing (e.g., [81]) in the design, manufacturing, and assembly of product variants [234]. High degrees of commonality between variants indicate high proportions of component reuse, implying more potential for companies to achieve greater economies of scale and scope [230]. ED can use any metric depending on the company interests. Gauss et al. [91] presented both module and product family evaluation criteria, including: modularity, utility cost, interaction, lead time, commonality, variety, or environmental perspectives. Furthermore, Alkan et al. [385] presented a method for assessing structural assembly complexity that can help designers to evaluate design alternatives that meet functional requirements.

The assessment of assembly complexity in early design phases is a viable proxy for a cost assessment as it combines the complexity of joints with their relation to the product architecture. The structural assembly complexity metric of Alkan et al. [385] further addressed as assembly complexity, considering component, interface, and topological complexity. Without going into depth for each term, the equation is presented in Eq. 3.26, after which each term is described:

$$C = C^P + C^I C^T \tag{3.26}$$

- **Component complexity** (C^P) expresses the sum of complexities of individual system components in the system. It describes the developmental and managerial effort of isolated components independent of the product architecture.
- **Interface complexity** (C^I) describes the sum of complexities of interfaces of every pair-wise connection between two components. The factors for calculating individual component and interface complexities originate from the Lucas method [147, 385].
- **Topological complexity** (C^T) expresses the inherent arrangement of connections by dividing the graph energy over the number of components. The graph energy measures the complexity of representing or reconstructing the adjacency matrix used in the interface complexity [54]. Highly dense networks correspond to more complex matrices for representing the adjacency matrix, resulting in high graph energy.

Fig. 3.41 presents the evaluation criteria with structural assembly complexity for whether to densify joining elements between joining scenarios A and B.

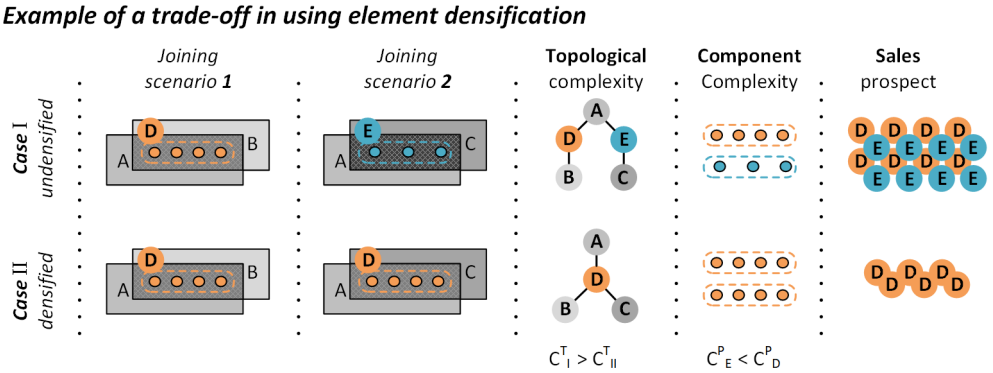


Fig. 3.41: Example of element densification of two joining scenarios $A\&B$. Cases I and II show the undensified and densified situation, respectively.

The *undensified* case I has two nodes representing the joining components D and E of joining scenarios 1 and 2, respectively. The *densified* case II has only one joining component D to join both joining scenarios. Component D is more complex than joining component E as it contains more joining elements $C^P_E < C^P_D$. Case II utilizes component D twice: $2 \times C^P_2$. However, case II has a more simple and centralized graph compared with case I, and thus, $C^T_1 > C^T_2$. Here, only one joining component D connects nodes A, B, and C, instead of both joining components $D\&E$ in case I.

Fig. 3.41 visualizes the densification by removing the joining module E. It removes an edge and a yellow node from the graph. Component D becomes a central node in the graph that joins components A, B, and C together. ED would remove joining module E from product documentation, assigning its function to another module D. It centralizes the architecture [54] creating bus components, which are integral-type modules that have connections with most other modules [213].

Structural assembly complexity enables an objective comparison of the densified (I) and undensified case (II); see Fig. 3.41. However, this requires a few changes to the assembly complexity of Alkan et al. [385], specifically changes to complexity terms for components C^P and topology C^T. The interface term C^I remains as is.

1. A temporary **joining component** is defined as a "*group of all joining elements on the same IoCR*." Joining components contain joining elements that all share the same product variants.
2. The part fastening attributes of the Lucas method [147] are not sufficient for expressing the complexity of joints. The component term C^P must consider the number of joining elements. Each joining component describes the complexity of the individual joining elements. For example, joining components with three, four, or five laser welding bonds must entail increasing complexity [436]. Furthermore, the complexity of each joining element can include a term for the weights of TU and SA.
3. Joining modules are part of the product architecture. Therefore, ED requires the integration of modules as nodes to express the topological complexity. Contrary to

Alkan et al. [385], this would only include the "essential" components A, B, and C (Fig. 3.41), and the topology term C^T needs to include joining components as nodes. Moreover, the graph virtually stacks all product variants to evaluate complexities for the configurations created from the IoCR. For joining scenarios 1 and 2, Fig. 3.41 depicts the topological complexity for all three components A, B, and C. Individual product variants would only integrate either A & B or A & C. It expresses the commonalization of the joining module over the joining scenario.

ED is beneficial when the assembly complexity of case II is lower than that of case I; see Fig. 3.41. The example indicates that the total component complexity C^P is lower in case I, but the topological complexity term C^T is lower in case II. The interface complexity term C^I is equal for both cases, as there are no changes to components or the joining technology [385].

$$C_{II} < C_I \tag{3.27}$$

$$C^P_{II} + C^I_{II}C^T_{II} < C^P_I + C^I_I C^T_I \tag{3.28}$$

$$C^P_{II} + C^I C^T_{II} < C^P_I + C^I C^T_I \tag{3.29}$$

$$C^P_I - C^P_{II} > C^I C^T_{II} - C^I C^T_I \tag{3.30}$$

$$\Delta C^P > -\Delta C^T \tag{3.31}$$

This trade-off (Eq. 3.31) states that it is beneficial to densify if the *increase* in component complexity from reusing fewer complex components is smaller than the *reduction* in topological complexity from removing joining components. The topological term can be negative for cases where there are no interchangeable joining components on the IoCR.

The previously discusses methodologies concentrate on commonalization. They are enablers for modularization of joining elements, of which a methodology is discussed in the following subsection.

3.3.9 Modularization with module grouping

Joining elements commonalized by TU, SA, and ED provide a solid basis for creating joining modules. Module creation collects joining elements used in the same products and places them together in modules. The modularization of joining elements is a necessary step for increasing manageability and reducing complexity in product documentation. To quickly recap, modularization standardizes components and processes. It enables the resulting models to be combined with many product variants [206]. Modularization deconstructs products into independent functional blocks, enabling parallel development and interchangeable parts to fulfill these functions.

The previous subsections describes methodologies to commonalize joining elements. These methodologies enable a more effective modularization, as was argued in Section 3.1.2. Commonality expresses similarity between products, aiming to decrease the number of components in a product family. An increase of shared components means a higher grade of commonality, which is expected to decrease cost and complexity. Module creation performs

commonalization and modularization to create groups of joining elements. Both approaches produce different results when optimized; see Fig. 3.42.

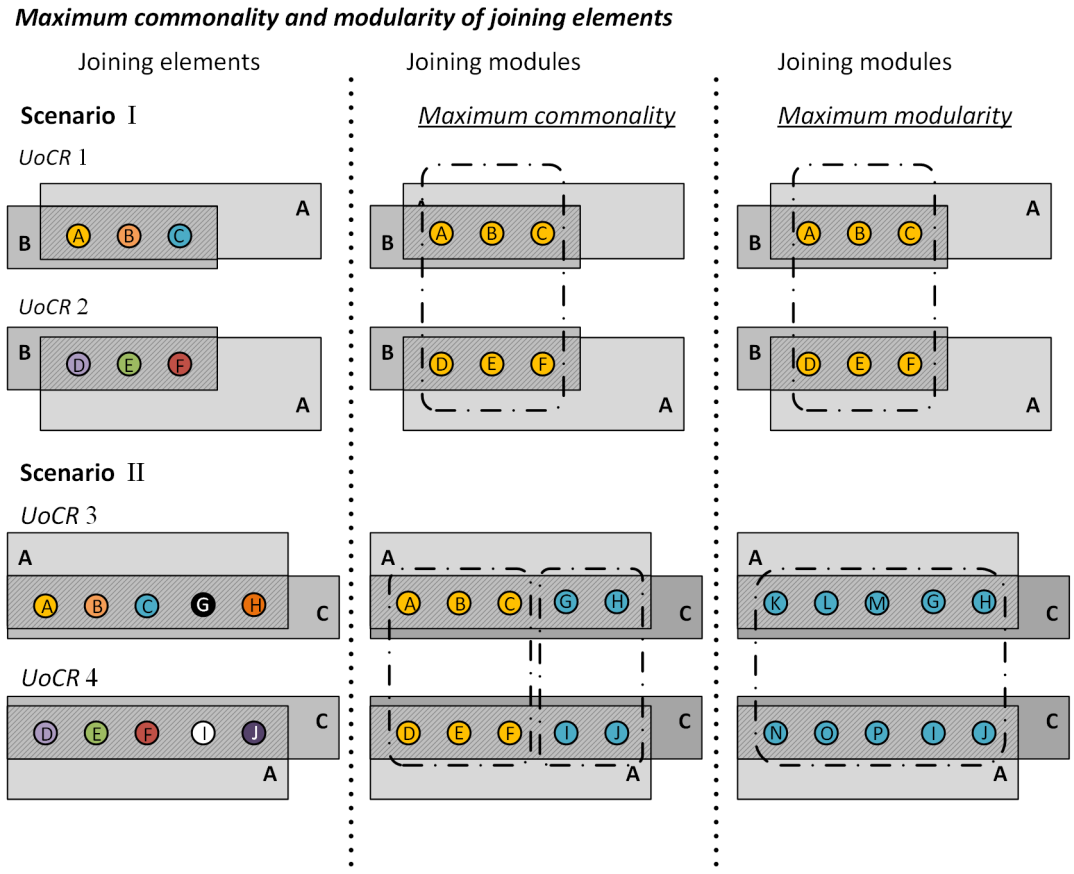


Fig. 3.42: Visualization of the difference between maximum commonality and maximum modularity when creating joining modules.

Fig. 3.42 also presents Venn diagrams with joining elements and components for each joining scenario. Every segment visualizes a shared group of components that share the same set of product variants. Maximum modularity in the right column indicates that joining elements can be redundant between two joining modules. However, this redundancy assigns entire joining modules to interfaces instead of combining multiple modules. However, module grouping may also include the components, as Fig. 3.43 illustrates.

Including components during modularization creates independent modules, enabling parallel development and outsourcing. These modules can be regarded as subassemblies and behave as a single unit. However, redundant joining elements decrease transparency and increase complexity.

Module creation grouping for maximum commonality results in joining modules being shared uniquely by product variants. This grouping implements one basic rule, namely that components included in the same set of product variants should be placed in the same module.

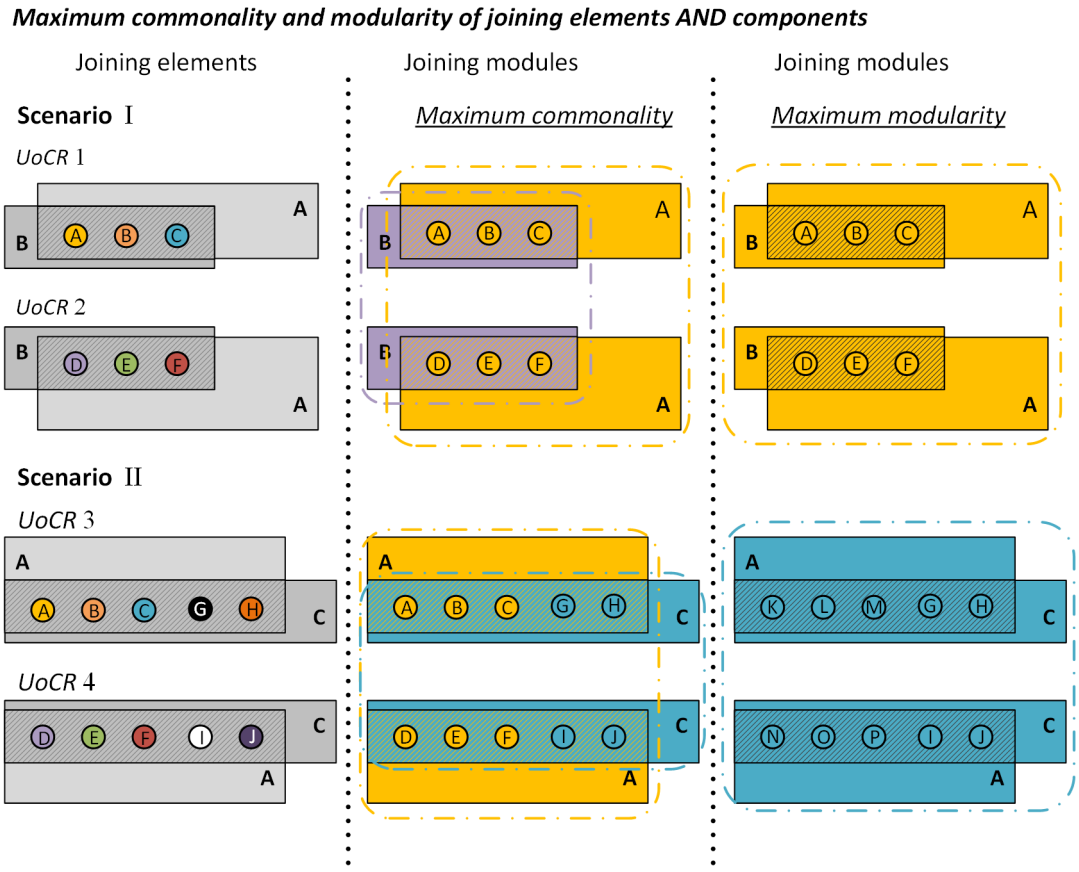


Fig. 3.43: Visualization of the difference between maximum commonality and maximum modularity when creating modules and subassemblies.

The maximum commonality of components results in nonredundant components. Joining modules may become small, causing joints to consist of multiple joining modules. Furthermore, joining modules may contain any arbitrary combination of joining elements in product variants due to shared occurrences. It may be that structurally and functionally nonrelated joining elements on different positions become a single joining module. High levels of commonality may increase the complexity in modules, module interfaces, and product architectures.

The grouping of joining elements must involve trade-off between modularity and commonality. A component shared by many product variants could be shared by fewer to allow it to integrate with other modules [437]. The functional requirements of components, joining elements, and products determine which approach is best suited. Baylis et al. [87] maximized commonality while considering modularity. They modularized sets of common components between products. Systematically reducing common components increased the modularization potential of individual products. A Pareto analysis enabled a trade-off between the two assessments. Although Baylis et al. [87] proposed one solution for balancing these

properties, methodologies such as those of Galizia and ElMaraghy [86] and Stocker et al. [84] can also create joining modules.

Regardless of the commonality–modularity trade-off, the results must be sustainable. Modular sustainability refers to the ability of modules to cope with changes. It may require the redefining or creation of joining modules. Joining modules require size and joining element variety to be considered during their design. Han et al. [80] stated that "*[c]ommonality is limited to common modules defined by fixed physical standards*." They opted for greater flexibility on interfaces and proposed uncertainty-oriented product platforms. The product development process itself is continuously reiterating, creating new variants asynchronously. Large and diverse modules are less sustainable, even though customer requirements might induce small changes.

This section proposed new AI methodologies to automate joining element design. To conclude the chapter, the following section discusses the VICTOR framework, the applicability of AI fields to automate joining element design, and proposed methodologies.

3.4 Discussion

This chapter has discussed the design of joining elements in manufacturing industries with high product variety. It has presented a framework called VICTOR with modular methodologies depending on company needs. The framework organizes various AI methods that it can implement to predict or modularize joining elements. They work as a divergent and convergent force on designs. Prediction approaches have the freedom to make designs, after which modularization methodologies collect results and bring them to a common sustainable denominator. This chapter also identified and discussed the applicability of AI methods to automate joining element design.

The applicability of AI methods relies on many factors, which differ for each company and industry. The manufacturing industry continually implements new joining technologies and component designs, inspired by trends such as lightweight design. These scenarios are difficult to model, and thus, they often fall outside of ML model boundaries. Therefore, human designers or optimization methodologies are required to solve unencountered problems, such as in the study of Florea et al. [46]. The VICTOR framework contains methodologies applicable for every specific use case regardless of product maturity. It can take over repetitive and time-consuming tasks in the user journey. In turn, this will enable designers to concentrate on their core competencies [2]. It is crucial to recognize that there is a proper time and place for all methodologies. Moreover, the applicability of AI fields (Section 3.2) reveals that ML cannot completely take over as the only AI method for the automation of joining element design.

The framework introduces ML for tackling multiple problems in automated joining element design. It takes advantage of historical data by learning the designs of successful product variants. It can also learn the relationships of joining scenarios to the product as a whole yet also predict joining scenarios. ML mimics manual design practices on various levels of cognitive abstraction. First, it may predict features such as joining technologies or the number and distribution of joining locations. Furthermore, it can directly predict joining locations through geometric ML. Lastly, it can learn from topology optimization algorithms

and apply generalized knowledge of numerous load case simulations to new products. All solutions align with the minimalistic, experience-based, and use case-based manual design methods. Increasing the complexity in ML models would increase the ability to predict joining elements but also increase the solution space. Hence, computational cost limits the implementation of ML methods for joining element prediction.

The prediction of joining locations is the most challenging task due to the enormous solution space induced by geometry. First, training one model for each joining technology helps with performance. Another strategy may be to input the joining technology as a multimodal feature and let the model learn the relationship between joining locations and its technology. Second, ML approaches require consistently structured data with a fixed number of inputs and outputs. However, joining scenarios may consist of varying numbers of components. Moreover, geometries vary in size and complexity, requiring the balancing of geometric details and computational cost. Euclidean data structures retain spatial dependencies in samples, creating a relatively straightforward process of reconstructing Cartesian coordinates. 2D image-based ML models more effectively limit computational cost as fully 3D counterparts. They can use the same classification and regression concepts. However, significant drawbacks are the projection of curved CRs onto 2D images as well as coordinate reconstruction, which requires the inverse of the projection. Furthermore, multiview-based methods with arbitrary camera positions suffer from the necessity of coordinate reconstruction.

The commonality and modularity methodologies reduce variety in joining element design by aggregating and standardizing joining elements. They aim to reduce product variety-induced complexity costs and aggregate uncertain predictions. Hence, they integrate DFA principles in early product development phases from a modular design perspective.

The methodologies consider every joining element as an individual component, enabling one to define joining modules on module interfaces. A five-step modular design model for joining elements (see Section 3.3.5) commonalizes joining aspects (technology as well as parameters and locations) and joints. Modularization may include joining elements alone or with components, for example, to create subassemblies.

Moreover, the methodologies fill gaps in product family modeling. These gaps were identified from the meta-process of Gauss et al. [91]. First, the modular design block of VICTOR defines interfaces for joining elements. These interfaces become functional modules themselves. To this extent, the interfaces using geometrical layouts to create a mapping of structural dependencies between components; see Fig. 2.8. Furthermore, they enable physical modules, such as a group of spot welds, to be independent and interchangeable. The physical joining modules are derived from commonalization methodologies that can reduce variety in joining elements. Subsequently, modularization methodologies can create these building blocks.

Furthermore, commonalization processes might create suboptimal solutions for individual joining scenarios and potentially violate performance-containing requirements. Standards and guidelines only validate the manufacturability of joining scenarios, but not their performance, such as strength or stiffness, using finite element methods. The uncontrolled or arbitrary removal and addition of joining elements might reduce performance [18]. Thus, after applying

commonalization and modularization, designers need to revalidate the designs to ensure their quality.

TU might leave multiple technologies open due to manufacturability limitations. Multiple technologies would require modularization steps in parallel UoCRs. Joining elements then map as variants and potentially still may share the same joining locations. For example, sophisticated documentation methods enable varying parameters [31]. They might have different technologies and parameters depending on a specific product variant. However, such implementations depend on the size and complexity of product documentation as well as a company's strategy for handling them.

The modularization of joining elements remains highly dependent on the quality of component design. Suboptimal designs or variety considerations may result in low-quality joining modules. However, the results of individual steps enable further analysis and insight into the complexity induced by joining element designs [433]. Analyses may recommend tuning the weights of modularized joining elements and their modules, thus assisting the modular sustainability of the product family.

After summarizing this chapter, the following chapter discusses the validation of several proposed methodologies in the VICTOR framework.

Chapter 4

Validation

This chapter describes the validation strategy of the VICTOR framework. Whereas Chapter 3 discussed how to generate joining element designs, this chapter applies this knowledge to generate designs using real data. Suppose that a firm has generated many joining element designs. These designs create a dataset that methods such as ML can train on. Additionally, they enable other methods to measure performance compared with successfully marketed designs.

The framework from Chapter 3 supports designers in selecting appropriate methodologies and tools for generating joining element designs. Moreover, it identifies underused techniques within these methodologies, such as ML for joining location prediction. These techniques may help overcome the various limitations in state-of-the-art joining element design, which currently include the consideration of successful designs, entire products, and other product variants.

Thus, the framework identified the promising applicability of various AI fields, such as SML (Section 3.2). This chapter aims to validate these findings through various methodologies. Fig. 4.1 visualizes how to place these methodologies in the framework.

The figure presents relevant AI fields and techniques (Fig. 3.8) and links them to the sections of this chapter. The next few paragraphs introduce the topics of validation.

For the prediction of joining technologies, locations, and parameters, SML is deemed applicable. However, no methodologies were found in the literature for these tasks (see Section 2.3.7). Hence, this chapter explores SML for predicting joining technology (Section 4.1) and locations (Sections 4.4 and 4.5). This chapter aims to validate whether the classification of joining technologies using SML as a rapid design tool is an alternative to multidisciplinary optimization methodologies. Its validation for predicting joining locations is more complex.

Recent computer vision techniques rapidly set new benchmarks in the performance of image segmentation and object detection. These are techniques that, with a few adaptations, might work for the prediction of joining locations (Section 3.2.2). This chapter aims to validate that SML (using the grid-based drawing concept from Section 3.2.2) enables the reuse of joining designs, and also to determine its boundary conditions. Therefore, it first must determine the more appropriate learning task: classification or regression (Section 4.4). Then,

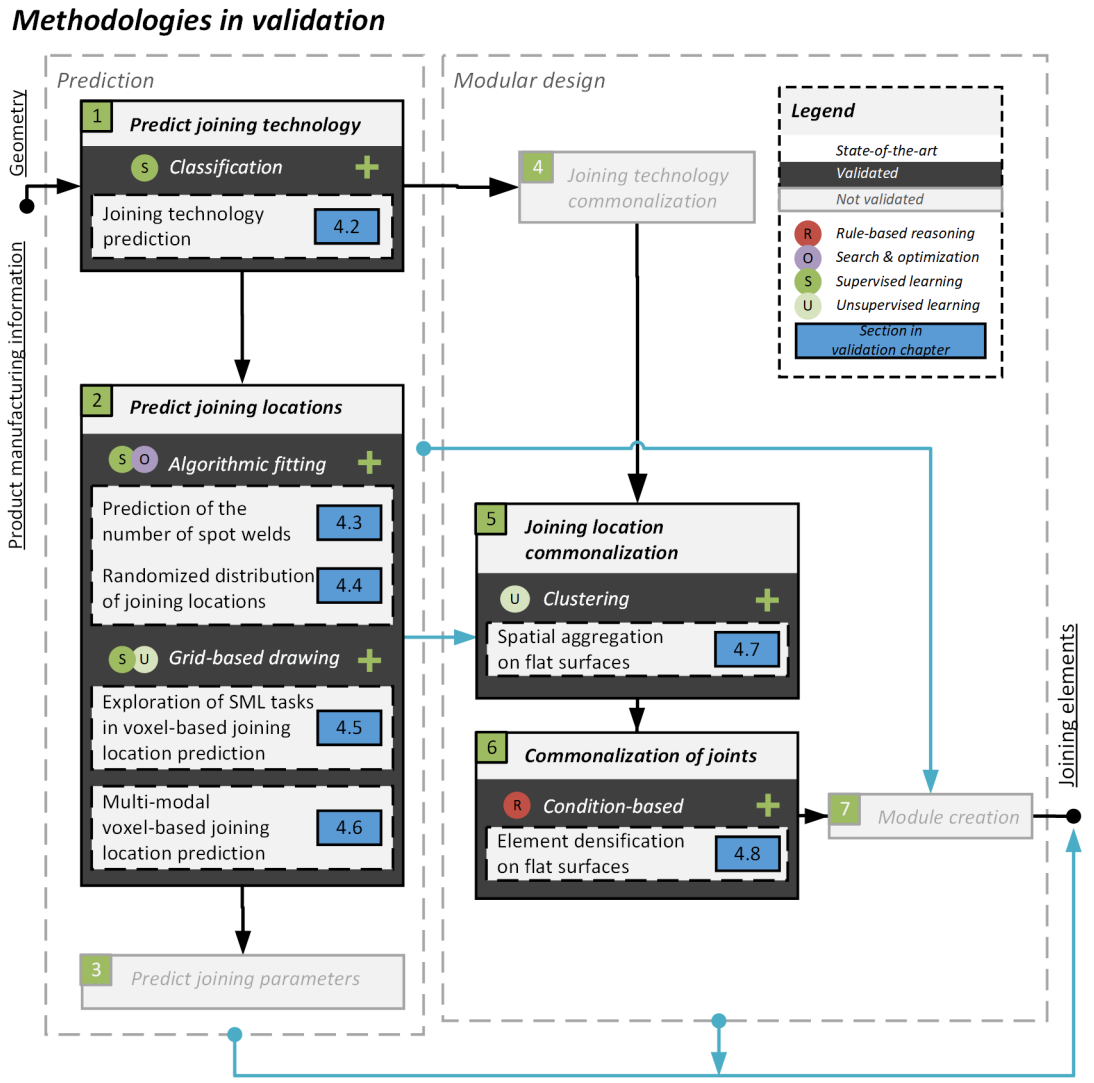


Fig. 4.1: Allocation of the proposed methodologies in the framework of Fig. 3.8.

the chapter validates the integration of nongeometric data into a computer vision-based prediction model (Section 4.5). Nongeometric data, such as material and function, contain additional information about the joint that is supplementary to geometries [376]. It enriches the information for prediction models, thereby improving their performance.

However, SML prediction models for joining locations require data samples that contain geometries. The representation of these geometries is vital in the balance between computational cost and informational content (see section 2.3.5). Therefore, this validation chapter also explores algorithmic fitting (Section 3.2.2) as a more lightweight prediction model. Algorithmic fitting can use S&O techniques for distributing joining locations, thus preventing the high developmental effort required in RBR methodologies [101]. Additionally,

the distribution does not rely on patterns within joining locations, which enables its application in earlier product design phases. The aim is to validate whether predicting the number of spot welds using SML (Section 4.2) and distributing these points using an evolutionary algorithm (Section 4.3) work as a simpler, faster implementation compared with only using SML.

Besides evaluating the applicability of AI for predicting joining elements, VICTOR also presents four modular design problems for joining elements (Section 3.2). The literature does not contain many methodologies for these design problems. Although, it describes methodologies that are equivalent or perform similar tasks as TU (Section 3.3.6) and module grouping (Section 3.3.9), for example, the studies of Ukala and Sunmola [237] and Stocker et al. [84]. respectively. Hence, methodologies proposed in the VICTOR also include SA (Section 3.3.7) and ED (Section 3.3.8), which are conceptually novel. SA aims to reduce the variability in joining locations caused by nondeterministic prediction models (Section 4.6). Moreover, ED aims to unify joints between multiple product variants (Section 4.7). Together, these are two methodologies systematically commonalize joining element designs to reduce the negative impacts of product variety.

All of the methodologies being validated must be evaluated according to the same set of criteria. The state of the art indicates that the main problems in the joining element design process are time-consumption and quality. In early product design in particular, where many boundary conditions and dependencies are still unknown, rapid results are vital to ensure that designers can continue their work. Speed enables quicker iterations and reduces lead times. Higher quality through, for example, the consideration of successful designs or other product variants prevents unnecessary design iterations. However, these properties are difficult to measure between the different boundary conditions for each methodology in this chapter. Hence, the methodologies receive a qualitative rating. In addition to the main issues in the joining element design process, comments are also made on the methodologies' development effort, computational cost, structural performance, and applicability to actual use cases. The assessment section of this chapter (Section 4.8) lists the methodologies according to the same set of properties used for the literature overview (Section 2.4).

However, this validation chapter does not assess the structural performance of predictions in depth (as introduced in Section 2.1) due to the developmental effort, the scope of this study, and practicality. Proper structural performance requires testing of the product as a whole as well as many use cases [25]. Moreover, as the scope of this study included product variety, the joints also had to be tested for multiple product variants. These variety tests were required to validate the joints under various stresses and strains.

Hence, and as aforementioned, a large real dataset enables the comparison of predictions with successful designs. Thus, regardless of how that dataset came into existence, this validation chapter assumes it to contain ground-truth "optimal" designs. However, small qualitative discussions are also presented on structural performance during the analysis of selected predictions. Additionally, the correlation between the number of spot welds and nongeometric data, as well as the prediction of the number of spot welds (Section 4.2), briefly addresses the ability to cope with tension and compression stresses. This chapter focuses on resistance spot welding for the following reasons:

- Modeling and product life cycle analyses are significantly different between joining technologies. Prediction models that would consider multiple joining technologies would induce difficulties due to variety and uncertainty. For example, technologies such as riveting and clinching create holes in components that change their structures and performance. Additionally, some curve-based joining locations, such as laser beam welding, are located on edges of components' flanges, creating a significantly different CR and boundary conditions.
- Resistance spot welding is one of the most commonly used joining technologies in the automotive industry [25]. This popularity creates a large dataset. Additionally, its popularity in the literature indicates global interest in the joining location prediction of spot welds.
- Learning approaches require historical data to train and test models. Although spot welding is a somewhat dated joining technology, its popularity over the years has created a large dataset. Furthermore, the modeling of spot welds is relatively simple. Often, mere Cartesian coordinates can represent spot welds.

Additionally, the design of joints using spot welding has been researched extensively (see Section 2.1.5), especially in the automotive industry [18]. The reasons for this interest are as follows:

- The automotive industry is a leading sector in product variety and modularization [438];
- There is extensive documentation of products [19];
- There are high-quality design requirements due to regulations and safety [8];
- There are a high number of joining elements in every product variant [18].

Therefore, this validation chapter uses a dataset from the automotive industry. The raw dataset includes 13 cars from Mercedes-Benz that differ in designs through properties such as carlines, model types, and steering sides. It includes the geometry and some parameters in CAD files, which were extracted from PDM systems. Hence, the raw dataset comes from actual successful vehicles on the roads today. The samples in the dataset consist of joining scenarios. Each scenario describes two CAD parts with joining elements, resulting in 6517 data samples. Joints that assemble more than two components were reduced into two-component joints. This simplification keeps the data structures constant and significantly reduces complexity in algorithms. Then, the combination of two components without joining elements represents the input state, and with the joining elements the output state (ground truth).

The particularities of each methodology in the validation require them to have subsets of the raw dataset. For example, to predict the number of spot welds, a dataset that only contains the appropriate joining scenarios is required. Fig. A.1 presents the steps used to create the datasets for each validation. Each methodology briefly addresses the derivation of the dataset in their respective section of this chapter.

Before validating the methodologies, a preliminary data analysis was performed (Appendix A.2). That section discusses the properties of both the joining technologies and the number of

joining locations in the nongeometric dataset. Some methodologies in this chapter have their results confirmed on the basis of the analysis.

Most methodologies were validated using separate datasets, methods, and performance measurements. However, to maintain structure in this chapter and to retain the linkage to VICTOR, each of the following sections describes one methodology and is divided into six subsections: approach, implementation, process, results, discussion, and evaluation. This division not only helps to find parallels and distinctions between the methodologies but also to structure each methodology.

Following the sequence of steps in the user journey (Fig. 4.1), the first section in validation is the prediction of joining technologies.

4.1 Joining technology prediction

Section 3.2 identified SML as a promising field for predicting joining technologies. The prediction model needs to recommend the appropriate technique through classification (Section 2.3.3). The multiclass classification model shall output conditional probabilities for each joining technology. The highest probability corresponds with the selected joining technology class. The goal of validating this methodology is to determine whether SML can predict the joining technology.

Approach

Joining technology prediction using SML enables rapid recommendations, especially in early product design. Current methodologies that, for example, implement MCDM optimize for the best joining technology. However, they require much information that is not necessarily available at a given moment in time (Section 2.1.3). As SML relies on patterns in data, it relies on similar use cases. Moreover, the selection of a joining technology must consider the component geometries. However, as the model performs a classification task, the data representation of these geometries may be abstract, such as size, weight, or encodings. Hence, data preparation may combine the geometric information with further structured data, such as materials and functions (see joint representation in Section 2.3.6).

The state of the art describes many different methods in SML (Section 2.3.3). To validate the methodology, it is beneficial to keep the prediction model simple. Decision trees are simple to understand and to preprocess, and they can handle the required data formats [279]. Additionally, designers can trace every decision in the tree to verify its outcome. Although decision trees are relatively simple to train, they require appropriate data and formatting to achieve good results.

Implementation

Fig. 4.2 depicts the data preparation process for predicting joining technology.

The image indicates the effect of each step on the number of data samples m and the number of features n_f. The latter is distinguished in the number of numerical features plus $(+)$ the number of nominal features. The following list presents the steps of this process:

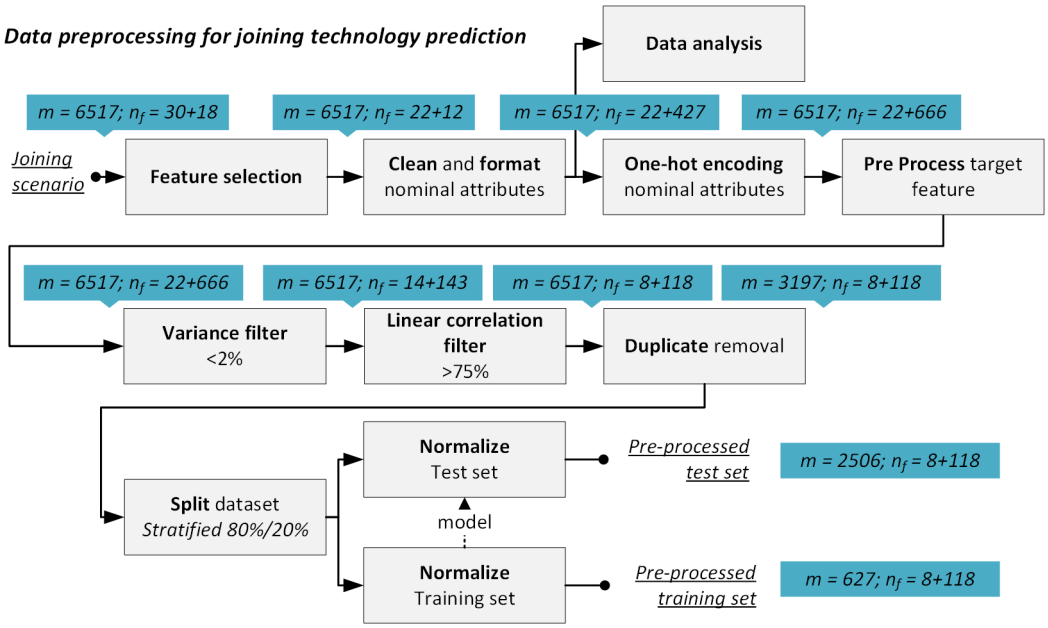


Fig. 4.2: Data preparation process for predicting the joining technology. It tracks the number of features n_f and number of data samples m throughout the process.

1. Feature selection filtered the raw dataset by removing nonrelevant attributes. This includes nonrelevant PMI in the CAD files, such as versions or multilingual nomenclature.

2. Nominal attributes may include text and numerical values. Cleaning and formatting repaired errors such as spelling mistakes in nomenclature and inconsistent material naming.

3. One-hot encoding [275] split all string values into individual words and codes that represent the information. Each class became a feature. Here, each data sample may receive one 1 for a given class in a feature. Moreover, label encoding enabled multiple words to be addressed if the original feature was a text. When each word in the text was a new feature, label encoding set true values ($\equiv 1$) at all of these words.

4. The preprocessing of target features handled outlier target classes. For example, when a given exotic joining technology occurred twice in the entire dataset, it might have an undesirable effect on the performance of the prediction model. Additionally, this processing step integrated subvariants of joining technologies into one, such as combining self-piercing riveting with blind riveting into riveting. This step increased the available data samples for the class of riveting, but also reduced the complexity of the prediction model.

   Classes below 15 data samples were filtered, resulting in 18 different target labels. These classes included labels representing combinations of technologies (e.g., adhesive bonding with spot welding).

5. Features with very weak relationships with the target classes were filtered. The variance filter removed features with low variance ($\leq 2\%$), which would likely distract learning algorithms. Furthermore, the correlation filter removed features with a high correlation ($\geq 75\%$) to the target classes as they were redundant in the prediction task. The correlation filter calculated coefficient pairs of selected features. For numeric-to-numeric features, it used the Pearson correlation coefficient [439], whereas for nominal-to-nominal features, it used Pearson's χ^2 test [440] normalized by Cramer's V [441] with $p = 0.05$.
6. The filtering of features may create duplicate samples in the dataset. The removal of duplicates step filtered these to prevent any biases when training the prediction model.
7. After preprocessing the data samples, they were split into a training dataset and a test dataset. Stratified splitting implies that the distribution of each target class is equal between both datasets.

   The training and test sets had 2506 and 627 data samples, respectively. Each sample had 126 attributes, of which eight were numeric.

Now, the dataset contained useful features for the prediction of joining technologies. However, to validate the prediction models, feature engineering might help to assess the importance of each feature in prediction. The most critical features are the most vital ones to create in early product design to enable joining technology recommender systems based on SML. Additionally, they enable the analysis of noise and additional complexity when compared against a model trained on all features.

Feedforward feature selection enables the evaluation of the importance of each feature. It finds and ranks the best features for the prediction tasks. Feedforward feature selection starts with individual features to determine the most important one. Then, it evaluates all combinations of two features, considering the previous best feature. It continues this process for all combinations of best features. Measuring the importance of features requires three methods: decision trees, random forests, and XGBoost to cross-validate their results. The process of feedforward feature selection is discussed further in the **process** section as well as Fig. 4.3.

For example, Fig. A.5 presents the feature importance for each of the classifiers, namely decision tree, random forest, and XGBoost, respectively. The following list presents the selected features using each method. These sets were cherrypicked using the elbow method [322]. The features were often similar between the methods, such as the thickness, moment of inertia, material, and coating.

- Decision tree ($12\times$): car line, a coating, base materials of both components, alloy of component 2, thickness of both components, depth in the product architecture of component 1, cross member in component 2 (nomenclature), closing sheet (nomenclature), a submodule, and a moment of inertia of component 2.
- Random forest ($9\times$): car line, surface areas of both components, one moment of inertia of both components, thickness of component 2, front seat (nomenclature), roof structure (nomenclature), and above (nomenclature).

- Random forest (16×): features of random forest (9×), pillar (nomenclature), product structure depth, main floor (nomenclature), front wall (nomenclature), fund seat (nomenclature), and two coatings.
- XGBoost (10×): one moment of inertia of both components, thickness of component 2, area of component 2, product architecture depth, part (nomenclature), casting, a submodule, and two coatings.

The feature sets include traditional information that corresponds with engineering considerations for the selection of joining technologies. For example, the moments of inertia and thicknesses indicate an expected load due to the dimensions of the components. Furthermore, the coatings of components may indicate corrosion from, for example, water leakage, which sets up for adhesive bonding. A prediction model will receive each set of features. The process of training of these models is addressed next.

Process

Fig. 4.3 presents the generic process for generating decision tree-based models for the prediction of the joining technology.

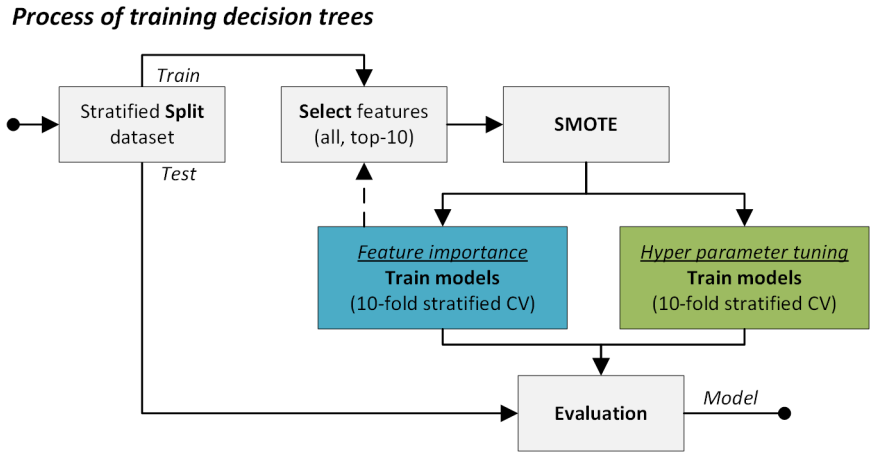


Fig. 4.3: Process of training classification and regression decision trees for feature engineering and hyper-parameter tuning for the tasks of predicting the joining technology and number of spot welds.

Fig. 4.3 displays both the process of feedforward feature selection and hyperparameter tuning for any model (e.g., random forest and XGBoost). The inputs are the preprocessed datasets with the aforementioned sets of features; see Fig. A.2. The following list discusses the steps in the process:

1. The dataset splits into a training and test dataset with an $80\%/20\%$ ratio. This step is redundant to the last step in Fig. 4.2.
2. The features on which to train are selected. These features may include the entire dataset or cherry-picked features. Additionally, they may result from a process to determine the most important features, such as the top 10.

3. Synthetic Minority Oversampling Technique (SMOTE) [442] is a statistical method for increasing data samples for classes with low frequencies. It generates new data samples by combining feature values of nearest neighbors ($n_{nn} = 5$) for every target class.
4. The decision trees are generated using k-fold cross-validation (CV) to verify the results. CV uses small test sets, where statistical uncertainty is more considerable for the estimated test error, preventing confidence in evaluations between models [275]. An implemented 10-fold CV splits the dataset into 10 equally sized subsets [443]. Every subset is used once as a target set the other times included in the training set. Averaging the test errors over k-folds returns the models' test error [275].

   Here, the forward feature selection method finds and ranks the optimal features for the prediction tasks. Besides forward feature selection, the hyperparameter tuning process performs a grid search on the properties of decision trees, such as their number and depth. Increasing these parameters enables the decision tree to increase the complexity of the task. The following list presents the values used for the hyperparameter tuning of regular decision trees, random forests, and XGBoost classifiers:

   - **Random forest** and regular decision trees
     Maximum tree depth: $[5, 10, 20, 30, \text{None}]$
     Number of trees: $[50, 100, 200, 250]$
   - **XGBoost**
     Maximum tree depth: $[2, 4, 6, 8, 10, 12, 14]$
     Learning rate: $[0.1, 0.2, 0.3]$
     Number of boosting rounds: $[50, 100, 150, 200, 250, 300, 350]$

5. Evaluation scores the best model after hyperparameter tuning on the hold-out test set. The measurements include the accuracy (A), F1-score ($F1$), and Cohen's kappa (κ) (Section 2.3.3).

Feature engineering resulted in four feature sets after feedforward feature selection. These sets implemented three different decision tree methods. Random forest and XGBoost are more complex and robust methods than the use of only one decision tree. Hence, the decision tree method is only used in feedforward feature selection. The empirical analysis resulted in the following six prediction models for validation:

1. **Random forest** classifier on *all* features of the dataset after data preparation;
2. **XGBoost classifier** on *all* features of the dataset after data preparation;
3. **Random forest classifier** using the 12 most important attributes from feature engineering with regular *decision trees* (FE-DT);
4. **Random forest** classifier using the nine most important attributes from feature engineering with *random forests* (FE-RF-9);
5. **Random forest** classifier using the 16 most important attributes from feature engineering with *random forests* (FE-RF-16);

6. **XGBoost** classifier using the 10 most important attributes from feature engineering with *XGBoost* (FE-XG).

Results

Table 4.1 presents the joining technology prediction results. The identifiers of the rows correspond to the numbers of the models in the list above. All models demonstrated promising results, as indicated by the test set scores being above 70% for all metrics. XGBoost seemed more promising for models that use all features with an F1-score of 85.8%. However, the random forest classifier using the nine most important features resulted in an F1-score on the test set of 94.8%. The Cohen's kappa scores led to a similar conclusion when accounting for the class frequencies.

			ACC		$F1$		κ	
ID	**F**	**DT**	**TR**	**TE**	**TR**	**TE**	**TR**	**TE**
1	all	RF	99.4	90.7	99.4	82.2	99.4	87.3
2	all	XG	99.4	**92.7**	99.4	**85.8**	99.4	**90.0**
3	FE-DT	RF	94.9	79.6	94.8	74.6	94.6	73.0
4	FE-RF-9	RF	98.8	**95.7**	98.8	**94.8**	98.7	**94.1**
5	FE-RF-16	RF	99.0	94.3	99.0	91.9	99.0	92.2
6	FE-XG	XG	96.3	83.8	96.3	78.4	96.1	77.7

Table 4.1: Results of the trained models for predicting the joining technology. Abbreviations: F – selected features set, DT – tree learning method, RF – random forest, XG – XGBoost, TR – training set, CV – cross validation set, TE – test set, ACC – accuracy, $F1$ – f1-score, and κ – Cohen's kappa.

Table 4.1 provides an overview of the models, while Fig. 4.4 presents the distribution for each predicted class of each model.

The figure normalizes the distribution of the spot welds to make the models comparable. The x-axis lists all target labels. The graph also visualizes the dominance of spot welds creating a skewed distribution for the learning task. All models seemed to replicate the target distribution of classes rather well. However, model *ID3*, which used a random forest with a decision tree-based feature set, followed the trend the least. Table 4.1 also indicates that this was the model with the worst overall performance.

Discussion

The results were promising. The random forest prediction model (*ID4*) that used the nine most important features achieved a high performance, whereas the random forest model that trained on all of the data (*ID1*) achieved a lower performance. It may be that the tree depth or number of trees for the random forest classifier was too low. This parameter tuning might have prevented the model becoming complex enough to explain the variance in the data. In other words, the model had size limitations. Another explanation might be that the other features contained noise, which could distract the model. The model (*ID1*) achieved 99.4% on the test set for all performance measurements. This performance might also indicate overfitting of the model on the training set, which would reduce the performance when predicting on unseen cases.

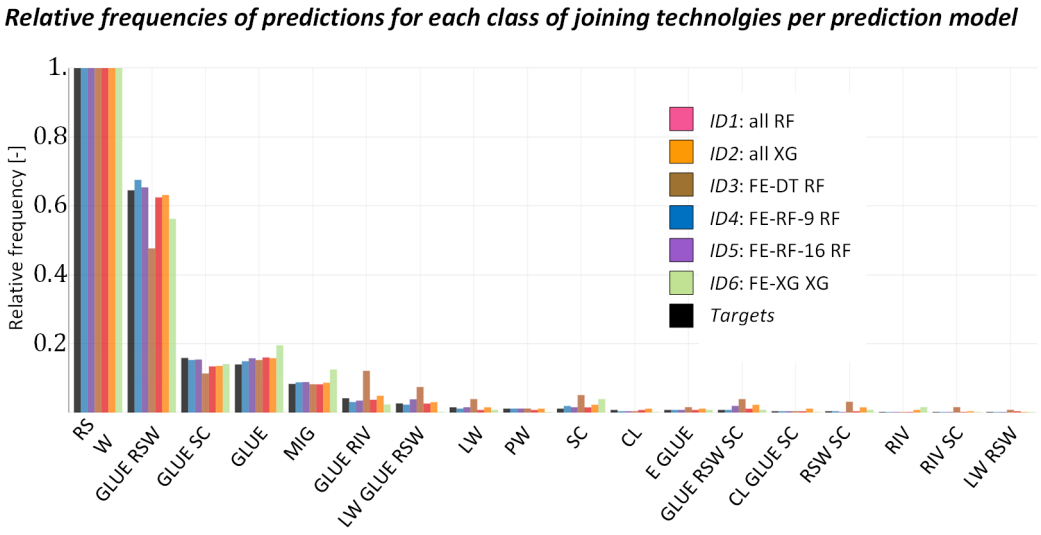


Fig. 4.4: The predictions of the joining technology normally distributed over the classes. Abbreviations of target classes: RSW – resistance spot welding, SC – cutting clinching, MIG – metal-inert gas welding, RIV – riveting, LW – laser welding, and PW – projection welding.

Nevertheless, reducing to nine of the 126 features enabled the performance to significantly increase.

Compared with the random forest models, the XGBoost models exhibited the opposite behavior after feature engineering. While, one XGBoost model (*ID2*) achieved a similar performance when using all of the features as Random Forest model (*ID1*), it lost performance when using the 10 most important features, dropping to 77.7%. The used features largely overlapped with those found by random forest-based feedforward feature selection. Possible reasons may include not only the hyperparameter settings but also the data. The variance in the data might lend itself better to training with random forests. This performance difference might not occur for other use cases and data formatting. Interestingly, the XGBoost models also required a significantly longer training time. Still, XGBoost is a technique to consider in joining technology prediction.

As mentioned, hyperparameter tuning might have a significant impact on the outcome. However, the training of models requires expert knowledge. Joining scenarios become more complex with combinations of technologies and subvariants thereof. The skewed distribution – in the dataset of this automotive use case – between bread-and-butter technologies, such as resistance spot welding and exotic ones such as riveting (see Fig. 4.4), makes training and implementation tricky. Preprocessing must aggregate certain variant technologies under more generic labels for a consistent dataset. This includes defining a set of technologies, such as resistance spot welding over an adhesive bond, as one surrogate technology. Moreover, two components may have multiple joints that all consist of multiple technologies. In short, the simplification of target features and data preprocessing has an effect on bringing such models into practice. Large simplifications might increase the performance significantly. However, this

performance increase reduces the precision of models. Here, precision describes the ability to predict specific types of joining technologies for a given data sample.

Furthermore, the sets of the most important features are largely generic and well-known to the industry, such as thicknesses and moments of inertia. However, other features are highly application-specific, such as coatings and terms in nomenclature. Keywords such as "front seat" and "roof structure" add value but are specific to this dataset. The nonstandardization of nomenclature, material, surface treatment, and coating documentation creates many features and cumbersome preprocessing. Hence, these features aid models in becoming specific to the use case.

Moreover, the selected features from feature engineering overlapped with those specified in table-based approaches, such as in the work of Bond et al. [41]. Their methodology enables designers to manually select joining technologies based on materials and joining characteristics, such as sheet thickness, joint dimensions, surface finish, and strength. Moreover, the selected features overlapped with the process groups of l'Eglise et al. [152]. The recurring features validated feature engineering and confirmed correlations in the dataset; in other words, the models could not find an arbitrary combination of features that coincidentally achieved a similar performance.

Evaluation

Table 4.2 summarizes the trained models.

ID	F	DT	Perf.	Speed	Dev.	Comp.	Struc.	Appl.
1	all	RF	2	2	2	1	–	2
2	all	XG	2	2	2	3	–	2
3	FE-DT	RF	1	3	3	1	–	1
4	**FE-RF-9**	**RF**	3	3	3	2	–	3
5	FE-RF-16	RF	2	3	3	2	–	3
6	FE-XG	XG	2	3	3	3	–	1

Table 4.2: Evaluation of the trained models for predicting the joining technology. Values $[1-3]$ correspond with low to high scores. The minus $(-)$ indicates no differentiation between models on criteria. Abbreviations: F – selected features set, DT – tree learning method, RF – random forest, XG – XGBoost, Perf. – performance, Dev – development effort, Comp. – computational cost, Struc. – structural performance, and Appl. – practical applicability.

It rates each model according to the six validation criteria discussed at the beginning of the chapter: performance, speed, development effort, computational cost, structural performance, and practical applicability. The ratings are qualitative as a result of the results and discussion. The range of values is $[1-3]$, where 1 is low and 3 is high. A minus sign $(-)$ presents nondifferentiating criteria. For example, structural performance is not relevant for joining technology prediction due to the type of task. The rating for each model in Table 4.2 has the following considerations:

- Models that rely on feature engineering have increased *computational cost*. An additional process is necessary for determining the most important features.

- Moreover, these models require increased *development effort* to set, analyze, and create the prediction models.
- However, the resulting prediction models from feature engineering are simpler. These models implement a smaller number of features. This reduction increases the *speed* of prediction and validation.
- Regular decision trees and random forests train significantly *faster* than XGBoost. This property of the method prevents quick validation and experimentation for large feature sets.
- The *industrial applicability* favors simple models in times of new materials and technologies. Additionally, model *ID4* has the highest performance.

Table 4.2 indicates that feature engineering using feedforward feature selection with random forests is the most likely model to have promising results. The model also aligns best with the boundary conditions of the use case. It relies only on a few features that are available in early product design, has high performance, and predicts rapidly. However, much research exists on new joining technologies and materials. SML models need to be retrained on strategic moments to cope with new trends and boundary conditions.

In short, joining technology prediction is a highly feasible task to perform using SML. The random forest-based models predicted with an F1-score up to 94.8%. Feature engineering can positively affect prediction performance, training speed, model complexity, and its ability to generalize. Because the most important features are generic in nature, models that utilize them can support designers during the early phases of product development.

Whereas this section regarded the prediction of joining technologies, the following section validates a subproblem in predicting joining locations; predicting the number of joining locations.

4.2 Prediction of the number of spot welds

VICTOR, the framework provided in Chapter 3, presents algorithmic fitting as a promising approach for creating joining locations. However, it proposes algorithmic fitting as a combined methodology. First, this requires data on the joining locations and how many there are, which are then distributed on CRs. The prediction of the number of spot welds enables a secondary algorithm to determine the joining locations. This number correlates with the structural performance. Typically, it implies more spot welds in areas that need to cope with higher stresses and strains. However, predictions of the number of spot welds mainly correlate with tension and compression. Other stresses require the consideration of geometry and joining locations; see Section 2.1. Therefore, this methodology can only consider a small part of structural performance. Hence, the main goal of this validation was to determine whether the number of spot welds can be predicted using SML.

The learning task can be regression and classification. The target feature (number of spot welds) is a numerical value. Regression can treat this value as continuous. Continuous variables enable the model to interpolate and extrapolate in cases where unseen data samples fall in

between or outside of others. Moreover, a classification task requires the number of spot welds (or a range) to be considered a class. It must simplify the classes in the target feature to prevent a skewed distribution of classes. Validation includes both learning tasks.

Limiting the scope to spot welds prevents other joining technologies affecting the prediction performance. It also prevents the model from determining the joining technology. For example, by including curve-based joining technologies such as adhesive bonding, a CR may have one joining location, namely a curve. However, spot welding of this joint could involve 15 joining locations. The popularity of spot welding in the automotive industry ensured that the dataset was large enough for validation.

Approach

Similar to joining technology prediction using SML, predicting the number of spot welds has the same boundary conditions. It enables rapid recommendations, even in early product design, by relying on successful designs of the past. Pelka et al. [376] performed a similar task by predicting the mutual distance of spot welds. However, they mainly considered the geometry of components. This consideration left out information about, for example, the component's function. Furthermore, detailed geometry might be unavailable in early product design phases. Similarly, as for joining technology prediction, the input for predicting the number of spot welds may combine geometric information with further structured data, such as material and function; see joint representation in Section 2.3.6. In addition to the similar dataset processing, the prediction methods can also be similar. Because the goal is to validate whether the prediction is possible, this methodology also implements decision trees, which can perform both classification and regression tasks.

Implementation

As previously mentioned, the implementation was similar to joining technology prediction, so the figure that visualizes the data preparation process is found in the Appendix (Fig. A.2). The number of spot welds used the same raw dataset of features as JTS. However, it only considers the joining scenarios that use spot welds; hence, the values for the number of data samples m and number of features n_f differ. The number of spot welds also had a skewed distribution. There were low frequencies of joining scenarios that have many spot welds; see Fig. A.3. To prevent a negative effect on the performance, the classifier models used an outlier class, which represented joining scenarios with more than 33 spot welds $n_{RSW} > 33$. The class had $m_{33+} = 135$ of the total $m = 3133$ samples.

The training and test set had 2100 and 525 samples, respectively. Each data sample had 131 attributes, of which 21 were numeric. This dataset had fewer features than the dataset used for predicting the joining technology. Not all of the features in data samples contained enough information to pass the variance and correlation filters. To validate the features used in the prediction models, feedforward feature selection was used to determine the most important ones. Fig. A.6 in the Appendix illustrates the feature importance for the XGBoost classifier. Feature engineering resulted in the following selected features for each model:

- Decision tree ($6\times$): surface area of component 1, a moment of inertia of component 2, thickness of component 1, depth in product architecture of component 1, and a coating.

- Random Forest (4×): surface area of both components, a moment of inertia of component 2, and thickness of component 2.
- XGBoost (8×): surface area of both components, a moment of inertia of component 2, thicknesses of both components, depth in product architecture of component 1, a coating, and two manufacturing methods of component 2.

These feature sets included the traditional information that corresponds with engineering considerations. They seemed more geometrically driven than the selected features for joining technology prediction. This also aligns with the work of Pelka et al. [376], who predicted the mutual distance of spot welds based on geometry. The features include moments of inertia and thicknesses, indicating more load and force-based considerations. The feature importance graphs revealed that only a few features have an impact. Hence, the feature sets are smaller than for joining technology prediction. Furthermore, creating all numeric and all nominal feature sets enabled the validation of the high geometrical influence in predicting the number of spot welds.

Process

Similar to the data preparation process, the process for prediction aligns with that of joining technology prediction. Fig. 4.3 presents this generic process, which was further described in Section 4.1.

The process for predicting the number of spot welds used eight prediction models. This methodology also validated a regression task. Additionally, it aimed to provide insights into the correlation between numeric features and the number of spot welds.

1. **Random forest classifier** on *all* features of the dataset;
2. **Random forest regressor** on *all* features of the dataset;
3. **XGBoost classifier** on *all* features of the dataset;
4. **Random forest classifier** using the six most important attributes from feature engineering with *regular decision trees* (FE-DT);
5. **Random forest classifier** using the four most important attributes from feature engineering with *random forests* (FE-RF);
6. **XGBoost classifier** using the eight most important attributes from feature engineering with *XGBoost* (FE-XG);
7. **Random forest classifier** using 21 *numeric* features (NUM);
8. **Random forest classifier** using 110 *nominal* features (NOM).

As stated, predicting the number of spot welds can be a classification or regression task. Comparisons between both types of predictors require the same metrics. Hence, the floating output value of the regressor models needs to be converted to an integer. Then, it may be regarded as labels enabling the evaluation of typical performance measurements for classification. Nevertheless, graphs that visualize differences to the target number enable an analysis of the error size for false predictions.

Results

Table 4.3 presents the results from predicting the number of spot welds using variants of decision trees.

			ACC		$F1$		κ	
ID	**F**	**DT**	**TR**	**TE**	**TR**	**TE**	**TR**	**TE**
1	all	RF	95.4	77.3	95.4	76.9	95.3	75.7
2	all	RF-reg	n/a	16.8	n/a	10.7	n/a	11.2
3	all	XG	99.4	**83.2**	99.4	**80.0**	99.4	**82.0**
4	FE-DT	RF	79.5	**54.3**	79.2	**57.9**	78.9	**51.5**
5	FE-RF	RF	73.0	40.8	72.3	48.3	72.2	37.3
6	FE-XG	XG	75.6	49.8	75.2	39.4	74.8	46.6
7	NUM	RF	89.0	**65.9**	88.8	**66.9**	88.6	**63.5**
8	NOM	RF	90.4	59.8	90.2	61.0	90.1	57.3

Table 4.3: Results of the trained models for predicting the number of spot welds. Abbreviations: F – selected feature set, DT – tree learning method, RF – random fores, XG – XGBoost, TR – training set, CV – cross-validation set, TE – test set, ACC – Accuracy, $F1$ – F1-score, κ – Cohen's kappa, NUM – numeric, and NOM – nominal.

The IDs overlap with the numbers of presented prediction models in the list above. The performances indicates that the predictions were promising yet not convincing. Model *ID3* utilized all 131 features, and an XGBoost classifier had the best F1-score with 80.0%. Feature engineering did not improve the prediction performances. The best feature engineered model was a random forest classifier (*ID4*) with the six most important features determined by regular decision trees, resulting in an F1-score of 57.9%. Interestingly, when using only numerical or nominal features, the F1-scores of random forest classifiers were 66.9% (*ID7*) and 61.0% (*ID8*), respectively. Furthermore, the random forest regressor (*ID2*) was the worst model with an F1-score of a mere 10.7%.

However, Table 4.3 does not offer insights into the false predictions. Predicting a class falsely is binary: one spot weld too few or too many is just as wrong as 20. Theoretically, the worst performing model (*ID2*) could still be performing close to other models. Hence, Fig. 4.5 visualizes the difference in the number of predicted spot welds.

Model *ID3* (XGBoost model on all features) produced the best results in Table 4.3 with a mean difference in the number of spot welds of $\Delta N_{RSW} = -.461$. The negative mean seemed to be a trend for all models, which predicted too few spot welds. The left tails were also thicker than the right tails of every model.

Fig. 4.5 confirms the suboptimal quality of the regression model (*ID2*). Only this model had a mean value that did not coincide with a zero difference. Models with high performance, such as the all-feature XGBoost model (*ID3*), were thin and had a sharp peak at 0. Worseperforming models exhibited opposite behavior; their curves widened outward. Nevertheless, the differences between the predicted and target numbers were small. Fig. 4.5 enabled the analysis of the model's prediction precision. To this extent, Fig. 4.6 maps how often each model predicted each number of spot welds.

Fig. 4.6 indicates the performance of each model for each class. The black line shows the target in relative occurrences. Models with higher performance should be able to follow

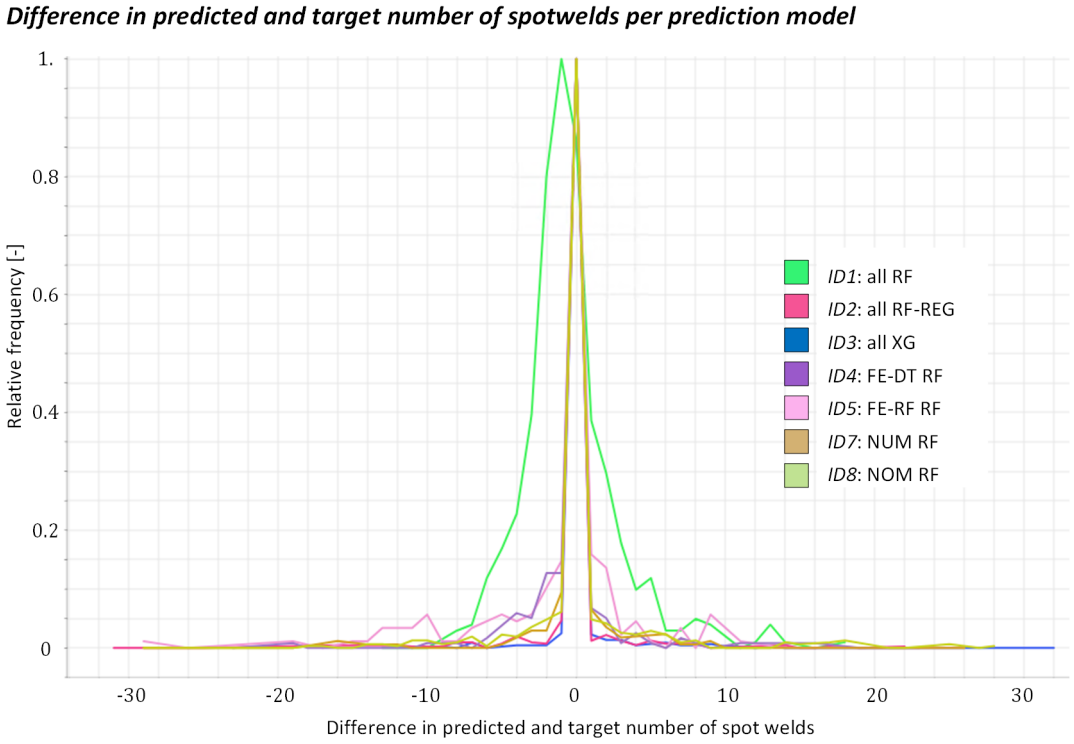


Fig. 4.5: The normalized difference between the number of spot welds in the target vs. predictors.

this line. The figure indicates a skewed class distribution. Most joining scenarios contain a number of spot welds below, for example, 10. Generally, the models seem to predict the joining scenarios with either a low or a high number of spot welds correctly. The mid-range around $[5 - 11]$ exhibits an increased difference between models, as indicated by their distance to the target line.

Discussion

The prediction of the number of spot welds using decision trees produced promising results. For example, the XGBoost model that used all features (*ID3*) achieved an F1-score of 80.0% (Table 4.3). However, the model seemed to need many features to achieve this result. Feature engineering provided undesirable results and attained in the best case an F1-score of 57.9%. One explanation is that it only resulted in a few important features. This restricted number did not allow the model to learn from enough information to perform the task well. On the other hand, it also signals that the number of spot welds depends on many features. Moreover, it may be that the required features were not present.

The experimental work of Pillai et al. [105] aimed to predict the mutual distance of spot welds. Their models also considered some form of represented loads and forces through considering geometry. Decision trees on structured data are advantageous as they train more quickly, are less complex, and require less processing. However, these trees use abstract

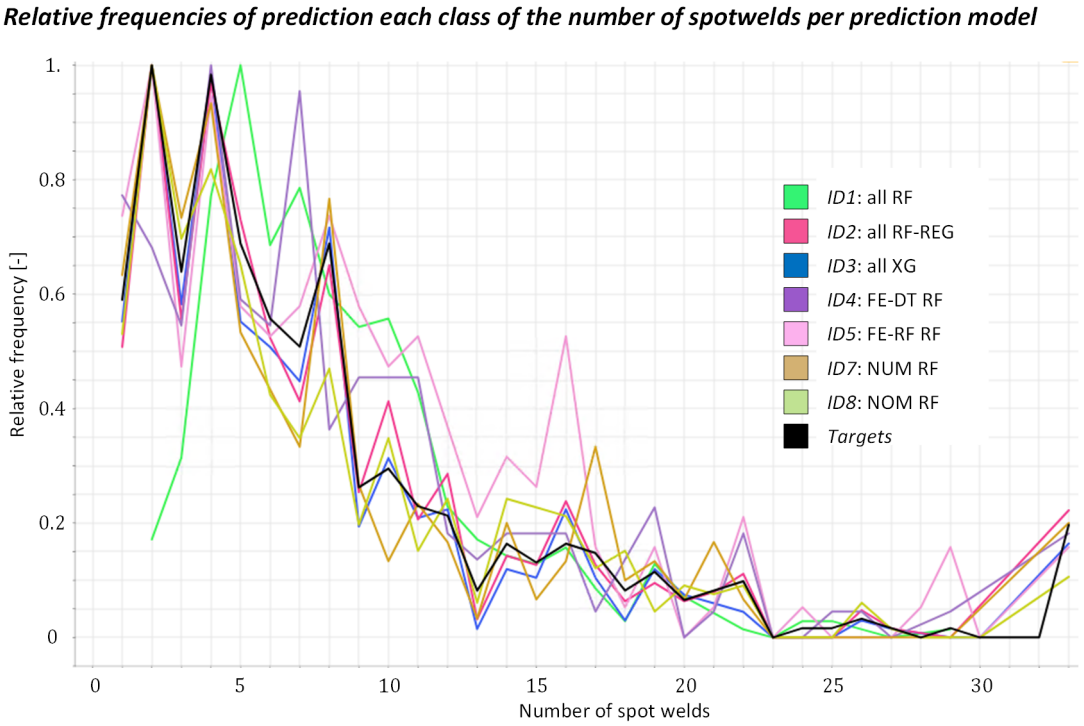


Fig. 4.6: The predictions of the number of spot welds distributed normally over the classes.

geometrical representations, such as volume, weight, and moments of inertia. These features may not contain enough detail about the joining scenario to predict the number of spot welds correctly. This statement is backed up when one examines Fig. A.4 (Section A.2). Here, many features correlate weakly with the number of spot welds, confirming this hypothesis.

For example, taking only numerical (*ID7*) or nominal (*ID8*) features both resulted in higher performance compared with the models that implemented feature engineering (Table 4.3). These models also indicated that the numerical 21 features achieved better results than the 110 nominal features. However, the number of nominal features was large compared with the total 2625 data samples. The prediction models may have extracted coincidences between features in samples to predict the number of spot welds.

These coincidences may also explain the overfitting of models. The difference between scores on the training and test set were large. For most models, it seemed to be above 20%. The dataset might be small for the prediction tasks, especially considering the large number of features.

Moreover, the number of target classes was 33, which is rather high for 2100 unique data samples. The under-representation of large joining scenarios may also have influenced the prediction quality. As Fig. A.3 shows, the number of joining scenarios for each class fell off as the number of spot welds increased. Most joining scenarios contain a number of spot welds below, for example, 10. Hence, models can achieve a high performance for predicting correctly on small joining scenarios.

Moreover, the continuous nature of the task may have affected the models. Figs. 4.5 and 4.6 visualize the distributions of predictions. Many models often predicted one or two spot welds from the target number. The evaluation scores indicated that such predictions were clearly wrong. However, Fig. 4.6 indicated that the predictions were not randomly wrong. Certain areas of the graph indicate that the models follow the black target line rather well. However, the mid-section seems troublesome. Possible explanations for this include fewer data samples as well as the higher geometrical complexity of components. The geometries could increasingly vary in size, thus affecting the numerical features, but without increasingly more spot welds such as plating on the outside of the vehicles. Furthermore, smaller scenarios might have fewer options for spot weld designs and could be more straightforward. Furthermore, large scenarios have lower frequencies that tend to contain less variability in the dataset after using SMOTE.

Evaluation

Nevertheless, predicting the number of spot welds is a promising pathway. The initial results suggested that prediction is feasible with only a few core features. The number of spot welds expresses a form of the expected loads and forces on components. These models enable quick results, supporting designers and potentially RBR methodologies, as in the work of Thompson and Salerno [50].

Table 4.4 summarizes the trained models. It has the same structure and setup as Table 4.2. However, the rating for each model in table 4.4 has one additional consideration:

- Besides performance, the rating considers *structural performance*, which concerns the difference to the target number of spot welds. It explains a model's ability to predict close to the required number.

ID	F	DT	Perf.	Speed	Dev.	Comp.	Struc.	Appl.
1	**all**	**RF**	2	2	2	1	2	2
2	all	RF-reg	1	2	2	1	1	1
3	all	XG	2	2	2	2	2	2
4	FE-DT	RF	1	3	3	2	1	1
5	FE-RF	RF	1	3	3	2	1	1
6	FE-XG	XG	1	3	3	3	1	1
7	NUM	RF	1	2	2	1	2	1
8	NOM	RF	1	2	2	1	1	1

Table 4.4: Evaluation of the trained models for predicting the number of spot welds. Values $[1-3]$ correspond with low to high scores, respectively. Abbreviations: F – selected features set, DT – tree learning method, RF – random forest, XG – XGBoost, Perf. – performance, Dev – development effort, Comp. – computational cost, Struc. – structural performance, and Appl. – practical applicability.

Although, the prediction of the number of spot welds is a partial solution, the methodology is transferable to other joining technologies. However, this implementation required an implementation for each joining technology. The results were not of such importance to enable the assumption that one model for predicting the number of joining

locations would suffice. The methodology seems too specific for discrete joining elements. Hence, a practical application would become highly use-case-driven, such as for predicting the number of spot welds in the automotive industry, whereas the aviation industry might benefit from such an implementation for riveting. In such cases, the number of joining locations for other technologies would require a different method.

Predicting the number of spot welds supports the prediction of joining locations. The following subsection validates an evolutionary algorithm to distribute joining locations on a CR.

4.3 Randomized distribution of joining locations

The number of spot welds is an input for distributing joining locations. Algorithmic fitting first requires a number before it can determine the joining locations. Section 3.2 identified that S&O is an AI field only used in topology optimization methodologies. Currently, RBR (Section 2.1.4) can only equally distribute a number of point-based joining elements on the centerline of a CR. Furthermore, topology optimization (Section 2.1.5) has a high computational cost and requires an entire product to consider all loading cases. S&O can enable more complex distributions to, for example, create better joining locations in terms of structural performance. Additionally, S&O can remain relatively simple and fast. These properties form a middle ground between current state-of-the-art approaches. The aim of validating this methodology was to determine whether a relatively straightforward S&O methodology can create meaningful results, especially as the algorithm lacks knowledge of successful designs.

Approach

Section 3.3.1 presented joining location prediction using randomized optimization. An evolutionary algorithm tries each iteration to increase the mutual and edge distances between joining locations. The section also presented an optimization objective, the main hyperparameters, and an initialization method.

However, the structural performance becomes harder to validate without using FEA. S&O aims to distribute joining locations while considering properties such as mutual and edge distances. However, it does not rely on successful validated designs. In this sense, it is uncertain whether the performance of the resulting joining locations will be good. Therefore, the discussion here addresses whether predicted joining locations are plausible and meaningful. The results of this methodology are not mechanically validated.

Implementation

This joining location distribution methodology uses a 2D dataset with spot welds; see Fig. A.1. It uses the geometries of components of all joining scenarios. Then, it reduces the dataset on joining scenarios that contain spot welds. Furthermore, to reduce the complexity of location distribution, this methodology only considers flat (2D) CRs. CRs that bend into 3D space require compensation for their curvature, making the algorithm unnecessarily complex. Additionally, it is not the aim to optimize for all types of CRs, but merely to determine

whether such an algorithm works for spot welds. Furthermore, the dataset contained a sufficient number of data samples for evaluating the performance of the methodology. Section A.4.1 describes the process of identifying flat CRs, which is illustrated by Fig. A.7.

In short, the data preparation process finds a flat CR and creates a snapshot of it. As a result, the CR becomes a shape on the image. This process is similarly visualized in the state-of-the-art chapter (Section 2.1); see Fig. 2.1). The preparation process creates two images for each flat CR: one with and one without spot welds. It draws spot welds as circles on the shape of the CR.

The distribution algorithm knows the required number of joining elements up front. This knowledge is taken from the target image and supplied to the algorithm. This step enables the specific evaluation of the distribution of joining locations.

The images have a size of 256×256 pixels with a $r_n = 2\ mm/pixel$ resolution. They are stored as RGB bitmaps. Bitmaps store information pixel-wise and enable simple pre- and postprocessing. For every pixel, it is clear to what class they belong. As a result, bitmaps enable filtering on each channel, similar to one-hot encoding. Compression methods such as JPG might create an unnecessary interpolation of values. The data representation has the following color coding:

- **Empty** – white – $(255, 255, 255)$.
- **Components** – blue – $(0, 0, 255)$.
- **CR** – green – $(0, 255, 0)$.
- **Spot weld** – red – $(255, 0, 0)$.

The algorithm needs a starting point, for which it takes the number of spot welds of the target. Next, the algorithm initializes joining locations randomly on the CR. It creates points by generating a random x and y value within the bounding box of the CR's shape. Then, the algorithm tests each point for whether it lies within the polygon shape. It stores only those within the CR of the initial set of joining locations L. This process repeats until it reaches the required number of joining elements n_{je}.

After initialization, the algorithm evaluates each iteration according to an improvement in the objective function (Eq. 3.1). This function involves weighing to balance the influence of the mutual and edge distances. The results of the algorithm are highly impacted by these weights. Empirical research found that for a nominal weight for edges $w_e^s = 1.0$, smaller weights for nearest neighbors were required $w_{nn}^s = 0.5$ or $w_{nn}^s = 0.9$.

Process

After setting the data representation, the hyperparameters follow. the algorithm runs for an empirically determined 6000 iterations. The learning rate starts at 20 to make rapid progress in the initial optimization phase. Every 1000 iterations, the learning rate halves. Hence, after 4000 iterations, the learning rate is 1, which is the smallest step size when representing a pixel on an image.

Section 3.3.1 presented a process for the randomized optimization of joining locations (Fig. 3.21). Table 4.5 presents the implemented algorithm based on Fig. 3.21.

Input	Polygon representing a contact region P Number of joining elements n_{je} Maximum number of iterations Learning rate Learning rate decay rate
Output	Set of joining locations L
1:	While the length of the set of joining locations $\|L\| < n_{je}$
2:	Initialize point p uniform randomly within bounding box of polygon P
3:	If point p within border of polygon P
4:	Add point p to the set of joining locations L
5:	Calculate value of objective function $\mathcal{L}_d^{best}$
6:	For i in number of iterations
7:	Copy the set of joining locations L in a temporary list L^{tmp}
8:	For p^{tmp} in the set of joining locations L^{tmp}
9:	Translate p^{tmp} with $[-1, 0, \text{or}, 1]$ times the learning rate
10:	Check if p^{tmp} remains within polygon P
11:	Calculate value of objective function with new joining locations $\mathcal{L}_d^{tmp}$
12:	If new joining locations are better than the old distribution $\mathcal{L}_d^{tmp} > \mathcal{L}_d^{best}$
13:	Set new joining locations as best $L := L^{tmp}$
14:	Reduce learning rate on half-life time
15:	For p in the set of joining locations L
16:	Check if p is conform the manufacturing requirements
17:	Return the resulting set of joining locations L

Table 4.5: Randomized location optimization algorithm for point-based joining elements.

The validation considered multiple prediction models to analyze the influence of weighing and optimization. Varying weights provided insight into the generic behavior of the models as well as the basis for future work. The results of RI enabled a benchmark comparison and validation of the optimization steps. To analyze optimization fairly, the result of RI was seeded. For each data sample, the state after RI was equal for all three models.

- **RI benchmarking** (*RIB*) is a benchmark model. The output of this model equaled the random state after initialization. This model aided in assessing the quality of optimization.
- **RI + nearest neighbor weight** $w_{nn}^s = 0.5$ (*AF1*) is also a model that optimizes the locations after RI. This model used the edge weight $w_e^s = 1.0$ and a mutual distance weight of $w_{nn}^s = 0.5$.
- **RI + nearest neighbor weight** $w_{nn}^s = 0.9$ (*AF2*) is a model that optimizes the locations after RI. It used the same edge weight but a higher mutual distance weight of $w_{nn}^s = 0.9$.

The output of the prediction models required postprocessing to acquire the performance measurements. Section 3.3.4 describes a method for reconstructing coordinates from grid-like

data structures (subsection 3.3.4.1), such as images. Furthermore, the section describes a method for coordinate-based performance measurements (subsection 3.3.4.2). Both of these methods need to set parameters.

- **Coordinate reconstruction**
  A typical spot weld creates a group of approximately 16 pixels in the target. Hence, to filter noise out of the predicted image, a minimal group size of $\tau_g = 4$ was required. The noise filter removed groups with fewer adjacent joining location pixels.
- **Performance measurements**
  The relevant measurements (from subsection 3.3.4.2) included accurateness, similarity, and correctness.
  - **Accurateness** (*A*; see Eq. 3.20)
    The *accurate* distance $s_{valid} = 40\ mm$ prevented the consideration of outliers in performance measurement. It is twice the minimal welding distance, implying that imprecise spot welds were still regarded in the evaluation.
  - **Similarity** (*S*; see Eq. 3.21)
    The Gaussian variable $\sigma = 1.6$ for the similarity evaluation. Eq. 3.21 has a value of 1 for perfect similarity. It approaches zero for a Euclidean distance of approximately $40\ mm$ with a resolution of $2\ mm/pixel$. Fig. 4.7 plots the similarity function with $\sigma = 1.6$.

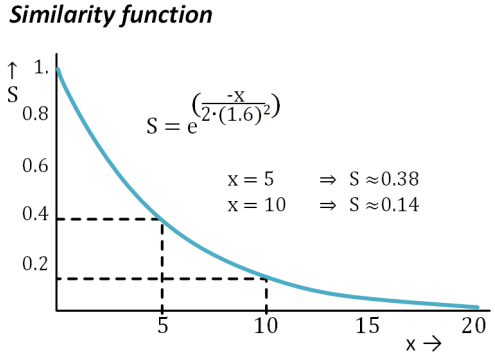


Fig. 4.7: Visualization of the similarity function for $S = e^{\frac{-x}{2\times\sigma^2}}$, where $\sigma = 1.6$ and $x = \frac{s_{eucl}}{r_n}$.

  - **Correctness** (*Q*)
    A correct prediction implied that all predictions were within the threshold $s_{correct} = 3\ mm$ of the targets without any leftovers.

Results

Table 4.6 presents the results for three models: RI benchmarking (*RIB*), a low nearest neighbor weight (*AF1*), and a high nearest neighbor weight (*AF2*).

The *RIB* model predicted 79% of the locations within $40\ mm$ (20 $pixels$) of their targets. Evolutionary optimization improved joining locations compared with the benchmark *RIB*

ID	w_e^s	w_{nn}^s	A	S	Q	ΔN
RIB	–	–	0.79	0.11	0.01	–0.10
AF1	1.0	0.5	**0.87**	**0.25**	**0.17**	–0.03
AF2	1.0	0.9	**0.87**	0.23	0.16	**0.01**

Table 4.6: Randomized location distribution results. Abbreviations: RIB – randomized initialization, AF – algorithmic fitting, A – accurateness, S – similarity, Q – correctness, and ΔN – difference between the number of predicted and target spot welds.

model as performance measurements improved, but only by a small amount. Models *AF1* and *AF2* differed only by 2% and 1% for similarity (S) and correctness (Q). The correctness of *AF1* stagnated at 17%.

Noteworthily, although the models all used the number of spot welds from the target, a difference existed between the number of predicted and target coordinates. This difference originated from overlapping joining locations that were recognized as one pixel group (see subsection 3.3.4.1). The model *AF2* revealed, due to its increased nearest neighbor weight $w_{nn}^s = 0.9$, that it was closer to zero difference $\Delta N_{je}^{\text{AF2}} = 0.01$.

Discussion

The discussion starts with several cherry-picked predictions. These help to offer a deeper insight into the performance of the algorithm on top of the generic results in Table 4.6.

A deeper analysis of specific data samples revealed the behavior, advantages, and limitations of using the randomized evolutionary algorithm. The optimization models (*AF1* and *AF2*) had almost identical results in terms of performance measurements; see Table 4.6. Figs. 4.8 and 4.9 present the input and output samples of the optimization process with two separate weight settings for the nearest neighbor: $w_{nn}^s = 0.5$ and $w_{nn}^s = 0.9$.

Both figures also present the performance metrics of each sample. The input images are the result of the RI process. The *RIB* model had two objective values due to the differences in weighing terms, making the results comparable for the two prediction models. The characters positioned on the left side of the image correspond with the characters in the list below that discusses each sample. Additionally, the sample identifiers overlap between the figures and the list.

a. Sample $x^{(6)}$ optimized five spot weld points. The results demonstrated that the outer points moved toward the outer edges. Model *AF1* placed all points on the centerline of the CR. It was observed that initialization grouped three points in the bottom part of the CR as the number of iterations was not quite high enough to separate them all. Model *AF2* pressured the outer points to move into the corners of the CR.

b. Sample $x^{(5056)}$ contained six points to distribute over the polygon. Both models collected the points at the bottom of the CR. The points for both models traversed the smaller area to get there. *AF1* placed the points in the bottom of the skeleton of the polygon. *AF2* created a similar result but placed the locations slightly further apart. The models could not construct the targeted straight line of locations.

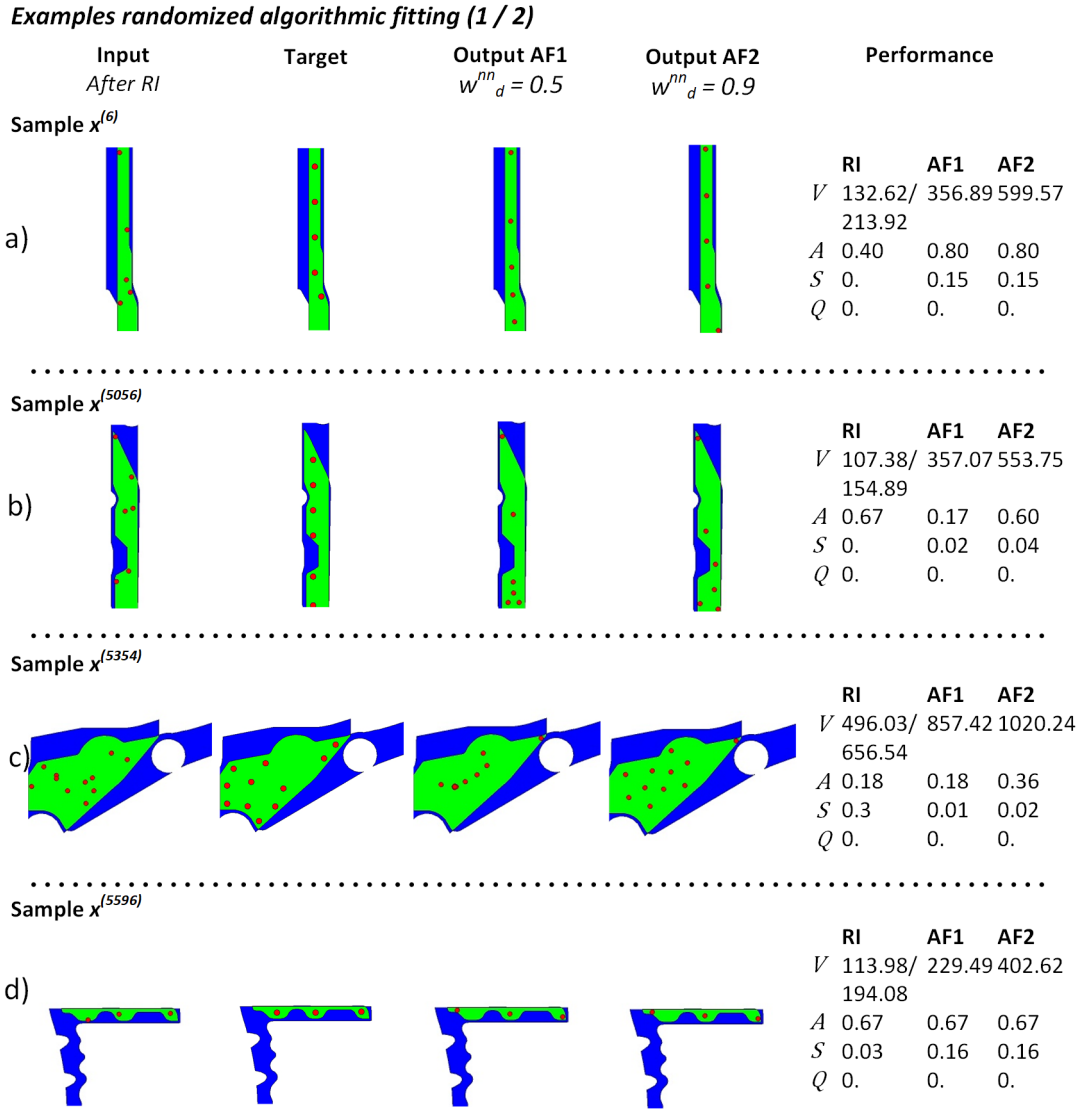


Fig. 4.8: Cherry-picked exemplary results for the discussion of data samples $[6, 5056, 5354, 5596]$ for images $[a - d]$, respectively. Contact regions are in green and blue areas represent the geometry of components.

c. Sample $x^{(5354)}$ had 11 spot welds that needed to be distributed over a large surface with no clear centerline. Model *AF1* tended to optimize the edge distance such that the points were equidistant from it. It seemed that the points moved toward the skeleton of the shape. The mutual distance term in the objective function lacked weighting to move them apart. This resulted in overlapping spot welds in the center of the polygon, thus reducing the number of predicted spot welds. *AF2* forced the locations to remain further apart and created a group that seemed arbitrarily positioned relative to the CR.

   This sample indicated that the prediction of *AF2* had a higher structural performance.

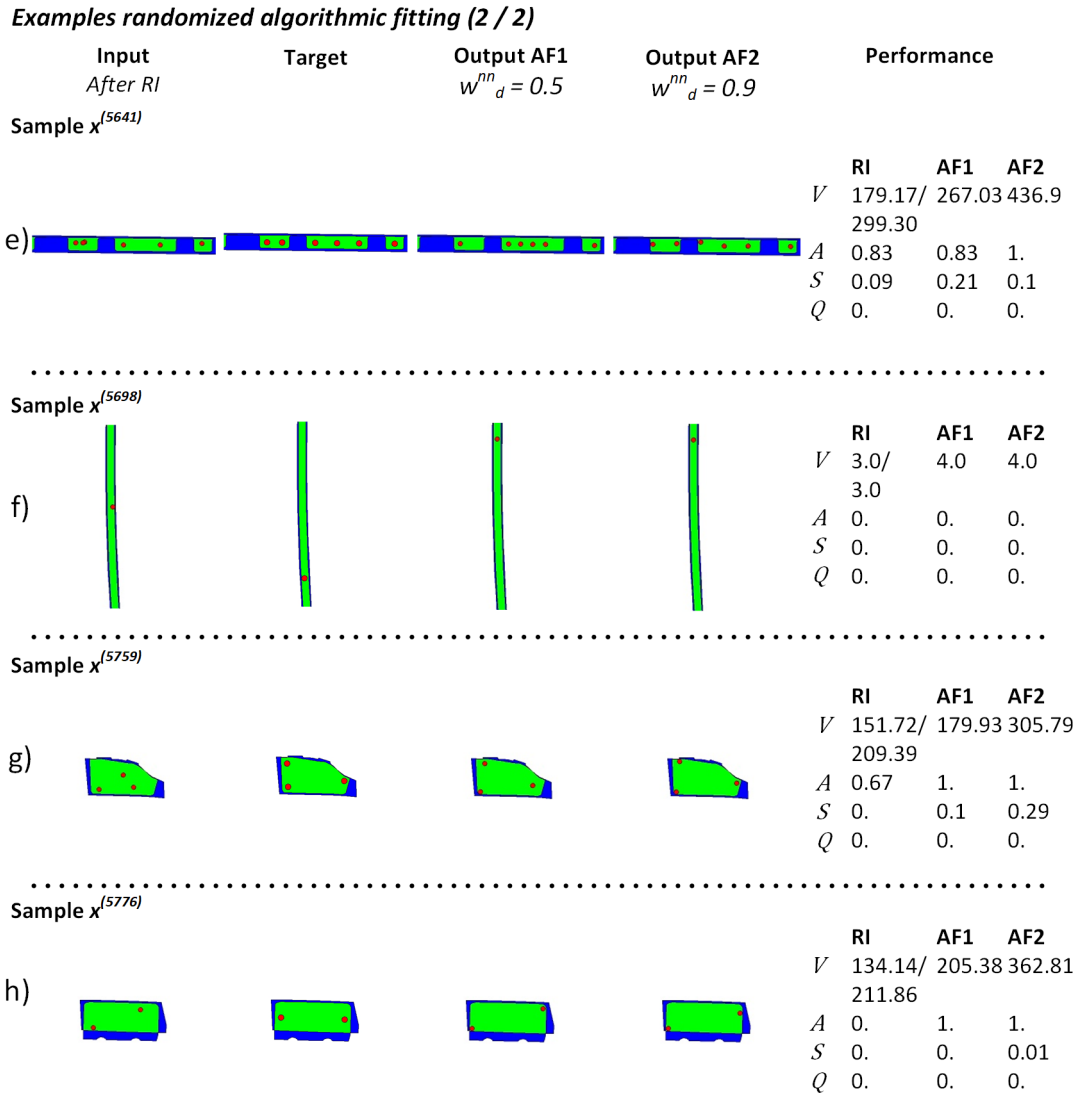


Fig. 4.9: Cherry-picked exemplary results for the discussion of data samples $[5641, 5698, 5759, 5776]$ for images $[e - h]$, respectively. Contact regions are in green and blue areas represent the geometry of components.

The locations were further from the center and better spread over the CR. This spreading is better for stresses, such as those from bending, torsion, or buckling.

d. Sample $x^{(5596)}$ had one CR with three larger areas for three joining elements. RIB predicted a location in each larger area. The algorithm could have difficulty moving the points out of each area for small learning rates. A large initial learning rate (step size) helps to jump points from one area to another. The results for both models indicated that outer points were located in corners of their respective areas. The algorithm did not consider the CR shape to put the locations in the center of the enlarged areas.

e. Sample $x^{(5641)}$ had six joining locations on three CRs with different area sizes. RIB placed three locations in the left smaller area. The algorithm ended up with more joining locations in the middle. Model *AF2* placed each of the right number of joining locations in each CR due to the support for creating equidistant locations, as the positioning of the CRs allows it to do. *AF1* placed a group of four locations in the center of the two outer locations.

f. Sample $x^{(5698)}$ only had one joining location. The optimizer slightly moved it to the centerline of the CR. No other location was available for distribution. Furthermore, the objective function only considered the nearest edge, preventing the joining element traversing to the center of the CR. Nearly all algorithm iterations were irrelevant, and a stopping criterion was implemented if the optimization stagnated for too long. This implied that once the value of the objective function did not increase for 2500 iterations, the algorithm stopped. Due to the same arbitrary step, the distance to the edge improved minimally. Hence, both models ended up with the same joining location.

g. Sample $x^{(5759)}$ had three joining locations, and both models seemed to find the target points rather well. This distribution had the largest distance between the locations. Coincidentally, it overlapped with the target distribution, resulting in an accurateness of 1.

h. Sample $x^{(5776)}$ resulted in two locations at the edges of the CR, where they mutually had the greatest distance to one another. Therefore, both models produced the same results. The current objective function cannot predict spot welds on a centerline in a rectangular CR, which suggests the need to consider the polygon's shape in the objective function.

Several samples exhibited unbalanced results. For example, the outlier points in samples $x^{(5056)}$ and $x^{(5354)}$ raise the question of whether an additional term that considers the distribution of locations may increase the performance. This section briefly explores the following two terms to add to the objective function of Eq. 3.1: the standard deviation in the mutual distances between locations, and the variance in the area of Voronoi regions.

- **Deviations in mutual point distance** refer to whether the nearest neighbors of points s_{nni} are close to the average distance between nearest neighbors μ_{nn}. The aim is to promote the algorithm to set equidistant points. The objective function requires the inverse to minimize the difference. To compensate for the small number, a large weight is required for the term w^{σ}_{nn}. Thus, the objective function of Eq. 3.1 becomes Eq. 4.1:

$$\mathcal{L}_d = \sum_{i=0}^{n_{je}} \left(w^s_e s_{ei} + w^s_{nn} s_{nni} + \frac{w^{\sigma}_{nn}}{\sqrt{\frac{1}{n_{je}} \sum_{i=1}^{n_{je}} (s_{nni} - \mu_{nn})^2}} \right), \text{ where } \mu_{nn} = \frac{\sum_{i=1}^{n_{je}} s_{nni}}{n_{je}} \tag{4.1}$$

- **Voronoi** regions refer to the spaces around points equidistant to other points [408]. All points within each region are closest to the point that defines it. The areas of Voronoi regions express a measure of distribution. Unequal sizes indicate that there are small groups around the object.

The term measures the difference of the area of a Voronoi region A_i^V to the average area of regions μ_{AV}. Minimizing the difference should increase the distribution of Voronoi regions and thus their points. As with mutual point distance, a large weight is required to compensate for the small value after inversion. Thus, the objective function of Eq. 3.1 becomes Eq. 4.2:

$$\mathcal{L}_d = \sum_{i=0}^{n_{je}} \left(w_e^s s_{ei} + w_{nn}^s s_{nni} + \frac{w_{AV}^\sigma}{\sqrt{\frac{1}{n_{je}} \sum_{i=1}^{n_{je}} (A_i^V - \mu_{AV})^2}} \right), \text{ where } \mu_{AV} = \frac{\sum_{i=1}^{n_{je}} A_i^V}{n_{je}} \tag{4.2}$$

The weighing terms w_{nn}^σ and w_{AV}^σ were determined empirically at $w_d^{stddev} = 10^4$. Furthermore, the models used the other weights of model *AF1*. Fig. 4.10 illustrates the effect of these models on two data samples $x^{(6)}$ and $x^{(5354)}$ (also depicted in Fig. 4.8).

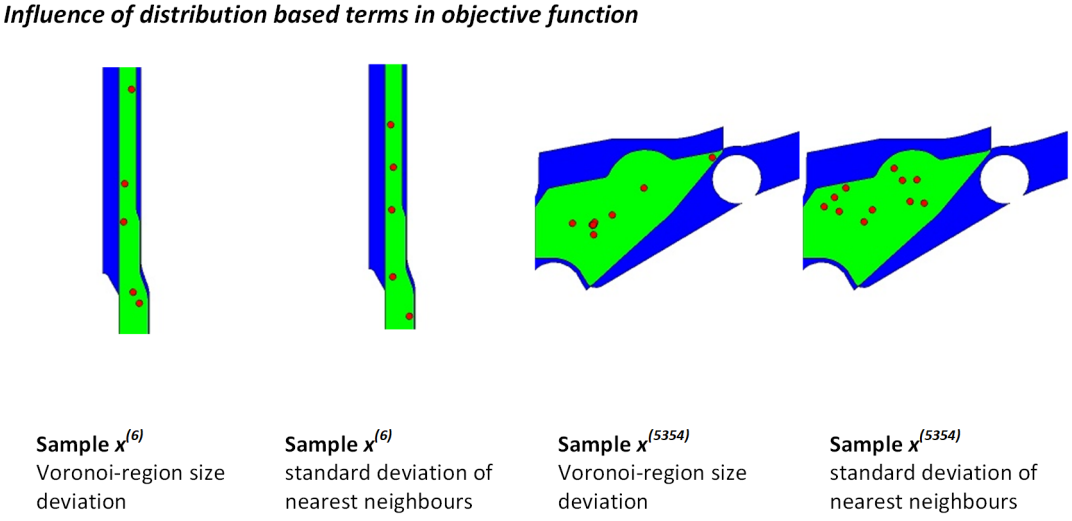


Fig. 4.10: Examples of adding the standard deviation to the objective function based on Voronoi-region sizes or nearest neighbor distances.

Notably, the deviation-based terms did not improve the results. Moreover, joining locations moved away from meaningful locations in sample $x^{(6)}$. The results of *AF1* and *AF2* seemed cleaner. Additionally, the deviation-based terms did not aid in producing more equidistant results. The Voronoi results seemed a very small improvement compared to the results after RI.

Similarly, Sample $x^{(5354)}$ exhibited no discernible pattern. Although the Voronoi-based prediction created locations at the skeleton of the CR's shape, this was similar behavior to *AF1*. The nearest neighbor-based prediction made two groups of locations without any better distribution over the entire CR.

Furthermore, the deviation-based terms did not seem to aid the distribution process. They might even clutter the simplicity of merely considering edge and mutual distances. Gerlach

[407] experimented with a hybrid optimization of distances and Voronoi regions, but without promising results. Lastly, an additional disadvantage of these extra terms was that they increased the algorithm's complexity, thereby slowing its execution time and increasing the difficulty of converging.

After analyzing additional terms for the objective function that aimed to improve the distribution of joining locations, the performance of models *RIB*, *AF1*, and *AF2* need to be addressed. First, the results in Table 4.6 also exhibited a difference in predicted and target coordinate measurement ΔN_{je}. Although the algorithm took the target number of spot welds, a difference existed in the results. The problem originated from data formatting. As model *AF1* in Fig. 4.8 indicates, poor optimization left multiple spot welds practically on top of one another. The locations derived from the red spots on target images. However, coordinate reconstruction (subsection 3.3.4.1) was unable to differentiate between these spots and handled them as one. Hence, by examining this metric, model *AF2* was more effective at distributing the locations due to the higher weight for this term. This weight resulted in a ΔN_{je} closer to zero.

Regarding another performance metric in Table 4.6, namely the accurateness (A), both models that optimized predict joining locations closer to the targets. The increase in similarity (S) confirmed this. However, the accurateness after RI was 79%. This high base accurateness originated from the data structure. As the data samples were $256\ pixels \times 256\ pixels$ with a resolution of $r_n = 2\ mm/pixel$ and area of $512\ mm \times 512\ mm$, the CRs were only a small part of this. As RI may only create joining locations within this shape, the probability was high that spot welds would end up within the threshold $s_{valid} = 40\ mm$ of a target location.

Still, 13% of the spot welds were not located within this $40\ mm$ threshold of the targets after optimization. The analysis of individual data samples revealed that the algorithm had difficulties with large CRs. These CRs had no straightforward equidistant pattern, nor could the algorithm create plausible patterns (e.g., see samples b. $x^{(5061)}$ and c. $x^{(5354)}$ in Fig. 4.8).

However, accurateness only measures a form of meaningfulness of the prediction. Similarity expresses how closely the predicted joining locations aligned with the target locations. The similarity was zero if the distance for each location exceeded $40\ mm$, where it was 0.5 for a distance of approximately $20\ mm$. The similarity after optimization ended for model (*AF1*) at 0.25. On average, most spot welds were one minimum spot welding distance ($\approx 20\ mm$) from their target.

However, a low similarity would not necessarily mean that the predictions are not meaningful. As Section 2.1 addressed, setting joining locations at the edges of CRs increases the ability to absorb stresses, such as bending, torsion, or buckling. Generally, the spot weld locations shown in Figs. 4.8 and 4.9 move away from the center, which was naturally caused by the nearest neighbor term in the object function. However, the results might still be useful in early product design for quickly creating joining locations. This hypothesis requires this methodology to be validated with respect to structural performance.

Regarding the low similarity score, the correctness (Q) for model (*AF1*) was 17% (Table 4.6). This score implied that the models could barely reproduce joining locations similar to the target. One explanation is that successful joining locations do not comply simply with the equidistant behavior of edges and nearest neighbors. This underscores that the design of joining

locations is not a trivial task and justifies the amount of scientific interest that it receives (see Section 2.1.5).

In addition, the weights affecting the edge and nearest neighbor distances exhibited very different results. Lowering the nearest neighbor weight w_{nn}^{s} to 0.5 revealed locations ending up on the skeleton of the CR; see samples b. $x^{(5061)}$ and c. $x^{(5354)}$ in Fig. 4.8. The objective function rewarded points that were equidistant to the polygon's border. Consequently, these results regularly violated manufacturability requirements for minimum mutual distances.

The evolutionary algorithm is robust when distributing locations along CRs originating from flanges (e.g., sample a. $x^{(6)}$ in Fig. 4.8). These long and small polygons require joining elements to be ordered in a line. Samples with large CRs will receive less meaningful results. The algorithm either aims to put joining locations on the skeleton of a CR or arbitrarily creates groups of equidistant locations with a few outliers.

Moreover, the nearest neighbor weight $w_{nn}^{s} = 0.9$ emphasized distances between points. The results revealed no overlapping and joining locations that were often equidistant. An interesting behavior was that groups of points started to exist together with outliers in far corners of the CR; see sample b. $x^{(5061)}$ and c. $x^{(5354)}$ in Fig. 4.8. There was one spot weld in the top corner, whereas the rest resided at the bottom.

Besides the weights of terms, the number of iterations is another delicate parameter. It affects the optimization time significantly, especially when the number of joining elements is large. The learning rate optimizes the execution time and aids the algorithm in traversing complex CRs. Polygons may have smaller connecting areas over which locations need to traverse. Their traverse is challenging when the learning rate is low, as demonstrated by Gerlach [407], who did not use a learning rate. Consequently, the algorithm might have required millions of iterations, which would have affected the optimization time greatly.

A small learning rate also affects the risk of falling into local optima. The algorithm might not overcome these optima, resulting in suboptimal outcomes. Large learning rates enable the jumping of locations between CRs; see sample f. $x^{(5641)}$ in Fig. 4.9. A high learning rate enables the redistribution of joining locations that can be stuck in a separated CR. After the initial distribution with large rates, learning rate decay slowly reduces step sizes to approximate a global optimum of the objective function. Besides the learning rate, another option is to run the algorithm several times. Locations get reinitialized and randomly redistributed, and then the optimal result can be the final prediction.

Regardless of whether weights or hyperparameters are being tuned, the algorithm has some structural weaknesses. The distance optimization sometimes forced joining locations to the outer borders of the CR. This effect was significant on CRs requiring two joining locations. Here, the mutual point distance shared between two points naturally weighed heavier compared with the edge distance optimization; see Fig. 4.11. As a result, the mutual distance term in the objective function effectively pushed the two joining elements apart. Such results would not comply with manufacturing requirements. A preventive erosion (inwards offset) of the CR polygon by the minimum edge distance would prevent the joining elements violating the edge distance requirement; see Fig. 4.11.

However, such a solution does not solve the behavior of the algorithm. For this, the second nearest neighbor and second nearest edge as additional terms might enable the algorithm to

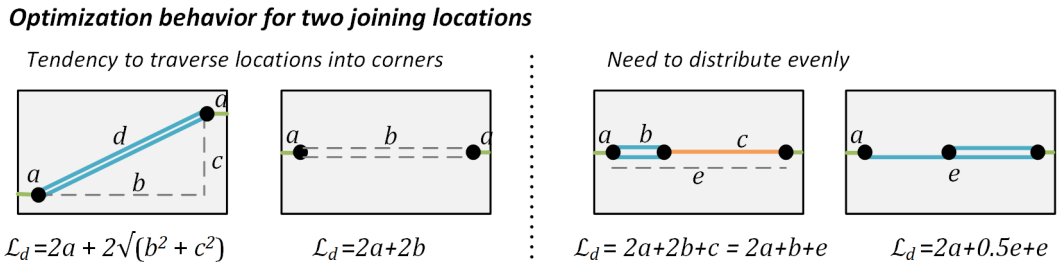


Fig. 4.11: Visualization of optimization behavior for contact regions with two spot welds and the solution for complying with the manufacturing requirements as edge distance.

better consider engineering requirements. Furthermore, one can decide to prevent measuring the same edge between two nearest neighbors. Lastly, one could add a rule set to the algorithm that considers manufacturing requirements as minimum and maximum distances for edges and nearest neighbors.

Evaluation

The randomized distribution of spot welds is a straightforward approach for creating joining locations, although the performance measurements were not convincing regarding the potential of this methodology. However, most predictions seemed to be meaningful. Theoretically, it could work in early product development phases as an initial design. However, this would require these predictions to be validated using, for example, FEA and simulations.

The best predictions come from small and long CRs that require a spot weld location in a straight line. However, these are simple use cases for RBR methodologies, such as the work of Thompson and Salerno [50]. In this regard, the randomized distribution algorithm would not necessarily improve the benchmark performance. Table 4.7 illustrates this.

ID	Perf.	Speed	Dev.	Comp.	Struc.	Appl.
RIB	1	3	1	1	1	1
AF1	2	2	2	2	2	1
AF2	2	2	2	2	2	1

Table 4.7: Evaluation of the models for predicting the joining locations of spot welds using randomized distribution. Values $[1-3]$ correspond with low to high scores, respectively. Abbreviations: RIB – randomized initialization, AF – algorithmic fitting, Perf. – performance, Dev – development effort, Comp. – computational cost, Struc. – structural performance, and Appl. – practical applicability.

The industrial applicability in the current form is low. The algorithm does not improve on a state-of-the-art rule-based approach. Various considerations led to the ratings in Table 4.7:

- The *performance* is unsatisfying after a comparison with the results of *RIB* (Table 4.6). The performance requires significant improvements for this methodology to become *applicable in industry*.

- The algorithm's *speed*, which is tied to the *computational cost*, is satisfying but only after implementing a learning rate decay.
- The *developmental effort* of the algorithm itself is not high. However, data preprocessing and further experimentation for improving it might increase this effort significantly.
- The algorithm seems to output meaningful spot weld locations. However, *structural performance* still requires validation.

The randomized distribution of joining locations is a partial solution for algorithmic fitting. Together with predicting the number of spot welds, this methodology can create joining locations. However, results for both methodologies are not convincing in terms of algorithmic fitting's potential. Connecting these methodologies in a series also chains their errors, lowering the overall performance.

Consequently, the following section validates a methodology that implements SML to predict joining locations.

4.4 Exploration of supervised machine learning tasks for voxel-based joining location prediction

Another AI field, besides S&O, that VICTOR identifies for predicting joining locations is SML (Section 3.2). Furthermore, VICTOR proposes using SML to perform grid-based drawing (Section 3.2.2). It involves prediction models coloring the cells in grids, such as images, and *drawing* joining locations on data samples.

ML can exploit patterns in data to predict unseen data samples. Thus, it can apply knowledge of successful joining location designs to new joining scenarios. This property can be vital for companies with large products and a high product variety, such as those in the automobile and aviation industries. Current state-of-the-art RBR and optimization approaches lack these considerations (Sections 2.1.4 and 2.1.5), and consequently, their results are prone to rework.

The aim of validating this methodology was to determine whether these patterns in joining locations are exploitable through SML. However, SML consists of two learning tasks, namely classification and regression. This section aims to determine which learning task better suits the use case.

Approach

The framework presents an approach for using EncDec architectures to predict joining locations (Section 3.3.2). These architectures enable a consistent mapping between input and output. This constant data structure helps to reconstruct Cartesian coordinates of the joining locations systematically. Additionally, these architectures enable state-of-the-art computer vision methods to be implemented, for tasks such as object detection and image segmentation. These methods provide validated ideas on, for example, networks and data formatting.

Hence, the *classification* concept corresponds to an image segmentation task; see subsection 3.3.2.1. This task aims to assign a class label to each grid cell (e.g., pixel or voxel).

It helps to locate objects in the input grids. Furthermore, the *regression* concept corresponds with probability mapping (see subsection 3.3.2.2). To each grid cell, it assigns the probability of it containing the target label.

Previously, predicting the number of spot welds (Section 4.2) already demonstrated their high reliance on geometry. Hence, to predict joining locations and prevent models from becoming unnecessarily complex, this methodology only considers geometry. Moreover, component geometry is a requirement for predicting joining locations. It purposefully neglects nongeometric data, such as materials, coatings, and functions aside. However, Section 4.5 considers these data.

The SML approach uses Euclidean structures to model data. More specifically, this methodology represents geometry using voxels. Compared with pixels in images, voxels have three dimensions, which quickly induces challenges with computational cost. Nevertheless, voxels enable the representation of 3D curved CRs. Additionally, they can consider more of the surrounding component geometry.

Eggink et al. [3] present the initial comparison between the regression and classification approach. This work aimed to explore both approaches and determine whether SML with encoder-decoder architectures is a viable solution to predict joining location.

Implementation

Many considerations exist when implementing SML to predict joining locations. As Section 3.3.2 already suggested, the key is to determine appropriate resolutions, grid sizes, and data formatting for the classification and regression concepts (subsection 3.3.2.3).

Fig. A.1 presents the dataset being derived for 3D spot weld location prediction. It takes only the geometry of each joining scenario that contains spot welds. Furthermore, the data preparation process is complex, leaving fewer data samples ($m = 3138$) than, for example, for 2D location prediction using the randomized distribution methodology ($m = 4634$).

The data preparation process is explained in Appendix A.4.2. It creates voxel grids (i.e., CCs) containing CRs and joining elements from joining scenarios consisting of two CAD geometries. However, the resolution and sizes for the CCs are not trivial. The grid dimensions x, y, and z follow from a preliminary analysis of the geometry of joining scenarios. Table 4.8 lists the mean and standard deviation for each axis of a *region of interest*.

Measurement	**x** $[mm]$	**y** $[mm]$	**z** $[mm]$
Mean	225.95	164.38	152.25
Std. dev	232.35	191.79	122.93

Table 4.8: Mean and standard deviation in $[mm]$ in the x-, y-, and z-directions of the intersection bounding box from the bounding boxes of both components.

The region of interest describes the intersection between bounding boxes of each component; see Section A.4.2. The CRs and joining elements reside within the intersection of bounding boxes. CCs neglect the other geometry in favor of the computational cost.

Large CCs contain joining elements to predict. However, after considering the cubic scaling of data dimensionality and the results in Table 4.8, the grid sizes were set to W, H, and D at

cubic grids of 200 $mm \times 200\ mm \times 200\ mm$. Furthermore, cubic grids simplify convolutional operations, the selection of strides and filter sizes, and data augmentation.

Additionally, the resolution for voxels r_n highly influences the level of detail, grid size, and computational cost. Fig. 4.12 illustrates that smaller voxels can represent geometry more accurately but more voxels are required to do so.

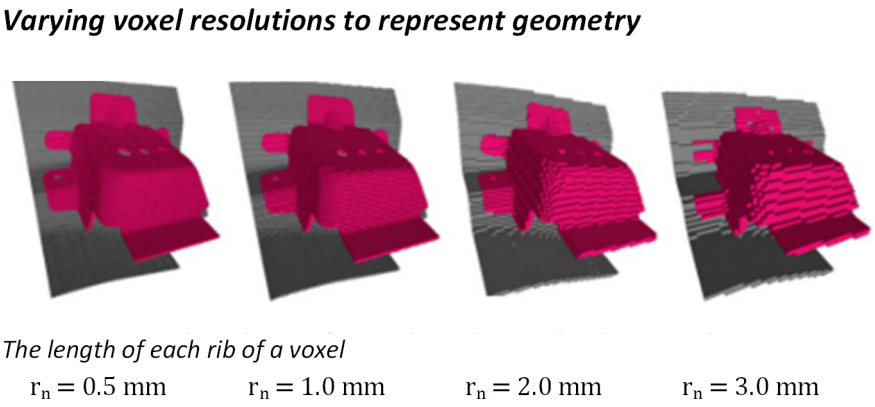


Fig. 4.12: Examples of different voxel resolutions; taken from Perez-Ramirez [412].

The use case in the automobile industry involves metal sheets that can have thicknesses of $0.5\ mm$. Proper sampling (e.g., considering the Nyquist frequency [444]) of geometry requires that $r_n = 0.25\ mm$, resulting in a $512 \cdot 10^6$ dimensional input grid. Doubling the resolution $voxels/mm$ scales grids by a factor of $8\ (= 2^3)$. Hence, this study opted not to model the sheet thicknesses due to calculation costs.

Empirical research found that pitches above $r_n = 2\ mm/voxel$ contained little information for training models. Hence, the resolution $r_n = 2\ mm/voxel$ and CC dimensions $200\ mm \times 200\ mm \times\ 200mm$ provided the input grid of size $W \times H \times D = 100 \times 100 \times 100$, resulting in 10^6 input dimensions. Consequently, each voxel represented a $2\ mm \times 2\ mm \times 2\ mm$ cube. The following list contains further implementation settings, most of which were discussed earlier in Section 3.3.2.

- The method involves the **modeling** of spot welds as cylinders with a $2\ mm$ radius according to their average nugget diameter [115]. Their length is derived from the sheet thicknesses of both components. The data representation of spot welds differs for the classification and regression tasks.
- The resolution and grid sizes define the input, target, and prediction **shapes** of the data samples. Referring to Eq. 3.3, the segmentation approach of input (i) has shape $x^{(i)} \in R^{100\times100\times100}$, where each grid cell has a value of $x^{(i)}_{w,h,d} \in \{0, 1, 2\}$.
- The **classification** approach has output y and prediction $\hat{y}$ shapes of $y^{(i)} = \hat{y}^{(i)} \in R^{100\times100\times100\times4}$. The fourth dimension is one-hot encoded over each class $y^{(i)}_{w,h,d,c} = \hat{y}^{(i)}_{w,h,d,c} \in \{0, 1\}$.

- The **regression** approach has the same shapes for input, target, and prediction grids $x^{(i)}, y^{(i)}, \hat{y}^{(i)} \in R^{100\times100\times100}$. Each cell has the same range of values in the interval between zero and one: $x^{(i)}_{w,h,d}, y^{(i)}_{w,h,d}, \hat{y}^{(i)}_{w,h,d} = [0, 1]$. The probabilities for the spatial regions are bounded by $\tau_1 = 0.008$ and $\tau_2 = 0.7$. The target values and probability ranges for the spatial matrix $\tilde{T}$ are listed in Table 4.9, where CR probability equals τ_2, $P_2^{\text{PROB}} = 0.008$.

Class (c)	**Target value** ($y_{w,h,d}$)	**Probability range matrix** ($\tilde{T}_c$)
0	0.	$y_{w,h,d} = 0$
1	0.0001	$0 < y_{w,h,d} < 0.008$
2	Eq. 3.11	$0.008 \leq y_{w,h,d} < 0.7$
3	1.	$y_{w,h,d} > 0.7$

Table 4.9: Target values and boundaries of spatial regions of the regression concept.

  The distance dependency function of Eq. 3.11 uses $\tau_d = 32$. This approach uses the same weights to scale each spatial region in the loss function as for the segmentation approach.

- **Weighting** all classes is critical for preventing equal rewards by predicting empty voxels or a joining location correctly. Eq. 3.5 calculates the weights for each class $c \in \{0, 1, 2, 3\}$ and ensures that $a_0 < a_1 < a_2 < a_3$. The additional scaling factor further arbitrarily prioritizes joining location assigned voxels $\beta_c \in \{0.1, 0.1, 0.2, 0.6\}$. After normalization, this results in $a_c \in \{0.004, 0.014, 0.225, 0.757\}$ for $c \in \{0, 1, 2, 3\}$.
- **Strides** l^s for creating CCs out of joining scenarios are set as equal to the sizes of the CCs themselves, $w_{cc},\ h_{cc},\ d_{cc}$. No limit is implemented for minimal striding information τ_s.
- **Coordinate reconstruction** and performance measurement use the same settings as specified for randomized distribution (Section 4.3). To filter the joining locations from the probability approach, the threshold for $\tilde{T}_3$ is set to $\tau_l = 0.6$.

The resulting CCs have a highly unequal class distribution; see Table 4.10.

Grids on average have 97% of voxels labeled as 0, representing empty geometry. Class 3, joining locations, occupy approximately 0.001% of the voxels in a grid. Component geometry (class 1) and CRs take up approximately 2.7% and 0.3% of the voxel grid. Ratios of the means for classes 0, 1, 2, and 3 are approximately $7824 : 214 : 25 : 1$, respectively.

Process

This section describes the method's implementation for training ML models. Fig. 4.13 illustrates the prediction process.

The preprocessing steps of the methodology involved splitting data into training and test datasets. Joining scenarios can be split into multiple CCs. Hence, preprocessing splits joining scenarios with stratified sampling [445]. Stratified sampling enables an equal distribution of the number of joining elements between the datasets. Splitting of the dataset created the training

Class	Median	Mean	Std. Dev.
0	978,359.5	970,146.2	30,708.0
1	19,196.5	26,574.0	28,008.1
2	1,557.5	3,155.6	7,031.0
3	84.0	124.1	175.0

Class	Min.	25%	50%	75%	Max.
0	695,315	965,639.5	978,359.5	988,149.75	999,982
1	0	10,306.75	19,196.5	30,489.5	304,048
2	0	919	1,557.5	2,768.75	154,504
3	2	47	84	143	3,356

Table 4.10: Class distribution of all target samples. Abbreviations: Std. Dev. – standard deviation, Min. – minimum value, 25% – 1^{st} quartile, 50% – 2^{nd} quartile, 75% – 3^{rd} quartile, and Max. – maximum value.

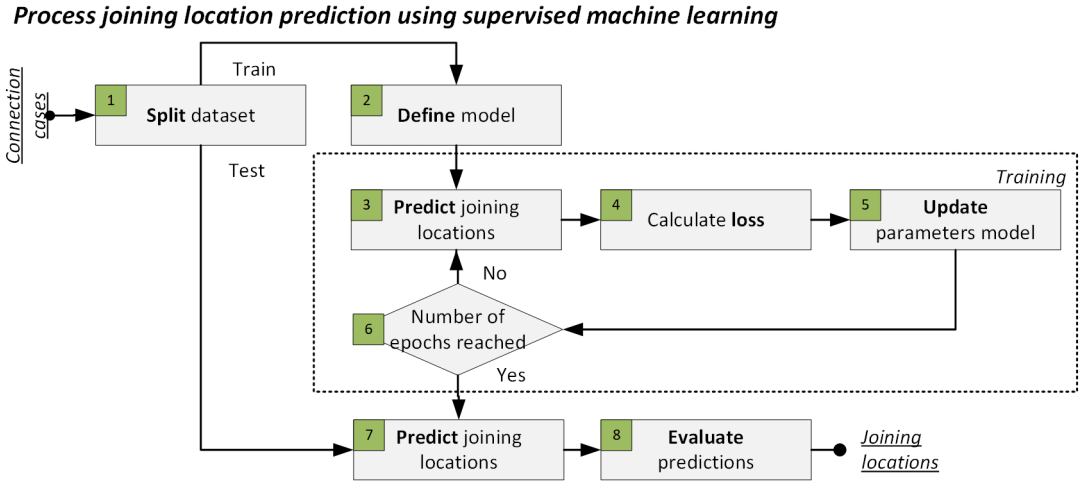


Fig. 4.13: Visualization of the prediction process for predicting joining locations with supervised machine learning.

D_{train} and test set D_{test} in a $80\% : 20\%$ ratio, respectively. Fig. 4.14 presents the distribution of the classes between the sets.

The area of each distribution added up to one, indicating that the datasets had roughly the same distribution of voxel classes. Hence, their CCs came from approximately the same distribution.

After creating the datasets, it was crucial to define the NN for predicting the joining locations. A network architecture with an EncDec structure considered spatial structures and feature-based hierarchies over larger spatial regions [360]. The implemented architecture was related to the synthesis network of Chen et al. [446], inspired by the VGG-like architecture for the deep CNNs used for image recognition tasks [447]. Fig. 4.15 presents the architecture, which used a smaller number of layers compared with the depth synthesis network of Chen et al. [446] to contain computational cost.

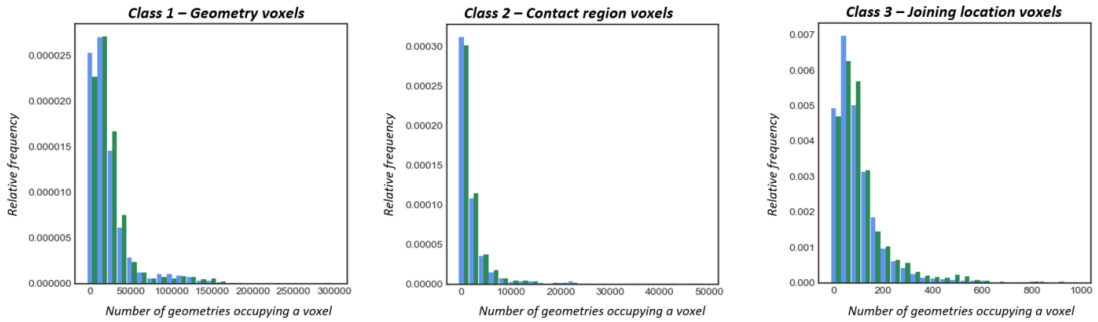


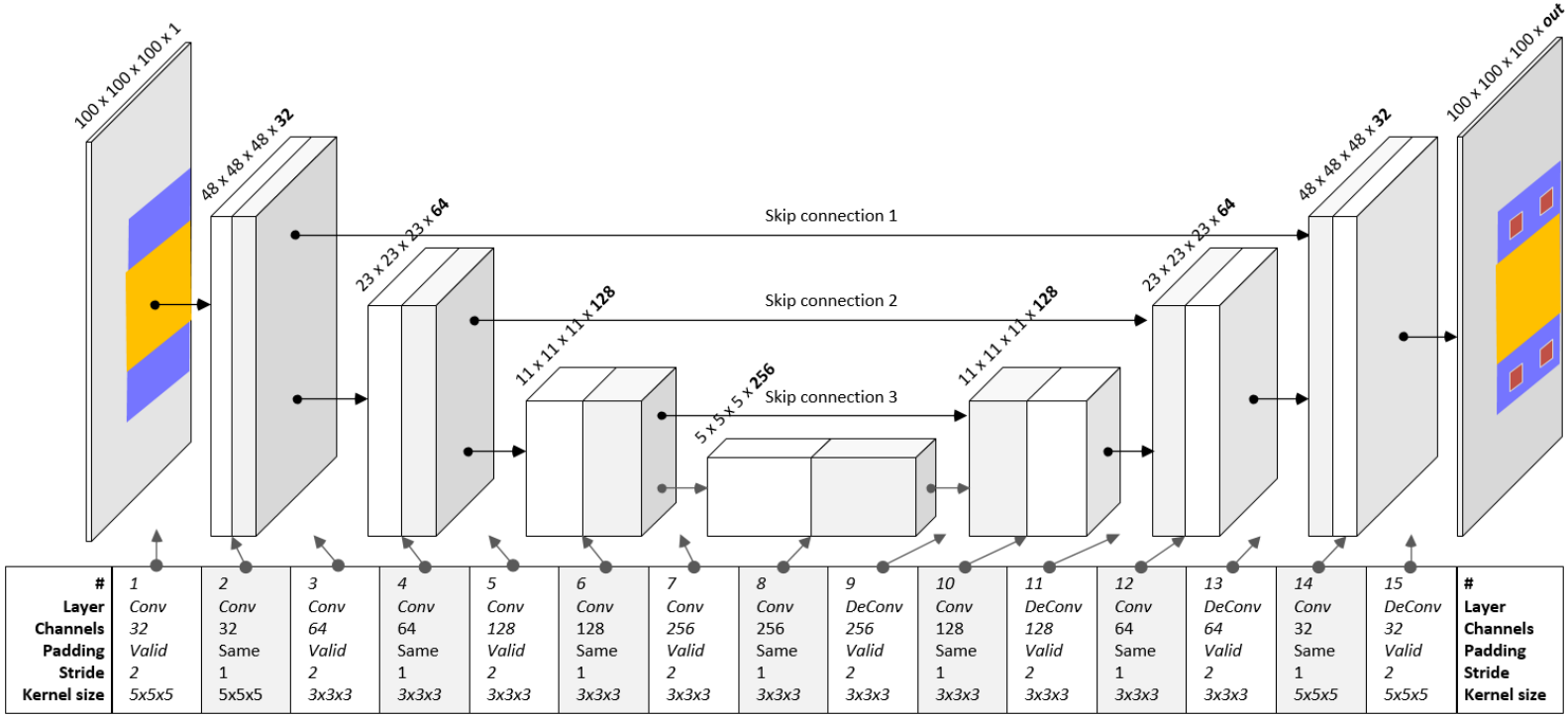
Fig. 4.14: Distribution of voxel class frequencies between the training + cross validation (blue) and test (green) sets.

Implemented networks based the architecture of Fig. 4.15 used Xavier initialization and a mini-batch of 10. Each convolutional layer had rectified linear unit (ReLU) activation functions and implemented batch normalization (see Goodfellow et al. [275]). Deconvolution layers do not have an activation function nor do they implement batch normalization. The architecture used skip connections to build deeper networks by reducing training problems from, for example, vanishing gradients [275]). Skip connections take outputs from an encoder layer and add them to outputs of a decoder layer of the same size. The Adam optimizer with standard settings was used [291]. Empirical testing determined an initial learning rate of $1 \cdot 10^{-4}$.

The network for the segmentation concept had a Softmax operation on the final layer (feature-map size = 4). It performed an Arg-max function across the feature maps to output the class with the highest assigned probability per voxel. The probability concept utilized min-max normalization in the final layer since a Sigmoid layer failed to converge during training. The network had approximately $5.4 \cdot 10^6$ trainable parameters.

The entire network had to process the information from the edge of the input data. Convolutional networks are sparse, meaning that the number of neurons connected to other neurons is relatively low compared with fully connected NNs [275]. Due to this sparsity, CNNs must be deeper (i.e., have more layers) to ensure the indirect connection of deep layers to the entire input. The receptive field refers to the scope of the input affected by hidden units in NNs [275]. Araujo et al. [448] presented mathematical derivations and a programming library to calculate the receptive fields of CNNs. The receptive field of the presented architecture (Fig. 4.15) was 97%, implying that almost the entire network was connected indirectly.

After defining the NN and before starting training, it was crucial to create a benchmark model. Benchmark models support the validation of concepts and implementations. They should enable the argument to be made for whether naive and random models may be superior to the developed methodology. Models trained on a loss function, and only the evaluation phase analyzed constructed coordinates due to the computational cost. Hence, during training, standard classification metrics, such as the F1-score, were used to evaluate model quality. The low frequency of joining location voxels enabled a 99.999% accuracy to be achieved by merely reconstructing the input. Similarly, a model that only predicts zeros would have a 97%



#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	#
Layer	*Conv*	*Conv*	*Conv*	*Conv*	*Conv*	*Conv*	*Conv*	*Conv*	*DeConv*	*Conv*	*DeConv*	*Conv*	*DeConv*	*Conv*	*DeConv*	**Layer**
Channels	*32*	32	*64*	64	*128*	128	*256*	256	*256*	128	*128*	64	*64*	32	*32*	**Channels**
Padding	*Valid*	Same	*Valid*	Same	*Valid*	Same	*Valid*	Same	*Valid*	Same	*Valid*	Same	*Valid*	Same	*Valid*	**Padding**
Stride	*2*	1	*2*	1	*2*	1	*2*	1	*2*	1	*2*	1	*2*	1	*2*	**Stride**
Kernel size	*5x5x5*	5x5x5	*3x3x3*	*3x3x3*	*3x3x3*	*3x3x3*	*3x3x3*	*3x3x3*	*3x3x3*	*3x3x3*	*3x3x3*	*3x3x3*	*3x3x3*	*5x5x5*	*5x5x5*	**Kernel size**

Fig. 4.15: The implemented encoder-decoder architecture, which was inspired by Chen et al. [446]. The number of outputs is 1 and 4 for the regression and classification approach, respectively.

accuracy. For simplicity, the randomized benchmarking model implemented the classification concept and changed a class 2 (CR) voxel into a class 3 (joining location) voxel with a certain probability. A probability of 15% provided the best results after empirical experimentation [421]. Besides the randomized benchmark model, validation included one regression and one classification model.

- **Randomized benchmarking** (*RB*) randomly transformed 15% of CR voxels in the input voxel-wise into joining location voxels in the output.
- **Segmentation** (*SEG*) performed a multiclass classification task with a weighted cross-entropy loss function (Eq. 3.4).
- **Probability mapping** (*PROB*) performed a regression task with a weighted mean squared error loss function (Eq. 3.12).

The implementation employed the TensorFlow v1.3 GPU library as an ML framework. Besides the learning rate adjustment, the training was stopped after 400 epochs as the F1-score did no longer optimized significantly. Furthermore, doing so limited the computational cost. A machine with $32\times$ Intel® Xeon® CPU E5-2667 v4 @ 3.20GHz with 256 GB RAM and an NVIDIA Quadro P6000 trained the models in approximately three days. The training time was too short for hyperparameter tuning of any sort. However, the hyperparameters used were the result of empirical research.

The performance measurements for determining the quality of the models were similar to those used for randomized distribution (Section 4.3). Section 3.3.4 describes the measurements in detail. Furthermore, subsection 3.3.2.5 explains the equivalency of creating coordinates between 2D and 3D approaches. Notably, this methodology uses Euclidean distance in the similarity metric; hence, a higher performance correlates with lower similarity scores.

Results

Table 4.11 lists the performance of the three models.

ID	**Cost**	$F1$	A	S	Q
RB	N/A	6%	N/A	N/A	N/A
SEG	$5.51 \cdot 10^1$	**96%**	**99%**	**0.07**	**86%**
PROB	$6.83 \cdot 10^2$	N/A	92%	1.01	26%

Table 4.11: Results of the test set of trained models for determining the suitability of concepts. Abbreviations: $F1$ – f1-score, A – accurateness, S – Euclidean dissimilarity, and Q – correctness.

The values clearly demonstrate that the segmentation model (*SEG*) had the best results for every performance. The randomized benchmarking model (*RB*), with its F1-score of 6%, demonstrated an inability to predict joining locations. The other performance measurements were not available due to the random distribution of the class 3 voxels. The regression concept (*PROB*) had a significantly higher similarity (S) measurement than segmentation (*SEG*). The

inability of the regression model to predict joining elements close to the targets resulted in a lower correctness score of 26%.

Discussion
First, this subsection discusses the regression approach from subsection 3.3.2.2. It continues by discussing the segmentation approach from subsection 3.3.2.1.

- **Regression approach**
  Fig. 4.16 displays four prediction extremes produced by the probability mapping concept: low cost $CC^{(3107)}$, high cost $CC^{(17)}$, low accurateness $CC^{(2107)}$, and high accurateness $CC^{(3400)}$. Coordinates, as presented in Fig. 4.17, are average indices within the input volume $H \times W \times D$ and do not express Cartesian coordinates at the joining scenario level. The grids in the images only display probabilities above 0.08%, corresponding with at least CR probabilities for the sake of clarity and readability. Both approaches implement Euclidean dissimilarity to measure the deviation of target joining locations.

  - $CC^{(3107)}$ had a meager cost. The input geometry was rather complex, but the model identified the small CR to predict spot welds.
  - $CC^{(17)}$ had a very high cost but predicted spot welds accurately. Cost scales with the confidence of predicted spot welds. The dissimilarity indicates the ability to localize them.
  - The model predicted four distributed spot welds in $CC^{(2107)}$ on the large vertical CR, where the target indicated one near the bottom (see dotted circles in Fig. 4.16). The dissimilarity was relatively high due to the pairing of the target coordinate to one of the predicted coordinates. Nevertheless, the model predicted the three remaining target coordinates correctly, resulting in 25% accurateness.
  - $CC^{(3400)}$ was a highly accurate prediction. Here, the probability mapping concept focused on spot welds by neglecting the reconstruction of CRs. In contrast to $CC^{(2107)}$ with spot welds on all CRs, for $CC^{(3400)}$ the model predicted spot welds only in the targeted locations.

  To summarize, the regression approach tended to scatter blue – increased probability voxels in most predictions arbitrarily. The noise in empty space and at the edges of CCs had little effect on cost. Furthermore, the model occasionally reconstructed undefined CRs (e.g., $CC^{(2107)}$) or ignored their reconstruction (e.g., $CC^{(3400)}$). Together, these properties make the regression approach less viable for predicting joining locations. Inherent uncertainty existed in the results, making postprocessing steps cumbersome.

  Moreover, the model suffered from data leakage, which usually positively affects performance measurements. However, this is not observed in the results. The performance of the regression model remains unconvincing, considering the data leakage. For example, as the reconstruction of CRs was difficult, the traceability of the manner in which the model made decisions was reduced. This made further

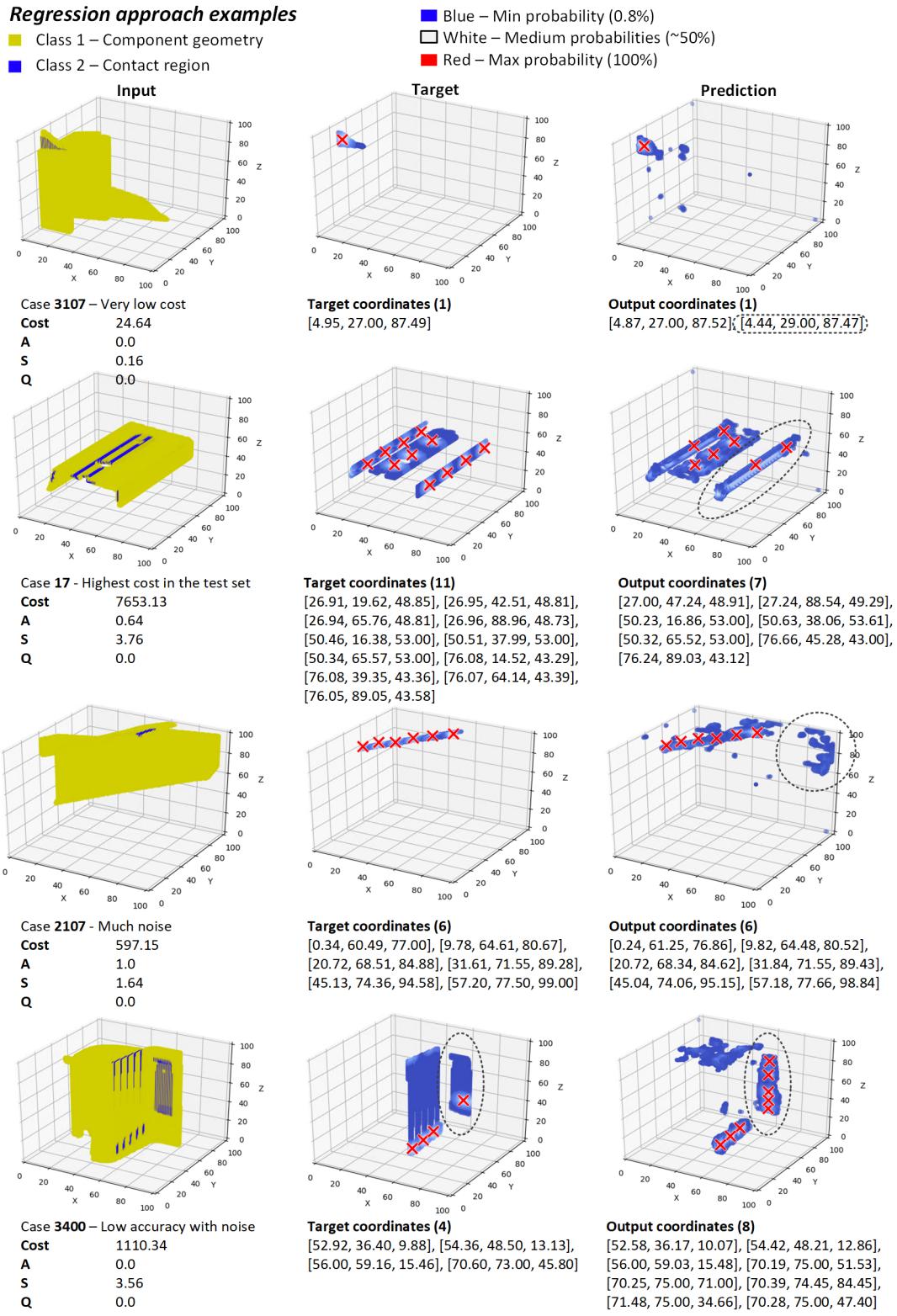


Fig. 4.16: Four prediction examples of a probability mapping concept using a regression approach; adapted from Eggink et al. [3].

optimization and understanding of the model more difficult. Lastly, Eggink et al. [3] found that the regression approach worked better with a smaller network. The model might have started overfitting in a larger network that fitted the segmentation approach better.

- **Classification approach**
  Fig. 4.17 presents four examples of prediction extremes made by the segmentation concept: lowest cost $CC^{(2401)}$, highest cost $CC^{(4320)}$, low accurateness $CC^{(2293)}$, and high accurateness $CC^{(2844)}$.

  - $CC^{(2401)}$ had the lowest cost with an optimal dissimilarity, implying that predicted coordinates exhibited no deviation from targets.
  - The highest cost of $CC^{(4320)}$ resulted in an inaccurate prediction. Here, the model did not reconstruct the CR correctly, preventing the prediction of spot welds.
  - $CC^{(2293)}$ was an exact prediction, but with missing spot welds exactly on CC splitting planes.
  - $CC^{(2844)}$ indicated that the model can reconstruct complex curved geometry accurately considering many rectangular shapes in the training data. Despite the high cost, it predicts coordinates correctly.

  To summarize, predictions using the classification approach are clean and with very little noise. One can observe that the model learns geometric dependencies. For example, CRs appear in overlapping geometry areas, and spot weld locations always lie on them. An ultimate example is $CC^{(4320)}$, which had no prediction of coordinates without reconstructed CRs. The classification approach makes a robust impression as it can recreate the targets highly accurately, also considering the data leakage problem.

Both models proved capable of predicting spot weld locations (Table 4.11). The table and figures indicate that segmentation tasks had more robust results with the used setup and training strategy. All performance metrics beat the probability mapping model. Although the network architecture may be more suitable for the segmentation approach, Nibali et al. [402] performed regression tasks using ResNet models designed for image recognition. The regression approach provided feasible results, but the output was noisy and the model did not reconstruct the geometry well.

Notably, the randomized benchmarking model (*RB*) had a very low F1-score, which was probably due to individual voxels being set to class 3. Although both learning tasks exhibited an ability to predict joining locations, a more robust benchmark model could better indicate their performance by, for example, taking the target number of spot welds and randomly distributing joining locations. This random model would be similar to the benchmark model in terms of validating randomized distribution (Section 4.3).

However, the model (*SEG*) had excellent scores (F1-score of 96% and correctness of 86%). After analysis, the training and test sets had similar CCs. This created data leakage, where models are evaluated on data samples they have trained on. The similar CCs had two main origins: product variety and splitting.

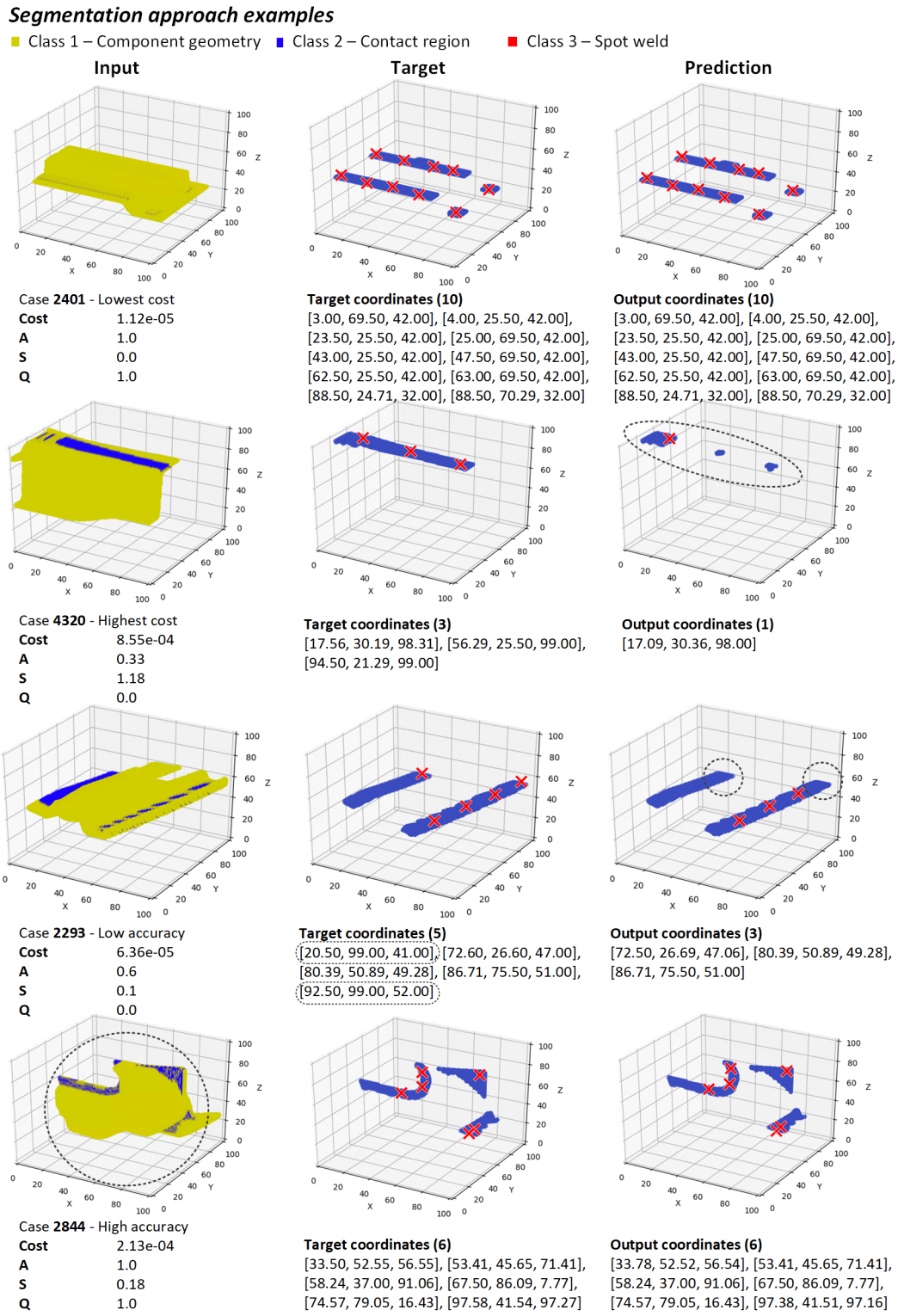


Fig. 4.17: Four prediction examples of a segmentation concept using a classification approach; taken from Eggink et al. [3].

- Product variety created a large dataset but also contains many similar joining scenarios.
- Splitting created small grids from joining scenarios, resulting in many repetitive CCs.

Hence, the models provided overly optimistic results for properly validating the concept. Nevertheless, Table 4.12 and the analysis of predictions in Figs. 4.16 and 4.17 indicate that the classification-based segmentation approach seemed more robust.

Evaluation
Table 4.12 lists the evaluation of the models.

ID	Perf.	Speed	Dev.	Comp.	Struc.	Appl.
RB	1	3	2	2	1	1
SEG	3	2	3	3	3	2
PROB	2	2	3	3	3	2

Table 4.12: Evaluation of the best SML learning task for predicting spot weld locations. Values $[1-3]$ correspond with low to high scores, respectively. Abbreviations: RB – randomized benchmarking, SEG – segmentation concept, PROB – probability-mapping concept, Perf. – performance, Dev – development effort, Comp. – computational cost, Struc. – structural performance, and Appl. – practical applicability.

As seen in the results in Table 4.11, the classification task through an image segmentation approach had the highest performance for all measurements. Furthermore, further criteria for evaluating the prediction models did not significantly differ from one another.

The results of the experiments are not sufficiently representative for objectively measuring performance due to data leakage problems. However, the results did reveal some structural behaviors of each concept. The more promising results for the segmentation approach created a basis for analyzing the concept in more depth. The segmentation approach exhibits an understanding of spatial dependencies in geometry. More specifically, it predicts spot welds on CRs. The spot welds are of the same size and distributed logically. This implies that if the segmentation model cannot reconstruct the CR, then it consequently will not be able to predict the required joining locations. The following section explores the segmentation concept further with multimodality and multiview approaches.

4.5 Predicting voxel-based joining locations considering nongeometric data

This section validates the implementation of nongeometric data in Euclidean data representations using branding. The previous section (4.4) concluded that the segmentation concept is more promising due to robust and precise predictions. Hence, this section continues with the implementation of a classification task. This methodology aims to validate whether branding is a suitable method for improving the prediction of joining locations by considering nongeometric data.

Approach

Including nongeometric data in the segmentation concept creates MMML. This involves bringing multiple data structures together to perform one task. Nongeometric data are structured enabling their representation in tables (e.g., see datasets for predicting joining technology [Section 4.1] or the number of spot welds [Section 4.2]). On the other hand, geometry is free-form and requires appropriate discretization to enable the training of ML models.

Subsection 3.3.2.6 roughly explained the approach for using branding as a technique to fuse nongeometric data and geometry. Through DR and clustering, labels can represent the nongeometric data of each data sample. A set of voxels represents each label. These labels are burned onto the input and target data samples; see Fig. 3.26. This section compares various branded models with benchmark (nonbranded) prediction models.

Additionally, the data leakage observed in Section 4.4 was reduced by using a cross-validation dataset, a similarity filter (subsection 3.3.2.4), and a new splitting strategy for large joining scenarios.

Implementation

This subsection discusses the implementation for reducing data leakage, including data augmentation and branding. Considering data leakage, **splitting** regards multiple CCs that originate from the same joining scenario. First, geometry for a new CC must at least occupy more than $\tau_s = 0.5$ along the splitting axis. This prevents the last CC from copying a large part of the previous one; see the potential overlap of the blue box in Fig. 3.24. Second, all CCs derived from a given joining scenario are moved into one dataset. There is an increased probability that these CCs have similarities, such as after the splitting of long beams. Moving them into one dataset prevents information leakage from these CCs to the test set.

Moreover, **data augmentation** can increase the number of data samples and potentially also the prediction model's performance. Empirical research provided additional information by rotating and mirroring CCs along the x- and z-axes. The sheet metals in a small scope of $200\ mm \times 200\ mm \times 200\ mm$ had slight differences, and augmentation may affect generalization to unseen cases. Due to internal symmetry axes, the similarity filter removed an additional 30% of the data after augmentation.

A further measurement for preventing data leakage is the **similarity filter**. It compares the geometry of CCs with one another. The implementation did not use dilation to buffer sheet metals. The similarity threshold $\tau_J = 0.3$ removed 23% of the CCs.

Branding requires labels for each data sample. Clustering nongeometric data categorizes joining scenarios, creating a branding concept for EncDec architectures. Fig. 4.18 presents the detailed preparation process for creating labels.

Preprocessing handles all typos, missing data, and one-hot encoding of nomenclature for the categorical (nominal) features. It is largely equivalent to the preparation process depicted in Fig. A.2. It creates binary vectors with one of the words included in the nomenclature, splitting the dataset into training, CV, and test sets at a ratio $70\% : 15\% : 15\%$. The datasets contained the same joining scenarios as for the training, CV, and test sets of geometrical ML. The similar datasets prevented data leakage through the branding of clustering results.

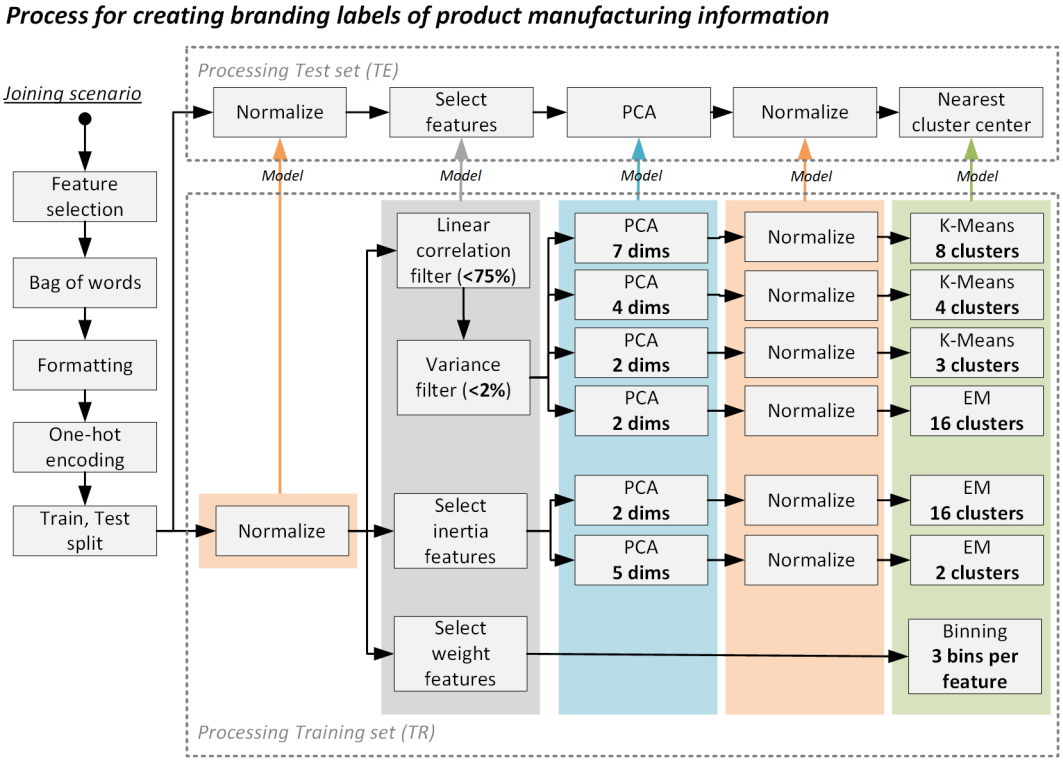


Fig. 4.18: Process for creating branding clusters and listing all multimodal models.

- Linear correlation and a variance filter removed features that added little additional information to each sample. After splitting, the training set had 5212 samples with 682 features. into Linear correlation filtered all coefficients greater than 75%, thus reducing the number of features to 540. Numeric-to-numeric features were filtered using Pearson's correlation coefficient with $p = 0.05$ and degrees of freedom $df = 5211$. Nominal-to-nominal features were evaluated for correlation using Pearson's χ^2 test [440] normalized by Cramer's V [441] with $p = 0.05$. The variance filter removed all features that explained under 2% of the variance and contained little information, thus reducing the number of features to 536.

- PCA reduced the dimensions of the dataset. Fig. 4.19 visualizes the scree-plot for determining the number of output dimensions for all nongeometric data.

  The plot does not clearly display an elbow. Hence, the PCA used identified elbows at 2, 4, and 7 eigenvectors. This reduction preserved 3.0%, 5.1%, and 7.5% of the information, respectively. Data that did not exhibit a clear angle in the scree-plot indicated that PCA might not be optimal. However, DR methods such as t-SNE did not create a meaningful clustering with, for example, large differences in the number of samples for each cluster. The small amount of information resulting from the PCA was likely due to the sparse features caused by the one-hot encoding of nominal features.

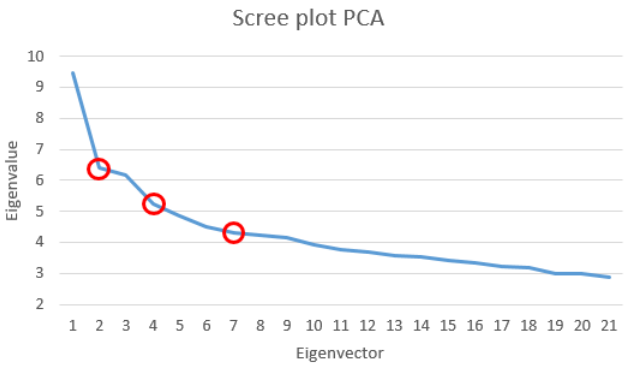


Fig. 4.19: Scree-plot of PCA for determining the number of dimensions for the dimensionality reduction of nongeometric data.

- After normalization of the PCA results, the number of dimensions was low enough for stable clustering. However, two cluster strategies are possible for branding. The first is to cluster all components individually and then combine them for every joining scenario. The second is to cluster the combinations of components for every joining scenario. The former approach creates $(n_{\text{clusters}})^2$ labels for joining scenarios. Not only does this rapidly scale with the number of clusters but also some component–cluster combinations may have a rather small sample size, preventing a model from learning. The second approach directly considers the combinations of components and their relations. A drawback is that clustering runs with almost twice as many features. Additionally, clusters may be more difficult to distinguish as combinations of very different components may be leveled by DR and clustering. Appendix A.5 discusses exemplary clustering results.

  Instead of the silhouette coefficient for cluster evaluation, the EM algorithm implemented a 10-fold cross-validation approach that checked whether the log-likelihood increased by increasing numbers of clusters [449]. The models of normalization, PCA, and created clusters were applied to transform the test data separately.

In addition to the extensive data preparation process, the data analysis revealed that the number of joining locations correlated most with weight and moments of inertia (Fig. A.4). Therefore, these features were also selected separately as features for clustering. The distribution of weights did not create clear clusters when using K-means or EM. Instead, a binning method enabled dividing the weights into three bins. Each bin contained 1,737 samples. The weight attributes came from both components. Therefore, the two features had the following bins: $[-0.917, -0.543]$, $(-0.543, 0.04]$, $(0.04, 5.555]$ for component 1, and $[-0.778, -0.591]$, $(-0.591, -0.042]$, $(-0.042, 3.427]$ for component 2. The moments of inertia consisted of six features for each component: I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}. PCA reduced these to two, and five dimensions after which the EM algorithm found 16 and two clusters, respectively.

After clustering nongeometric data to create the labels, branding necessitated identifying

the label's regions (subsection 3.3.2.6). Branding requires fixed regions of voxels to correspond to specific clusters. Choosing the right edge reduces lost information due to branding. Fig. A.10 in the Appendix plots geometric information summed over all voxel grids. Most voxels contain approximately 50 times the geometry. Relative to the approximately 6000 CCs, this is approximately 1% of the time. The total number of voxels on the ribs (with thickness $l_{rib} = 3$) was 10,656, or approximately 1% of the grid's total. This clearly demonstrates that little information existed toward the outside of the grid. Consequently, Fig. 4.20 visualizes the frequencies of geometry at the edges of the grid. It depicts only voxels that represent fewer than 10 times the geometry. Dark red voxels have the input value of 0 throughout the entire dataset. The edge that is darkest, and hence with the least information, runs along the z-axis from $(100, 100, 0)$ to $(100, 100, 100)$.

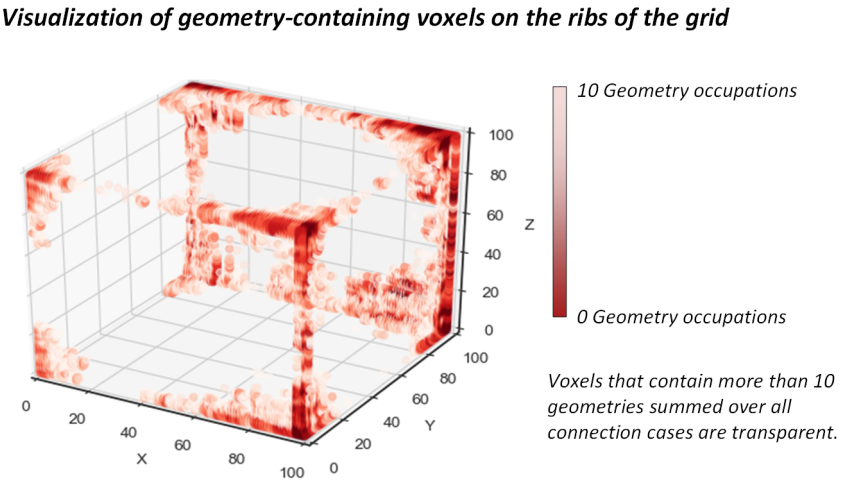


Fig. 4.20: Analysis of informational content in voxel grids for determining the edge for branding. The figure highlights barely used voxels (dark red) at the edges of the grid.

Branding used this entire edge to position labels. The dimensions of each label were $l_{rib} \times l_{rib} \times l_{brand}$ and they were spaced with $l_{space} = 1\ voxel$. The number of clusters $n_{clusters}$ determined the length of the brand label l_{brand}; see Eq. 4.3. The number of brands and their spaces determined the length of the feature vector l_{fv}. The edge length l_{edge} equaled the grid height h_{cc}. Lastly, cluster cl_i (0-indexed) of $n_{clusters}$ was branded from voxel index i_b until and including voxel index i_e.

$$l_{brand} = \left\lceil \frac{(l_{edge} - ((n_{clusters} - 1) \times t_{space}))}{n_{clusters}} \right\rceil \tag{4.3}$$

$$l_{fv} = n_{clusters} \times (l_{brand} + l_{space}) - t_{space} \tag{4.4}$$

$$i_b = cl_i(l_{brand} + t_{space}) \tag{4.5}$$

$$i_e = i_b + l_{brand} \tag{4.6}$$

Branding assigned class 1, component geometry, to all voxels of the label. It retained the

spatial structures that the model learned between the branded and nonbranded approaches. An additional fifth class (i.e., class 4) would increase the output dimension, forcing the model to predict an extra 10^6 voxels. Using class 2 or 3 for branding would interfere with the relationship between these classes as joining elements appeared.

Process

Fig. 4.21 depicts the updated process from Fig. 4.13.

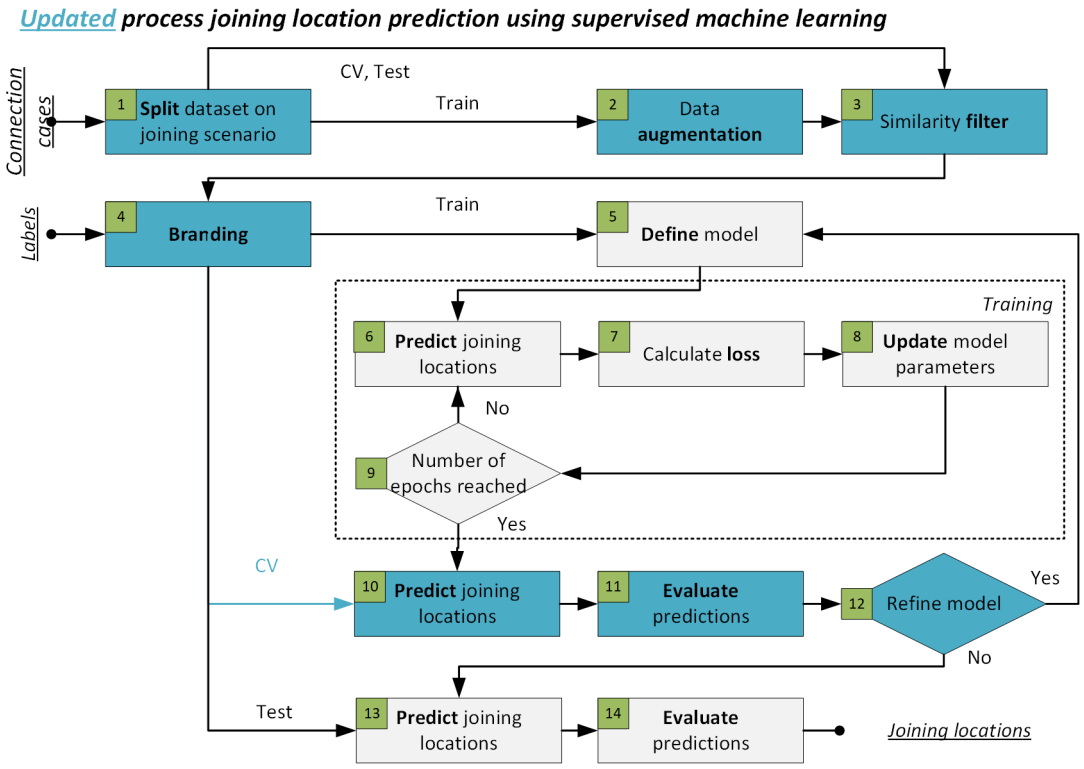


Fig. 4.21: Visualization of the updated prediction process for predicting joining locations with supervised multimodal machine learning.

The updated steps are in blue: data augmentation, similarity filter, branding, splitting, and the additional feedback loop using the CV set. The training, CV and test sets D_{train}, D_{CV}, and D_{test} were split at a ratio of 70%, 15%, and 15%, respectively.

After the datasets had been prepared, the training of the models remained largely the same. The process validated each model after it was trained on the cross-validation set. This loop enables changing the architecture or hyperparameters if, for example, the performance is not sufficient. It also enables multiple models to be trained without having feedback on the final quality of the test set.

This methodology was used to validate 11 models, which were separated into three groups: the benchmark, clustering & branding, and feature selection & branding models. The groups offer insight into how they affect the models' performance.

1. The first **benchmark** classification model with an augmented dataset. It did not perform MMML. It differed in results between models *ID2* and *ID3* due to the RI of weight.
2. The second **benchmark** classification model with an augmented dataset. It did not perform MMML. It differed in results between models *ID1* and *ID3* due to the RI of weight.
3. The third **benchmark** classification model with an augmented dataset. It did not perform MMML. It differed in results between models *ID1* and *ID2* due to the RI of weight.
4. The fourth **benchmark** classification model **without data augmentation** *ID4*. It did not perform MMML. It provided insight into the usefulness of data augmentation.
5. The first **multimodal model** through **data reduction and clustering** *ID5*. It found two main dimensions using PCA (Fig. 4.19) and three clusters using K-means (Fig. A.9).
6. The second **multimodal model** through **data reduction and clustering** *ID6*. It found four main dimensions using PCA (Fig. 4.19) and four clusters using K-means.
7. The third **multimodal model** through **data reduction and clustering** *ID7*. It found seven main dimensions using PCA (Fig. 4.19) and eight clusters using K-means.
8. The fourth **multimodal model** through **data reduction and clustering** *ID8*. It found two main dimensions using PCA (Fig. 4.19). These dimensions were the same as for model *ID5*. Furthermore, it determined 16 clusters using EM (Fig. A.9).
9. The first **multimodal model** through **feature selection** *ID9*. It supplied the models with the number of spot welds from the target of the joining scenario. These 33 clusters were the same as for the prediction of the number of spot welds (Section 4.2).
10. The second **multimodal model** through **feature selection** *ID10*. It supplied the models with binned weights of components. Each component had three bins. Considering two components per joining scenario, this resulted in nine labels.
11. The third **multimodal model** through **feature selection** *ID11*. It supplied the models with moments of inertia after their dimension reduction with PCA to two dimensions. Furthermore, it determined 16 clusters using EM.

All models trained for 400 epochs to create a cut-off point for evaluation as well as to contain computational cost (models trained for approximately three days). The models trained for 400 epochs as from this point the cross-validation error surpassed the test set error, while its F1-scores remained almost constant.

Results

Table 4.13 lists the results of the segmentation concept and compares them with those of models that also had access to nongeometric data.

Generally, only slight differences existed between models' performances on the test sets. They all remained within a few percent, let alone the benchmark model without augmentation. Moreover, models *ID1*, *ID2*, and *ID3* differed by $\pm 2\%$ in their F1-scores. This variation overlapped that of the MMML models as well as for metrics such as similarity and

MODELS						TRAIN		CV	
ID	**MMML**	**DR**	**CM**	**nC**	**E**	**C**	$F1$	**C**	$F1$
1	–	–	–	–	365	33.39	0.99	1852.78	0.45
2	–	–	–	–	365	32.14	0.99	1699.58	0.46
3	–	–	–	–	290	33.39	0.99	1656.80	0.45
4	–	–	–	–	295	31.42	1.00	2653.83	0.31
5	ALL	2	K	3	275	39.80	0.99	1698.77	0.47
6	ALL	4	K	4	350	39.31	0.99	1499.55	0.46
7	ALL	7	K	8	375	36.36	0.99	1983.37	0.46
8	ALL	2	EM	16	390	34.31	0.99	1922.39	0.46
9	N_{RSW}	–	–	33	305	32.89	0.99	1832.82	0.47
10	WEIGHT	–	B	9	225	34.87	0.99	1727.40	0.46
11	INERTIA	2	EM	16	385	33.50	0.99	1754.72	0.46

	TEST					
ID	**C**	$F1$	A	S	Q	ΔN
1	2238.94	0.38	0.62	0.45	0.22	–0.62
2	2038.28	0.40	0.63	**0.47**	0.23	–0.49
3	2030.98	0.39	0.63	0.45	0.23	–0.39
4	2731.41	0.30	0.58	0.38	0.13	–0.39
5	1998.67	**0.42**	**0.65**	0.46	0.23	**–0.30**
6	**1794.79**	0.41	0.64	0.45	0.22	–0.43
7	2601.20	0.40	0.63	0.45	0.23	–0.34
8	2352.27	0.39	0.65	0.46	0.23	–0.42
9	2208.68	0.40	0.64	**0.47**	**0.24**	–0.40
10	2105.80	0.40	**0.65**	0.45	0.23	–0.38
11	2144.69	0.39	0.62	0.46	0.22	–0.62

Table 4.13: Results of the trained models for determining the suitability of concepts. Abbreviations: MMML – selected features for multimodal machine learning, DR – output dimensions of PCA, CM – used clustering method, K – K-means, B – binning, EM – Expectation Maximization, E – number of epochs, nC – number of output clusters, TR – training set, CV – cross-validation set, TE – test set, C – cost, $F1$ – f1-score, A – accurateness, S – similarity (Gaussian with $\sigma = 1.6$), Q – correctness, and ΔN – difference in number of spot welds between output and target.

correctness. Furthermore, the performance of the models stagnated well before the 400 epochs. Notably, the test set contained 899 unseen samples. Hence, each percentage improvement would correspond with approximately nine additional correctly predicted samples.

Model *ID5* had the highest F1-score of 41%, accurateness of 65%, and Gaussian similarity value of 0.47, and also the highest number of spot welds on the test set. Notably, Gaussian similarities of 0.45 and 0.47 with $\sigma = 1.6$ correspond with Euclidean distances of $4.09\ mm$ and $3.87\ mm$, respectively. Furthermore, models *ID2* and *ID9* achieved this. Model *ID9* had the highest correctness of 24%. Models *ID1* and *ID11* exhibited the worst values for correctly predicting the number of spot welds (-0.62). Model *ID10* (weight) added the least information

to the training data. Although, the weights are features that most highest correlate with the number of spot welds (see Fig. A.4). Although, this model did not yield interesting results. Lastly, model *ID4* trained on a smaller, nonaugmented dataset, which resulted in significantly lower results. Fig. 4.22 presents the distribution of the predicted number of spot welds to the targets for models *ID2*, *ID4*, and *ID5*.

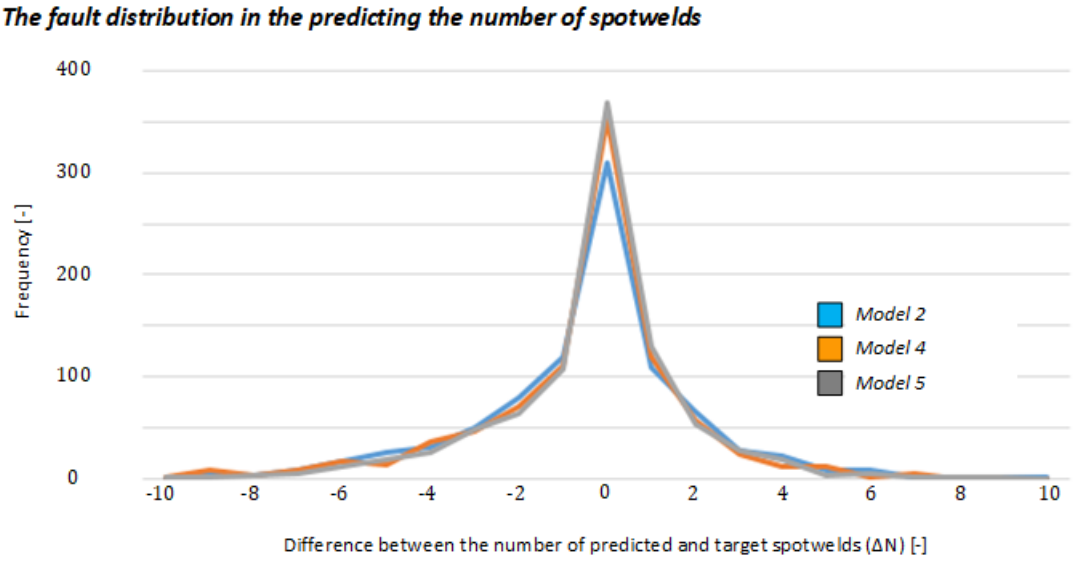


Fig. 4.22: Plot of the differences in the predicted number of spot welds for models *ID2*, *ID4*, and *ID5*.

The curves of the models exhibited a considerable overlap, as was expected after referring to Table 4.13. The graph generally indicates that the models were more likely to predict too few spot welds, as the curves on the left-hand side reduced more slowly than on the right. Model *ID5* predicted 369 of the 899 test samples with the correct number of spot welds. This value was 310 and 353 for models *ID2* and *ID4*, respectively.

On the other hand, model *ID5* predicted 177 cases (20%) with a difference of more than ± 2 spot welds. All models in the results table (Table 4.13 exhibit negative values for the number of spot welds. The models also unanimously predicted too few spot welds. As the distribution in Fig. A.3 indicates, having fewer spot welds was much more likely. The chance of predicting a smaller number of spot welds was also higher. Furthermore, the noise filter during coordinate creation might filter small voxel groups that would otherwise be joining locations. However, lowering this requirement might affect the accurateness and similarity negatively.

Interestingly, the performance metrics did not all explain the models' quality equally well. Fig. 4.24 already indicated a lack of correlation between cost and F1-score. It is assumed that a lower cost would imply a better F1-score as voxel-wise differences between predictions and targets reduce. Table 4.14 presents the correlation of the performance metrics in the predictions of the test set for model *ID5*.

The table shows the linear correlation using the Kendall's product-moment coefficient [450] with $p < 0.05$. The F1-score of the joining location class was a better predictor of

Model 5	**Cost**	$F1$	MCC	A	S	Q	ΔN
Cost	X	–0.38	–0.37	–0.22	–0.37	–0.305	0.132
F1-score (class 3)		X	0.99	0.65	0.94	0.83	–0.11
MCC (class 3)			X	0.65	0.94	0.82	–0.10
Accurateness (A)				X	0.61	0.55	0.12
Similarity (S)					X	0.79	–0.05
Correctness (Q)						X	–0.07
Diff. number (ΔN)							X

Table 4.14: The correlation between the performance metrics of the test set results of model *ID2*. Abbreviations: $F1$ – f1-score, A – accurateness, S – similarity (Gaussian with $\sigma = 1.6$), Q – correctness, and ΔN – difference in the number of predicted and target coordinates.

coordinate-based performance than cost. The latter barely correlated with any of the other metrics. Currently, the loss function considers voxel-wise differences, neglecting relations between voxels. The loss function might benefit from considerations of the other characteristics. For example, although coordinate-based metrics have a high computational cost, a term similar to the intersection over unions (Jaccard index) might aid the model during training.

Moreover, the performances of MMML and the benchmark models only differed slightly (Table 4.13). Hence, Table 4.15 presents the results from applying models *ID5*, *ID6*, and *ID7* regarding whether branding has an effect on performance.

ID	**SET**	**Cost**	$F1$	A	S	Q	ΔN
5	Branded	1998.67	**0.42**	0.65	0.46	0.23	–0.30
5	Benchmark	**1991.26**	0.41	0.65	0.46	0.23	**–0.29**
6	Branded	1794.79	**0.41**	0.64	0.45	0.22	**–0.43**
6	Benchmark	**1788.79**	0.40	0.64	**0.46**	0.22	–0.44
7	Branded	2601.20	0.40	0.63	0.45	0.23	–0.34
7	Benchmark	2601.20	0.40	0.63	**0.46**	0.23	–0.34

Table 4.15: Results of MMML model *ID5* on a nonbranded dataset. Abbreviations: $F1$ – f1-score, A – accurateness, S – similarity (Gaussian with $\sigma = 1.6$), Q – correctness, and ΔN – difference in number of spot welds between output and target.

These models used the benchmark augmented dataset (of models $[1 - 3]$ in Table 4.13). Consequently, the MMML models had no access to the branded information. If they relied on labels, the performance should have dropped. Comparing these predictions with the performances in Table 4.13, it becomes evident that no significant difference existed between the models. They were able to predict joining locations with and without the branded meta data equally well. Moreover, Fig. 4.23 presents that model *ID5* reconstructed the label (yellow stripe in the left graphs).

The label indicated that the information traveled through the network. However, model *ID5* made the same mistake on sample $CC^{(3593)}$ by predicting four instead of the required one spot welds. All of these four spot welds were located on the same spot.

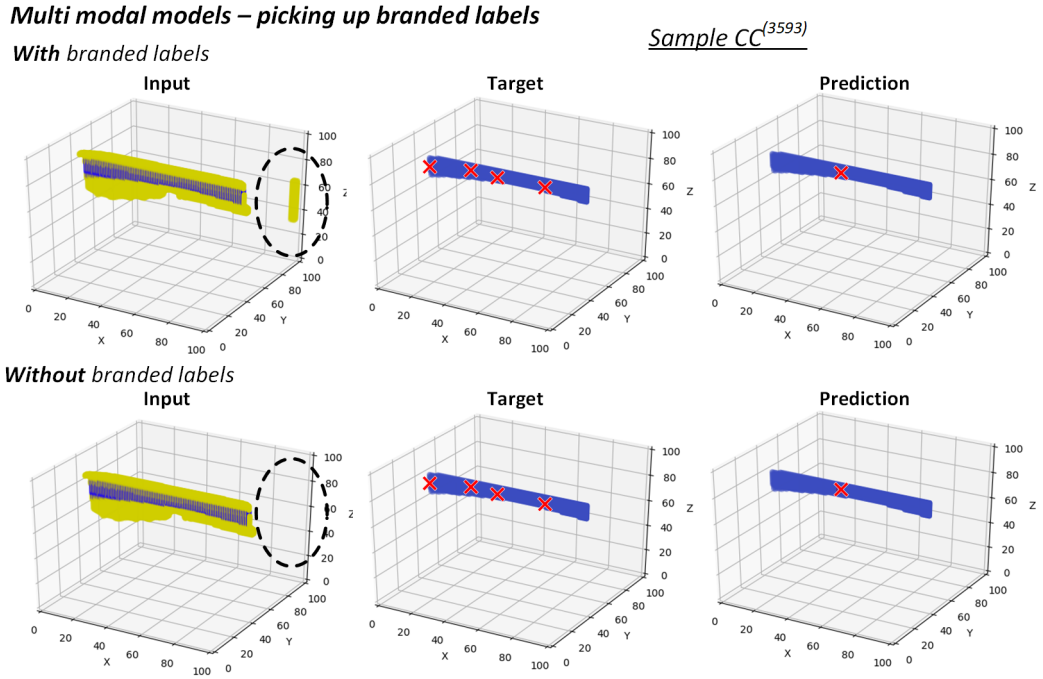


Fig. 4.23: Predictions of model *ID5* on the branded and nonbranded dataset for sample $CC^{(3593)}$.

Therefore, the models did not integrate the informational content of labels into their predictions. The following experiment validated the prediction of labels using nongeometric data. It sought to demonstrate whether a prediction model can create a mapping between nongeometric data and cluster labels. A random forest classifier predicted cluster labels from the normalized training data before DR (see the first *Normalize* step in the process of Fig. 4.18). Then, the classifier needed to imitate the PCA and clustering step. The random forest classifier used the settings described in Section 4.1. Classifiers having high performance would suggest that it is possible to learn the differences between the branded labels.

The classifiers were trained on the labels used in models *ID7* and *ID8*. They had different numbers of dimensions after PCA as well as different clustering algorithms (K-means and EM); see Table 4.16.

ID	MMML	DR	CM	nC	Accuracy	κ
7	all	7	K	8	**0.984**	**0.980**
8	all	2	EM	16	0.942	0.935

Table 4.16: Validation on the training set of the branding labels by using random forest regressors to model the dimensionality reduction and clustering for models *ID7* and *ID8*. Abbreviations: MMML – selected features for multimodal machine learning, DR – output dimensions of PCA, CM – used clustering method, K – K-means, EM – Expectation Maximization, nC – number of output clusters, and κ – Cohen's kappa.

The dataset was split into an $80\% : 20\%$ ratio between the training and test set. The results indicated that the random forest classifier could map the normalized input data to the output

clusters effectively. This indicated that this mapping is learnable and that the models could classify nongeometric data from unseen cases with high performance for model *ID7*. The somewhat lower performance for the EM clustering result was likely due to its ambiguity as clusters consider variance distribution. Nguyen demonstrated that random forest regressors can mimic nonlinear DR methods on nongeometric data of spot welds, such as t-SNE [421].

Besides the performance metrics themselves, the models' performance between the datasets was also examined. All models exhibited variance in costs and F1-scores between the training and cross-validation sets; see TR and CV in Table 4.13. This variance probably originated from the overfitting of the training data. Variance describes a lack of a model for generalizing knowledge [275]. It results in significantly lower performance on unseen samples in the test set. Specifically, the lack of data augmentation in model *ID4* indicated that the dataset was relatively small. Fig. 4.24 clearly visualizes the variance problem.

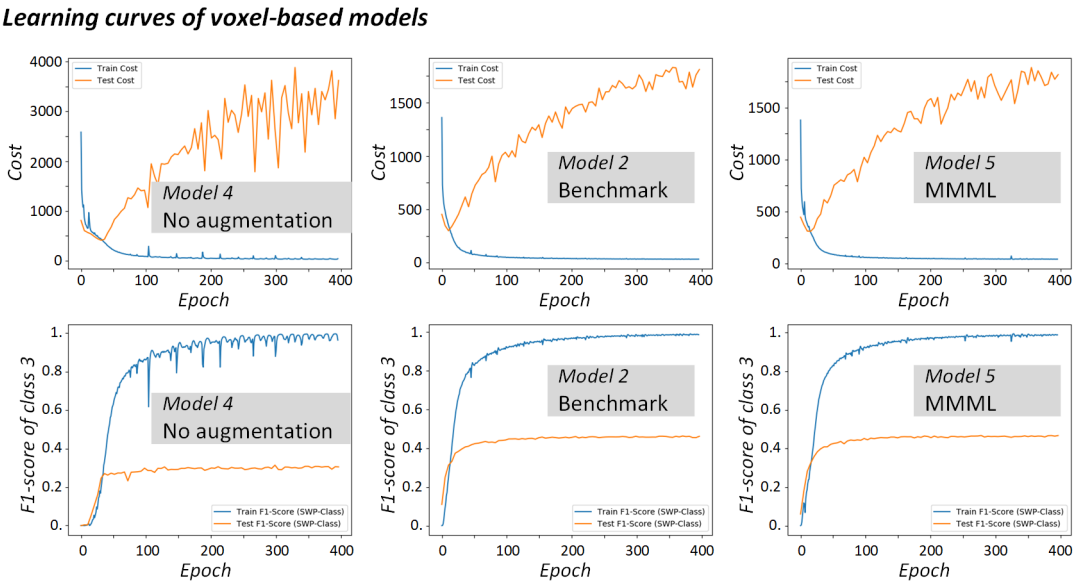


Fig. 4.24: Exemplary training curves measuring the cost and F1-score of class 3 for models *ID2*, *ID4*, and *ID5*.

The blue and orange lines represent the model's performance on the training and cross-validation set, respectively. The cost plots show the orange curve quickly rising after a few epochs, whereas the blue training curve lowers. Hence, longer learning on the training set reduces the capacity of the model to predict more accurately on the cross-validation or test set. However, the plots with F1-scores exhibit early stagnation but not necessarily overfitting on the training data, implying that the models would rapidly perform significantly worse, as the cost curve suggests.

Moreover, splitting joining scenarios into CCs might influence data leakage and variance problems. For example, long beams found in automobile structures might create many similar CCs. Although the similarity filter removes many overlapping data samples, slight differences might still come through. Similarly, small joining scenarios might consist of specific parts, such as reinforcements or halters. These scenarios tend to be more diverse as they are specific

solutions between larger parts. Table 4.17 depicts the performance of model *ID5*; see Table 4.13 on subsets of CCs of split joining scenarios. The split CCs have a slightly better performance, but the difference is not significant enough to state that those CCs are often similar.

ID	**Subset**	m	A	S	Q
5	all	899	0.65	0.47	0.23
5	split	208	**0.68**	**0.47**	**0.24**
5	not-split	691	0.64	0.46	0.22

Table 4.17: Results on the test set of model *ID5* for determining the influence of splitting. Abbreviations: m – number of samples, A – accurateness, S – similarity (Gaussian with $\sigma = 1.6$), and Q – correctness.

Discussion

This subsection analyzes the differences between models *ID2* and *ID5*, representing benchmark non-MMML and MMML, respectively.It focuses on the differences in predictions in the test set. Figs. 4.25 and 4.26 present five CCs with their input, target, and predictions for model *ID2* and *ID5*. The exemplary CCs were cherry-picked based on CR shapes, spot welding distances, and geometry complexities.

- $CC^{(70)}$ had four locations in the target grid (Fig. 4.25). Both models predicted the spot weld locations almost perfectly. The cost was very low and coordinates differed only within hundredths of a millimeter. The spot welds lay equidistant in the middle of the CR. The input image from the next sample $CC^{(1703)}$ originated from the test set of model *ID5*, showing the branded label in the top right corner.
- $CC^{(1703)}$ had a complex component geometry and the CR lay on one side in the grid (Fig. 4.25). The target required three coordinates. Both models created too many spot welds by attempting to fill the entire CR. Furthermore, they exhibited uncertainty as the joining locations were not points, but rather more like stains. They were smeared over the surface. Moreover, there were small individual voxels, or noise, in between the larger spot weld voxel groups. These did not translate to coordinates due to the filtering function in the performance measurement.
- $CC^{(3593)}$ was a CC with a large CR, yet it required one target coordinate (Fig. 4.25). Both models aimed to distribute multiple spot welds over the CR. Performance metrics such as F1-score, MCC, and similarity were acceptable as both models predicted the one required coordinate.
- $CC^{(4771)}$ had two CRs that required four target spot welds in total (Fig. 4.26). The models seemed to distribute spot welds on the larger region equidistantly, thus predicting too many joining locations. Model *ID2* predicted one spot weld too many; however, it had a higher cost, yet a lower F1-Score, MCC, and similarity than the result of model *ID5*. Only the accurateness metric and the difference between the number of spot welds indicated the “overprediction” of model *ID5*.
- $CC^{(5913)}$ contained two long rectangular CRs with a high mutual distance between spot welds in the target grid; see Fig. 4.26. None of the models output equidistant spot welds.

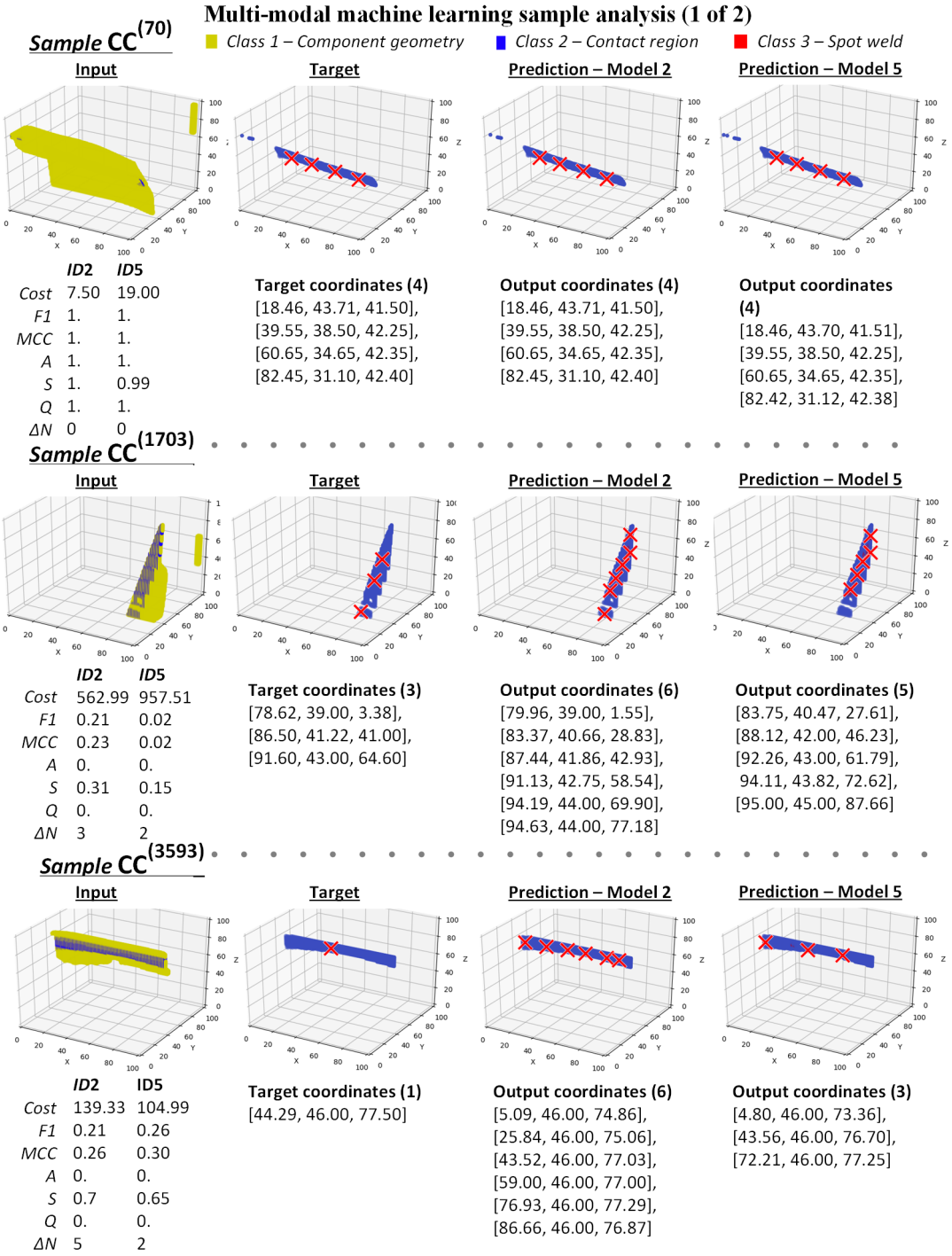


Fig. 4.25: The analysis of samples $CC^{(70)}$, $CC^{(1703)}$, and $CC^{(3593)}$. The figure plots the input, output, and predictions of model *ID2* (no-MMML) and model *ID5* (MMML), and also lists the performances of the models.

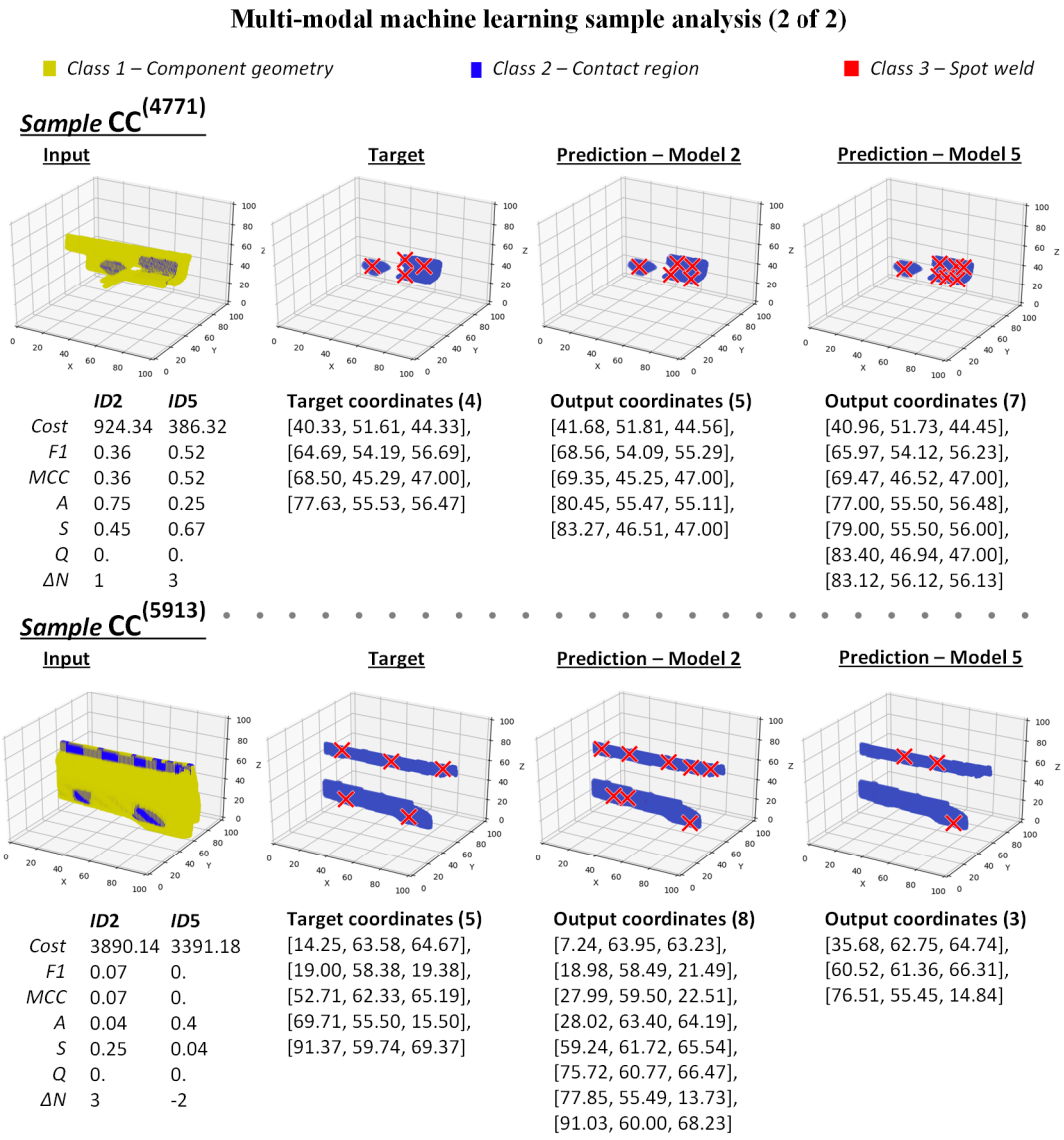


Fig. 4.26: The analysis of samples $CC^{(4771)}$ and $CC^{(5913)}$. The figure plots the input, output, and predictions of model *ID2* (no-MMML) and model *ID5* (MMML), and also lists the performances of the models.

Interestingly, model *ID2* predicted three too many and model *ID5* two too few spot welds. All performance metrics exhibited very low results, but still the output of the models seemed feasible. The CRs were properly reconstructed, and spot welds only appeared on the blue areas. Furthermore, no real sign of noise existed.

All models provided plausible results. However, a detailed analysis was required to find differences between the predictions. The models meticulously reconstructed the CRs and predicted spot welds that only occurred within them. Hence, they created a consistent

input–output mapping of the geometry. The EncDec network recreated the joining component geometry and CRs accurately. Furthermore, joining locations were very often tiny cylinders of approximately the target size. The locations only appeared on the CR area with an appropriate edge distance to the flanges. Most predictions did not contain noise in the form of tiny spot weld voxel groups. These properties of the models ensured that many of the predictions seemed feasible and even manufacturable.

However, several problems must be addressed. First, the grid size of a CC is a cube with $20\ cm$ ribs. This creates a small scope of geometrical context for the model. This scope also incurs similarity problems in datasets, such as shifted joining locations on large CRs or data leakage (Section 4.4). One solution might be to use the grid-based drawing approach on 2D images. For example, Dhameliya [451] presented exploratory work that predicted spot weld locations using a smaller network that considers larger geometric surface areas.

Moreover, structures over many product variants often contain similar components and combinations. They use specific components and structure orientations to optimize safety requirements, such as crash-worthiness. These considerations limit the data augmentation possibilities. For example, the rotation of a CC might duplicate another nonaugmented sample.

Thus, large components do not to fit the defined size of data samples. However, splitting causes the samples not to relate to one another. Long short-term memory models can combine data samples to reconstruct large components. An autoencoder similar to that of Essien and Gannetti [452] could load all data samples before prediction. This would enable entire joining scenarios to be considered at once and would also potentially increase the prediction quality.

Besides the grid-size, the resolution also limits geometric information for models. The implemented resolution is $r = 2\ mm/voxel$. This is a trade-off between the level of geometrical detail and computational cost. This resolution prevents representing the thicknesses of most sheet metals. Engineers actively consider component thicknesses and materials as a decisive factor in determining minimal spot weld distances [350]. Higher component thicknesses and stronger materials correlate with increased forces and loads on joints [118]; see Section A.4. Theoretically, MMML enables supplying models with such information, such as component weight, albeit unfortunately without significant improvements to their performance.

The resolution, grid size, product variety, and data augmentation create the need for independent data samples. The similarity filter aims to determine these data samples. However, it is a cumbersome endeavor with a high computational cost. Similarity filtering requires all CCs to be compared with another, and it is still an empirical task to determine the degree of similarity. The task involves estimating the amount of additional informational value a given CC adds to the dataset based on a similarity measurement. However, instead of similarity filtering on voxel grids, filtering on component geometry (e.g., using shape descriptors) of joining scenarios may create more robust datasets.

Furthermore, the similarity filter aims to reduce data leakage that models in the previous section suffered (Table 4.11). These models had excellent scores, but the performance of benchmark models (Table 4.13 with identifiers $[1 - 4]$) dropped significantly. This drop indicates the size of the data leakage.

Nevertheless, the models still predicted plausible spot weld locations. The results indicated that the models found patterns for specific geometries and could map them to new unseen CCs. On the other hand, design freedom on larger geometries may bring uncertainty into the model as solution spaces increase. Distributing the joining locations may be difficult as the target data includes asymmetric solutions that would likely throw off ML models.

As the performance measurements dropped after similarity filtering, the branding implementation could not compensate for it. MMML did not perform to expectations. Moreover, the performances did not significantly differ from the benchmark models. There are various explanations for the lack of performance increase, including increased complexity with identical conditions, suboptimal data reduction, clustering, and branding applicability.

- Models need to learn that branding **increases the informational content** of samples. The model needs to relate labels to both geometry and resulting joining locations. This requires a dataset that includes similar geometries that relate differences in branding labels to differences in spot weld distributions. However, similar geometries require filtering to prevent data leakage, and consequently, the variance of data samples might be too small, making branded labels irrelevant.

  Furthermore, branding increases the complexity of models. Besides the geometry-based prediction task that occurs in the benchmark models, prediction models must determine the relation of a label to joining locations. Hence, these models must perform more complicated prediction tasks with the same architecture. Consequently, discarding information with PCA and inconsistencies in clusters may even distract models from predicting joining locations, thereby lowering their performance.

- **Subpar data reduction and clustering** might create counterproductive labels. Model *ID7* had the highest remaining information content after DR. These clusters represented 7.5% of the variance of the nongeometric data.

  Additionally, clusters might be ambiguous. Thus, CCs might receive different labels to geometrically similar CCs. This might in particular create issues for clusters that contain a smaller number of samples. For example, model *ID4* had eight branding labels. Three labels contain only 3%, 6%, or 7% of the examples. These data samples may be underrepresented in the dataset and may be interpreted as noise by the model.

  However, decision trees reveal that clusters seem predictable from nongeometric data, implying an achievable input–output mapping. Still, the dataset requires enough samples to learn the particularities of labels, their relation to CCs, and their influence on joining locations.

- The branding concept might **not** be **applicable** for the task or dataset of predicting joining locations. The few studies that have implemented branding (e.g., [376]) have used pre-trained models of image recognition. This study applied the concept in the field of geometrical ML. The implemented EncDec structure, hyperparameters, or dataset may not enable the integration of information from the brand labels onto the geometric considerations for generating joining locations.

Regardless of multimodalities, some generic issues need to be addressed. The introduction of a separate holdout (test set) enabled the visualization of overfitting issues (Fig. 4.24), such

as the large gap between the cross-validation and test set curves. There may be two main explanations for the variance.

- Differences in the data distribution between datasets: Samples in the training set described different CCs to those on the cross-validation and test sets. However, Fig. 4.14 reveals that the classes between the training & validation (blue) and test data (green) for all classes overlapped. However, the figure also presents the distribution in the frequency of classes; for example, it neglects the distribution of certain joining location patterns or component properties.
- A small dataset may not contain enough variety in samples for the model to generalize well. A more extensive dataset increases the informational content, enabling better generalization and thus performance. Additionally, other data augmentation strategies might also boost the performance.

Furthermore, performance measurements, such as accurateness, similarity, and difference in the number of predicted joining locations, explain much about a CC. However, they are blunt instruments for model evaluation, as observed in the CC analyses. The metrics suffer from an abstraction of performance, sometimes making it difficult to interpret quality. Additionally, the model assumes target samples to be ground truths, although they are mainly experience-based manual designs. Measuring the performance relative to the target does not necessarily express the quality of the predicted joining locations. For example, the CCs $CC^{(3593)}$, $CC^{(4771)}$, and $CC^{(5913)}$ indicated that the model predicted feasible and manufacturable designs, but without directly overlapping joining locations. The objective measurement of the quality of an ML model would require an FEA that evaluates the joining locations within the entire product.

Interestingly, the difference in the number of predicted and target spot welds was lower for the best MMML voxel-based model *ID5* $\Delta N_{je}^{vox} = -0.30$ than for the best number-of-spot-welds predictor *ID3* (XGBoost; see Section 4.2) $\Delta N_{je}^{\mathrm{XGB}} = -0.461$. Naturally, the datasets differed in size, formatting, and scope (CCs versus joining scenarios). Still, the models predicted the correct numbers in both cases rather well. The XGBoost predictor had no information about geometry, such as the CR size. Moreover, the MMML-based predictor was clueless regarding the material, function, and size of the components. Both models' relative success in predicting the number of spot welds confirmed the contribution of both nongeometric and geometric data in joining element design.

Evaluation

Table 4.18 presents the evaluation results of the 11 models. The following list explains the ratings in the table:

- As seen in the results table (Table 4.13), no significant difference existed in (*structural*) *performances* between models.
- However, there was additional *development effort* for the branding approach (*ID5* to *ID11*). The effort was even higher for creating labels through DR and clustering (*ID5* to *ID8*).

- The *speed* of prediction and *computational cost* followed the same considerations as for development effort.
- Based on the current implementation, the benchmark models (*ID1* to *ID3*) are the most promising direction for *application in industry*.

ID	Perf.	Speed	Dev.	Comp.	Struc.	Appl.
1, 2, 3	2	2	1	2	2	3
4	1	1	1	1	1	1
5	2	3	3	3	2	1
6	2	3	3	3	2	1
7	2	3	3	3	2	1
8	2	3	3	3	2	1
9	2	2	2	2	2	1
10	2	2	2	2	2	1
11	2	2	2	2	2	1

Table 4.18: Evaluation of the best SML learning task for predicting spot welds locations. Values $[1-3]$ correspond with low to high scores, respectively. Abbreviations: Perf. – performance, Dev – development effort, Comp. – computational cost, Struc. – structural performance, and Appl. – practical applicability.

All models confirmed the hypothesis that ML models can predict joining locations. Hence, ML techniques possess the capability to automate joining element design. Contrary to state-of-the-art approaches, these models predicted locations by considering existing engineering knowledge stored within market-validated data. Thus, they integrate the flexibility of rule-based approaches while considering products as a whole, similar to topology optimization (Section 2.1.5). The voxel-based approach enabled the prediction of joining locations directly in 3D space.

Moreover, models that rely solely on geometry produced meaningful predictions. However, considering nongeometric data (in the current setup) did not improve performance. Moreover, it seemed that the models neglected the branded labels. They needed larger datasets and more computational power for further validation. This led to a variance problem as well as a small geometric scope. Consequently, these boundary conditions prevented the models achieving performances that would support designers.

Regardless of the quality of predicted joining locations, the following section validate a methodology to commonalize them.

4.6 Spatial aggregation on flat surfaces

The aforementioned methodologies concerned the prediction of joining aspects. However, the structure block of the framework also presents commonalization design problems (Section 3.1.3). To this extent, VICTOR evaluates the applicability of AI fields for commonalizing joining locations (Section 3.2.5). Section 3.3.7 presented this methodology.

Commonalization aids in reducing the variability in joining locations when considering multiple product variants. For example, if two product variants consist largely of the same components, there can be similar joining scenarios. These combinations of components can have a joint in the same position. Practically, this joint is the same between the two product variants, although, one of the components it joins may differ between them. Prediction models create joining locations on CRs for one product variant. Hence, these locations may differ between product variants. These locations require commonalization to reduce their variety between product variants. This section describes the validation of the commonalization methodology for joining locations.

Approach

Product development requires methodologies such as SA to run on strategic moments, and thus, after designers have made several new joint designs. Consequently, commonalization is not in the user journey (Fig. 2.5). However, SA aggregates joining element designs as the results of multiple user journeys.

In short, this methodology is not a design approach, and it does not generate new data. However, it does organize designs for improving coping with the effects of product variety. The methodology systematically aggregates joining locations of multiple product variants, creating fewer that can be shared between them. Consequently, the methodology may reduce the joining performance for an individual product variant. However, it increases their manageability for all product variants. As a result, joining location commonalization has two main tasks (Fig. 4.27):

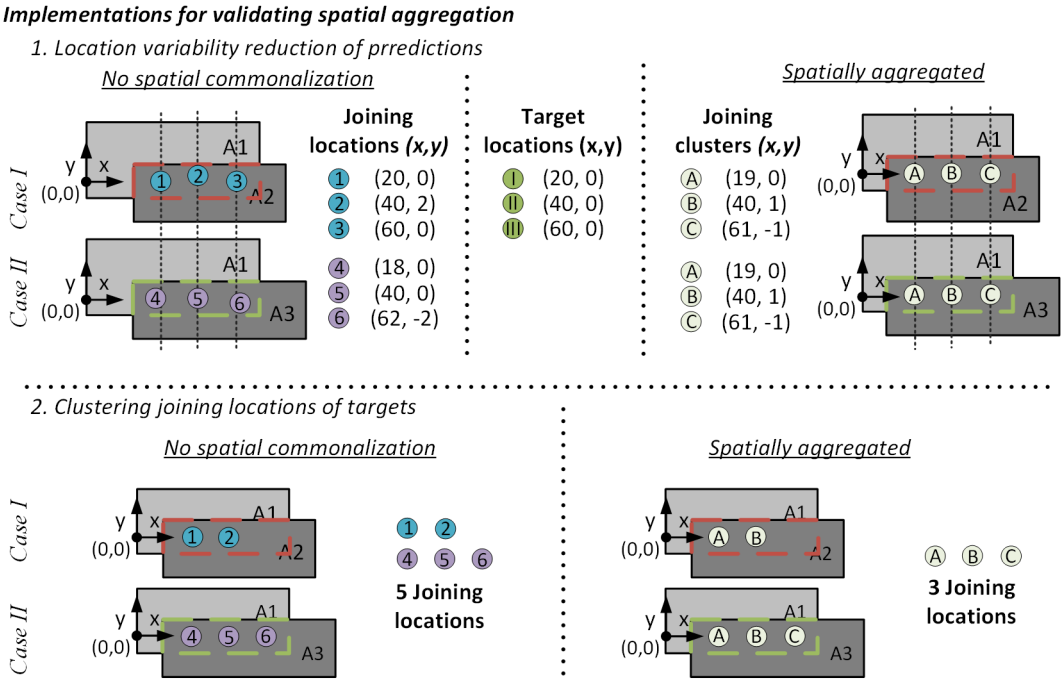


Fig. 4.27: Properties and validation purpose of the two datasets for commonalization.

1. Determining shareable joining *locations* between product variants: The methodology determines the joining *locations* to commonalize for sets of product variants.
2. Reducing variability in joining *locations*: The methodology determines surrogate joining locations for those in close vicinity after considering multiple variants at once.

The input and output of the methodology are joining locations assigned to product variants. However, the overall number of unique joining locations should reduce after the methodology. This reduction occurs due to shared joining locations between product variants. This validation focuses again on spot weld locations. Their number, simple geometry, and use case are the main factors.

The methodology employs a UML method to cluster joining locations. UML enables commonalization due to the geometric freedom in designs. For example, fields such as RBR would require a high development effort to achieve similar results. The K-means clustering algorithm has various properties that serve SA (listed in Section 3.3.7). The cluster center represents the commonalized surrogate joining locations.

The aim of validating this methodology was to determine whether SA is beneficial for modular design. Moreover, because joining location commonalization is a novel method, performing SA on a dataset with real data enables its potential to be determined.

Therefore, similar to the randomized distribution of joining locations (Section 4.3), the SA employed a 2D approach. The largest advantage is that CRs and the resulting UoCR (Fig. 3.33) are straightforward to derive by, for example, using overlapping polygons. A 3D approach would include complex curved surfaces, adding little additional value toward validating the methodology.

Implementation

SA used the same dataset of the 2D approach to distribute spot weld locations (Section 4.3); see Fig. A.1. Fig. A.7 presents the process used to create the 2D data samples (Section A.4.1). However, it required additional processing to determine overlapping CRs. The following list describes these process steps; see also Fig. 4.28. The first three steps explain CRs being found. Finding overlapping CRs is an equivalent process.

1. All flat surfaces on each mesh of all joining components were determined.
2. A pair-wise comparison was conducted between each joining component regarding whether it has parallel flat surfaces that touch another.
3. CRs were created when these coinciding surfaces might hold joining elements.
4. A pair-wise comparison was conducted between all CRs regarding whether they were parallel to and touched one another
5. A union of all pair-wise overlapping CRs was created using transitive closure [453]. Transitive closure found all adjacent CRs. This step links CRs that do not directly overlap, but only through another CR.

However, overlapping CRs do not consider the design characteristics of products. For example, products may be symmetrical, and although their components are mirrored between

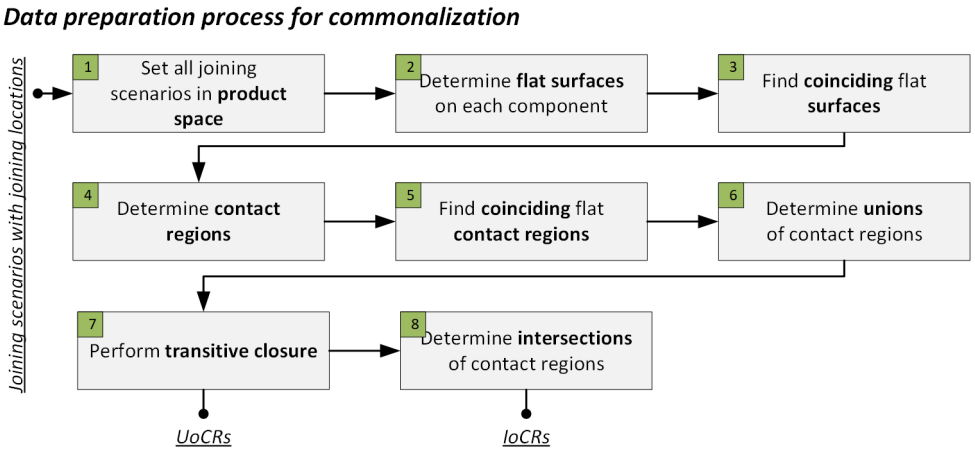


Fig. 4.28: Data preparation process for creating 2D overlapping contact regions.

both sides of a product, their CRs and joining requirements might be similar. Transforming specific joining scenarios into new origins can consider such design characteristics. This process can add CRs to UoCRs, thereby increasing the commonalization potential. The combination of the product's geometric characteristics (e.g., mirroring and rotation), functional requirements (e.g., crash-worthiness and plating), and design parameters (e.g., materials and dimensions) can help to identify joining scenarios for SA. An increasing scope considers more joining elements at once as well as increases the modularization potential. The following three types of scopes can be distinguished [2]; see Fig 4.29:

- The **global scope** transforms (all) UoCRs into one origin (i.e., the global origin of the product). Here, all CRs overlap, creating one UoCR. This scope creates the lowest number of resulting joining elements by affecting the most product variants at once.
- The **local scope** considers the UoCRs in global product space without any transformations. UoCRs remain in place and do not consider product characteristics to commonalize more joining scenarios at once. Joining elements are more optimal in each product variant, but at the cost of additional joining modules, possibly including unnecessary complexity.

  The algorithm runs at a local scope considering all joining elements at their location in the final assembled product (product space). The local scope prevents the transformation of data, which eases the interpretation of results.
- The **domain scope** takes a subset of joining scenarios and transforms them into the same origin using a shared property, such as function, space, or assembly station. For example, a distinction between structural and aesthetic functions lets joining elements fulfill their function correctly. This hybrid between local and global enables increased modularization potential without over-commonalization.

The scope and definition of commonalization tasks (Fig. 4.27) enable the datasets for

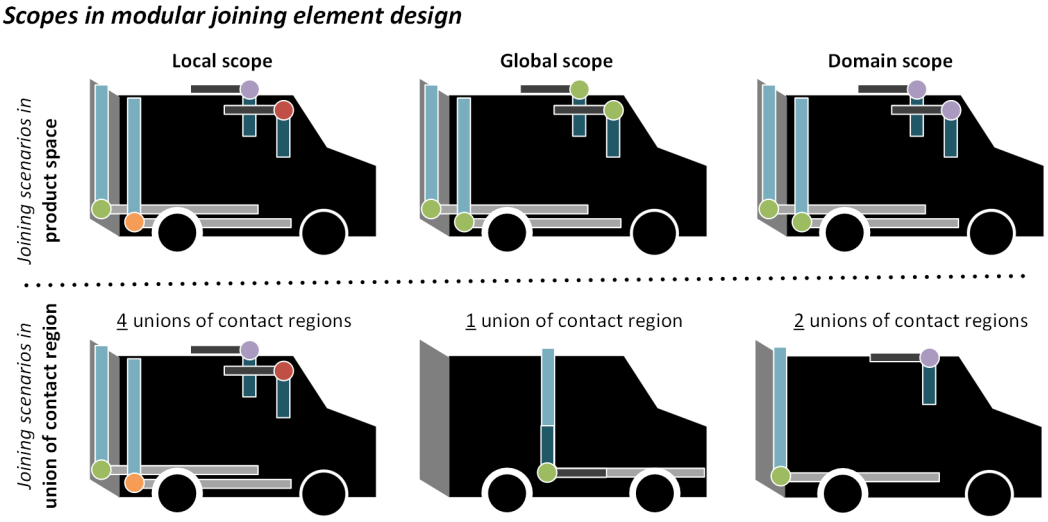


Fig. 4.29: Visualization of the use of scopes to control the degree of commonalization. Colors of joining modules indicate their uniqueness. A local scope does use transformations to increase the number of joining scenarios into unions of contact regions. A global scope would transform everything into one union of contact regions. Lastly, domain scopes aggregate joining scenarios into unions of contact regions, which may have functional, geometric or other similarities.

validation to be determined. Each dataset enables a task to be validated. Table 4.21 summarizes the properties of these datasets.

1. **Determining shareable locations** originates from a dataset with target coordinates (*2D*). The expected result is that the methodology finds all joining locations with the same coordinates in the product space.

   The entire dataset of 2D spot welding joining scenarios has ($m^{\mathrm{2D}} = 4634$) data samples. This implementation focuses on the collection and calculation of shared locations for reusable joining elements. It reconstructs the common joining locations from joining scenarios that have been taken apart in two-component joints. Using the entire dataset provides insights into the potential, degree, and value of commonalizing joining elements.

   The 2D dataset contains 1698 joining scenarios that have 4634 CRs creating 339 UoCRs. Each UoCR has on average $n_{cr}^{\mathrm{UoCR}} = 2.24$ CRs. The number of spot welds per CR in the UoCR is $n_l^{cr} = 3.50$. Joining scenarios have on average 5.88 joining locations.

2. To **reduce location variability**,a dataset with predicted coordinates (*TE-VOX*) is used. The aim is to reduce small perturbations in locations due to various prediction processes.

   An example is the output of the benchmark model (*ID2*) from Section 4.5. Comparing it with the targets provides insight into the reduction of prediction variability through combining multiple predictions.

   A few steps are required to acquire coordinates for 2D data samples from the 3D voxel predictions. First, Eq. 3.18 can transform the predicted coordinates from voxel grids to

global space. Next, the component combination identifiers of predictions are matched to those in the UoCR. Then, 2D data samples are created according to the regular process (Section A.4.1). However, instead of target coordinates, the predicted coordinates are used.

The 899 CCs in the test set of the voxel-based approach come from 360 joining scenarios. Preprocessing identifies $n_{\text{UoCR}} = 30$ UoCRs. Each UoCR has on average $n_{cr}^{\text{UoCR}} = 2.27$ CRs. The number of spot welds perDR in the UoCR is $n_l^{cr} = 2.59$. Joining scenarios have on average 3.84 joining locations.

This dataset uses the test set data samples of the 3D voxel-based approach. The dataset is smaller ($m^{\text{TE-VOX}} = 899$) than the dataset m^{2D} for the previous task. Consequently, the number of UoCRs is significantly reduced. Hence, the number of UoCRs forces a per-sample analysis on the effects of SA.

Besides dataset preparation, the algorithm (Table 4.19) has various parameters for controlling the results. However, this implementation does not use additional weighing parameters for each joining location, such as those proposed through Eq. 3.25. Hence, each location has an equal weight on the resulting outcome. Furthermore, the maximum cluster size s^c is set to $10\ mm$, which prevents large movements of locations and reduces the impact on structural performance.

Process

For each SA task, there is one dataset. The process implements two models with equal settings (Fig. 4.27). The only difference between the models is the dataset:

- The **creation of joining element platforms** (*SA1*) uses the entire dataset of 2D spot welding joining scenarios.
- The **reduction of prediction variability** (*SA2*) evaluates the predicted versus target coordinates, commonalizing joining locations predicted by the voxel-based classifier model *ID2*.

As the input and output of the methodology describe joining locations, this may be confusing. For the sake of clarity, the output and "spatially aggregated joining locations" are referred to as joining clusters hereinafter.

This implementation used the algorithm presented in Table 4.19 on each UoCR. This algorithm describes the implementation of the process flow in Fig. 3.38.

To validate SA's impact, the percent commonality index ($\%CI$) of Siddique is used [236].However, only the term "commonality of connections" ($\%CI^l$) is used; see Eq. 2.6. The component and assembly term are ruled out. SA does not impact these terms, enabling them to be left out of the measurement. Eq. 4.9 presents the calculation of the commonality of connections. The percent commonality of connection $\%CI^l$ is the fraction of the number of joining clusters that occur in more than one joining scenario n^c, where $n_c^{js} > 1$ and the total number of joining clusters n^c.

Input	Superset with joining scenarios $j \in JS$ Sets of joining locations for each set $p \in j$ Minimum number of clusters k_{min}
Output	Set of joining clusters C containing aggregated joining locations p
1:	For $k \in \{k_{min}, \ldots, \lvert JS \rvert\}$
2:	Let set C have k clusters with first k_{min} clusters from p of $\text{argmax}_j \lvert j \rvert \in JS$ and random selected p for all other clusters c
3:	Calculate Euclidean distances $d_{p,c}$ for all $p \in JS$ to all $c \in C$
4:	For each location $p \in JS$
5:	Sort C in ascending order on distances d_p from p to c
6:	For every cluster $c \in C$
7:	If point p within maximum cluster size $d_{p,c} \leq r_{max}$
8:	If no locations of same JS, $c \cap \{p \supset j\} = \emptyset$
9:	Assign p to cluster c
10:	If all locations assigned, $C \supseteq JS$
11:	For each cluster $c \in C$
12:	Update centroid $c := \sum\{a_p^l p \in c\} / \sum\{a_p^l \in c\}$
13:	Iterate between line *3* and line *12* until convergence
14:	If all locations of joining scenario $p \in j$ for $j \in JS$ are okay with standards
15:	Return clusters C

Table 4.19: Spatial aggregation clustering algorithm for one set of joining scenarios from one union of contact regions.

$$\%CI^l = \frac{100 * \text{common connections}}{\text{common + unique connections}} \tag{4.7}$$

$$\%CI^l = \frac{\text{number of joining clusters shared by multiple joining scenarios}}{\text{total number of joining clusters}} \tag{4.8}$$

$$= \frac{n^c \text{ , where } n_c^{js} > 1}{n^c} \tag{4.9}$$

Results

This section presents the results of SA on a predicted and target dataset. Fig. 4.20 lists the results of both models. This subsection addresses the results for each model separately.

- **Results of SA on 2D target coordinates** (*SA1*)

  The algorithm started with $n^l = 2747$ joining locations. SA combined them into $n^c = 1755$ joining clusters (-36.0%). The commonality of connections expressed the following fraction of joining clusters: $\%CI^l = 881/1755 = 25.4\%$.

ID	DS	n^l	n^c	$\%CI^l$	$\bar{s}^l$	$\bar{s}_s$
SA1	2D	2747	1758	49.8%	0.07	0.99
SA2	TE-VOX	174	138	25.4%	0.04	0.27

Table 4.20: Spatial aggregation results on the target dataset (SA1) and the predictions of the voxel-based model *SA2*. Abbreviations: DS – dataset, TE-VOX – test set prediction of model *ID2* in the voxel-based approach, 2D – entire 2D spot weld dataset, n^l – initial number of joining locations, n^c – number of resulting joining clusters, $\%CI^l$ – percent commonality index, $\bar{s}^l$ – mean translated Euclidean distance of joining locations in each cluster, and $\bar{s}_s$ – mean translated Euclidean distance of joining locations in each cluster without clusters containing one location.

The mean translated Euclidean distance of each joining location was $\bar{s}^l = 0.07\ mm$ in each cluster. After excluding all clusters with one joining location, this became $\bar{s}^l = 0.14\ mm$. The mean Euclidean distance of all joining locations translate within a joining scenario to $\bar{s}_s = 0.99\ mm$.

- **Results of SA on predictions** (*SA2*)

  The algorithm started with $n^l = 174$ joining locations. After SA, $n^c = 138$ joining clusters remained. Hence, it reduced 36 spot welds corresponding to 20.7%. The number of shared joining locations (35) over the total number of locations (138) expressed the following commonality of connections: $\%CI^l = 35/138 = 25.4\%$. Table 4.20 lists the results of both validation datasets.

  The mean translated Euclidean distance of each joining location was $\bar{s}^l = 0.04\ mm$ in each cluster. After excluding all clusters with one joining location, this became $\bar{s}^l = 0.15\ mm$. This distance resembled a small movement of points, indicating that the predicted points occurred in almost the same position. It seemed that the ML model found patterns and applied them consistently in its predictions. Further, the average Euclidean distance of all joining locations translated within a joining scenario to $\bar{s}_s = 0.27\ mm$.

Discussion

This subsection first analyzes cherry-picked examples in depth. Notably, the sample images used a different color scheme compared with those in the previous prediction sections (red for component geometry and blue for spot welds). The new color scheme aimed to indicate the difference between design and commonalization tasks. Fig. 4.30 presents three interesting data samples: $UoCR^{(2)}$, $UoCR^{(9)}$, and $UoCR^{(11)}$.

- $UoCR^{(2)}$ consisted of two CRs from the joining scenarios $JS^{(207)}$ and $JS^{(234)}$. The irregularity of prediction can be observed in the overlapping locations on the right-hand side. The left $CR^{(104)}$ has five joining locations while the right $CR^{(130)}$ has four, as the first row of images shows.

  The second row maps the joining locations of the other CR over (white circles with black borders). SA creates the blue circles, which represent the joining cluster that contained one location of both CRs. Here, black points also represent joining clusters; however, they only consist of one joining location.

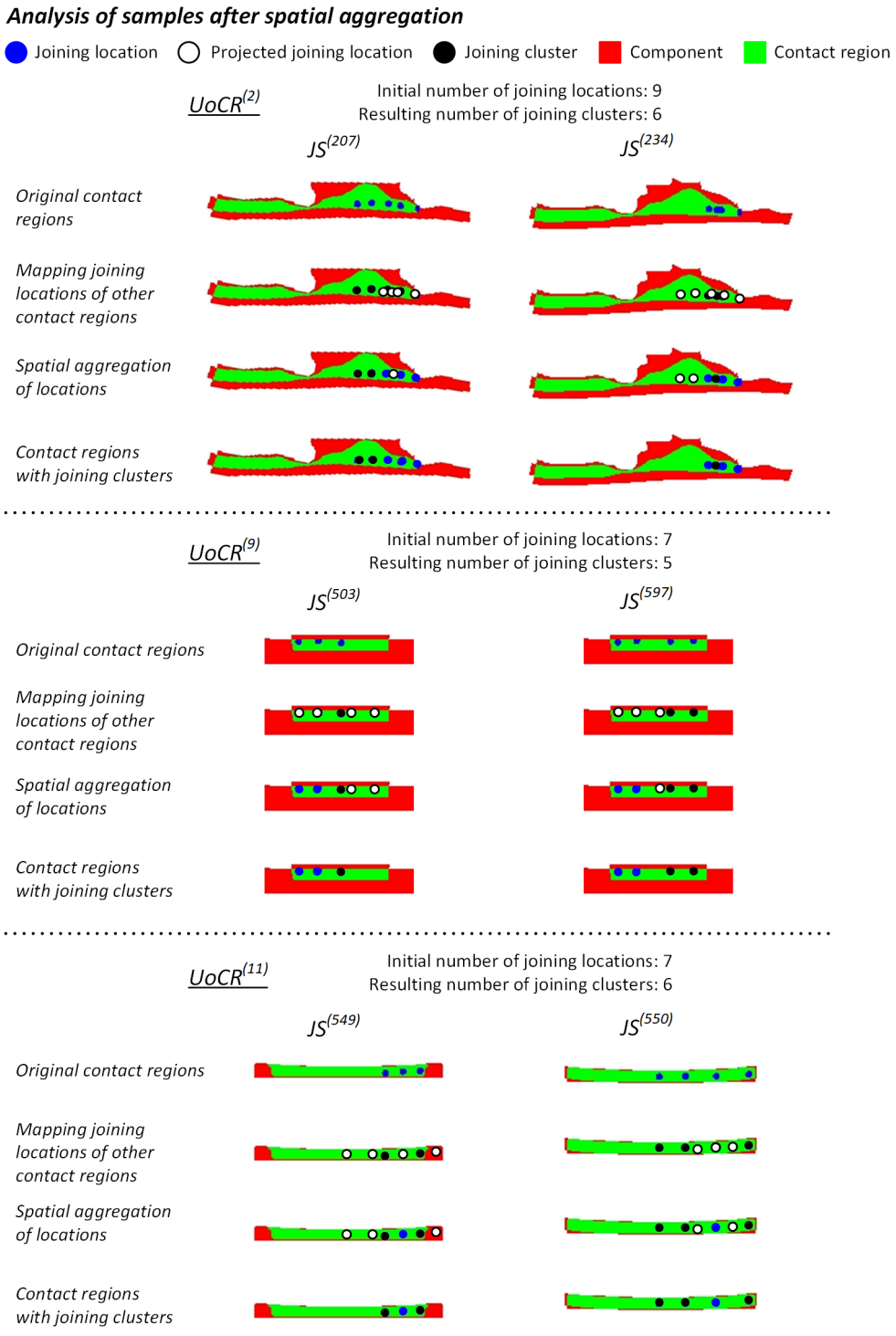


Fig. 4.30: Analysis of spatial aggregation results of the voxel-based dataset.

SA output three joining clusters to share between the two CRs. Additionally, three joining clusters remained specific to each.

- $UoCR^{(9)}$ had two CRs from the joining scenarios $JS^{(503)}$ and $JS^{(597)}$. SA reduced seven initial joining locations to five as it enabled the bottom two points to be shared between the joining scenarios. The ML model already predicted these points on top of one another (i.e., coinciding points) and did not require a new joining cluster coordinate to be found.
- $UoCR^{(11)}$ had two CRs from the joining scenarios $JS^{(549)}$ and $JS^{(550)}$. The algorithm found one shared joining cluster for a joining element in the middle and reduced the number of joining locations from seven to six.

This discussion has neglected the data samples of model *SA2*, which considered the predicted joining elements. The analyzing these samples would not bring additional insights. They were similar in their structure and results to the data samples provided in Fig. 4.30. The results in Table 4.20 indicate that the geometrical movement $\bar{s}^l$ were minimal. Moreover, due to matching 2D flat CRs from the 3D CCs, the samples were incomplete.

The sample analysis of the SA results revealed inconsistencies in overlapping points; see Fig. 4.30. For example, samples $UoCR^{(2)}$ and $UoCR^{(11)}$ exhibited shared joining clusters located in between nonshared joining clusters. The shared joining clusters might have originated from prediction consistencies. However, the goal of SA was to reduce variability and create a meaningful basis for modularization. Future joining modules for the latter $UoCR^{(11)}$ will overlap. The joining elements of one module sit in between joining modules. This effect reduces transparency, as designers need to validate requirements for each product variant. These inconsistencies and results invite designers to rethink their designs. Such considerations would align with Kuhn et al. [433], who stated that complexity analysis can summon transparency, feedback, and adaptability.

Moreover, the commonalization of joining locations may reduce the modularization potential. Again, focusing sample $UoCR^{(11)}$ in Fig. 4.30, plain modularization would result in one module with three spot welds and one module with four. Nevertheless, SA identified a joining cluster in the middle. Modularization could result in three modules with one, two, and three spot welds. Naturally, a task for modularization is to consider this. However, it indicates that the increased commonality of joining locations may negatively affect product variety-induced complexity.

Now, the focus of this discussion shifts from the sample analysis to the overall performance (Table 4.20). The algorithm reduced the number of predicted spot welds by 20.7%. This reduction helped efforts in planning, documentation, and management. This value was lower than for the 2D CRs. However, the smaller dataset had fewer overlapping CRs. For example, the 4634 samples of m^{2D} created 339 UoCRs, whereas there were 30 UoCRs for the 899 samples of $m^{\text{TE-VOX}}$. There is a factor 11 between the resulting UoCRs. Hence, the effectiveness of the methodology was severely impacted by the number of overlapping product variants in the dataset.

Furthermore, the potential of sharing locations for 2D CRs achieved approximately 50%. The algorithm achieved these values with a local scope, leaving all locations on the global

product coordinates. A domain-based scope may find relations in the product architecture that might meaningfully increase the percent commonality index ($\%CI^l$).

An increase in commonality came at the cost of moving joining locations. The models on average did not move joining locations by much ($\bar{s}^l < 1\ mm$). Consequently, adding parameters as specified in Eq. 3.25 would not have influenced the results significantly. Furthermore, the average number of CRs in a UoCR is just over 2. Hence, most joining clusters created by two joining locations are their midpoint. Moreover, joining clusters that contain one joining location have a distance of $s^l = 0\ mm$, which reduces the mean significantly (see the difference between $\bar{s}^l$ and $\bar{s}_s$).

Although the moving distances of joining locations are low, model *SA1* will probably never reach zero. An explanation lies in the design of joining locations. By constructing coordinates on surfaces, designers may set them on sheet metal's top or bottom surface. However, the preprocessing step maps all joining locations on a coinciding plane between the surfaces (Fig. A.7), which creates a translated distance that appears in the results.

The methodology's performance highly depends on the use case. Moreover, the results and their interpretation rely on the dataset. The automotive industry designs vehicles with many curved shapes, limiting 2D overlapping CRs for product variety. A solution may be to ignore overlapping CRs and consider joining locations freely in space. Neglecting CRs reduces functional and geometric constraints but increases the potential for commonalization. The methodology becomes prone to over-commonalization, thus potentially reducing the overall modularity.

Evaluation

This evaluation does not include a table as the evaluations of the prediction methodologies did. Both models represent the task of joining location commonalization, not the dataset or the model. Model *SA1* focused on creating shareable joining locations between product variants, whereas model *SA2* focused on reducing variability in joining locations after prediction. There was no benchmark available for comparing their results. However, this evaluation addresses the validation criteria in the text.

- First, the *performance* was difficult to validate. The unexpected large dependency on the dataset suggested that the methodology needs more testing. For example, it could be tested multiple times during a product development trajectory.
- Hence, its *applicability in industry* is debatable.
- In any case, the methodology found joining locations and made joining clusters effectively. The distance of moved joining locations after clustering was low. Hence, it was assumed that the *structural performance* was not greatly impacted.
- Furthermore, SA is a relatively *quick* algorithm. The number of joining locations in a UoCR was relatively low for clustering algorithms.
- However, the *developmental effort* is high. Determining all CRs between all relevant components in overall product variants is complex to implement. Moreover, regarding geometry, it has a high *computational cost*.

In short, SA can support designers in commonalizing joining elements. The methodology converges the variability in predictions and enables the sharing of joining locations between product variants. However, the methodology is highly dependent on the use case. Hence, it only brings value when performed at strategic moments during product development.

As a logical step after commonalizing joining locations, the following section validates a methodology to commonalize entire joints.

4.7 Element densification using flat surfaces

The previous section discussed the commonalization of joining locations, which concerned a single joining aspect. This section discusses the commonalization of joints. The framework defines this design problem to consider overall joining aspects at once (Section 3.1.3). It reduces the variety in joining elements. VICTOR determines the most applicable AI fields to be deterministic (Section 3.2.6), such as RBR. A condition-based implementation can help designers to control and understand the adaptations to joining designs by the methodology.

Section 3.3.8 presents the methodology. In short, ED aims to increase sharing of the joining elements. These joining elements are designed on overlapping CRs, which describe a geometric boundary for joining elements (see Fig. 3.33). Hence, when interchanging a component between product variants, the overlapping CR remains. Consequently, it may be beneficial to share the joining elements on these overlapping CRs. This section describes the validation of commonalizing joints. The aim is to determine the functioning of this methodology, its potential, and its limitations.

Approach

Whereas SA in the previous section (Section 4.6) aimed to commonalize joining locations, ED considers entire joints, although both methodologies have a similar use case. Their application is required on strategic moments in product development after many new joining designs. Prediction methodologies are divergent in designs. Commonalization methodologies are convergent, meaning they run from the bottom up.

ED is a logical step to perform after commonalizing individual joining aspects. By reducing variety in technologies, locations, and parameters, less variety remains to be considered on the level of joints. For example, performing ED after SA prevents the densification of two joining locations with near-zero distance as they lack identification as one element. In other words, the densification algorithm must consider the manufacturing requirements of all joining aspects. Moreover, the algorithm might need to perform similar tasks to achieve the same results. Similar to SA, the methodology identifies tasks for the commonalization of joints through ED through the following actions:

1. The determination of shareable joining *elements* between product variants: The methodology determines the joining *elements* to commonalize for sets of product variants.
2. The reduction of variability in *joints*: The methodology determines surrogate joints for shared overlapping CRs by considering multiple variants.

The input and output of the methodology are joining elements. However, the overall number of unique joining elements should be reduced after the methodology is applied. This reduction occurs due to shared joining elements between product variants. This validation focuses again on spot welded joints. Their number, simple geometry, and use case are the main factors. Picking one joining technology reduces the many manufacturability requirements that need consideration in the algorithm. Additionally, the high product variety and use of spot welds in the use case support this scope. The aim of validating this methodology was to determine whether ED can reduce variety. Hence, a dataset with real data can determine the benefits of this methodology.

The methodology employs an RBR approach. This enables the systematic substitution of joining elements on joints, such as for unifying their designs on a shared overlapping CR (also defined as the IoCR). Condition-based approaches can enable designers to make final decisions. However, it is the traceability and certainty of such approaches that support designers. Substituting joining elements has a large influence on the product, not only in terms of structural performance but also of manufacturing capabilities and resources.

Implementation

ED was run on the same flat CR joined by a spot welding raw dataset as SA. However, instead of the UoCR, this methodology implemented the IoCR; see Fig. 3.33. The IoCR describes the shared CR over a unique set of product variants. Commonalizing joining elements between IoCRs enables them to share the assigned variants.

As ED performed similar tasks to SA, it used the same datasets. The implementation subsection in Section 4.6 explained each dataset, and Table 4.21 below summarizes them.

DS	m	n_s	$n_{\mathbf{UoCR}}$	n_{cr}^{UoCR}	n_l^{cr}	n_l^s
2D	4634	1698	339	2.24	3.50	5.88
TE-VOX	899	360	30	2.27	2.59	3.84

Table 4.21: Properties of datasets for element densification. Abbreviations: DS – dataset. TE-VOX – test set voxel-based approach, 2D – entire 2D spot weld dataset, m – number of samples, n_s – number of joining scenarios, n_{UoCR} – number of unions of contact regions, n_{cr}^{UoCR} – mean number of contact regions for each UoCR, n_l^{cr} – mean number of joining elements for each contact region, and n_l^s – mean number of joining locations for each scenario.

However, the datasets did not validate each task separately. Both datasets jointly enabled the evaluation of the ED algorithm.

1. To **determine shareable elements**, a dataset with target coordinates was required. The expected result was that all joining elements that can fit other joining scenarios without violating manufacturing requirements would be found.
2. To **reduce variety in joints**, a dataset with predicted coordinates was required, which was small. Hence, it provided insight into performing ED early in product design. At this level of product maturity, many joining elements have not yet been designed. Moreover, the prediction model created joining locations on CCs. Due to similarity filtering, among other reasons, joining scenarios may not contain complete joints. ED

may complete these joints, such as when other predictions coincidentally contain joining elements for overlapping CRs.

Besides dataset preparation, the algorithm (Table 4.22) entails a trade-off for controlling the results. Substituting joining elements into other joints is beneficial for companies. ED reduces the complexity due to an increased modularization potential. However, this needs to make up for the increase in manufacturing time and cost of producing more joining elements; see Eq. 3.31. The methodology proposes the calculation of the difference in assembly complexity as a proxy for the trade-off. However, the main goal of validating this methodology was to determine its functioning and behavior. The implementation of the trade-off was neglected for simplicity reasons and the focus remained on the main goal. Hence, the validation considered all densifying tasks (each trade-off) as beneficial.

Process

The algorithm iterated through all intersections in a UoCR. It determined intersections of CRs that contained unequal numbers of joining elements between the affected joining scenarios. For such cases, ED added joining elements to those with fewer elements. However, it added them only when the new joining elements did not violate one or two manufacturability requirements: edge $< 5\ mm$ and mutual distance $< 20\ mm$.

Notably, the addition of joining elements is a simplification of the methodology presented in Section 3.3.8. The methodology originally presented the substitution of joining elements. Substitution takes all joining elements of one joining scenario (within the IoCR) and puts them in another, while discarding the ones that were in it. Addition seeks available space on a CR for setting a joining location from one on the other. This change in functioning was due to simplification in the implementation.

The algorithm was run after SA. The addition of joining elements became simpler after shareable joining locations were identified, which reduced variability and supported the process of finding shareable joining elements. The validation considered two models – one for each dataset. The process implemented the following two models with equal settings:

- **Densification on target spot welds** (*ED1*), used the entire dataset of 2D spot welding joining scenarios (*2D*);
- **Densification on predicted spot welds** (*ED2*), which commonalized locations from the dataset (*TE-VOX*) predicted by the voxel-based classifier model *ID2* from Section 4.5.

This implementation used the algorithm presented in Table 4.22 on each UoCR. This algorithm describes the implementation of the process flow in Fig. 3.40.

The validation of ED's impact employed the percent commonality index ($\%CI$) of Siddique [236]. Similar to SA, it only used the term "commonality of connections" ($\%CI^l$); see Eq. 4.9.

Results

Table 4.23 presents the results of the implementation as well as some results of SA for comparison. This enables a discussion of the effectiveness of ED.

Input	Set of joining scenarios JS Sets of joining clusters $JC \in JS$ Sets of joining locations $JL \in JC$
Output	Set of joining clusters C for each joining scenario $JC \in JS$
1:	Sort JC on number of JL in descending order
2:	Sort JS by containing JC with most JL; then most JC on JS
3:	For each intersection of contact regions $iocr \in IoCRs$
4:	For all equal joining technologies u grouped in $iocr$
5:	For each joining scenario $s_A \in iocr$
6:	Get JCs laying on $iocr$ of joining scenario s
7:	Number of JCs n_{jc} in $s_A \in iocr$ is highest $\text{argmax}_{s_A}\{n^s_{jc}\}$
8:	For JCs on $jc \in s_A$ are manufacturable on other JS s_B
9:	If densifying trade-off (Eq. 3.31) is profitable
10:	Add jc to joining scenario s_B

Table 4.22: Joining element densification algorithm for one set of joining scenarios from one union of contact region.

		SA				***ED***			
ID	**DS**	n^l	n^c	n^c_{shared}	$\%CI^l$	n^l	n^c	n^c_{shared}	$\%CI^l$
ED1	2D	2747	1758	875	49.8%	2788	1758	899	51.3%
ED2	TE-VOX	174	138	35	25.4%	182	138	42	30.4%

Table 4.23: Element densification results on test set predictions of voxel-based Model *ID2* and the target dataset. Abbreviations: DS – dataset, TE-VOX – test set voxel-based approach, 2D – entire 2D spot weld dataset, SA – spatial aggregation, ED – element densification, n^l – initial number of joining locations, n^c – number of resulting joining clusters, n^c_{shared} – number of shared joining clusters over multiple scenarios, and $\%CI^l$ – percent commonality index.

- **Results of ED on a 2D dataset** (*ED1*).

  The algorithm was run on 18 of 339 UoCRs (5.3%). The percent commonality index of connections ($\%CI^l$) increased after SA was run (49.8% vs. 51.3%). The samples in the 2D dataset were consistent and the result of successfully marketed products. Hence, designers already considered product variety. Consequently, ED added $n^l_{\text{ED}} - n^l_{\text{SA}} = 41$ new joining locations to joining scenarios.

  Furthermore (not in table), joining scenarios shared $n^c_{\text{sharedED}} - n^c_{\text{sharedSA}} = 24$ new joining clusters that otherwise represented one joining location. In other words, 24 joining clusters from SA that contained only one joining location now reoccurred in multiple joining scenarios.

- **Results of ED on the predicted dataset** (*ED2*)

  ED was run on three UoCRs of the voxel-based predictions corresponding to 10% (refer to Table 4.20). Furthermore, the total created joining locations increased from 174 to 182. However, they were still managed by 138 joining clusters as the algorithm did not create new clusters. ED added eight joining clusters to new joining scenarios. Furthermore,

seven joining clusters that represented one joining location after SA now represented two or more. The percent commonality index of connections increased to $\%CI^l = 30.4\%$.

Discussion

This subsection first analyzes cherry-picked examples in depth. Fig. 4.31 presents three interesting data samples from model *ED2*: $UoCR^{(126)}$, $UoCR^{(266)}$, and $UoCR^{(292)}$.

All cases indicated that SA organized the spot welds due to the overlapping points. The images depict *un*densified CRs of scenarios on the left-hand side and densified on the right-hand side. The middle blue image represents all IoCRs generated by the algorithm.

- Sample $UoCR^{(126)}$ had two contact regions that overlapped almost completely. The joining scenario $JS^{(2369)}$ received one additional joining element. The CR seemed to be designed to have a joining location there. The $CR^{(2369)}$ most likely switched out a spot weld for, for example, a rivet, which was filtered during dataset generation. This might also explain various other asymmetrical samples visualized throughout the report. Nevertheless, ED took the opportunity to reuse a joining cluster of $CR^{(1237)}$ to stitch the components back together.
- Sample $UoCR^{(266)}$ had four highly similar joining scenarios. The CRs exhibited minimal differences, as indicated by the orange circles on the left-hand, undensified side. The high similarity originated from components with newer versions used in more recent car lines. Components may undergo non-joining-related changes, promoting the reuse of previously designed joining elements. ED added the middle spot weld to three other CRs and, as a result, all CRs had 13 spot welds.
- Sample $UoCR^{(292)}$ had six varying joining scenarios. Here, $CR^{(4685)}$ was similar to $CR^{(5067)}$ of sample $UoCR^{266}$. The local scope of preprocessing prevented these overlaps being found. ED processed CRs on their location in the final product, which may in this case be the left and right sides of the product. The six CRs aligned vertically as an exploded view of all intersections of CRs. This sample visualizes the complexity of results for SA. Joining scenarios shared clusters in varying components, sizes, and distributions. ED added four joining locations in total, as indicated by the orange circles on the right-hand side.

Similar to the SA model *SA2*, this discussion excludes the data samples of model *ED2* that considered the predicted joining elements. The analysis of these samples would not bring any new insights. Moreover, the results were incomplete due to the fewer number of spot welds for each joining scenario.

Now that individual data samples have been discussed, some generic remarks should be made. The execution of ED was not frequent. The large 2D dataset with 4634 sample CRs had 18 overlapping combinations to densify. Furthermore, the purpose of ED is to set up joining elements for modularization, which promotes the sharing of joining clusters. The results demonstrated this in the number of joining locations n^l increasing by eight and 41 for *ED1* and *ED2*, respectively. These numbers could increase as individual product variants may have CRs with more joining elements. However, it also requires enough space for adding joining

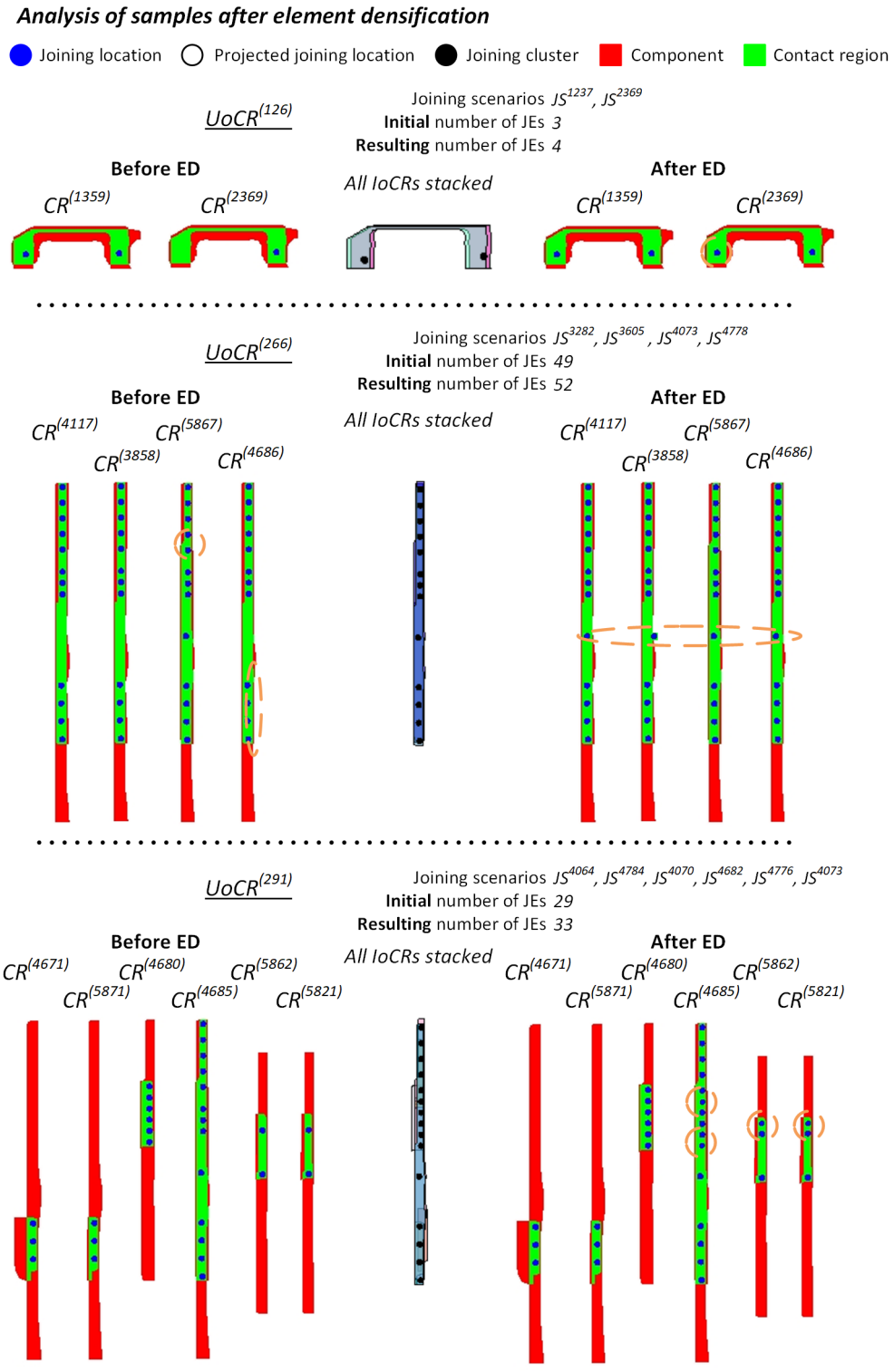


Fig. 4.31: Analysis of element densification results for the 2D dataset.

elements. Furthermore, the percent commonality index $\%CI^l$ increased as product variants shared the joining clusters multiple times. This effect increased $\%CI^l$ by 5.0% and 1.1% for *ED1* and *ED2*, respectively.

Nevertheless, the low frequency of the algorithm's execution indicated stringent boundary conditions. The execution frequency might become even lower after implementing the assembly complexity trade-off. Furthermore, the IoCRs are relatively small surface areas that bring joining scenarios together. A solution might be to implement a domain scope. Changing scope would enable the integration of more CRs into the IoCR. However, this scope may lead to a limited increase in commonality. This theoretical limitation is due to the IoCR. The intersections split CRs, and hence, the unique shareable CRs remain small.

Another solution for increasing the algorithm's effectiveness is to implement a substituting task. Currently, geometric boundaries limit the addition of joining elements. However, the mutual distances between joining elements also require consideration. For example, two joining elements might be positioned in the middle of a small CR. A third joining element would only fit after the two others move to make space. The addition of joining elements would not make it possible to cope with this situation. However, substitution would remove the two joining elements and replace them with three from a different joining scenario.

Furthermore, ED seemed forgiving on forgotten joining elements. The filtering on spot welds induced a missing joining location in $UoCR^{(126)}$. The algorithm interpreted this as a chance to densify. This may aid designers in creating robust joining modules during the generation of multiple product variants with slightly changing requirements. Moreover, $UoCR^{(266)}$ exhibited the unintended capability of the algorithm in design reuse. Updates of versions or new component combinations using the same CR may inherit designed joining elements.

Interestingly, this RBR algorithm picks and places available joining elements from the database on new joining scenarios. The method is called algorithmic fitting in the framework in Fig. 3.8. However, this methodology would require a tailored selection procedure for joining elements to be picked and placed. Sample $UoCR^{(291)}$ indicated that large CRs received all joining elements; see $CR^{(4685)}$. A consequence of high product variety may be new widely overpopulated joining scenarios. The prediction of joining features and the number of joining locations as SML approaches may aid this system.

Evaluation

Similar to SA, this evaluation subsection does not include a table as those for the prediction methodologies did. The models differ in terms of dataset input. Hence, the reflection is presented in writing and employs the validation criteria mentioned at the beginning of this chapter.

- The *performance* was unsatisfying. The execution frequency was too low due to the intersections of CRs. Furthermore, the performance was difficult to validate. The unexpected large dependency on the dataset indicated that the methodology required more testing and validation after modularization.

- Additionally, the *structural performance* was not greatly impacted due to the few joining elements that were added.
- Besides, ED itself is a *quick* algorithm. Thus, it was relatively straightforward to add joining elements.
- However, the data preparation was *computationally* and *developmentally costly*. It was essential to determine all CRs between all relevant components for overall product variants. This cost increases when implementations are considered in 3D space. Moreover, including a trade-off that calculates assembly complexity might increase the running time, developmental effort, and computational cost as well.
- Consequently, the *applicability in industry* is debatable. Regardless, during densification, the algorithm worked as intended.

ED identified and applied joining elements on joining scenarios to increase the commonalization of joints. The algorithm required SA to clean the variability in joining locations. However, the algorithm did not increase commonality significantly. The geometrical constraint of the IoCR prevented it. Nevertheless, with a few adaptations, ED seems to be an approach that could work for algorithmic fitting. It could pick and place joining elements from databases into a new joining scenario.

The previous sections validate various methodologies. Although, all evaluations remained within the scope of the experiment. The following sections summarizes the results and compares them with the state-of-the-art methodologies.

4.8 Assessment of the methodologies

This section summarizes the results of this validation chapter. For each methodology, an evaluation of various models has been presented. However, these evaluations now need to be placed in perspective. Hence, this section addresses how the methodologies relate to the use case. It also discusses the framework. In doing so, it summarizes the results of the chapter. First, Table 4.24 lists the validated methodologies according to the criteria in the literature overview (Section 2.4). The list at the end of this section summarizes the assessment of the validated methodologies.

The framework presented in Chapter 3 distinguished several design problems in joining element design. For each problem, the framework evaluated various AI fields to support automated joining element design (Section 3.2). Using these AI fields and research gaps (identified in Chapter 2), it presented novel AI methodologies for automating joining element design (Section 3.3).

This chapter has validated the methodologies for four design problems (Fig. 4.1). The main goal has been to reduce time-consumption and increase the quality of designs. The framework identified SML as an underused technique for achieving these goals. Additionally, SML may rectify current limitations, such as the considerations of successful designs, products as a whole, and other product variants.

After the data analysis (Appendix A.2), the first methodology used decision trees to predict joining technologies (Section 4.1). The best model achieved a satisfactory

Methodology	CL	AI	DE	DI	O	S	G	H	CO	AD	3D	K
Joining technology prediction	JT;JP	SML	N	I	3	F	G	4	2	4	Y	Y
Algorithmic fitting	JL	S&O	N	D	2	D	I	3	3	2	Y	Y
Grid-based drawing	JL	SML	N	I	3	F	G	3	4	3	Y	Y
Spatial aggregation	PP	S&O	Y	D	2	F	I	2	2	2	Y	N
Element densification	PF	RBR	Y	D	2	F	I	2	2	2	Y	N

Table 4.24: Assessment overview of novel validated methodologies. Column abbreviations: reference (R), class (CL), main AI-field (AI), deterministic (DET), deductive / inductive (DI), optimality (O), scope (S), genericity (G), holicity (H), computational cost (CO), automation degree (AD), geometry (3D), and knowledge-based (K). Section 2.4 explains the properties and values in detail.

performance, which confirmed that SML can be employed as a recommender system as an alternative to multidisciplinary optimization methodologies. Moreover, the random forest model required a few features for predicting joining technologies. However, all decision trees required heavy simplification of the data. Hence, their application would make more sense in earlier product design phases. Detailed optimization methodologies enable higher-quality predictions in later phases.

The next validated design problem was joining location prediction (Section 3.2.2). The framework identified various approaches for using AI, of which this chapter has validated two, namely algorithmic fitting (Sections 4.2 and 4.3) and grid-based drawing (Sections 4.4 and 4.5). Algorithmic fitting used an evolutionary S&O approach to distribute joining locations.

Section 4.3) validated randomized distribution, which optimized spot weld locations considering edge and mutual distances. The algorithm is lightweight and straightforward, making it useful in early design phases. However, the results were debatable. Most predictions seemed meaningful; for example, the current implementation had the best predictions for small and long CRs, which required spot welds in a straight line. Consequently, it did not necessarily improve the benchmark of RBR methodologies.

Additionally, algorithmic fitting requires a prediction method for the number of joining locations to distribute. The prediction of the number of spot welds (Section 4.2) used decision trees as an SML technique. A random forest method required all features to classify the numbers; however, the data lacked detailed geometric considerations, thus limiting the performance. Combining the prediction of the number and distribution of spot weld locations stacked the uncertainties of both methodologies, creating an unsatisfactory algorithmic fitting approach. Additionally, the methodologies considered one joining technology, requiring high developmental effort for generalization. This approach requires more research to overcome the benchmark joining location prediction methodologies. In short, algorithmic fitting is a simple approach, but its results lack structural performance and knowledge-based considerations.

Grid-based drawing uses SML to exploit patterns in successfully marketed designs. It can reuse joining designs but with different prediction tasks. Section 4.4 presented the implementation of image segmentation as a technique for classifying spot weld locations. Furthermore, it validated a probability mapping technique as a regression task. Both of the SML implementations could predict joining locations meaningfully. Classification provided

more robust and accurate results, probably due to an increased understanding of spatial dependencies. However, these models only considered geometry.

Therefore, the integration of nongeometric data in a computer vision-based classification model was validated (Section 4.5). The branding technique was used to implement DR and clustering to create labels of nongeometric data. However, the models did not recognize the additional information of the branded labels. Hence, the geometry-only classification model was the most promising. Regardless, this methodology is complex and has many variables. For example, the product variety in the dataset caused problems with data leakage. The advantage of a large dataset due to variety was negatively affected by its ambiguity and similarities. As a result, much research is still required to overcome the variance and scaling problems.

Besides prediction, this chapter has validated two novel commonalization methodologies of joining locations (Section 4.6) and joints (Section 4.7). They aimed to cope with the negative effects of product variety through increasing commonality in joining element design. Additionally, they prepared joining element designs for modularization.

First, SA aimed to reduce the geometric variability of joining location predictions. Additionally, it determined shareable joining locations between product variants. SA used the unsupervised learning technique K-means to cluster spot weld locations on overlapping CRs. This methodology led to promising results. It was relatively quick and effective at solving both tasks. However, this highly depends on the size of the dataset. Thus, the algorithm must be used on strategic moments during product development.

Second, ED employed RBR to determine shareable joining elements. This enabled the reduction of the variety of joints. The methodology added joining elements to shared overlapping CRs between variants, which it did quickly and effectively. However, the overlapping CRs created a constraint on the algorithm's execution. These boundary conditions occurred rarely, making it difficult to evaluate the performance and harvest its benefits. Additionally, the dataset must allow for such a methodology to work. However, an unexpected result was found, namely that ED can lend itself to a rule-based prediction methodology. Theoretically, it can reuse joining elements by picking and placing them into new joining scenarios.

To summarize the aforementioned assessment, the lists in Appendix A.7 concludes the validation of the methodologies with their pros and cons. To conclude this validation chapter, the framework identified promising AI fields for automated joining element design. Additionally, it proposed novel methodologies using these fields. This chapter has validated several methodologies, several of which had promising results. After additional research and validation, they may be found to be alternatives to benchmark approaches. They addressed the main problems in joining element design, namely time-consumption and suboptimal results, for specific use cases. For example, they reduce the time required by either optimizing the prediction speed with fast algorithms (e.g., randomized distribution) or by reducing the number of design iterations. Additionally, commonalization methodologies can determine shareable joining elements, automating the modular design necessities.

After summarizing the assessment of validated methodologies, the next chapter concludes the dissertation study, provides an overview of results, and presents an outlook for further research.

Chapter 5

Conclusion and outlook

Joining element design in the manufacturing industry is a complex and multi-disciplinary field [15, 116]. Today, customer demands are diversifying and companies must cope with the trend of increasing product variety [6]. As a result, the sheer number of joining elements in one product variant and the relationships between them are causing the complexity of their design to surpass human capabilities [6, 31, 59]. This creates the need for joining element design to be automated, which will accelerate development processes, reduce unnecessary design iterations, and optimize design quality. Automation can relieve designers of cumbersome and repetitive tasks, freeing them to concentrate on more creative and holistic problems [101].

This chapter has five sections. Section 5.1 answers the research questions regarding how to implement AI for designing joining elements while considering product variety. Next, Section 5.2 addresses the results that provide support for these answers. Then, Section 5.3 lists the contributions of this thesis. Furthermore, Section 5.4 provides recommendations on potential future research directions. Lastly, Section 5.5 presents a closing summary on this dissertation study.

5.1 Research questions and objectives

This section discusses the research questions and objectives of this dissertation. It relates the meaning and implications of the results to each question. This section starts with the subquestions, before answering the central research question. The first subquestion was as follows:

How are products developed and how does this impact joining element design?

Product development is a continuous endeavor [11]. Companies already begin the process of creating new and updated versions of products when they are developing and producing their predecessors. Moreover, companies that manufacture many different variants of the same product may have multiple parallel development processes [13]. To cope with the tremendous effort and complexity in development required to offer a high variety of products, companies may develop variants sequentially [135]. Their development starts with a few base variants, after which each added variant emerges from small additional modifications [58]. Hence, the

continual development of new products combined with the sequential design of product variants create a highly complex development. Companies can outsource tasks to optimize their internal processes [8]. Outsourcing enables them to focus on their core activities, through which they can add the most value to their products [8].

As new product variants are created sequentially and continuously, so too are the designs of their joining elements. Joining elements depend on the components they join; hence, they depend on the selected components in each product variant. For example, an interchanging component between two product variants might require a change of the joining technologies that would otherwise join the affected components.

Moreover, joining element design plays roles in the design, engineering, and manufacturing phases of development [116]. They are multidisciplinary tasks with numerous and varied stakeholders. Consequently, creating quality joining element designs becomes both difficult and complex [454], especially for companies with high product variety.

Historically, joining element design was a manual task on 2D drawings. Today, highly sophisticated, powerful 3D CAD software supports the creation of benchmark designs; however, designers still use workflows based on 2D methodologies [26]. They work with traditional experience and trial-and-error approaches that are prone to causing unnecessary design iterations and costly rework [18]. There are a (limited) number of solutions available for automating the design of joining elements. However, these solutions do not consider the requirements for designing products with high variety. As a result, major problems exist in joining element design concerning both the time-consuming nature of the design process and the resulting quality of designs [1].

- **Time-consumption**: The current design of joining elements is unnecessarily time-consuming. A major reason is the lack of automation in their design. Designing joining elements manually may entail highly cumbersome and repetitive work. Additionally, joining element designs require validation for every product variant regarding, for example, their manufacturability. These validations becomes highly complex for products with high variety. Consequently, faulty designs will require rework, implying an additional flow through the design process.
- **Suboptimal solutions**: Joining element design is multidisciplinary and becomes more difficult due to new and changing customer demands, such as higher performance requirements and new technological innovations. These circumstances constrain designers from finding global optima. Furthermore, the solution space becomes too large to find the best possible designs. Moreover, as these tasks can be outsourced, designers who may not have insight into the product as a whole will experience even more difficulty in creating quality designs.

After discussing how products are developed and how it impacted joining element design, the second subquestion was as follows:

How can joining element design be automated?

Today, several tools and methodologies support joining element design. Examples of such methodologies include multidisciplinary decision-making for selecting the best joining

technology and topology optimization for determining joining locations. However, these state-of-the-art methodologies do not consider previous successful designs nor the designs of other product variants. The methodologies treat joining elements as individual design problems and tend to neglect viewing joints from the perspective of the products as a whole. Therefore, their results are joining designs that are optimized for individual product variants. However, these designs may be not optimal when considering the entire product. The collection of all joining element designs will become unnecessarily diverse and complex over time. This variety may become a cause for delays, quality issues, and costly rework throughout product development [6]. The consideration of other successful designs, such as by using a knowledge-base, would enable the generation of more complete designs that could potentially reduce the aforementioned disadvantages of variety.

To tackle these challenges from the state of the art in automating joining element design, this study adopted a broad perspective on AI. Currently, there are various popular AI fields in the manufacturing industry [101], categorized by RBR, CBR, S&O, and ML. Organizing the state-of-the-art methodologies indicated that joining element design lacks the use of ML. Saliently, ML exploits historical data to predict new solutions. It can find patterns and apply them to solve unseen problems [275]. Moreover, ML considers a whole large dataset at once. Hence, it may automatically have a more holistic view of the product compared with current methodologies. These properties solve the need to consider both successful designs and holistic considerations in joining element design.

To simplify the implementation of AI, distinguishing different aspects of joining element design is of great help. Each joining element has information about its technology, geometry, and parameters. The joining technology describes the type of manufacturing process, such as clinching, riveting, or spot welding. The joining geometry represents the shape and location, such as a point for setting a spot weld or a line for an adhesive bond. Lastly, the joining parameters entail the necessary information for manufacturing, including the diameter of a clinching point or the type of glue for adhesive bonding. Consequently, different AI techniques suit the automation of different joining aspects. The applicability of each AI technique depends on the availability, structure, and formatting of the data.

SML, as an AI technique, can handle the task of classifying joining technology, joining parameters, and the prediction of the number of joining locations. The methods validated in this study include the use of decision trees and NNs. Their implementations can remain relatively straightforward and simple. For example, it might only need a few key features to predict the joining technology with a sufficient quality for early design phases. Moreover, these approaches may theoretically also predict the joining parameters as their design tasks are quite similar.

However, automating the design of joining locations, also referred to as joining location prediction, is the most difficult task compared with designing the other joining aspects. It requires the consideration of both geometric and nongeometric data for generating coherent designs, making this a multimodal prediction task. MMML requires much effort to implement successfully, even with current technologies. Therefore, this dissertation has presented three approaches with different levels of cognitive automation to predict joining locations. Cognitive automation refers to the number and complexity of tasks that the approaches take over from designers. The core of these approaches is the prediction of geometrical features.

The inclusion of nongeometric data is optional for potentially increasing the quality of their predictions.

- **Algorithmic fitting**: This approach aims to position a given number and size of joining elements systematically on CRs. First, it must predict one or several features, such as the number or length of joining locations. This prediction can be classified by an SML model. Then, the algorithm can distribute joining locations according to the predicted number. Furthermore, the algorithm may implement a pick-and-place approach for fitting already designed joining elements from a database.

  This study implemented an evolutionary optimization algorithm that distributes the locations by equalizing their mutual and edge distances. The algorithm produced promising results even for a randomized implementation. However, the validation also revealed that the prediction performance fell off rapidly for nontrivial design problems with high solution spaces, such as CRs with large surface areas.

- **Grid-based drawing**: Due to advances in computer vision techniques, ML models can understand and transform imagery. These models can predict the regions that represent joining locations in grid-like (Euclidean) structures, such as images or voxel grids. Practically, these models draw joining locations on CRs, either in 2D (pixels) or 3D space (voxels). In particular, NNs with architectures that can consider spatial coherence are applicable for such tasks, including CNNs and GANs. These NNs can perform both a classification or regression task. However, the classification implementations, through image segmentation techniques for example, seemed more robust and promising than the regression approaches. They were able to reconstruct geometry more consistently from the input samples. As a result, they seemed to have a better understanding of the relationship between geometry and joining locations, and to that extent, of parts and joining elements.

  Considering nongeometric data in Euclidean supervised models requires additional processes. ML models require consistent, discretized data structures that are not provided by the multimodality of geometric and nongeometric data. However, an MMML approach, through early fusion for example, can integrate nongeometric data with geometry. The study explored using an approach also known as branding, which adds labels to images. Each label may represent a cluster of nongeometric data that, for example, describes reinforcement parts or types of material. The implementations included various expressions of nongeometric data. However, unfortunately, models did not seem to pick up on the additional labeled information, partly due to large DRs causing a loss of information and a suboptimal dataset for MMML.

- **Inexhaustible simulation**: This is the only ML technique that could theoretically surpass human performance in joining element design. In a simulated environment, an algorithm can try several different joining location distributions inexhaustively. Each distribution is evaluated by, for example, several FEAs on one or more performance metrics. Over many iterations, the reinforcement learning algorithm learns the impact of joining locations in a distribution by analyzing the resulting performances of joints.

To summarize the answer to this research question, a variety of AI techniques can predict joining elements. Specifically, ML is a promising upcoming field in the manufacturing industry. It possesses the necessary properties to consider successful designs and other product variants. However, there is no one-whole methodology. Dividing joining element prediction into three subtasks reduces the complexity of automation. These subtasks consider the joining technology, location, and parameter prediction. Here, joining location prediction is the most challenging task due to geometric considerations. As a result, this study presented three approaches that tackle the challenge on different levels of machine cognition. These levels align with the resources, capabilities, and maturity of the product design phase.

After discussing how the design of joining elements can be automated, the third and final subquestion was as follows:

How should product variety be considered in joining element design?

Many strategies exist for coping with product variety, among which modular product design is highly popular [79]. It entails approaches such as standardizing interfaces and sharing components between multiple product variants. Doing so can reduce the negative effects of product variety inside companies without limiting the ability to meet customer demands [6]. Modular design approaches, such as modularization and commonalization, are particularly suitable for joining element design. Commonalization aims to increase the sharing of the same parts and properties between multiple product variants, whereas modularization aims to dissect products into large integrated interchangeable chunks called modules.

Many regard commonalization as a general strategy for product platforms. However, the literature lacks methodologies that purposefully include it in design processes. Actively considering commonalization during design can reduce variety in designs [85] as well as the need for design changes in later phases. For example, as joining element design considers modularity, the changes required by modular design methodologies might be fewer in number. Besides parts, commonalization can also be implemented on properties and joining elements, such as by reducing the variety in materials, suppliers, or technologies. Increasing the shared properties of parts and joining elements between product variants makes it easier to cope with product variety.

Modularization has received much interest in the scientific world. Most methodologies aim to define either strategic modules for many product variants or technical modules with highly interconnected components [75]. However, no modularization methodologies specifically address joining element design. They only optimize for assembly operation-related criteria, such as parts handling and insertion. This lack of research has resulted in unnecessary design iterations as products will have unnecessary variety in joining elements between their variants. Moreover, highly detailed management of joining elements tends to affect the assembly operations of products negatively.

Furthermore, unnecessary variety in designs makes modular design more difficult. By containing the variety in the design of new joining elements, approaches such as commonalization and modularization become more effective. They require fewer modifications to the original designs as each modules needs to account for less variety. Through promoting ML approaches for designing joining elements, the probability that

predictions will follow the pattern in the dataset increases, thereby reducing the chance of generating exotic, low-frequency design solutions. This is in contrast to optimization methodologies that would determine the optimal individual solution for each design problem. However, variety containment during design is also possible with methodologies less sophisticated than ML. Even setting up a database to pick and reuse designs from other product variants would enable the rapid prediction of joining elements. In short, automated joining element design can consider product variety by implementing ML.

Joining element designs require a certain variety to properly join each product variant. Joining designs are specific to a local use case, hence, product variants may consist of many differentiating designs. However, the great variety in designs of joining elements rapidly invoke variety problems. Furthermore, the variety requires joining element designs diverge for both larger products and an increasing number of product variants. For example, an ultimate remedy is to commonalize such that all joints are designed with one overall technology. However, these degrees of commonalization are often not feasible. There is a trade-off between optimal joint design within each product variant and between all other variants.

This trade-off requires a convergence of designs, i.e., reducing the variety of designs to a few key design which are shareable. Modular product design – and within its extent commonalization and modularization – enables this. Modular product design enables the creation of interchangeable modules to fulfill each functional requirement demanded by customers [206]. Modularization reduces the level of detail of products and with it the variety in designs. The challenge that the literature has neglected is considering joining elements as objects when defining modules.

To enable modular product design to fully integrate joining elements, this dissertation defined and considered joining elements as individual components. This consideration enables their standardization and reuse over multiple product variants. It also enables joining elements to be picked and placed on modules' interfaces. Furthermore, this definition enables the modularization of joining elements as objects and not as optimization criteria. Modularization methodologies should treat joining elements as components and can create integrated modules of all objects. Thus, a reusable group of joining elements is a joining module, similar to a reusable group of parts. Joining modules create new boundary conditions in part design as the interfaces between parts require standardization. Standardization enables the interchangeability of joining modules. Different joining modules could join the same combination of parts. Hence, the standardization of interfaces is a requirement that enables the consideration of product variety in joining element design.

Considering joining elements as individual components also enables algorithms to commonalize them. Similar to predicting each joining aspect, they may commonalize each joining aspect individually. For example, commonalizing only joining technologies can reduce the number of different robots, tasks, and operations in manufacturing. As joining elements do not directly derive from customer demands, changes to their design are simpler than for components as their main consideration is the performance of joints. However, commonalizing joining elements requires them to remain manufacturable. Additionally, they need to at least have the same structural performance. Hence, every change requires an extensive validation of designs for all affected product variants.

The answers to the three aforementioned subquestions enables to answer the main research question, which was as follows:

How can joining element design be automated for high-variety products?

Optimal joining element designs consider the entire product life cycle over all products. Their multidisciplinary and holistic design requirements are highly diverse and may change throughout product development. Furthermore, optimal designs are relative to the environment of the application. They may differ for each company, use case, and moment in time. Hence, they require balancing over many, often uncertain and dynamic, requirements.

Moreover, these requirements will increase in number and difficulty, yet some, such as the relationship of product variety and joining elements, have received little research attention. However, they are necessary for designing successful products. Therefore, answering this research question involved two aspects, namely the intelligent automation of joining elements and their modular design.

- The generation of joining elements must consider other designs. The use of AI, particularly ML, enables knowledge of successful designs to be re-instantiated and the design process to be accelerated. It requires a database of designs to find patterns and then to apply them to new problems. However, ML and AI techniques generally require much more research in order to achieve human design performance. The skills of human designers remain superior to automation, especially for joining element design problems with a large solution space. Hence, initially achieving more optimal designs would require a combination of human creativity and expertise with machine support. This combination enables designers to concentrate on the most difficult problems, of which automation can solve the most cumbersome and repetitive.
- Modular product design keeps joining elements manageable and transparent. It creates an environment where both designers and algorithms advance toward optimal joining data for high-variety products. Increasing the number of joining elements will cause their designs to diverge. Therefore, it is not sufficient to consider successful products during prediction; rather, products must undergo an active strategy to create modular designs. Moreover, ML approaches require databases of relevant samples, which may not be available during new product development. As a result, modularization and commonalization approaches must clean up the created variety; that is, they need to reduce the variety within any joining element design, regardless of whether it concerns the joining technology, locations, or parameters. Moreover, through substituting joining elements in CRs, the variety in groups of joining elements (i.e., potential joining modules) can be reduced. Enabling the reuse of joints over multiple product variants. In this sense, commonalization enables modularization to determine the necessary modules for coping with the variety in joining element designs.

This dissertation has presented a framework called VICTOR that integrates the design automation of joining elements with their modular design. The framework organizes automated joining element design into various design problems, including the prediction and

commonalization of each joining aspect. For example, a function may concern joining location prediction or the commonalization of joining technologies.

The design problems were determined by structuring the literature. In doing so, this study considered the relevant studies on state-of-the-art joining element design, modular design, and AI. Furthermore, the design problems were aligned with the steps of the joining element design process. Hence, each design problem can be performed using various methodologies, between which the techniques and methods differ. This study's evaluation of the methodologies for a given design problem relied on recommendations, experimental results, and discussions with experts.

The framework captures the design problems in the structure block. This block is built upon a prediction block and a modular design block. These blocks describe the fundamental ideas for enabling automated joining element design. They define the boundary conditions for the design problems. These blocks also act as divergent and convergent forces on designs. Prediction approaches have the freedom to make designs, whereas modular design methodologies collect results and bring them to a common sustainable denominator.

Moreover, the sequence block is placed on top of the structure block. This block describes the relations of the design problems. It presents sequences for integrating automated joining element design into the user journey of designers. The sequence block aids not only designers but also researchers through guiding them through the design problems, as not all problems can be solved sequentially. For example, not every problem is relevant at a given moment in time. As a result, the framework's nature helps to select the relevant AI fields and methodologies for each design problem.

The framework structures the task of automated joining element design by dividing it into seven design problems, which are not independent. However, splitting them enables application-specific solutions to be generated. Additionally, the framework enables experimentation with new technologies to achieve a higher degree of automation.

Using the framework, this study evaluated popular AI fields in manufacturing within each of the seven design problems. It discussed the advantages and disadvantages of methodologies within these fields and assessed their applicability for automated joining element design. Hence, the framework enables designers and researchers to select the most applicable methodologies for their design problems.

For assessing the applicability for AI, the framework presents new methodologies. These methodologies explore the use of new AI techniques. However, they also aim to fill current research gaps. These new methodologies include joining technology, location prediction, and the commonalization of joining aspects. The prediction of joining locations is of particular interest as it is a highly difficult task, which is due to the many unresolved issues in state-of-the-art methodologies. These problems include the consideration of successful designs, fast predictions, and use of state-of-the-art AI techniques. Furthermore, this study found no modular design methodologies explicitly for joining elements.

Now that this study's research questions have been answered directly, the following section moves on to listing the results of the newly proposed methodologies.

5.2 Overview of the results

The proposed methodologies in VICTOR originated through the identification of new AI fields in the framework. This section discusses how they meet expectations and fit the theory. The following subsections follow the same order as those in the validation chapter, which is indicated in the following list. The last subsections address considerations for generalizing the methodologies as well as the limitations of this dissertation study.

- Joining technology prediction – using an SML approach
- Prediction of the number of spot welds (algorithmic fitting approach) – prediction of the number of joining locations using SML
- Randomized joining location distribution (algorithmic fitting approach) – prediction of joining locations using an optimization approach
- Voxel-based joining location prediction (grid-based approach) – prediction of joining locations using SML while only considering geometric information
- MMML (grid-based approach) – prediction of joining locations using SML considering both geometrical information and PMI
- SA – commonalization of joining locations using UML
- ED – commonalization of joints using a RBR approach
- Generalization – generic remarks on the validated models

Joining technology prediction
The goal of this implementation was to determine whether nongeometric data contain sufficient information that can be employed to predict the joining technology using SML. A random forest classifier with the nine most correlated features was able to predict the joining technology with an F1-score of 94.8%. It used conventional features such as thickness and surface area, but also application-specific nomenclature such as “roof structure” and “above.”

Its performance was very high, although the model used simplifications to cope with the many subtechnologies and combinations of technologies in joints. Such simplifications may grow problematic as joining technologies are and will become increasingly more specific and varied in the future. For example, designers are increasingly using combinations of technologies to improve joining quality. Furthermore, the growing variety of the joining technologies themselves is reducing the number of data samples for each technology. This reduction in samples is increasing the difficulty of training models and, as a result, datasets must rely on simplifications even more heavily. This effect will limit the potential performance of predicting joining technologies. Consequently, it is vital to consider both the application during feature engineering and to keep the models up to date in a dynamic development environment.

Nevertheless, this validation revealed that SML is a proper alternative to multidisciplinary optimization methodologies. Joining technology prediction can aid designers during early phases of product development. Fast predictions accelerate the design process and may give

designers a head start. Using this information, designers and automation methodologies may select the final technology (combination) that is most suitable for the design problem.

Prediction of the number of joining locations

Algorithmic fitting requires the prediction of joining features, such as the number of joining locations, before distributing them over a CR. This study's validation of this methodology aimed to determine whether geometric features, such as the number of spot welds in a joint, are predictable from nongeometric data using SML. In other words, it aimed to determine whether correlations exist between nongeometric and geometric data and, if so, whether models can exploit them.

An XGBoost classifier using all nongeometric data obtained the best results in predicting the number of spot welds with an F1-score of 80.0% and an average difference of -0.46. Feature engineering did not improve the performances of the prediction models. The results indicated that the number of joining locations is difficult to predict from nongeometric data only. For example, the voxel-based spot weld location classifier had a lower difference in the number of predicted spot welds of –0.30.

However, a direct performance comparison between the decision tree and voxel-based model was not possible, as the models had significantly different boundary conditions. However, the results indicated the relative ease of a geometric-based model for achieving this performance. The number of spot welds also depended on other possible joining technologies in the joining scenario. For example, some joints with components that have similar properties might have additional adhesive bonding as well as spot welding. This would create a similar challenge to that in joining technology prediction, where the number and variations of technologies are also difficult to represent in ML models.

Furthermore, the number of joining elements is correlated with the loads, forces, and other design requirements. For example, increasing numbers of elements tend to correlate with increasing loads as joints require increasingly more strength to hold structures. These complex circumstances limit the use of such approaches to early product design phases, where they add value due to their simplicity and rapid design generation. Consequently, this allows one to combine such models with lightweight algorithmic fitting approaches to predict joining locations.

Hence, decision trees are unsuitable for predicting the number of spot welds based on nongeometric data alone. The models were relatively complex compared with those in joining technology prediction. These results enabled this study to argue that such tasks require prediction models that consider geometry. In short, algorithmic fitting is a simple approach, and its results lack structural performance and knowledge-based considerations.

Randomized joining location distribution

The number of spot welds are an input. Algorithmic fitting uses an evolutionary algorithm to create joining locations. The aim of this validation experiment was to determine whether a relatively straightforward optimization algorithm can create meaningful joining locations. Randomized distribution can significantly accelerate the initial design of joining elements, but it requires further analysis and research into the objective function.

An implemented evolutionary algorithm aimed to randomly optimize the distance of spot weld locations to the edge as well as to one another. The results revealed an accurateness of 87%, similarity of 25%, and correctness of 17%. The algorithm used 2D samples, which tend to contain less complex problems than 3D samples. For example, flat (2D) CRs for spot welding often imply a strip where the algorithm can distribute the elements along a centerline. For such CRs, it achieved particularly good results; however, for CRs with a higher solution space, such as large square surfaces, it experienced great difficulty. Nevertheless, a blunt algorithm implementing random distribution positioned 87% of the joining locations on meaningful locations within $40\ mm$ of their target coordinate.

This methodology creates meaningful predictions, and it is fast and straightforward; however, these are also its limitations. Thus, it is not suitable for any type of CR. Additionally, inconsistencies in the results led to it not improving the benchmark RBR methodologies. Consequently, randomized distribution might be useful in early design phases, but only after additional research and development efforts.

Voxel-based joining location prediction

This approach applies image segmentation, a 2D computer-vision technique, to 3D voxels with the aim of drawing joining locations on a voxel grid. This methodology enables patterns in data to be exploited to predict on unseen data samples. It implements a large CNN with an EncDec architecture to predict joining locations. The main aim of the validation was to determine which SML task (i.e., classification or regression) best suits the use case of spot welds in the automotive industry.

The results of using this SML approach for predicting joining locations were satisfying. The experimental results indicated that both regression and classification tasks both predict spot weld locations by only considering voxel-based geometry. However, the classification approach exhibited a more promising performance than regression as it explicitly incorporated geometric reconstruction. This behavior led to more robust and feasible predictions, whereas the regression approach often output noise in the predictions. Classification resulted in an F1-score of 40%, accurateness of 63%, similarity of 47%, and correctness of 23% on the test set. Notably, these scores cannot be directly compared with the results of randomized distribution as both the dataset and data formatting were significantly different.

The model was able to predict complicated joining location distributions as long as the training data covered them. All of the implemented voxel-based models suffered from variance problems, implying that the dataset did not allow them to generalize well. This is partly a consequence of product variety. This study assumed that product variety was an enabler of SML by providing large datasets. However, it also had a significant drawback due to high similarities between product variants and, as a result, also between parts and joints. Part variants with only minor deviations create similar data samples, which makes approaches prone to data leakage, as was observed in the exploratory concepts. The models exhibited falsely good performance as they were evaluated on the same samples that they were trained on. Hence, high product variety requires filtering of similar data samples. The careful cleaning and preparation of the dataset are also crucial. Precise effort is required to balance the size of the dataset and the additional information brought by every data sample.

These models were difficult to train. The results originated from empirically determined sets of hyperparameters, architectures, and loss functions. The high computational costs limited the ability to experiment and to optimize the models. Nevertheless, the segmentation-based SML approach could predict advanced patterns and produced robust results. The models demonstrated NNs' promising ability to predict engineering data. They exhibited an understanding of geometry; for example, joining locations only appeared on CRs.

Nevertheless, the voxel-based methodology worked for both regression and classification. Moreover, SML with an EncDec architecture was able to predict joining locations while exploiting patterns in the data. The results indicated that joining location designs reoccur for similar data samples. This indicates that this methodology fills the research gap in joining element design by considering the knowledge of successfully marketed products.

Multimodal machine learning

The aforementioned methodology only utilized geometry to predict joining locations. It neglected data, such as PMI. By contrast, MMML integrates geometric and nongeometric data, supplying models with more information. The inclusion of nongeometric data theoretically enables models to increase their prediction performance. The implemented multimodal models were evolved from voxel-based classification models through applying a branding approach. Branding adds a label to data samples representing the cluster that they belong to.

Unfortunately, the models did not lead to significant improvements. The best-performing model employed PCA to reduce the number of nongeometric data dimensions to seven, after which K-means clustering determined eight clusters. It achieved an F1-score of 42% and accurateness of 65%. These performance metrics were only slightly better than those of models that only considered geometry. As tests on the benchmark revealed that RI can explain differences of $\pm 2\%$, the branding approach did not significantly improve the models.

Although branding creates additional complexity in prediction tasks, the models did not seem to apply the additional information to affect the joining locations. Various reasons may exist for this lack of a performance increase, three of which are described as follows:.

- First, clusters might contain too much overlapping information, which would cause the model to fail to distinguish the properties of each cluster. However, even when this study reduced the features to only include weights of parts (i.e., the feature that correlates the most with the number of joining elements), there were no significantly different results.
- Second, samples might not contain significantly different joining locations between the branded labels. This behavior would make branding only suitable for use on matured datasets, as then only small particularities in the labels would create different outcomes.
- Third, variables such as network architecture or the branding approach itself might cause MMML to be unsuccessful. Thus, the integration of 3D geometry with nongeometric data requires further research.

Hence, the geometry-only classification model was the most promising. Regardless, this methodology is complex and has many variables. Nevertheless, the results demonstrated the

potential for SML in engineering design. The models understood spatial dependency and were able to create structurally meaningful joining element designs. However, further research is required to overcome the variance and scaling problems.

Spatial aggregation

Besides the prediction of joining aspects, the validation chapter also discussed two commonalization methodologies: (1) SA used to commonalize joining locations, which reduces their location variability when considering multiple product variants; and (2) SA used to determine shareable joining locations between product variants. This study validated the methodologies on two different datasets.

The first dataset contained predictions of the voxel-based joining location approach to determine the ability of SA to reduce the variation of predictions due, for example, to noise. The algorithm reduced approximately 25% of the initially defined locations. The second dataset contained ground truth values to determine SA's ability to create joining locations over multiple product variants. Here, the algorithm reduced approximately 50% of the locations when considering flat contact surfaces in the database.

The results demonstrated that the commonalization of joining elements is an enabler for modularization. Unfortunately, a consequence is that commonalization may also reduce modularity when only subsets of elements are shared between product variants. SA can significantly reduce effort in planning, documentation, and management by reusing the same location.

The algorithm was quick and effective. However, the methodology also highly depended on the dataset's size and quality. Therefore, the algorithm must be used on strategic moments during product development.

Element densification

This methodology commonalizes joints. ED aims to add individual joining elements to joints. These additions enable the reuse of entire joints. The algorithm is designed for high-variety products as it requires overlapping CRs between product variants to contain different numbers of joining elements.

In the validation experiment in this study, ED increased the shared joining elements of the same two datasets used in SA by 5.0% and 1.1%, respectively. These values seem relatively small, but may equate to a large impact for designers in large products with high product variety, such as those in the automobile industry. However, the methodology requires modularization results to measure the effect on its ability to cope with product variety. ED enables the creation of simpler joining modules. The commonalization of an entire joint may potentially result in one joining module being reused over the affected product variants. Additionally, the implementation only added joining elements as the substitution of joining elements may be more effective at increasing their shareability. Substitution seems less prone to the geometric constraints of overlapping CRs. However, this requires additional research.

ED itself was quick and effective at its task; however, the overlapping CRs created a constraint on the algorithm's execution. These boundary conditions occurred rarely, making it difficult to evaluate the performance and harvest its benefits. Additionally, the dataset must

allow for such a methodology to work.

Besides the commonalization of joints, ED employed an active design reuse strategy. It selected joining elements from databases and applied them on the CRs of new product variants. This behavior of the methodology was unexpected as the design of the algorithm used a modular product design philosophy. Adapting ED to perform as an algorithmic fitting approach for predicting joining locations requires further research.

Generalization

The simplest solutions often suffice for a problem. Many design problems are complex and thus require design iterations by default to achieve quality results. The aim of automation should be to create new designs fast, but not necessarily to achieve a global optimum with the initial prediction. It is the cumbersome, repetitive tasks that create most mistakes and rework. The prediction of both the technology and number of spot welds can create a solid starting point for designers and algorithms. Such technologies are simple to train, update, and optimize.

However, more complex ML models that exploit Euclidean data structures also have promising results. They can distribute joining locations arbitrarily on more complex geometry and thus lead to more creative designs. However, this ability comes at the cost of significantly higher computational cost and effort for training the models. The computational cost and limitations on the sizes of data samples limit the indefinite scaling of this method to include large components.

Regardless, the predictions of the presented methodologies in this study required validation in a real or simulated environment. The proposed performance only measured differences with successful data. However, it did not imply that the predictions were necessarily wrong. For example, the structural performance was difficult to measure and required detailed analysis. FEA and evaluations by expert designers can help in assessing the quality of these methodologies and optimizing them accordingly.

Moreover, the generalization of models to other industries and applications is more straightforward for simple feature predicting models. The models implemented in this study relied on data from the automobile industry, which traditionally has a high product variety and many spot welds per product variant. However, the dataset was barely large enough for the number of trainable parameters in NNs and the necessary information in each data sample. Other industries might not have such a database, and hence, they would be limited by even smaller datasets. However, such methodologies are transferable to any industry and any type of assembly. As a result, generalization might originate from creating models that span many industries and applications.

Limitations

This research only focused on validating whether the developed concepts are feasible. It did not aim to determine the best AI technique or concept nor its best implementation. The aim of validating for feasibility prevented making statements regarding the best methodology for automating joining element design. Even for the given use case, no complete comparison nor recommendation was possible.

Time and computational constraints also restrained this study from cherry-picking concepts

for validation, including the creation of a viable dataset for each concept. The dataset, concept, time, and study may lack some depth due to the experimental results.

Furthermore, many companies do not publish their internal processes and resources to retain market advantages. Joining element design, which is typically an in-house endeavor, is difficult to describe from an outside view. Hence, companies' processes in joining element design along with considerations and stakeholders can differ significantly.

Moreover, the findings were derived from resistance spot welding data from the automobile industry. Although this is of much research interest, it is a specific use case that contains typical designs for that industry. The findings can therefore only demonstrate that the feasible implementations are valid for automating joining element design for this particular use case.

In addition, product development is a creative process, and many things only become clear after multiple design iterations. Still, this study aimed to find a generic process for representing joining element design. Therefore, the process is not directly applicable for some companies that must distill their needs from this dissertation.

Summarizing the overview of results section, the validation of the methodologies demonstrated that the framework can identify methodologies and technologies for automating joining element design. Various validated methodologies were capable of performing their tasks. They have their use cases and fit within the design problems of the framework. The following six criteria enabled the assessment of the proposed methodologies: performance, speed, development effort, computational cost, structural performance, and applicability to industry. These criteria enabled an initial evaluation. More in-depth research would need to consider other criteria, such as sensitivity and user friendliness. Nevertheless, a vast number of other – possibly unidentified – technologies exist. The VICTOR framework will enable these technologies to be structured and placed into context.

The study presented a new framework and many new methodologies that support automated joining element design. The following section succinctly lists the contributions of this dissertation study.

5.3 Contributions

As result of this dissertation study, five papers were published in scientific conferences. The papers covered the larger parts of this study's methodology and validation chapters. These papers, listed in chronological order, concerned the following:

1. The state of the art and the research gap [1];
2. The framework and the applicability of AI [2];
3. Voxel-based joining location prediction using CNNs [3];
4. Modular product design with joining elements [4];
5. 2D joining location prediction using a GAN [5].

In addition to these papers, this dissertation provides several overarching contributions that follow from reflections on the gaps in the existing research:

- This study explored the intersection between joining element design, modular product design, and AI. It reframed the design process by taking a different perspective on the current practice of engineering and design. As a result, the study proposed a framework that organizes both joining element and modular design methodologies. Moreover, within the framework, it evaluated the properties of AI for automated joining element design.
- This study adds to the young research area of engineering and creating process data using ML. The literature has integrated AI into the manufacturing industry, although only slightly in the design and engineering fields. Besides a new engineering field, little ML and geometry research has been conducted into generative design as most applications concern object detection, recognition, or pose estimation. Relevant literature is even thinner in terms of data representations in 3D space.
- Engineering includes designing geometrical features in the context of products-as-a-whole where components fulfill functions and have individual requirements and properties. Integrating PMI with geometry through MMML scratches a field that has until now seen the most implementation in medical domains.
- Product variety management has implications for joining elements, which experience high complexity due to the interchangeability of components and modules. This study has presented a modular design philosophy for joining elements and proposed commonalization and modularization methodologies. Until now, the literature has focused primarily on modularizing components with rudimentary modeling of assembly and connection information. Commonalization has seen only a few approaches, even for components.

In addition to the contributions of this study, there were many ideas that could not be included. Hence, the following section presents an outlook for further research.

5.4 Outlook

This section lists recommendations for future research related to the automated generation of joining elements in manufacturing industries with high product variety. This work explores the overlap between joining element design, modular product design, and AI.

- The framework focuses on automated joining element design for high product variety. As previously stated, many perspectives require attention during joining element design. Therefore, this framework is by no means complete and static. Other perspectives, such as ecology or economics, can evolve the framework further.
- The applicability evaluation of AI methods in joining element design was limited to the main fields. However, each field has multiple methodologies, and evaluating them in more detail may reveal new insights and identify novel applications.
- The newly proposed methodologies need to be validated in a productive environment as well as analyzed by expert designers. Additionally, the proposed performance metrics

describe the quality of models in a theoretical manner. It is still difficult to meaningfully describe predicted designs using these metrics. For example, implementing prediction models in designers' workflows will enable the measurement of the (structural) performance and validation of predictions with, for example, FEA.

- Further generalization and integration of such research approaches may result in fully automated design and the manufacturing of products without human intervention. As long as humans remain in the loop, modularization approaches must keep information understandable, clear, and transparent. Full automation will enable individual product variants to have more individual optimal designs. This would increase internal product variety regardless of the perceived external variety, yet also enable products to have higher quality and performance.
- All SML models have empirically determined hyperparameters. Without a doubt, the results can be improved through experimenting with network architectures (e.g., UNet [410] or GANs [331]), hyperparameter tuning and data formatting (e.g., Yolo [319]), data representations (e.g., point clouds [455]), weights (e.g., inverse of frequency [414]), and loss functions (e.g., dice loss [456]). It can also follow from analyses of layers in NNs or different datasets. This also includes other tricks for containing computational cost and increasing training speed or resolution (e.g., dropout [457]). Lastly, the benchmark YOLO framework [319] for predicting joining locations also seems a promising concept, as Perez-Ramirez proposed [412].
- The validation of models considers all types and shapes of joining elements and the predicted locations of multiple joining technologies, including curves in one or multiple models. Furthermore, research into a 2D approach that includes projections for predicting joining locations on curved surfaces is required. Besides joining elements, the methodologies require validation outside of the automotive industry. This would provide insight into the generalizability of the methods as well as the transferability of this knowledge.
- The behavior and potential of models with multimodality compared with a sole geometry model remain unclear. The relationships between the clusters and their influence on joining locations are also unclear. The branding concept may require additional exploration in applicable combinations of DR with clustering methods and feature selection. Besides branding, an implementation using multimodality might use 2D images with 3D shape descriptors or entirely different concepts that include late fusion. Additionally, multimodality can affect joining locations positively through implementing the latent-guided approach for GANs. Successful clustering might create the basis for setting up domains within to create new joining locations in new, unexplored areas of product design.

  Voxel-based (3D) methodologies received much attention in this study for the sake of evaluating the applicability of ML for joining element design. However, simpler 2D image-based methodologies might have fewer problems regarding computational cost as well as enable larger data samples. Relevant exploratory works by Dhameliya [451] (multiview-based CNNs) and Gerlach [5] (GANs) require further research for

evaluating the potential of their methodologies. Additionally, the use of depth buffers might supply 2D methods with additional geometric information [350].

- The prototyping of the reinforcement learning approach in the case of computational cost is not an issue. This approach would theoretically provide the best results and require a tremendous setup in terms of the automation and validation of pre- and postprocessing algorithms. Furthermore, The use of FEA as an adversary in every optimization loop has a enormous computational cost. To optimize calculation times, SML models may predict stresses and strains by, for example, using the standardized modeling of beams, as scholars such as Issler [120] have employed.
- This study did not implement proposed methodologies such as TU and module grouping. Similarly, the effects of the local, global, and domain scopes for CRs remain unknown. Furthermore, the performance of the commonalization and modularization approaches compared with benchmark modular product design requires research. Thereto, the optimization potential in costs is unclear. For products with high variety in particular, how to rapidly determine an optimum while also considering joining elements remains also unclear.
- Modular design using graphs and heuristics should also be explored further. The presented approaches are relatively simple and employ straightforward, state-of-the-art techniques. For example, SA would also work with an EM method with appropriately set boundary conditions. SA and ED require validation after modularization to determine their true benefits and performance.

To conclude the dissertation, the following section presents a brief closing summary.

5.5 Closing summary

This dissertation has presented a process for automating joining element design for product variety. Today, joining element design is time-consuming and highly practical, which results in unnecessary rework. Therefore, this study evaluated state-of-the-art methodologies and presented a framework. The framework first described design problems for automating the various aspects of joining element design. Second, it organized methodologies for each design problem. Lastly, the framework identified new AI applications for each design problem. After evaluating AI fields, this study presented several concepts for predicting various aspects of joining elements.

Furthermore, this study validated several concepts, including joining technology prediction with decision trees, joining location prediction with a random optimization algorithm, and several NN implementations. Moreover, this study explored the commonalization of joining elements. In short, the validation experiments produced promising findings, which can be used in the automation of joining element design in industries with high product variety. However, further research is required to optimize and implement this study's findings into a productive environment.

Appendix A

Appendices

The appendix hold some supplementary information for the validation of methodologies in chapter 4. It contains the following sections:

- Section A.1: Datasets.
- Section A.2: Preliminary data analysis.
- Section A.3: Feature engineering.
- Section A.4: Geometric data preparation processes.
- Section A.5: Exemplary clustering results of product manufacturing information for branding.
- Section A.6: Frequencies of geometry occupying voxels.
- Section A.7: The summary of the validation of the methodologies.

A.1 Datasets

The particularities of each methodology in validation requires them to have subsets of the raw dataset. For example, to predict the number of spot welds, it needs a dataset that only contains the appropriate joining scenarios. Fig. A.1 shows the used steps to create the datasets for each validation. Each methodology addresses briefly deriving the dataset in their respective section of this chapter.

A.2 Preliminary data analysis

This section briefly analyzes the non-geometric features of the data samples. Furthermore, a uni-variate and bi-variate analysis help gain insight for decision making in later sections. The section is based on results from work of Nguyen [323].

Wirth & Hipp [458] propose a circular 6 step process for data analysis and prediction models: data collection, data exploration, data cleaning, model, model evaluation, and feature engineering. Data collection focuses on gathering necessary and available information while

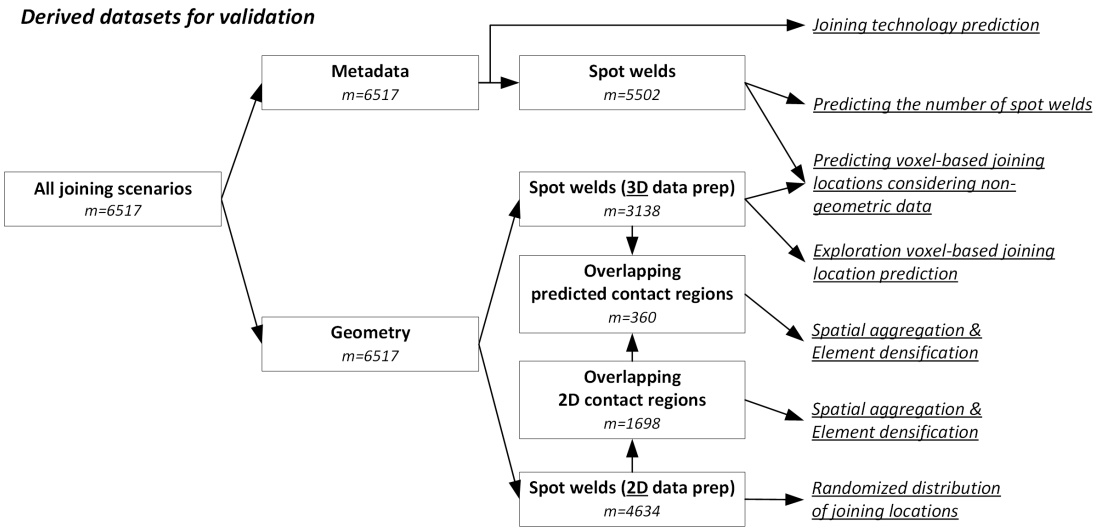


Fig. A.1: The created datasets for validation of prediction methods. The variable m represents the number of joining scenarios in the dataset.

considering project goals and challenges. Data exploration analyzes data to understand dependencies between and distributions of variables [459]. Data cleaning and preprocessing depend on data and its application but typically include removing outliers and noise, structural modifications to variables, and handling of missing values. The model step considers the selection, configuration, and transformation of prediction models. Model evaluation determines the quality of the model using predefined and use-case-specific metrics. Lastly, feature engineering changes data preprocessing activities or settings of the model to improve the quality of predictions.

Non-geometric data of joining scenarios describe available information for each component. Data exploration gives insight into relations between joining elements and components. Especially the relation between non-geometric data and shape descriptors on one side and the number of spot welds is of interest. It describes the scalability and considerations models need to make during prediction. The number of spot welds would be the straightforward evaluation of the model to determine its ability to understand performance requirements. The work of Nguyen lays the basis for this data analysis [421]. Fig. A.2 shows the process of data collection, exploration, and cleaning. It shows the number of data samples m and features n_f for the prediction of the number of spot welds N_{RSW} and joining technology u_s.

Feature selection removes non-relevant attributes. Clean and format nominal attributes correct spelling and formatting issues. Nomenclature contains a free text field and creates difficulties for many machine learning algorithms that demand fixed input sizes. Creating a bag of words [460] splits each word of every nomenclature resulting in a dictionary for the entire dataset. A binary vector has true values ($\equiv$ 1) where words in the nomenclature coincide with those in the dictionary. The dictionary removes, e.g., punctuation and non-informational words such as “and” & “or”.

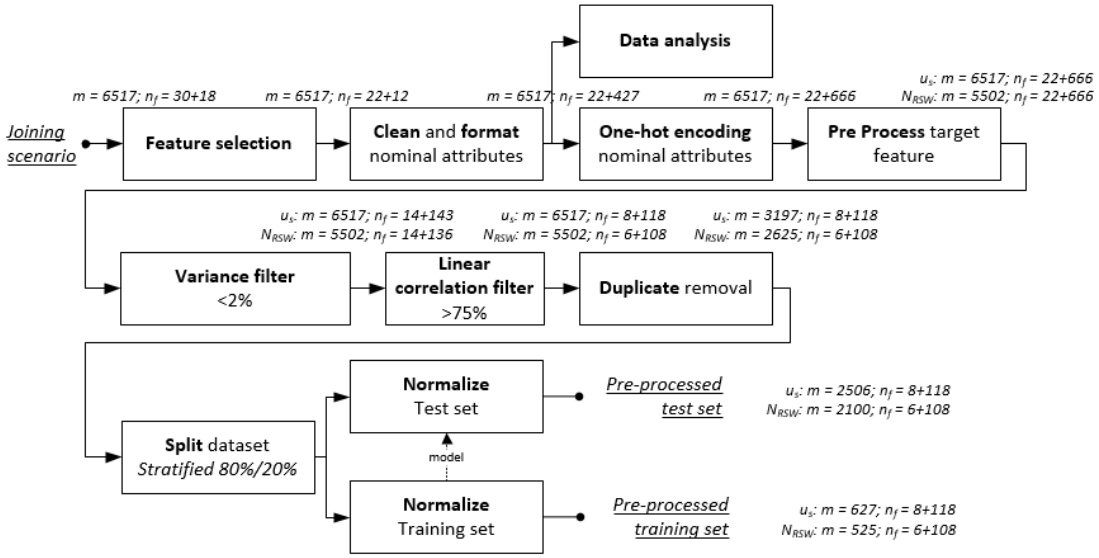


Fig. A.2: Implementation of process for data analysis and prediction model creation.

Uni-variate analysis explores a single target variable. Fig. A.3 shows the distribution of the number of joining elements over all joining scenarios. There is no normal distribution and are outliers in the maximum values (91 and 97). The number of joining elements n_{je} with highest frequencies are 4, 2 and 8 with (512, 428, and 321), respectively. The available metric features of each component: thickness, weight, height, area, volume, density, the center of gravity, and moments of inertia. Categorical features of each component: nomenclature, material, coating, and submodule [31].

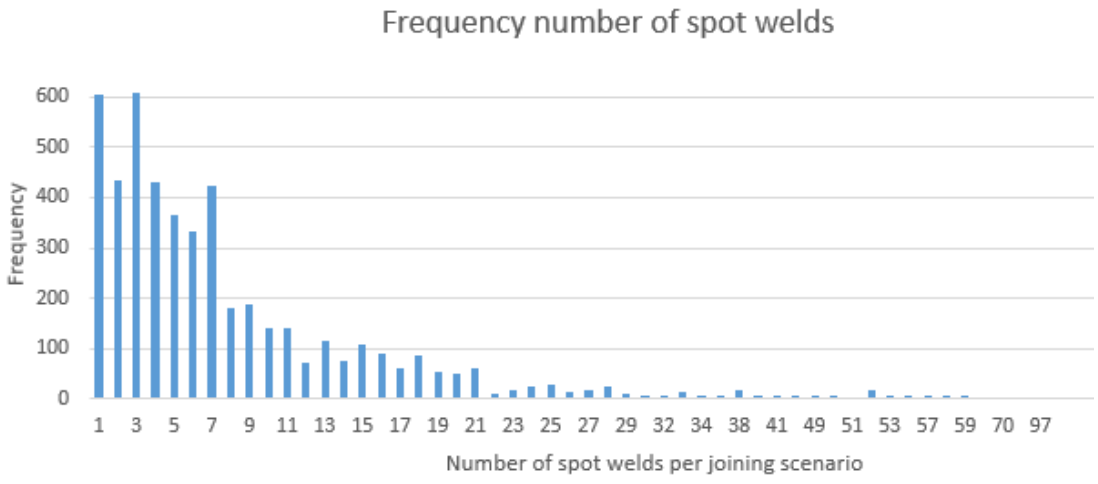


Fig. A.3: Frequencies of the number of joining elements on each joining scenario.

Bi-variate analysis considers correlations between two variables. Various correlation coefficients have been introduced for numeric correlations, most notably Pearson [440], Spearman [461] and Kendall [450]. Pearson expects a normal distribution of data, which is absent [462] (see Fig. A.3, determined using a Lilliefors test with $\alpha = 0.05$, and $p - value < 2.2e - 16$, also after Log and Box-Cox transformations. Spearman is sensitive to data inconsistencies [463], which is often the case in manufacturing industry [101]. Kendall has a higher robustness and asymptotic efficiency [463]. Hence, Kendall is the most appropriate coefficient to express metrical correlation for the number of joining elements [421].

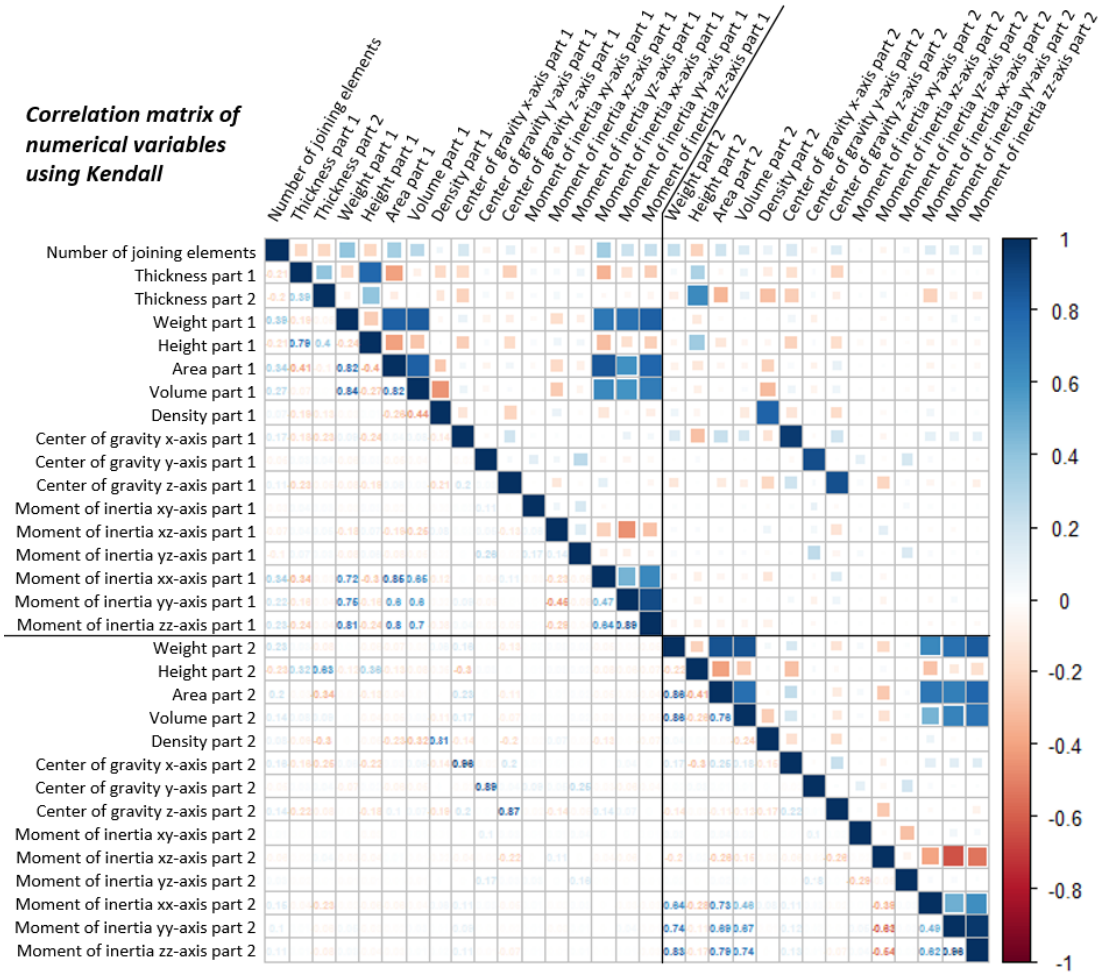


Fig. A.4: Correlation matrix of numerical variables using Kendall with $\alpha = 0.05$. Taken from Nguyen [462]

Fig. A.4 shows the correlations between numerical variables for both components in a joining scenario. The number of joining elements has the strongest correlations with the weight, area, and moment of inertia of the x-axis. The second component has lower correlation values with n_{je} probably due to the order of documentation [66], as designers may select bus components first. Moments of inertia, weight, area, and volume correlate with each

other as they essentially use the same variables in their calculation. The density and center of gravity correlate between the first and second components due to the fulfillment of the same function at the same location in a product variant. Similar components often join with another.

The fig. A.4 indicates that many variables correlate with the number of spot welds. It is logical that thicknesses, weight and moments of inertia seem to correlate more with the number of spot welds than other variables. Larger components weigh more and to their extent tend to need a higher joining performance. By keeping the technology the same, the number of joining locations probably should increase.

These results are expected, when referring to the paragraph on structural performance in section 2.1. For example, weights derive from volume times density. Volume largely depends on thicknesses together with surface area. Together, the moments of area are described by the thicknesses and the dimensions of components. For example, take a rectangle of Fig. 2.3. Also, note the equation 2.2 to calculate the second moment of area. The latter measures the ability of cross-sectional shapes to resist bending caused by loading. The base dimensions of a component aggregate into moments of inertia. The equation only illustrates a small example. In this sense, the results of the correlation matrix (Fig. A.4) are rather trivial. Still, it confirms that the dataset contains these relations which prediction models can learn.

Various tests can analyze the relationship of the number of joining elements n_{je} to categorical variables. One way ANOVA assumes that the continuous dependent variable n_{je} follows a known distribution [464], such as a normal- or Poisson distribution, however, is absent for the dataset [462]. Unfortunately, the test's residues need to have a normal distribution, which is not the case. The Kruskal-Wallis test [465] is non-parametric and applied in case the assumptions of ANOVA fail. The test evaluates for significant differences between a continuous dependent variable n_{je} and a categorical independent variable, e.g., material. However, the test only considers the correlation between the variables, not their categories [466].

Table A.1 shows results of the Kruskal-Wallis and Dunn tests for the nominal independent variables: nomenclature, module, and material. There are significant differences between the groups of the variables and the number of joining elements. The Dunn test [467] analyzes correlations of individual groups within an independent variable and the continuous variable n_{je} [466]. Dunn tests results ($p < 0.05$) for nomenclature, sub-module, and material resulted in almost all groups with differences, too many groups to evaluate, and no interesting results respectively [462].

Independent variable	**Sign. diff.**	**Kruskal-Wallis left**		**Kruskal-Wallis right**	
		χ^2	**df**	χ^2	**df**
Nomenclature	y	1943.9	607	2183.1	690
Module	y	2344.3	1253	26582	1367
Material	y	364.18	72	612.49	79

Table A.1: Results Kruskal-Wallis and Dunn post hoc tests. The dependent variable is the number of spot welds. The p-value for the Kruskal-Wallis test is $p < 2.2e-16$. χ^2 - Chi-squared. df - Degrees of freedom.

A.3 Feature engineering

The following exemplary figure was used to determine the feature importance for predicting the joining technology 4.1. It is seen that after a few key features, the additive value drops rapidly. Fig. A.5 shows the most important features through training of Random Forests to predict the joining technology. Validation picked the best 9 and 16 to evaluate 2 models.

The following exemplary figure was used in determining the key features for predicting th number of spot welds 4.2. Fig. A.6 shows the most important features through training of XGboost to predict the number of spot welds. Validation picked the best 8. Notably, the value of features drops significantly quicker as in the figures to predict the joining technology.

A.4 Geometric data preparation processes

This section presents two data preparation process. The first section A.4.1 presents creating data samples for methodologies that required 2D flat surfaces. These methodologies include randomized joining location distribution (section 4.3), spatial aggregation (section 4.6), and element densification (section 4.7).

The second section A.4.2 presents the process to create voxel-based connection cases from joining scenarios. These connection cases are used in methodologies as exploration of supervised machine learning tasks for voxel-based joining location prediction (section 4.4), and predicting voxel-based joining locations considering non-geometric data (section 4.5).

A.4.1 Implementation of process to create 2D data samples

Fig. A.7 visualizes the identification of contact regions and reproducible steps to create the 2d dataset. Each step is explained in the figure.

A.4.2 Implementation of process to create connection cases 3D

This sub-section describe the generation of data samples for the voxel-based methodologies in sections 4.4 and 4.5. Fig A.8 visualizes the process. The list below explains on each step.

1. Load and set the 2 component-meshes of a joining scenario (STL-files) into vehicle space. This sets the components in place.
2. Create bounding boxes with a slight padding around each component.
3. Perform an intersection between the bounding boxes. This step identifies the area of interest. The area where the bounding boxes intersect contains the contact regions. As such, it also contains the joining elements residing on these contact regions.
4. Use the intersected bounding box to cut out the relevant geometry of each component.
5. Voxelize each piece of cutout geometry.
6. Set the voxels in a grid with the same dimensions. This ensures that the meshes coincide again after voxelization.

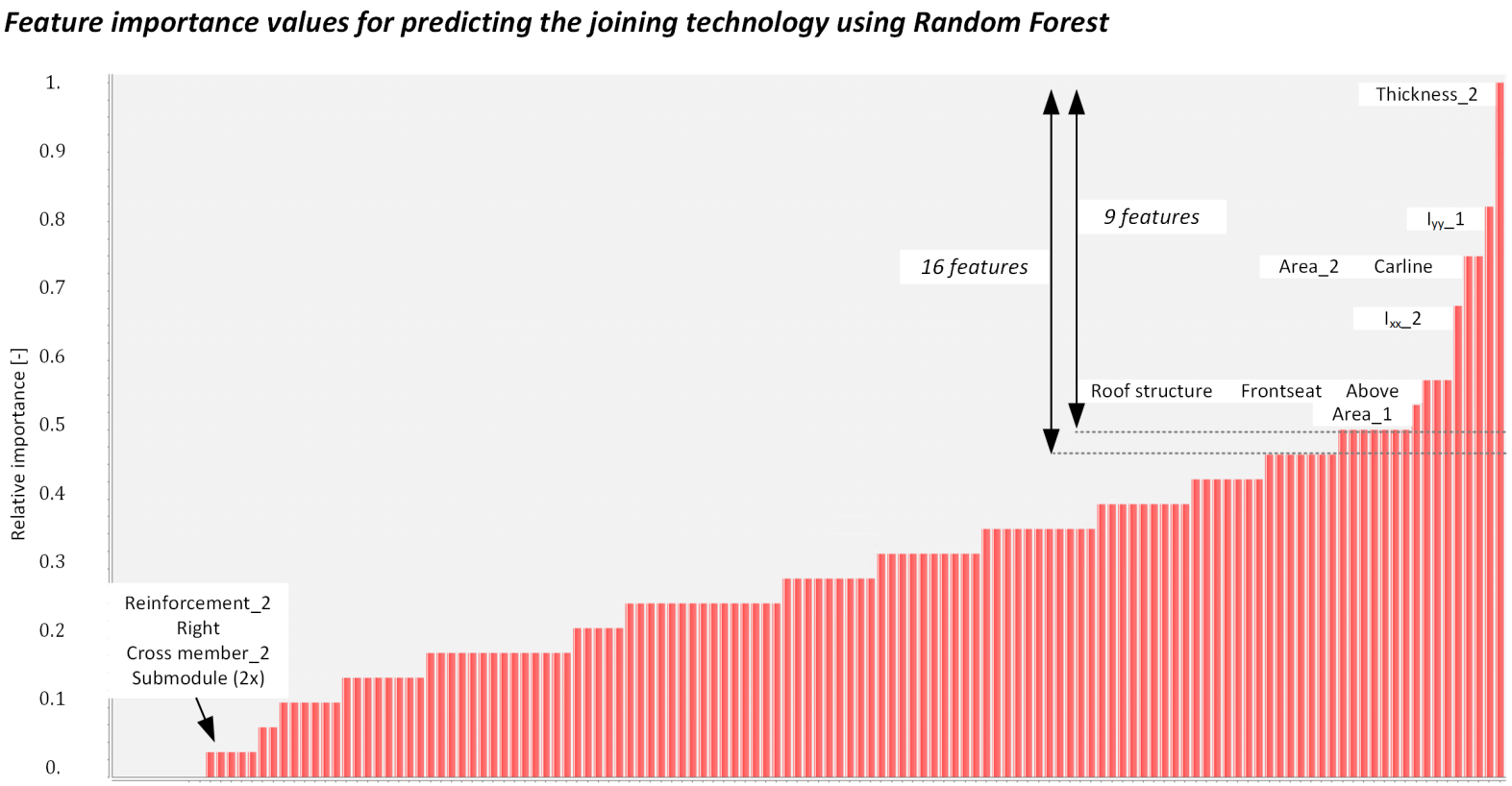


Fig. A.5: Most important features through training of random forests to predict the joining technology

Feature importance values for predicting the number of spotwelds using XGBoost

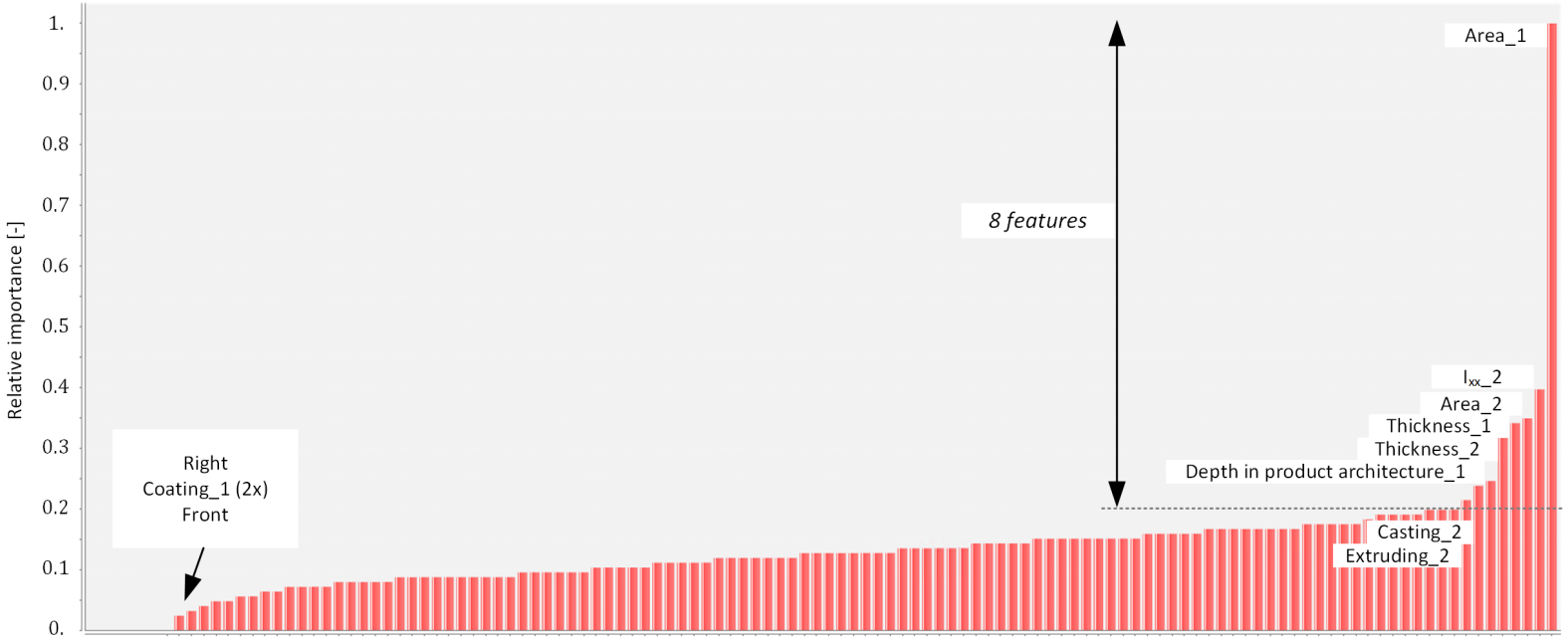


Fig. A.6: Most important features through training of XGBoost to predict the number of spot welds

7. Voxels have 1 values in grid-cells that contain geometry and 0 otherwise. Hence, adding the voxel matrices together creates contact regions. Here 1-values coincide between both meshes, creating cells with the value 2. These represent contact regions.
8. Some connection cases are too large. Hence, they require splitting and padding to create standardized grid sizes.

A.5 Exemplary clustering results of product manufacturing information for branding

A K-means algorithm clustered the data and optimized for k clusters using the silhouette coefficient; see the exemplary Tables A.2 and A.3.

Cluster	Sil. coef.	freq.	freq. [%]
0	0.48	622	12
1	0.57	1055	20
2	0.65	3536	68

Table A.2: Performance of clustering through K-means (three clusters) after a PCA dimensionality reduction (two dimensions). The silhouette mean is 0.62.

Cluster	Sil. coef.	freq.	freq. [%]
0	0.31	1275	24
1	0.42	1109	21
2	0.32	140	3
3	0.26	763	15
4	0.33	367	7
5	0.49	603	12
6	0.35	647	12
7	0.44	309	6

Table A.3: Performance of clustering through K-means (eight clusters) after a PCA dimensionality reduction (seven dimensions). The silhouette mean is 0.38.

Fig. A.9 on the left-hand side presents the result of DR through PCA to two dimensions and k-means finding three clusters.

Here, the silhouette mean was 0.62. However, to determine eight clusters over seven PCA dimensions created more ambiguous results, as the silhouette mean was 0.38. Although the clustering quality for seven dimensions might be lower, this does not necessarily imply that the informational content for MMML would also be lower. Moreover, Fig. A.9 on the right-hand presents the results of using EM with 15 clusters on two PCA dimensions.

A.6 Frequencies of geometry occupying voxels

Fig. A.10 shows geometrical information summed over all voxel grids. The top graph plots the number of voxels with a certain frequency of geometries measured over the entire dataset. Most voxels contain around 50 times geometry. Relative to the approximately 6000 CCs, this is about 1% of the time. The bottom graph shows a similar graph but filtered for the edges of the grid with a thickness of $l_{rib} = 3$. The total number of voxels on these edges is 10656, about 1% of the grid's total. It clearly shows the little information towards the outside of the grid.

A.7 The summary of the validation of the methodologies

The following lists in Appendix concludes the validation of the methodologies with their pros and cons:

- **Joining technology prediction**
  - \+ Quick predictions
  - \+ Decision tree (random forest) needs a few key features
  - \+ Models' decisions are interpretable by designers
  - – Reliant on data preprocessing
  - – Heavy simplification of the task required
  - – Application for early product design
  - – Difficult to adjust for new trends

- **Algorithmic fitting**
  - \+ Quick predictions
  - – Prediction of number of joining locations requires detailed geometry
  - \+ Distribution of joining locations is meaningful for small joining scenarios
  - – Application for early product design
  - – Current implementation does not improve the RBR benchmark
  - – Implementation in 3D space is difficult
  - – Implementation for curve-shaped joining elements is significantly more complex than that for point-based joining elements
  - – Currently relies on a sequence of two methodologies that are both promising but not convincing

- **Grid-based drawing**
  - \+ Very promising results
  - \+ Classification task is more robust than regression
  - \+ Considers successful designs
  - \+ Can predict both point-based and curve-based joining locations with the same model
  - – High development effort; models are difficult to train
  - – High computational cost; data samples must be small and have a low resolution
  - – Multimodal models implementing nongeometric data seem to neglect branded labels
  - – Prone to data leakage and variance problems

- **Spatial Aggregation**
  - \+ Quick results
  - \+ Effectively finds shareable joining locations
  - – High development effort in data preparation
  - – Effectiveness depends on the size of the dataset
  - – Needs to run on strategic moments

- **Element densification**
  - \+ Quick results
  - – Effectiveness depends heavily on the dataset due to geometric constraints
  - – Needs to run on strategic moments
  - – High development effort in data preparation
  - – Results are complex to interpret
  - – Potentially a strong design reuse prediction algorithm

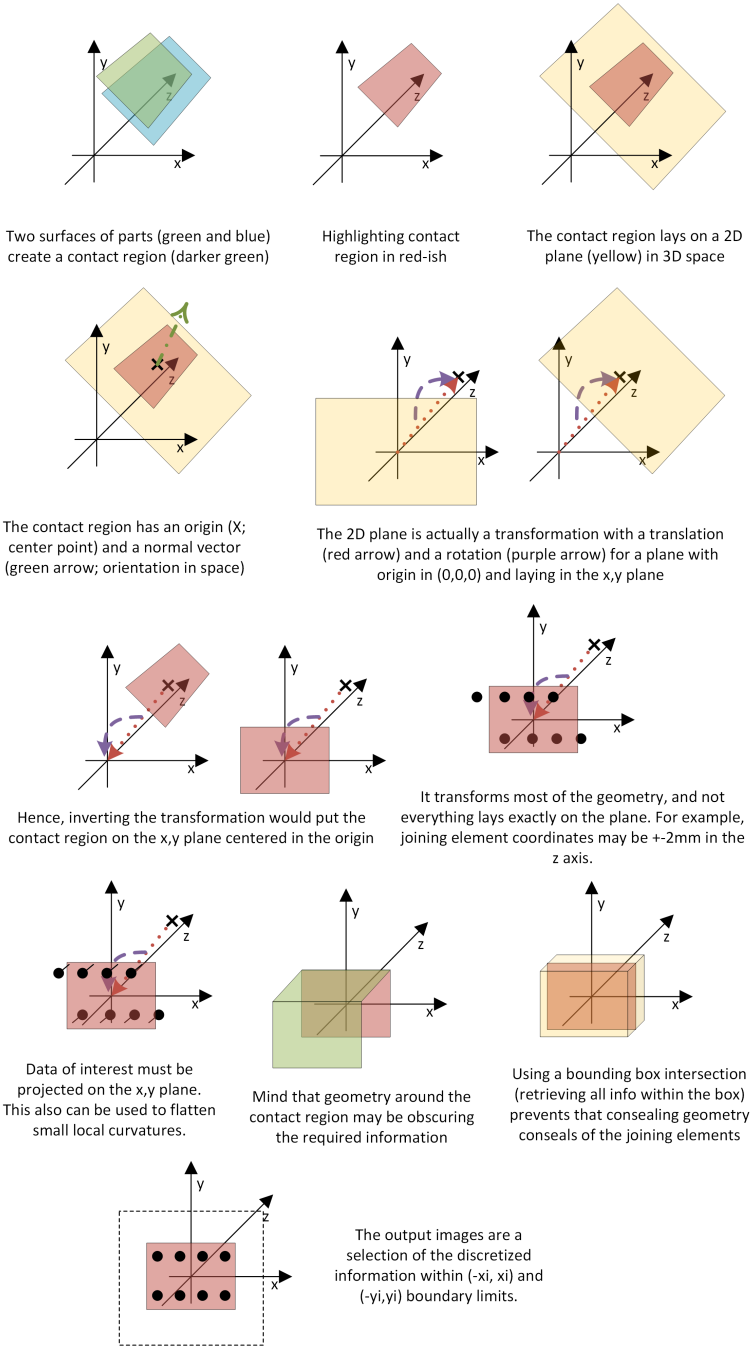


Fig. A.7: Creation of 2D dataset samples from 3D meshes.

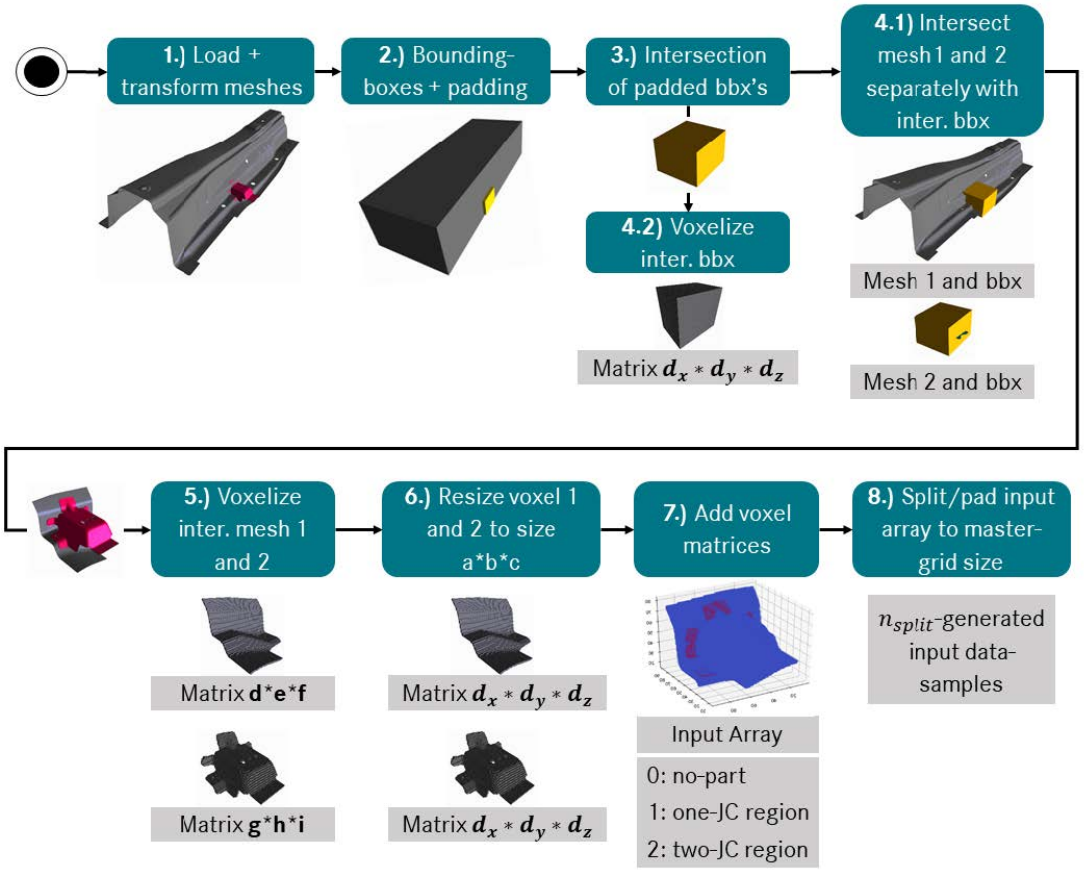


Fig. A.8: Process to generate connection cases. Taken from Perez-Ramirez [412].

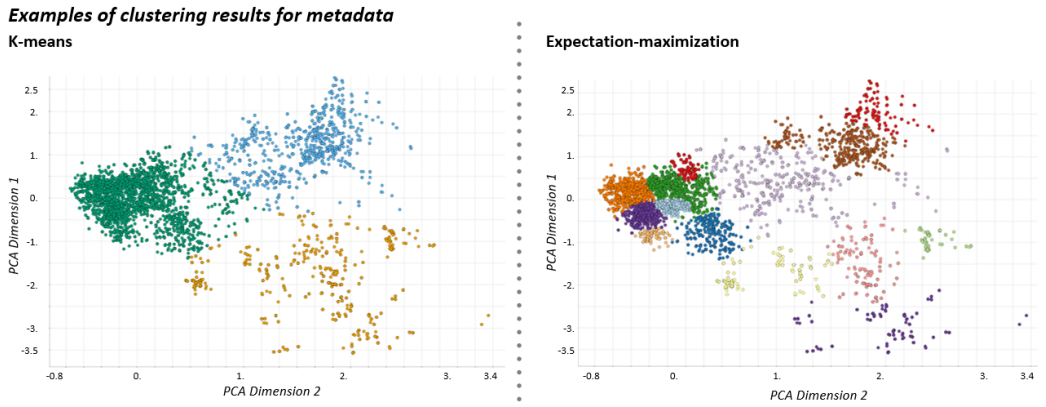


Fig. A.9: Exemplary clustering results through K-means (three clusters) and Expectation-Maximization (16 clusters) after a PCA dimensionality reduction (two dimensions).

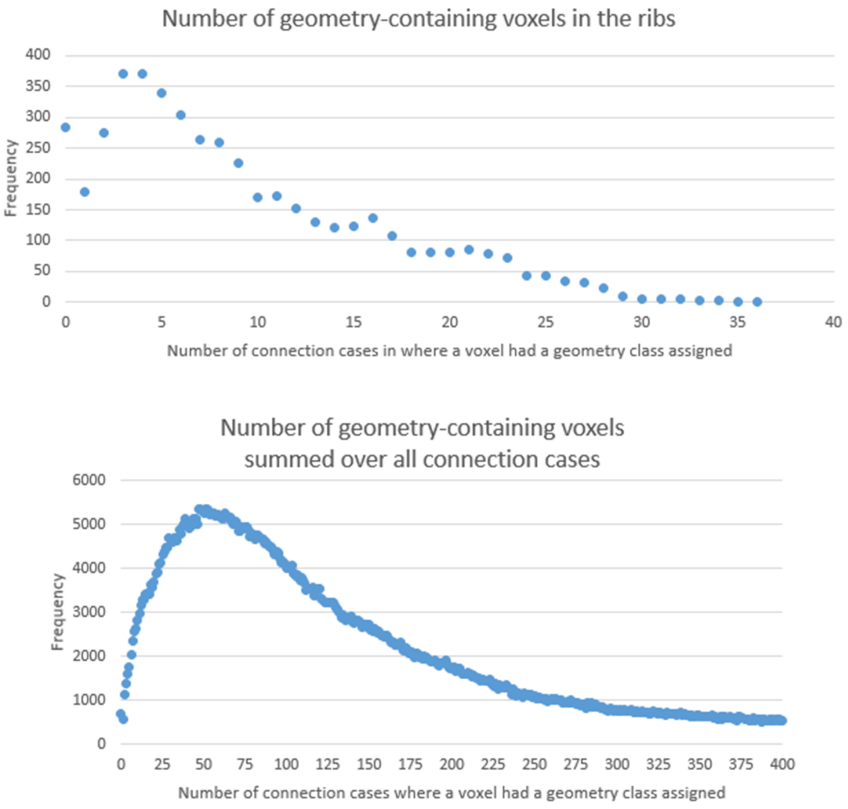


Fig. A.10: Analysis of informational content in voxel grids to determine edge for branding. The top graphs show frequencies of the number of geometry occupying voxels have.

Bibliography

[1] D. Eggink and M. W. Groll. “Joining element design and product variety in manufacturing industries”. In: *13th CIRP Conference on Intelligent Computation in Manufacturing Engineering*. Vol. 88. 2019, pp. 76–81. DOI: 10.1016/j.procir.2020.05.014.

[2] D. H. D. Eggink, M. W. Groll, D. F. Perez-Ramirez, J. Biedert, C. Knödler, and P. Papentin. “Towards automated joining element design”. In: *Procedia Computer Science CIRP*. Vol. 159. 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, 2019, pp. 87–96. DOI: 10.1016/j.procs.2019.09.163.

[3] D. Eggink, D. F. Perez-Ramirez, and M. W. Groll. “Automated joining element design by predicting spot-weld locations using 3D convolutional neural networks”. In: *2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. Piscataway, NJ: IEEE, 2020, pp. 1–9. ISBN: 978-1-7281-7037-4. DOI: 10.1109/ICE/ITMC49519.2020.9198601.

[4] D. Eggink and M. W. Groll. “Modularization of joining elements in high variety manufacturing industries”. In: *Procedia CIRP* 100 (2021), pp. 67–72. ISSN: 22128271. DOI: 10.1016/j.procir.2021.05.011.

[5] T. Gerlach and D. H. Eggink. “Generative Adversarial Networks for spot weld design”. In: *26th Annual Conference of the IEEE Industrial Electronics Society (IES)*. Västerås, Sweden: IEEE, 2021, pp. 1–8. DOI: 10.1109/ETFA45728.2021.9613282.

[6] H. ElMaraghy, G. Schuh, W. ElMaraghy, F. Piller, P. Schönsleben, M. Tseng, and A. Bernard. “Product variety management”. In: *CIRP Annals* 62.2 (2013), pp. 629–652. ISSN: 00078506. DOI: 10.1016/j.cirp.2013.05.007.

[7] C. Yongsheng and L. Haijiang. “Feature model and case retrieval for body-in-white part”. In: *The International Journal of Advanced Manufacturing Technology* 54.1-4 (2011), pp. 231–237. ISSN: 0268-3768. DOI: 10.1007/s00170-010-2910-1.

[8] G. Parry and A. Graves, eds. *Build To Order: The Road to the 5-Day Car*. London: Springer-Verlag, 2008. ISBN: 978-1-84800-224-1. DOI: 10.1007/978-1-84800-225-8.

[9] V. K. Soo, P. Compston, and M. Doolan. “The Impact of Joining Choices on Vehicle Recycling Systems”. In: *Procedia CIRP* 69 (2018), pp. 843–848. ISSN: 22128271. DOI: 10.1016/j.procir.2017.11.009.

[10] J.-H. Zhu, W.-H. Zhang, and L. Xia. “Topology Optimization in Aircraft and Aerospace Structures Design”. In: *Archives of Computational Methods in Engineering* 23.4 (2016), pp. 595–622. ISSN: 1134-3060. DOI: 10.1007/s11831-015-9151-2.

[11] M. Eigner and R. Stelzer. *Product Lifecycle Management: Ein Leitfaden für Product Development und Life Cycle Management*. Heidelberg: Springer-Verlag, 2009. ISBN: 978-3-540-44373-5. DOI: 10.1007/978-3-540-68401-5.

[12] E. Lutters, F. J. van Houten, A. Bernard, E. Mermoz, and C. S. Schutte. "Tools and techniques for product design". In: *CIRP Annals* 63.2 (2014), pp. 607–630. ISSN: 00078506. DOI: 10.1016/j.cirp.2014.05.010.

[13] T. Tolio, D. Ceglarek, H. A. ElMaraghy, A. Fischer, S. J. Hu, L. Laperrière, S. T. Newman, and J. Váncza. "SPECIES—Co-evolution of products, processes and production systems". In: *CIRP Annals* 59.2 (2010), pp. 672–693. ISSN: 00078506. DOI: 10.1016/j.cirp.2010.05.008.

[14] H. Tasalloti, H. Eskelinen, P. Kah, and J. Martikainen. "An integrated DFMA–PDM model for the design and analysis of challenging similar and dissimilar welds". In: *Materials & Design* 89 (2016), pp. 421–431. ISSN: 02613069. DOI: 10.1016/j.matdes.2015.10.012.

[15] K. Martinsen, S. J. Hu, and B. E. Carlson. "Joining of dissimilar materials". In: *CIRP Annals* 64.2 (2015), pp. 679–699. ISSN: 00078506. DOI: 10.1016/j.cirp.2015.05.006.

[16] Deutsches Institut für Normung e.V. *Fertigungsverfahren Fügen*. 2003.

[17] Deutsches Institut für Normung e.V. *Schweißen und verwandte Prozesse – Liste der Prozesse und Ordnungsnummern*. Berlin, 2010.

[18] S. Yang, L. Yan, and C. Qi. "An adaptive multi-step varying-domain topology optimization method for spot weld design of automotive structures". In: *Structural and Multidisciplinary Optimization* 5 (2018), pp. 129–149. DOI: 10.1007/s00158-018-2068-2.

[19] A. Kreis, M. Hirz, and P. Rossbacher. "CAD-Automation in Automotive Development – Potentials, Limits and Challenges". In: *Computer-Aided Design and Applications* 18.4 (2020), pp. 849–863. ISSN: 16864360. DOI: 10.14733/cadaps.2021.849-863.

[20] Q. Su, L. Liu, and D. E. Whitney. "A Systematic Study of the Prediction Model for Operator-Induced Assembly Defects Based on Assembly Complexity Factors". In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 40.1 (2010), pp. 107–120. ISSN: 1083-4427. DOI: 10.1109/TSMCA.2009.2033030.

[21] A. C. K. Choi, D. S. K. Chan, and A. M. F. Yuen. "Application of Virtual Assembly Tools for Improving Product Design". In: *The International Journal of Advanced Manufacturing Technology* 19.5 (2002), pp. 377–383. ISSN: 0268-3768. DOI: 10.1007/s001700200027.

[22] S. N. Samy and H. ElMaraghy. "A model for measuring products assembly complexity". In: *International Journal of Computer Integrated Manufacturing* 23.11 (2010), pp. 1015–1027. DOI: 10.1080/0951192X.2010.511652.

[23] S. N. Samy Badrous. "Complexity of products and their assembly systems". Doctoral. University of Winsor, 2011.

[24] S. H. Zhang, Z. R. Wang, Z. T. Wang, Y. Xu, and K. B. Chen. "Some new features in the development of metal forming technology". In: *Journal of Materials Processing Technology* 151.1-3 (2004), pp. 39–47. ISSN: 09240136. DOI: 10.1016/j.jmatprotec.2004.04.098.

[25] A.-B. Ryberg and L. Nilsson. "Spot weld reduction methods for automotive structures". In: *Structural and Multidisciplinary Optimization* 53.4 (2016), pp. 923–934. DOI: 10.1007/s00158-015-1355-4.

[26] D. J. Epstein. *Range: Why generalists triumph in a specialized world*. New York: Riverhead Books, 2019. ISBN: 0735214484.

[27] A. Lowe, C. McMahon, and S. Culley. "Information access, storage and use by engineering designers: part 1". In: *TheJournal of the Institution of Engineering Designers* 30.2 (2004), pp. 30–32.

[28] Y. Kwon, T. Wu, and J. O. Saldivar. "SMWA: A CAD-based Decision Support System for the Efficient Design of Welding". In: *Concurrent Engineering* 12.4 (2016), pp. 295–304. DOI: 10.1177/1063293X04042470.

[29] S. Sunnersjo. *Intelligent Computer Systems in Engineering Design: Principles and Applications*. 1st ed. 2016. Vol. 51. Studies in Systems, Decision and Control. Cham: Springer International Publishing, 2016. ISBN: 9783319281230. DOI: 10.1007/978-3-319-28125-4. URL: http://gbv.eblib.com/patron/FullRecord.aspx?p=4333607.

[30] P. Groche, E. Bruder, and S. Gramlich, eds. *Manufacturing Integrated Design*. Cham: Springer International Publishing, 2017. DOI: 10.1007/978-3-319-52377-4.

[31] M. Groll. *Interconnection based product and process documentation*. Enschede: University of Twente, 2008. ISBN: 978-90-365-2630-2. (Visited on 02/12/2018).

[32] J. Kaspar, S. A. Choudry, D. Landgrebe, and M. Vielhaber. "Concurrent and geometry-dependent selection of material and joining technology — An initial utility-based systematic decision-making tool". In: (2018), pp. 1–8. DOI: 10.1109/SYSCON.2018.8369549.

[33] T. Jania. "Änderungsmanagement auf Basis eines integrierten Prozess- und Produktdatenmodells Änderungsmanagement auf Basis eines integrierten Prozess- und Produktdatenmodells mit dem Ziel einer durchgängigen Komplexitätsbewertung". PhD. Universität Paderborn, 2004.

[34] J. Madrid, A. Forslund, R. Söderberg, K. Wärmefjord, S. Hoffenson, J. Vallhagen, and P. Andersson. "A Welding Capability Assessment Method (WCAM) to support multidisciplinary design of aircraft structures". In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 12.3 (2018), pp. 833–851. ISSN: 1955-2513. DOI: 10.1007/s12008-017-0429-5.

[35] M. H. Landherr. "Integrierte Produkt- und Montagekonfiguration für die variantenreiche Serienfertigung". PhD. Universität Stuttgart, 2014.

[36] D. Stienstra. *Introduction to Design for (Cost Effective) Assembly and Manufacturing*. 2014.

[37] D. Tsutsumi, D. Gyulai, A. Kovács, B. Tipary, Y. Ueno, Y. Nonaka, and L. Monostori. "Towards joint optimization of product design, process planning and production planning in multi-product assembly". In: *CIRP Annals* 67.1 (2018), pp. 441–446. ISSN: 00078506. DOI: 10.1016/j.cirp.2018.04.036.

[38] J. Landahl and H. Johannesson. "Product variety and variety in production". In: *Proceedings of International Design Conference*. Vol. 2. 2018, pp. 817–828. DOI: 10.21278/idc.2018.0208.

[39] S. A. Choudry, S. Müller, U. Alber, F. Riedel, and D. Landgrebe. "A Multidimensional Assessment and Selection Methodology: Optimized Decision-making of Joining Technologies in Automobile Body Development". In: *Procedia Manufacturing* 21 (2018), pp. 281–288. ISSN: 23519789. DOI: 10.1016/j.promfg.2018.02.122.

[40] S. K. Das and A. K. Swain. "Knowledge-Based Application of Liaison for Variant Design". In: *Product Lifecycle Management for Digital Transformation of Industries: 13th IFIP WG 5.1 International Conference, PLM 2016*. Ed. by R. Harik, L. Rivest, A. Bernard, B. Eynard, and A. Bouras. Cham: Springer International Publishing, 2016. DOI: 10.1007/978-3-319-54660-5{\textunderscore}33.

[41] D. Bond, F. A. Suzuki, and R. K. Scalice. “Sheet metal joining process selector”. In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 42.5 (2020), p. 51. ISSN: 1678-5878. DOI: 10.1007/s40430-020-02310-9.

[42] R. A. R. Ghazilla, Z. Taha, S. Yusoff, S. H. A. Rashid, and N. Sakundarini. “Development of decision support system for fastener selection in product recovery oriented design”. In: *The International Journal of Advanced Manufacturing Technology* 70.5-8 (2014), pp. 1403–1413. ISSN: 0268-3768. DOI: 10.1007/s00170-013-5373-3.

[43] J. Kaspar and M. Vielhaber. “Integrated Cross-Component Lightweight and Material-Oriented Development Methodology – The Embodiment Design Cycle”. In: *Procedia CIRP* 70 (2018), pp. 481–486. ISSN: 22128271. DOI: 10.1016/j.procir.2018.03.001.

[44] V. Florea, V. Shah, S. Roper, G. Vierhout, and I. Y. Kim. “Coupled Multi-Material and Joint Topology Optimization: A Proof of Concept”. In: *Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2018.

[45] C. Woischwill and I. Y. Kim. “Multimaterial multijoint topology optimization”. In: *International Journal for Numerical Methods in Engineering* 115.13 (2018), pp. 1552–1579. ISSN: 00295981. DOI: 10.1002/nme.5908.

[46] V. Florea, M. Pamwar, B. Sangha, and I. Y. Kim. “Simultaneous single–loop multimaterial and multijoint topology optimization”. In: *International Journal for Numerical Methods in Engineering* 121.7 (2020), pp. 1558–1594. ISSN: 00295981. DOI: 10.1002/nme.6279.

[47] M. Kiani and A. R. Yildiz. “A Comparative Study of Non-traditional Methods for Vehicle Crashworthiness and NVH Optimization”. In: *Archives of Computational Methods in Engineering* 23.4 (2016), pp. 723–734. ISSN: 1134-3060. DOI: 10.1007/s11831-015-9155-y.

[48] A. Cui, S. Z. Zhang, L. J. Xu, and H. Z. Liu. “Topology Optimization and Robust Analysis of Welding Spot Layout for a Heavy Duty Truck Cab Based on Element Strain Energy Density”. In: *Advanced Materials Research* 887-888 (2014), pp. 1284–1289. ISSN: 1662-8985. DOI: 10.4028/www.scientific.net/AMR.887-888.1284.

[49] Siemens PLM. *Syncrofit*. 2018. URL: https://www.plm.automation.siemens.com/global/de/products/nx/syncrofit.html (visited on 10/15/2018).

[50] T. R. Thompson and A. J. Salerno. “Automatic generation of joint locations”. WO2009158562A1.

[51] J. Schaffer and H. Schleich. “Complexity Cost Management”. In: *Build To Order*. Ed. by G. Parry and A. Graves. London: Springer-Verlag, 2008, pp. 155–174. ISBN: 978-1-84800-224-1.

[52] S. J. Hu, X. Zhu, H. Wang, and Y. Koren. “Product variety and manufacturing complexity in assembly systems and supply chains”. In: *CIRP Annals* 57.1 (2008), pp. 45–48. ISSN: 00078506. DOI: 10.1016/j.cirp.2008.03.138.

[53] S. Ripperda and D. Krause. “Costs in modularization approaches: A co-citation analysis”. In: *13th International Design Conference, DESIGN 2014*. 2014, pp. 965–974.

[54] K. Sinha. “Structural Complexity and its Implications for Design of Cyber-Physical Systems”. Dissertation. Engineering Systems Division, 2014.

[55] M. Götzfried. “Managing complexity induced by product variety in manufacturing companies: Complexity evaluation and integration in decision-making”. Dissertation. St. Gallen, Univ., 2013.

[56] S. N. Samy and H. ElMaraghy. “A model for measuring complexity of automated and hybrid assembly systems”. In: *The International Journal of Advanced Manufacturing Technology* 62.5-8 (2012), pp. 813–833. ISSN: 0268-3768. DOI: 10.1007/s00170-011-3844-y.

[57] J. Jiao, T. W. Simpson, and Z. Siddique. “Product family design and platform-based product development: A state-of-the-art review”. In: *Journal of Intelligent Manufacturing* 18.1 (2007), pp. 5–29. ISSN: 0956-5515. DOI: 10.1007/s10845-007-0003-2.

[58] S. Bednar and V. Modrak. “Mass customization and its impact on assembly process’ complexity”. In: *International Journal for Quality Research* 8.3 (2014), pp. 417–430.

[59] B. Alkan, D. Vera, M. Ahmad, B. Ahmad, and R. Harrison. “Design Evaluation of Automated Manufacturing Processes Based on Complexity of Control Logic”. In: *Procedia CIRP* 50 (2016), pp. 141–146. ISSN: 22128271. DOI: 10.1016/j.procir.2016.05.031.

[60] D. C. Fettermann and M. E. Soares Echveste. “New product development for mass customization: a systematic review”. In: *Production & Manufacturing Research* 1.1 (2014), pp. 266–290. ISSN: 2169-3277.

[61] J. A. V. Antunes. *Production systems: Concepts and practices for lean production projects and management.* Porto Alegre: Bookman, 2008.

[62] P. Child, R. Diederichs, F. Sanders, and S. Wisniowski. “The Management of Complexity”. In: *The McKinsey Quarterly* 4 (1991), pp. 52–68.

[63] G. Schuh. *Produktkomplexität managen: Strategien - Methoden - Tools*. München , Wien: Hanser Verlag, 2005.

[64] J. C. Cooper. “Logistics Strategies for Global Businesses”. In: *International Journal of Physical Distribution & Logistics Management* 23.4 (1993), pp. 12–23. ISSN: 0960-0035. DOI: 10.1108/09600039310041473.

[65] G. Schuh, M. Riesener, and S. Rudolf. “Identifying Preferable Product Variants Using Similarity Analysis”. In: *Procedia CIRP* 20 (2014), pp. 38–43. ISSN: 22128271. DOI: 10.1016/j.procir.2014.05.029.

[66] A. Kaiser and W. Küchlin. “Automotive Product Documentation”. In: *Engineering of Intelligent Systems*. Ed. by G. Goos, J. Hartmanis, J. van Leeuwen, L. Monostori, J. Váncza, and M. Ali. Vol. 2070. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 465–475. ISBN: 978-3-540-42219-8. DOI: 10.1007/3-540-45517-5{\textunderscore}51.

[67] R. B. Stone, K. L. Wood, and R. H. Crawford. “A heuristic method for identifying modules for product architectures”. In: *Design Studies* 21.1 (2000), pp. 5–31. DOI: 10.1016/S0142-694X(99)00003-4.

[68] M. Schmidt, J. Schwöbel, and M. Lienkamp. “Developing Key Performance Indicators for Variant Management of Complex Product Families”. In: *NordDesign*. 2018.

[69] H. Wildemann. *Variant Management: Guidelines for the Reduction, Control and Avoidance of Complexity in Products and Processes*. Munich, Germany, 2016.

[70] C. Y. Baldwin and K. B. Clark. “Modularity in the Design of Complex Engineering Systems”. In: *Complex Engineered Systems*. Ed. by J. S. Kelso, D. Braha, A. A. Minai, and Y. Bar-Yam. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. ISBN: 978-3-540-32831-5.

[71] A. Kampker, P. Burggräf, C. Deutskens, A. Maue, and R. Förstmann. “Integrated Product and Process Development: Modular Production Architectures based on Process Requirements”. In: *Procedia CIRP* 20 (2014), pp. 109–114. ISSN: 22128271. DOI: 10.1016/j.procir.2014.05.039.

[72] L. Nijmeijer. “Reducing complexity costs by implementing a modular design”. Master thesis.

[73] J. B. Dahmus, J. P. Gonzalez-Zugasti, and K. N. Otto. “Modular product architecture”. In: *Design Studies* 22.5 (2001), pp. 409–424. DOI: 10.1016/S0142-694X(01)00004-7.

[74] M. Kashkoush and H. ElMaraghy. “Optimum Overall Product Modularity”. In: *Procedia CIRP*. Ed. by R. Söderberg. Vol. 44. 2016, pp. 55–60. DOI: 10.1016/j.procir.2016.01.023.

[75] J. Ma, G. E. O. Kremer, and M. Li. “A key components-based heuristic modular product design approach to reduce product assembly cost”. In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 12.3 (2018), pp. 865–875. ISSN: 1955-2513. DOI: 10.1007/s12008-017-0448-2.

[76] E. D. Arnheiter and H. Harren. “Quality management in a modular world”. In: *The TQM Magazine* 18.1 (2006), pp. 87–96. DOI: 10.1108/09544780610637712.

[77] D. M. Anderson. *Build-to-order & mass customization: The ultimate supply chain management and lean manufacturing strategy for low-cost on-demand production without forecasts or inventory / David M. Anderson*. Cambria, Calif.: CIM Press, 2004. ISBN: 9781878072306.

[78] W. A. Swan. “Proposes Standardization of Car Sizes”. In: *The Automobile* 31 (1914), pp. 76–77.

[79] T. W. Simpson, J. Jianxin, Z. Siddique, and K. Hölttä-Otto, eds. *Advances in Product Family and Product Platform Design: Methods & Applications*. New York Heidelberg Dordrecht London: Springer, 2014. ISBN: 978-1-4614-7936-9. DOI: 10.1007/978-1-4614-7937-6.

[80] X. Han, R. Li, J. Wang, G. Ding, and S. Qin. “A systematic literature review of product platform design under uncertainty”. In: *Journal of Engineering Design* 31.5 (2020), pp. 266–296. DOI: 10.1080/09544828.2019.1699036.

[81] A.-K. Weiser, B. Baasner, M. Hosch, M. Schlueter, and J. Ovtcharova. “Complexity Assessment of Modular Product Families”. In: *Procedia CIRP*. Ed. by L. Wang and T. Kjellberg. Vol. 50. 2016, pp. 595–600. DOI: 10.1016/j.procir.2016.04.192.

[82] Z. Wang, M. Zhang, H. Sun, and G. Zhu. “Effects of standardization and innovation on mass customization: An empirical investigation”. In: *Technovation* 48-49.1 (2016), pp. 79–86. ISSN: 01664972. DOI: 10.1016/j.technovation.2016.01.003.

[83] T. AlGeddawy, S. N. Samy, and H. ElMaraghy. “Best design granularity to balance assembly complexity and product modularity”. In: *Journal of Engineering Design* 28.7-9 (2017), pp. 457–479. DOI: 10.1080/09544828.2017.1325859.

[84] J. Stocker, M. Chavushoglu, M. Felgenhauer, and M. Lienkamp. “Efficient Module Design for Chassis-Mounted Components of Commercial Vehicles”. In: *NordDesign*. Vol. 91. 2018.

[85] T. AlGeddawy and H. ElMaraghy. “Reactive design methodology for product family platforms, modularity and parts integration”. In: *CIRP Journal of Manufacturing Science and Technology* 6.1 (2013), pp. 34–43. ISSN: 17555817. DOI: 10.1016/j.cirpj.2012.08.001.

[86] F. G. Galizia, H. ElMaraghy, M. Bortolini, and C. Mora. “Product platforms design, selection and customisation in high-variety manufacturing”. In: *International Journal of Production Research* 58.3 (2020), pp. 893–911. ISSN: 0020-7543. DOI: 10.1080/00207543.2019.1602745.

[87] K. Baylis, G. Zhang, and D. A. McAdams. “Product family platform selection using a Pareto front of maximum commonality and strategic modularity”. In: *Research in Engineering Design* 29.4 (2018), pp. 547–563. ISSN: 0934-9839. DOI: 10.1007/s00163-018-0288-5.

[88] M. Kashkoush and H. ElMaraghy. "Designing modular product architecture for optimal overall product modularity". In: *Journal of Engineering Design* 28.5 (2017), pp. 293–316. DOI: 10.1080/09544828.2017.1307949.

[89] P. Daie and S. Li. "Matrix-based hierarchical clustering for developing product architecture". In: *Concurrent Engineering* 24.2 (2016), pp. 139–152. DOI: 10.1177/1063293X16635721.

[90] W. Ren, J. Wen, Y. Guan, and Y. Hu. "Research on assembly module partition for flexible production in mass customization". In: *Procedia CIRP*. Ed. by L. Wang. Vol. 72. 2018, pp. 744–749. DOI: 10.1016/j.procir.2018.03.023.

[91] L. Gauss, D. P. Lacerda, and P. A. Cauchick Miguel. "Module-based product family design: Systematic literature review and meta-synthesis". In: *Journal of Intelligent Manufacturing* 5.3 (2020), pp. 320–348. ISSN: 0956-5515. DOI: 10.1007/s10845-020-01572-3.

[92] Y. Lu. "Artificial intelligence: A survey on evolution, models, applications and future trends". In: *Journal of Management Analytics* 6.1 (2019), pp. 1–29. ISSN: 2327-0012. DOI: 10.1080/23270012.2019.1570365.

[93] M. R. Frank, D. Autor, J. E. Bessen, E. Brynjolfsson, M. Cebrian, D. J. Deming, M. Feldman, M. Groh, J. Lobo, E. Moro, D. Wang, H. Youn, and I. Rahwan. "Toward understanding the impact of artificial intelligence on labor". In: *Proceedings of the National Academy of Sciences of the United States of America* 116.14 (2019), pp. 6531–6539. DOI: 10.1073/pnas.1900949116.

[94] J. E. Bessen, S. Impink, R. Seamans, and L. Reichensperger. "The Business of AI Startups: (November 29, 2018). , Available at SSRN: https://ssrn.com/abstract=3293275 or". In: *SSRN Electronic Journal* 159.2 (2018), p. 157. ISSN: 1556-5068. DOI: 10.2139/ssrn.3293275.

[95] H. Salehi and R. Burgueño. "Emerging artificial intelligence methods in structural engineering". In: *Engineering Structures* 171 (2018), pp. 170–189. ISSN: 01410296. DOI: 10.1016/j.engstruct.2018.05.084.

[96] S. J. Russell and P. Norvig. *Artificial intelligence: A modern approach / Stuart J. Russell and Peter Norvig*. 3rd ed. Boston, Mass.: Pearson, 2010. ISBN: 9780132071482.

[97] C. Metz. "Tech Giants Are Paying Huge Salaries for Scarce A.I. Talent." In: *New York Times* (2017), p. 1. URL: https://www.nytimes.com/2017/10/22/technology/artificial-intelligenceexperts-.

[98] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben. "Machine learning in manufacturing: Advantages, challenges, and applications". In: *Production & Manufacturing Research* 4.1 (2016), pp. 23–45. ISSN: 2169-3277. DOI: 10.1080/21693277.2016.1192517.

[99] S. Doltsinis, P. Ferreira, and N. Lohse. "Reinforcement Learning for Production Ramp–Up: A Q–Batch Learning Approach." In: *2012 11th International Conference on Machine Learning and Applications (ICMLA 2012)*. Piscataway NJ: IEEE, 2012, pp. 610–615. ISBN: 978-1-4673-4651-1.

[100] J. Dekhtiar, A. Durupt, M. Bricogne, B. Eynard, H. Rowson, and D. Kiritsis. "Deep learning for big data applications in CAD and PLM – Research review, opportunities and case study". In: *Computers in Industry* 100 (2018), pp. 227–243. ISSN: 01663615. DOI: 10.1016/j.compind.2018.04.005.

[101] S. Hagemann and R. Stark. "Automated Body-in-White Production System Design: Data-based Generation of Production System Configurations". In: *ICFET '18 Proceedings of the 4th International Conference on Frontiers of Educational Technologies*. 2018, pp. 192–196. DOI: 10.1145/3233347.3233373.

[102] T. H. Davenport and R. Ronanki. "Things AI Can Already Do for Your Company". In: *Harvard Business Review* 1 (2018), pp. 108–116. URL: https://hbr.org/2018/01/artificial-intelligence-for-the-real-world.

[103] S. Banga, H. Gehani, S. Bhilare, S. Patel, and L. Kara. *3D Topology Optimization using Convolutional Neural Networks*. 2018.

[104] E. Hamouche and E. G. Loukaides. "Classification and selection of sheet forming processes with machine learning". In: *International Journal of Computer Integrated Manufacturing* 31.9 (2018), pp. 921–932. DOI: 10.1080/0951192X.2018.1429668.

[105] A. Pillai, U. Reuter, and M. Hiele. "Estimation of spot weld design parameters using deep learning". In: *12th European LS-DYNA Conference 2019*. 2019, pp. 1–16.

[106] S. Oh, Y. Jung, S. Kim, I. Lee, and N. Kang. "Deep Generative Design: Integration of Topology Optimization and Generative Models". In: *Journal of Mechanical Design* 141.11 (2019), p. 436. ISSN: 1050-0472. DOI: 10.1115/1.4044229.

[107] H. Haberhauer. *Maschinenelemente*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. DOI: 10.1007/978-3-662-53048-1.

[108] S. A. Choudry, S. Sandmann, and D. Landgrebe. "A Methodical Approach for an Economic Assessment of Joining Technologies under Risk – Optimized Decision-making in Automobile Body Development". In: *Procedia CIRP* 69 (2018), pp. 31–36. ISSN: 22128271. DOI: 10.1016/j.procir.2017.11.045.

[109] M. W. Geda and C. K. Kwong. "Simultaneous consideration of assemblability and disassemblability for fastening method selection". In: *Proceedings of the International Conference on Industrial Engineering and Operations Management*. IEOM Society International, 2018.

[110] Merriam-Webster. *Definition for join*. Springfield, MA, USA, 2021. URL: https://www.merriam-webster.com/dictionary/join (visited on 06/15/2021).

[111] Robert W. Messler, ed. *Joining of Materials and Structures: From Pragmatic Process to Enabling Technology*. Elsevier, 2004. ISBN: 978-0-7506-7757-8. DOI: 10.1016/B978-0-7506-7757-8.X5000-3.

[112] F. C. Campbell. *Joining: Understanding the basics / edited by F.C. Campbell*. Materials Park, Ohio: ASM International, 2011. ISBN: 978-1-61503-825-1.

[113] D. Eggink. "Development of a data model for connection elements". Master thesis. Stuttgart: University of Twente, 2017.

[114] International Organization for Standardization. *Resistance welding – Spot welding of aluminium and aluminium alloys – Weldability, welding and testing*. 2007.

[115] International Organization for Standardization. *Resistance welding – Procedure for spot welding of uncoated and coated low carbon steels*. 2015.

[116] J. Kaspar, S. A. Choudry, and M. Vielhaber. "Concurrent Selection of Material and Joining Technology – Holistically Relevant Aspects and Its Mutual Interrelations in Lightweight Engineering". In: *Procedia CIRP*. Ed. by L. Wang. Vol. 72. 2018, pp. 780–785. DOI: 10.1016/j.procir.2018.03.093.

[117] J. H. Kim, L. S. Wang, K. Putta, P. Haghighi, J. J. Shah, and P. Edwards. "Knowledge based design advisory system for multi-material joining". In: *Journal of Manufacturing Systems* 52.3 (2019), pp. 253–263. ISSN: 02786125. DOI: 10.1016/j.jmsy.2019.03.003.

[118] W. C. Young, R. G. Budynas, A. M. Sadegh, and R. J. Roark. *Roark's formulas for stress and strain*. 8th ed. / Warren Young, Richard Budynas and Ali Sadegh. New York: McGraw-Hill Professional, 2012. ISBN: 9780071742474.

[119] Soetens, F. *Connections (Part 1.1): Eurocode 9. Design of aluminium structures*. Brussels, 2008-2-18. DOI: 10.3403/BSEN1999-1.

[120] L. Issler, H. Ruoß, and P. Häfele. *Festigkeitslehre: Grundlagen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. ISBN: 978-3-540-40705-8. DOI: 10.1007/978-3-540-73485-7.

[121] J. Hou, J. Zhu, J. Wang, and W. Zhang. "Topology optimization of the multi-fasteners jointed structure considering fatigue constraints". In: *International Journal for Simulation and Multidisciplinary Design Optimization* 9 (2018), A4. DOI: 10.1051/smdo/2018003.

[122] A. H. Burr and J. B. Cheatham. *Mechanical analysis and design*. 2nd ed. Englewood Cliffs, N.J.: Prentice Hall and London : Prentice-Hall International, 1995. ISBN: 978-0023172656.

[123] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote. *Engineering Design: A Systematic Approach*. London: Springer-Verlag, 2007. ISBN: 978-1-84628-319-2. DOI: 10.1007/978-1-84628-319-2.

[124] J. Gausemeier and S. Moehringer. "VDI 2206- A New Guideline for the Design of Mechatronic Systems". In: *2nd IFAC Conference on Mechatronic Systems*. Vol. 35. 2002, pp. 785–790. DOI: 10.1016/S1474-6670(17)34035-1.

[125] O. Andersson, D. Semere, A. Melander, M. Arvidsson, and B. Lindberg. "Digitalization of Process Planning of Spot Welding in Body-in-white". In: *Procedia CIRP* 50 (2016), pp. 618–623. ISSN: 22128271. DOI: 10.1016/j.procir.2016.05.082.

[126] H. M Lankarani. "Modeling and characterization of spot weld material configurations for crash analysis". PhD. Wichita State University, 2014. DOI: 10.4172/2168-9792.S1.004.

[127] X. Gu, G. Sun, G. Li, L. Mao, and Q. Li. "A Comparative study on multiobjective reliable and robust optimization for crashworthiness design of vehicle structure". In: *Structural and Multidisciplinary Optimization* 48.3 (2013), pp. 669–684. DOI: 10.1007/s00158-013-0921-x.

[128] R. O. de Alba Alvarez, N. S. Ferguson, and B. R. Mace. "A robust spot weld model for structural vibration analysis". In: *Finite Elements in Analysis and Design* 89 (2014), pp. 1–7. DOI: 10.1016/j.finel.2014.04.010.

[129] M. Ouisse and S. Cogan. "Robust design of spot welds in automotive structures: A decision-making methodology". In: *Mechanical Systems and Signal Processing* 24.4 (2010), pp. 1172–1190. ISSN: 08883270. DOI: 10.1016/j.ymssp.2009.09.012.

[130] Q. I. Bhatti, M. Ouisse, and S. Cogan. "An adaptive optimization procedure for spot-welded structures". In: *Computers & Structures* 89.17-18 (2011), pp. 1697–1711. ISSN: 00457949. DOI: 10.1016/j.compstruc.2011.04.009.

[131] R. Crabtree, N. Baid, and M. S. and Fox. "Design Engineering: Problems in Coordination , Tokyo Japan". In: *Proceedings of the JSME/ASME Design Theory Workshop*. 1993.

[132] F. I. Kubota, J. Hsuan, and P. A. Cauchick-Miguel. "Theoretical analysis of the relationships between modularity in design and modularity in production". In: *The International Journal of Advanced Manufacturing Technology* 89.5-8 (2017), pp. 1943–1958. ISSN: 0268-3768. DOI: 10.1007/s00170-016-9238-4.

[133] M. Felgenhauer, J. Stocker, and M. Lienkamp. "New Approach for Architecture Design of High Variable Vehicle Portfolios". In: *Volume 11: Systems, Design, and Complexity*. ASME, 2017, V011T15A019. ISBN: 978-0-7918-5846-2. DOI: 10.1115/IMECE2017-71155.

[134] F. Alizon, K. Khadke, H. J. Thevenot, J. K. Gershenson, T. J. Marion, S. B. Shooter, and T. W. Simpson. "Frameworks for Product Family Design and Development". In: *Concurrent Engineering* 15.2 (2007), pp. 187–199. DOI: 10.1177/1063293X07079326.

[135] F. S. Wong and D. C. Wynn. "A New Method to Assess Platform Changes Over Successive Generations of Product Variants from Multiple Design Perspectives". In: *Proceedings of the Design Society: International Conference on Engineering Design* 1.1 (2019), pp. 2931–2940. ISSN: 2220-4342. DOI: 10.1017/dsi.2019.300.

[136] S. Bednar, J. Modrak, and Z. Soltysova. "Assessment of Assembly Process Complexity and Modularity in Mass Customized Manufacturing". In: *Geometric Inspection Planning as a Key Element in Industry 4.0*. Ed. by G. Moroni and S. Petrò. 2018, pp. 245–256. ISBN: 978-3-319-56430-2. DOI: 10.1007/978-3-319-56430-2{\textunderscore}17.

[137] S. Mittal and F. Frayman. "Towards a generic model of configuration tasks". In: *11th International Joint Conference on Artificial Intelligence*. Morgan Kaufman Publishers, 1989, pp. 1395–1401.

[138] M. Zhang, G.-x. Li, J.-z. Gong, and B.-z. Wu. "Predicting configuration performance of modular product family using principal component analysis and support vector machine". In: *Journal of Central South University* 21.7 (2014), pp. 2701–2711. ISSN: 2095-2899. DOI: 10.1007/s11771-014-2232-9.

[139] A. O'Brien. *Welding handbook*. 9. ed., 2. print. Miami, Fla.: American Welding Soc, 2007. ISBN: 9780871717290.

[140] G. Boothroyd. "Product design for manufacture and assembly". In: *Computer-Aided Design* 26.7 (1994), pp. 505–520. ISSN: 00104485. DOI: 10.1016/0010-4485(94)90082-5.

[141] M. M. Andreasen, S. Kähler, and T. Lund. *Design for assembly*. Bedford: IFS, 1983. ISBN: 9780903608350.

[142] G. Boothroyd and P. Dewhurst. *Product design for assembly*. Boothroyd Dewhurst Incorporated, 1987.

[143] H. J. Warnecke and R. Bäßler. "Design for Assembly — Part of the Design Process". In: *CIRP Annals* 37.1 (1988), pp. 1–4. ISSN: 00078506. DOI: 10.1016/S0007-8506(07)61572-8.

[144] T. OHASHI, M. IWATA, S. ARIMOTO, and S. MIYAKAWA. "Extended Assemblability Evaluation Method (AEM). Extended Quantitative Assembly Producibility Evaluation for Assembled Parts and Products". In: *JSME International Journal Series C* 45.2 (2002), pp. 567–574. ISSN: 1344-7653. DOI: 10.1299/jsmec.45.567.

[145] M. G. Miller, J. D. Summers, J. L. Mathieson, and G. M. Mocko. "Manufacturing Assembly Time Estimation Using Structural Complexity Metric Trained Artificial Neural Networks". In: *Journal of Computing and Information Science in Engineering* 14.1 (2014), p. 43. ISSN: 15309827. DOI: 10.1115/1.4025809.

[146] J. E. Owensby and J. D. Summers. “Assembly Time Estimation: Assembly Mate Based Structural Complexity Metric Predictive Modeling”. In: *Journal of Computing and Information Science in Engineering* 14.1 (2014), p. 5. ISSN: 15309827. DOI: 10.1115/1.4025808.

[147] V. Chan and F. A. Salustri. *DFA: The Lucas Method*. 2005.

[148] C. Favi and M. Germani. “A method to optimize assemblability of industrial product in early design phase: From product architecture to assembly sequence”. In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 6.3 (2012), pp. 155–169. ISSN: 1955-2513. DOI: 10.1007/s12008-012-0147-y.

[149] F. Rieg and R. Steinhilper. *Handbuch Konstruktion*. 2., aktualisierte Auflage. München: Carl Hanser Verlag, 2018. ISBN: 978-3-446-45224-4.

[150] K.-H. Grote, B. Bender, and D. Göhlich. *Taschenbuch für den Maschinenbau*. 25., neu bearbeitete und aktualisierte Auflage. Berlin: Springer Vieweg, 2018. ISBN: 9783662548042. URL: https://doi.org/10.1007/978-3-662-54805-9.

[151] G. La Rocca. “Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design”. In: *Advanced Engineering Informatics* 26.2 (2012), pp. 159–179. ISSN: 14740346. DOI: 10.1016/j.aei.2012.02.002.

[152] T. L’Eglise, P. de Lit, and P. Fouda. “A multicriteria decision-aid system for joining process selection”. In: *Proceedings of the 4’" IEEE International Symposium on Assembly and Task Planning*. 2001, pp. 324–329. DOI: 10.1109/ISATP.2001.929043.

[153] M. Noryani, S. M. Sapuan, M. T. Mastura, M. Y. M. Zuhri, and E. S. Zainudin. “Material selection criteria for natural fibre composite in automotive component: A review”. In: *IOP Conference Series: Materials Science and Engineering* 368 (2018), p. 012002. ISSN: 1757-8981. DOI: 10.1088/1757-899X/368/1/012002.

[154] H. Prüß, C. Stechert, and T. Vietor. “Methodik Zur Auswahl von Fügetechnologien in Multimaterialsystemen”. In: *Design for X - DfX-Symposium*. Vol. 21. 2010.

[155] J. M. Chien, K. C. McKinstry, C. Baek, A. Horvath, and D. Dornfeld. “Multi-objective Analysis on Joining Technologies”. In: (2012), pp. 287–292. DOI: 10.1007/978-3-642-29069-5{\textunderscore}49.

[156] J. A. Mesa, D. Illera, I. Esparragoza, H. Maury, and H. Gómez. “Functional characterisation of mechanical joints to facilitate its selection during the design of open architecture products”. In: *International Journal of Production Research* 56.24 (2017), pp. 7390–7404. ISSN: 0020-7543. DOI: 10.1080/00207543.2017.1412530.

[157] M. W. Geda, C. K. Kwong, and H. Jiang. “Fastening method selection with simultaneous consideration of product assembly and disassembly from a remanufacturing perspective”. In: *The International Journal of Advanced Manufacturing Technology* 101.5-8 (2019), pp. 1481–1493. ISSN: 0268-3768. DOI: 10.1007/s00170-018-3027-1.

[158] J. Kaspar, S. A. Choudry, and M. Vielhaber. “Systemic Assessment and Selection of Material and Joining Technology Exemplarily Applied on the Automotive Bodywork”. In: *Procedia CIRP* 91.20 (2020), pp. 201–206. ISSN: 22128271. DOI: 10.1016/j.procir.2020.02.168.

[159] C. LeBacq, Y. Brechet, H. Shercliff, T. Jeggy, and L. Salvo. “Selection of joining methods in mechanical design”. In: *Materials & Design* 23.4 (2002), pp. 405–416. ISSN: 02613069. DOI: 10.1016/S0261-3069(01)00093-0.

[160] S. Kadkhoda, A. Hassan, and E. Asadollahi-Yazdi. "Process and resource selection methodology in design for additive manufacturing". In: *The International Journal of Advanced Manufacturing Technology* 104.5-8 (2019), pp. 2013–2029. ISSN: 0268-3768. DOI: 10.1007/s00170-019-03991-w.

[161] M. F. Ashby, Y. Bréchet, D. Cebon, and L. Salvo. "Selection strategies for materials and processes". In: *Materials & Design* 25.1 (2004), pp. 51–67. ISSN: 02613069. DOI: 10.1016/S0261-3069(03)00159-6.

[162] D. Marini and J. R. Corney. "Process selection methodology for near net shape manufacturing". In: *The International Journal of Advanced Manufacturing Technology* 106.5-6 (2020), pp. 1967–1987. ISSN: 0268-3768. DOI: 10.1007/s00170-019-04561-w.

[163] C. Renzi, L. Di Angelo, and F. Leali. "Automotive Design Engineering: Material and Processes Selection Problems". In: *Design Tools and Methods in Industrial Engineering*. Ed. by C. Rizzi, A. O. Andrisano, F. Leali, F. Gherardini, F. Pini, and A. Vergnano. Cham: Springer International Publishing, 2020.

[164] T. Jeandin and C. Mascle. "A New Model to Select Fasteners in Design for Disassembly". In: *Procedia CIRP* 40 (2016), pp. 425–430. ISSN: 22128271. DOI: 10.1016/j.procir.2016.01.084.

[165] A. Chaimae, M. Imad, and A. Asmae. "An Artificial Intelligent based System to Automate Decision Making in Assembly Solution Design". In: *International Journal of Advanced Computer Science and Applications* 11.7 (2020). ISSN: 21565570. DOI: 10.14569/IJACSA.2020.0110772.

[166] M. J. Hoefer and M. C. Frank. "Automated Manufacturing Process Selection During Conceptual Design". In: *Journal of Mechanical Design* 140.3 (2018), p. 031701. ISSN: 1050-0472. DOI: 10.1115/1.4038686.

[167] DIN EN ISO. *Blindniete - Begriffe und Definitionen*. Berlin, 2001. DOI: 10.31030/9130682.

[168] C. Friedrich, H. Hubbertz, H. Roser, and P. Walker. "Towards Automated Design of Fastening Systems for Accelerated Product Development". In: *Volume 2B: Advanced Manufacturing*. ASME, 2014, V02BT02A009. ISBN: 978-0-7918-4644-5. DOI: 10.1115/IMECE2014-38150.

[169] J. Sim and K.-Y. Kim. "Hybrid Nugget Diameter Prediction for Resistance Spot Welding". In: *Procedia Manufacturing* 17 (2018), pp. 395–402. ISSN: 23519789. DOI: 10.1016/j.promfg.2018.10.062.

[170] K.-Y. Kim and F. Ahmed. "Semantic weldability prediction with RSW quality dataset and knowledge construction". In: *Advanced Engineering Informatics* 38.01 (2018), pp. 41–53. ISSN: 14740346. DOI: 10.1016/j.aei.2018.05.006.

[171] Z. Yuguang, X. Kai, and S. Dongyan. "An improved artificial neural network for laser welding parameter selection and prediction". In: *The International Journal of Advanced Manufacturing Technology* 68.1-4 (2013), pp. 755–762. ISSN: 0268-3768. DOI: 10.1007/s00170-013-4796-1.

[172] A. Esawi and M. Ashby. "Computer-based selection of joining processes". In: *Materials & Design* 25.7 (2004), pp. 555–564. ISSN: 02613069. DOI: 10.1016/j.matdes.2004.03.002.

[173] M. Pärttö and P. Saariluoma. "Explaining failures in innovative thought processes in engineering design". In: *Procedia - Social and Behavioral Sciences* 41 (2012), pp. 442–449. ISSN: 18770428. DOI: 10.1016/j.sbspro.2012.04.053.

[174] Cenit. *FastJoint Assembly: Management von Verbindungselementen in Catia*. 2018. URL: http://www.cenit.com/de_DE/plm/digitale-fabrik/software/fastjoint.html (visited on 10/15/2018).

[175] Net. *PTC Creo Intelligent Fastener Extension: Eindämmung der unkontrollierten Vervielfachung von Verbindungselementen*. URL: https://www.net-online.de/ptc-creo-intelligent-fastener-extension-ifx/ (visited on 10/15/2018).

[176] V. Pakalapati, V. Katkar, and R. Babar. "CAE based 'Multi Objective Optimization approach for Spot Weld Connections Layout' in Automotive Structure". In: SAE Technical Paper Series. SAE International400 Commonwealth Drive, Warrendale, PA, United States, 2011. DOI: 10.4271/2011-01-0794.

[177] D. Baxter, J. Gao, K. Case, J. Harding, B. Young, S. Cochrane, and S. Dani. "An engineering design knowledge reuse methodology using process modelling". In: *Research in Engineering Design* 18.1 (2007), pp. 37–48. ISSN: 0934-9839. DOI: 10.1007/s00163-007-0028-8.

[178] S.-K. Hong, B. I. Epureanu, and M. P. Castanier. "Joining of components of complex structures for improved dynamic response". In: *Journal of Sound and Vibration* 331.19 (2012), pp. 4285–4298. DOI: 10.1016/j.jsv.2012.05.004.

[179] D. Guirguis and M. F. Aly. "An evolutionary multi-objective topology optimization framework for welded structures". In: *2016 IEEE Congress on Evolutionary Computation (CEC)*. Piscataway, NJ and Piscataway, NJ: IEEE, 2016, pp. 372–378. ISBN: 978-1-5090-0623-6. DOI: 10.1109/CEC.2016.7743818.

[180] H. Long, Y. Hu, X. Jin, H. Yu, and H. Zhu. "An optimization procedure for spot-welded structures based on SIMP method". In: *Computational Materials Science* 117 (2016), pp. 602–607. ISSN: 09270256. DOI: 10.1016/j.commatsci.2015.08.058.

[181] F. N. Evren and I. Ozkol. "Optimization of connection analysis in aircraft structures". In: *2017 8th International Conference on Mechanical and Aerospace Engineering (ICMAE)*. IEEE, 2017, pp. 552–555. ISBN: 978-1-5386-3305-2. DOI: 10.1109/ICMAE.2017.8038707.

[182] V. Savic and S. Xu. "Fixed weld reduction method for optimal spot weld pattern design". In: *SAE 2003 world congress & exhibition*. SAE, 2003.

[183] H. Hasegawa, H. Sasaki, H. Uehara, and K. Kawamo. "The optimisation of spot-weld positions for vehicle design by using hybrid meta-heuristics". In: *International Journal of Vehicle Design* 43.1-4 (2007), pp. 151–173. ISSN: 0143-3369. DOI: 10.1504/IJVD.2007.012301.

[184] A. H. Ertas and F. O. Sonmez. "Design optimization of spot-welded plates for maximum fatigue life". In: *Finite Elements in Analysis and Design* 47.4 (2011), pp. 413–423. DOI: 10.1016/j.finel.2010.11.003.

[185] J. S. Eom, B. H. Ju, N. Choi, J. M. Park, B. C. Lee, and H. B. Byun. "Optimization on the Spot Welded Configuration of Vehicle Components Considering the Structural Performance". In: *Key Engineering Materials* 326-328 (2006), pp. 957–962. ISSN: 1662-9795. DOI: 10.4028/www.scientific.net/KEM.326-328.957.

[186] G. Geißler and T. Hahn. "Process development for multi-disciplinary spot weld optimization with Cax-loco, Ls-opt and Ansa". In: *4th ANSA & META International Conference*. 2011.

[187] T. Jiang and M. Chirehdast. "A Systems Approach to Structural Topology Optimization: Designing Optimal Connections". In: *Journal of Mechanical Design* 119.1 (1997), p. 40. ISSN: 10500472. DOI: 10.1115/1.2828787.

[188] S. Chavare, K. Thomson, and N. Sharma. "Use of Parametric Approach to Optimize Structural Adhesives and Spot Welds in CAE Based Optimization". In: SAE Technical Paper Series. SAE International400 Commonwealth Drive, Warrendale, PA, United States, 2017. DOI: 10.4271/2017-01-0254.

[189] T. Saito, Y. Tamai, and J. Hiramoto. "A Study of Topology Optimization for Joint Locations of Automotive Full Vehicle". In: *Advances in Structural and Multidisciplinary Optimization*. Ed. by A. Schumacher, T. Vietor, S. Fiebig, K.-U. Bletzinger, and K. Maute. Cham: Springer International Publishing, 2018. DOI: 10.1007/978-3-319-67988-4{\textunderscore}138.

[190] A. Oinonen, P. Tanskanen, T. Björk, and G. Marquis. "Pattern optimization of eccentrically loaded multi-fastener joints". In: *Structural and Multidisciplinary Optimization* 40.1-6 (2010), pp. 597–609. DOI: 10.1007/s00158-009-0392-2.

[191] J. Hou, J. Zhu, F. HE, W. Zhang, and W. GUO. "Stiffeners layout design of thin-walled structures with constraints on multi-fastener joint loads". In: *Chinese Journal of Aeronautics* 30.4 (2017), pp. 1441–1450. ISSN: 10009361. DOI: 10.1016/j.cja.2017.05.005.

[192] O. Ambrozkiewicz and B. Kriegesmann. "Simultaneous topology and fastener layout optimization of assemblies considering joint failure". In: *International Journal for Numerical Methods in Engineering* 122.1 (2021), pp. 294–319. ISSN: 00295981. DOI: 10.1002/nme.6538.

[193] V. Florea, M. Pamwar, B. Sangha, and I. Y. Kim. "3D multi-material and multi-joint topology optimization with tooling accessibility constraints". In: *Structural and Multidisciplinary Optimization* 60.6 (2019), pp. 2531–2558. DOI: 10.1007/s00158-019-02344-1.

[194] M. Eichstetter, S. Müller, and M. Zimmermann. "Product Family Design With Solution Spaces". In: *Journal of Mechanical Design* 137.12 (2015). ISSN: 1050-0472. DOI: 10.1115/1.4031637.

[195] J.-H. Zhu, W.-J. Guo, W.-H. Zhang, and T. Liu. "Integrated layout and topology optimization design of multi-frame and multi-component fuselage structure systems". In: *Structural and Multidisciplinary Optimization* 56.1 (2017), pp. 21–45. DOI: 10.1007/s00158-016-1645-5.

[196] D. Kahneman and G. Klein. "Conditions for intuitive expertise: A failure to disagree". In: *The American psychologist* 64.6 (2009), pp. 515–526. DOI: 10.1037/a0016755.

[197] C. Friedrich, H. Hubbertz, and G. Dinger. "Standardized Calculation of Bolted Joints for Future Requirements". In: *Volume 3: Design and Manufacturing*. ASME, 2011, pp. 475–480. ISBN: 978-0-7918-5489-1. DOI: 10.1115/IMECE2011-63774.

[198] J. Ma and G. E. O. Kremer. "A systematic literature review of modular product design (MPD) from the perspective of sustainability". In: *The International Journal of Advanced Manufacturing Technology* 86.5-8 (2016), pp. 1509–1539. ISSN: 0268-3768. DOI: 10.1007/s00170-015-8290-9.

[199] J. Bonvoisin, F. Halstenberg, T. Buchert, and R. Stark. "A systematic literature review on modular product design". In: *Journal of Engineering Design* 27.7 (2016), pp. 488–514. DOI: 10.1080/09544828.2016.1166482.

[200] K. T. Ulrich and S. D. Eppinger. *Product design and development*. 5th ed. New York: McGraw-Hill Higher Education and London : McGraw-Hill [distributor], 2012. ISBN: 9780073404776.

[201] J. Daaboul, C. Da Cunha, A. Bernard, and F. Laroche. "Design for mass customization: Product variety vs. process variety". In: *CIRP Annals* 60.1 (2011), pp. 169–174. ISSN: 00078506. DOI: 10.1016/j.cirp.2011.03.093.

[202] K. Salonitis. "Modular design for increasing assembly automation". In: *CIRP Annals* 63.1 (2014), pp. 189–192. ISSN: 00078506. DOI: 10.1016/j.cirp.2014.03.100.

[203] A. Jose and M. Tollenaere. "Modular and platform methods for product family design: Literature analysis". In: *Journal of Intelligent Manufacturing* 16.3 (2005), pp. 371–390. ISSN: 0956-5515. DOI: 10.1007/s10845-005-7030-7.

[204] W.-Y. Liang and C.-C. Huang. "The agent-based collaboration information system of product development". In: *International Journal of Information Management* 22.3 (2002), pp. 211–224. ISSN: 02684012. DOI: 10.1016/S0268-4012(02)00006-3.

[205] A. M. Shaik, V. V. S. K. Rao, and C. S. Rao. "Development of modular manufacturing systems—a review". In: *The International Journal of Advanced Manufacturing Technology* 76.5-8 (2015), pp. 789–802. ISSN: 0268-3768. DOI: 10.1007/s00170-014-6289-2.

[206] K. Ulrich. "The role of product architecture in the manufacturing firm". In: *Research Policy* 24.3 (1995), pp. 419–440. ISSN: 00487333. DOI: 10.1016/0048-7333(94)00775-3.

[207] S. K. Chandrasegaran, K. Ramani, R. D. Sriram, I. Horváth, A. Bernard, R. F. Harik, and W. Gao. "The evolution, challenges, and future of knowledge representation in product design systems". In: *Computer-Aided Design* 45.2 (2013), pp. 204–228. ISSN: 00104485. DOI: 10.1016/j.cad.2012.08.006.

[208] G. Gereffi, J. Humphrey, and T. Sturgeon. "The governance of global value chains". In: *Review of International Political Economy* 12.1 (2005), pp. 78–104. ISSN: 0969-2290. DOI: 10.1080/09692290500049805.

[209] M. H. Meyer and A. P. Lehnerd. *The power of product platforms: Building value and cost leadership*. New York and London: Free Press, 1997. ISBN: 9780684825809.

[210] U. Harlou. *Developing product families based on architectures: Contribution to a theory of product families*. Lyngby: Department of Mechanical Engineering, Technical University of Denmark, 2006. ISBN: 978-87-90130-07-7.

[211] K. Otto, K. Hölttä-Otto, T. W. Simpson, D. Krause, S. Ripperda, and S. Ki Moon. "Global Views on Modular Design Research: Linking Alternative Methods to Support Modular Product Family Concept Development". In: *Journal of Mechanical Design* 138.7 (2016), p. 071101. ISSN: 1050-0472. DOI: 10.1115/1.4033654.

[212] J. Park and T. Simpson. "Toward an activity-based costing system for product families and product platforms in the early stages of development". In: *International Journal of Production Research* 46 (2008), pp. 99–130. ISSN: 0020-7543.

[213] S. Jung and T. W. Simpson. "An integrated approach to product family redesign using commonality and variety metrics". In: *Research in Engineering Design* 27.4 (2016), pp. 391–412. ISSN: 0934-9839. DOI: 10.1007/s00163-016-0224-5.

[214] Z. Liu, Y. S. Wong, and K. S. Lee. "Modularity analysis and commonality design: A framework for the top-down platform and product family design". In: *International Journal of Production Research* 48.12 (2009), pp. 3657–3680. ISSN: 0020-7543. DOI: 10.1080/00207540902902598.

[215] J. Pakkanen, P. Huhtala, T. Juuti, and T. Lehtonen. "Achieving Benefits with Design Reuse in Manufacturing Industry". In: *Procedia CIRP* 50 (2016), pp. 8–13. ISSN: 22128271. DOI: 10.1016/j.procir.2016.04.173.

[216] F. Borjesson and K. Hölttä-Otto. "A module generation algorithm for product architecture based on component interactions and strategic drivers". In: *Research in Engineering Design* 25.1 (2014), pp. 31–51. ISSN: 0934-9839. DOI: 10.1007/s00163-013-0164-2.

[217] M. V. Martin and K. Ishii. "Design for variety: Developing standardized and modularized product platform architectures". In: *Research in Engineering Design* 13.4 (2002), pp. 213–235. ISSN: 0934-9839. DOI: 10.1007/s00163-002-0020-2.

[218] M. Li, Y. Zhang, B. Zeng, H. Zhou, and J. Liu. "The modified firefly algorithm considering fireflies' visual range and its application in assembly sequences planning". In: *The International Journal of Advanced Manufacturing Technology* 82.5-8 (2016), pp. 1381–1403. ISSN: 0268-3768. DOI: 10.1007/s00170-015-7457-8.

[219] J. Jiao and M. M. Tseng. "A methodology of developing product family architecture for mass customization". In: *Journal of Intelligent Manufacturing* 10.1 (1999), pp. 3–20. ISSN: 0956-5515. DOI: 10.1023/A:1008926428533.

[220] F. J. Emmaty and S. P. Sarmah. "Modular product development through platform-based design and DFMA". In: *Journal of Engineering Design* 23.9 (2012). DOI: 10.1080/09544828.2011.653330.

[221] C.-S. Wang. "Web-based modular interface geometries with constraints in assembly models". In: *Computers & Industrial Engineering* 56.4 (2009), pp. 1675–1686. ISSN: 03608352. DOI: 10.1016/j.cie.2008.10.018.

[222] H.-E. Tseng, T.-S. Chang, and Y.-C. Yang. "A connector-based approach to the modular formulation problem for a mechanical product". In: *The International Journal of Advanced Manufacturing Technology* 24 (2004), pp. 161–171. DOI: 10.1007/s00170-003-1656-4.

[223] V. Godthi, J. J.-W. Yoo, C. Bryant Arnold, T. W. Simpson, and S. Kumara. "Product Design Using Interface-Based Module Description". In: *Volume 5: 22nd International Conference on Design Theory and Methodology; Special Conference on Mechanical Vibration and Noise*. ASMEDC, 2010, pp. 47–58. ISBN: 978-0-7918-4413-7. DOI: 10.1115/DETC2010-28850.

[224] L. Li, G. Q. Huang, and S. T. Newman. "Interweaving genetic programming and genetic algorithm for structural and parametric optimization in adaptive platform product customization". In: *Robotics and Computer-Integrated Manufacturing* 23.6 (2007), pp. 650–658. ISSN: 07365845. DOI: 10.1016/j.rcim.2007.02.014.

[225] G. Schuh, S. Rudolf, and T. Vogels. "Development of Modular Product Architectures". In: *Procedia CIRP* 20 (2014), pp. 120–125. ISSN: 22128271. DOI: 10.1016/j.procir.2014.05.042.

[226] E. Cerqueira, R. Voltolini, and M. Borsato. "Product Platform: Trends and Opportunities". In: *Transdisciplinary Engineering for Complex Socio-technical Systems*. Ed. by K. Hiekata et al. IOS Press, 2019. DOI: 10.3233/ATDE190126.

[227] H. ElMaraghy. "Changing and Evolving Products and Systems: Models and Enablers". In: *Changeable and Reconfigurable Manufacturing Systems*. Ed. by H. A. ElMaraghy. London: Springer London, 2009, pp. 22–45.

[228] T. Fujimoto. *Competing to be really, really good: The behind-the-scenes drama of capability-building competition in the automobile industry / Takahiro Fujimoto ; translated by Brian Miller*. 1st English ed. Tokyo: International House of Japan, 2007. ISBN: 9784924971219.

[229] A. Ericsson and G. Erixon. *Controlling design variants: Modular product platforms / by Anna Ericsson, Gunnar Erixon*. Dearborn, MI: Society of Manufacturing Engineers, 1999. ISBN: 978-0872635142.

[230] M. M. Tseng, Y. Wang, and R. J. Jiao. "Mass Customization". In: *CIRP Encyclopedia of Production Engineering*. Ed. by S. Chatti and T. Tolio. Berlin, Heidelberg: Springer Berlin Heidelberg and Imprint: Springer, 2020. ISBN: 978-3-642-35950-7. DOI: 10.1007/978-3-642-35950-7{\textunderscore}16701-3.

[231] S. E. Ahnert, I. G. Johnston, T. M. A. Fink, J. P. K. Doye, and A. A. Louis. "Self-assembly, modularity, and physical complexity". In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 82.2 (2010). DOI: 10.1103/PhysRevE.82.026117.

[232] S. K. Moon, K. J. Park, and T. W. Simpson. "Platform design variable identification for a product family using multi-objective particle swarm optimization". In: *Research in Engineering Design* 25.2 (2014), pp. 95–108. ISSN: 0934-9839. DOI: 10.1007/s00163-013-0166-0.

[233] K. N. Otto and K. L. Wood. *Product design: Techniques in reverse engineering and new product development / Kevin N. Otto, Kristin L. Wood*. Upper Saddle River, N.J.: Prentice Hall and London : Prentice-Hall International, 2001. ISBN: 0-13-021271-7.

[234] M. V. Martin and K. Ishii. "Design for Variety: A methodology for understanding the costs of product proliferation". In: *The 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*. Irvine, California, 1996.

[235] Z. Pirmoradi and G. G. Wang. "A Review of Recent Literature in Product Family Design and Platform-Based Product Development". In: *Advances in Product Family and Product Platform Design*. Ed. by T. W. Simpson, J. Jianxin, Z. Siddique, and K. Hölttä-Otto. New York Heidelberg Dordrecht London: Springer, 2014, pp. 1–46. ISBN: 978-1-4614-7936-9.

[236] Z. Siddique and D. W. Rosen. "On the applicability of product variety design concepts to automotive platform commonality". In: *1998 ASME Design Engineering Technical Conferences*. Atlanta, Georgia, 1998.

[237] A. N. Ukala and F. T. Sunmola. "A Rule-Based Approach for Product Assembly Complexity Review in the Context of Virtual Engineering". In: *Procedia Manufacturing* 51.11 (2020), pp. 557–564. ISSN: 23519789. DOI: 10.1016/j.promfg.2020.10.078.

[238] S. Chowdhury and Z. Siddique. "Shape Commonalization to Develop Common Platform for Product Family". In: *Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Engineering Technical Conferences & Information in Engineering Conference* (2009).

[239] J. Hsuan and P. K. Hansen. "Platform development: Implications for portfolio management". In: *Gestão & Produção* 14.3 (2007), pp. 453–461. DOI: 10.1590/S0104-530X2007000300003.

[240] K. Ulrich and K. Tung. "Fundamentals of Product Modularity". In: *Proceedings of the 1991 ASME Winter Annual Meeting Symposium on Issues in Design/Manufacturing Integration*. Atlanta, 1991, pp. 45–73.

[241] D. Braha. “Partitioning Tasks to Product Development Teams”. In: *Volume 4: 14th International Conference on Design Theory and Methodology, Integrated Systems Design, and Engineering Design and Culture*. American Society of Mechanical Engineers, 2002, pp. 333–344. ISBN: 0-7918-3624-X. DOI: 10.1115/DETC2002/DTM-34031.

[242] P. J. Newcomb, B. Bras, and D. W. Rosen. “Implications of Modularity on Product Design for the Life Cycle”. In: *Journal of Mechanical Design* 120.3 (1998), pp. 483–490. ISSN: 10500472. DOI: 10.1115/1.2829177.

[243] D. Campagnolo and A. Camuffo. “The concept of modularity in management studies: A literature review”. In: *International Journal of Management Reviews* 105.Special Issue (2009), p. 1. ISSN: 14608545. DOI: 10.1111/j.1468-2370.2009.00260.x.

[244] G. Erixon. *Modular function deployment: A method for product modularisation*. Vol. 98,1. TRITA-MSM. Stockholm: The Royal Inst. of Technology, Dept. of Manufacturing Systems, Assembly Systems Division, 1998. ISBN: 9789925736058.

[245] C. Blees. *Eine Methode zur Entwicklung modularer Produktfamilien*. 1. Aufl. Vol. Bd. 3. Hamburger Schriftenreihe Produktentwicklung und Konstruktionstechnik. Hamburg: TuTech-Verl., 2011. ISBN: 978-3-941492-35-6.

[246] K. Sinha and E. S. Suh. “Pareto-optimization of complex system architecture for structural complexity and modularity”. In: *Research in Engineering Design* 29.1 (2018), pp. 123–141. ISSN: 0934-9839. DOI: 10.1007/s00163-017-0260-9.

[247] J. Pandremenos, J. Paralikas, K. Salonitis, and G. Chryssolouris. “Modularity concepts for the automotive industry: A critical review”. In: *CIRP Journal of Manufacturing Science and Technology* 1.3 (2009), pp. 148–152. ISSN: 17555817. DOI: 10.1016/j.cirpj.2008.09.012.

[248] S. Hankammer and F. Steiner. “Leveraging the Sustainability Potential of Mass Customization through Product Service Systems in the Consumer Electronics Industry”. In: vol. 30, pp. 504–509. DOI: 10.1016/j.procir.2015.03.007.

[249] F. Zirpoli and M. C. Becker. “The limits of design and engineering outsourcing: Performance integration and the unfulfilled promises of modularity”. In: *R&D Management* 41.1 (2011), pp. 21–43. ISSN: 00336807. DOI: 10.1111/j.1467-9310.2010.00629.x.

[250] Z.-k. Li, S. Wang, and W.-w. Yin. “Determining optimal granularity level of modular product with hierarchical clustering and modularity assessment”. In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 41.8 (2019), p. 536. ISSN: 1678-5878. DOI: 10.1007/s40430-019-1848-y.

[251] K. Sinha, E. S. Suh, and O. de Weck. “Integrative Complexity: An Alternative Measure for System Modularity”. In: *Journal of Mechanical Design* 140.5 (2018), p. 123. ISSN: 1050-0472. DOI: 10.1115/1.4039119.

[252] Y. Asaga and H. Nishigaki. “Modularization Method Based on New Layout Design in Conceptual Design Stage: Application of Multi-material Lightweight Structures Utilizing Machine Learning”. In: *Proceedings of the 21st International Dependency and Structure Modeling Conference, DSM 2019*. Ed. by H. Stowe, T. R. Browning, S. D. Eppinger, J. Tran, and P. Montijo. Monterey and United States: Design Society, 2019. ISBN: 978-191225406-4. DOI: 10.35199/dsm2019.5.

[253] A. Chaimae, A. Asmae, and M. Imad. “An Integrated Product-Process Design Approach Considering Material Selection and Product Assembly”. In: *Journal of Advanced Manufacturing Systems* 19.04 (2020), pp. 675–699. ISSN: 0219-6867. DOI: 10.1142/S0219686720500389.

[254] R. Helmer, A. Yassine, and C. Meier. "Systematic module and interface definition using component design structure matrix". In: *Journal of Engineering Design* 21.6 (2010), pp. 647–675. DOI: 10.1080/09544820802563226.

[255] S. Jung and T. W. Simpson. "New modularity indices for modularity assessment and clustering of product architecture". In: *Journal of Engineering Design* 28.1 (2016), pp. 1–22. DOI: 10.1080/09544828.2016.1252835.

[256] A. Shamsuzzoha, S. Piya, M. Al-Kindi, and N. Al-Hinai. "Metrics of product modularity: Lessons learned from case companies". In: *Journal of Modelling in Management* 13.2 (2018), pp. 331–350. ISSN: 1746-5664. DOI: 10.1108/JM2-12-2016-0127.

[257] K. Hölttä-Otto, N. A. Chiriac, D. Lysy, and E. Suk Suh. "Comparative analysis of coupling modularity metrics". In: *Journal of Engineering Design* 23.10-11 (2012), pp. 790–806. DOI: 10.1080/09544828.2012.701728.

[258] Z. Soltysova, S. Bednar, and A. Behunova. "Complexity Approaches to Complexity Measurement of the Production Process". In: *Proceedings of the SmartCity360 2016*. Ed. by M. Balog, L. Knapcikova, J. Soviar, P. Dorcak, F. Pollak, D. Caganova, P. Fazio, and K. Aydin. EAI, 2017. ISBN: 978-1-63190-149-2. DOI: 10.4108/eai.14-2-2017.152169.

[259] H. G. Wahdan and S. S. Kassem. "Product Modularization Using Cuckoo Search Algorithm". In: *Operations Research and Enterprise Systems*. Ed. by B. Vitoriano and G. H. Parlier. Cham: Springer International Publishing, 2017.

[260] C. Y. Baldwin and K. B. Clark. "Managing in an age of modularity". In: *Harvard Business Review* 75.5 (1997), pp. 84–93.

[261] J. Venn. *Symbolic logic*. 2nd rev. and rewritten ed. Providence, R.I.: AMS Chelsea, 2006. ISBN: 0821841998.

[262] T. Bahns, G. Beckmann, N. Gebhardt, and D. Krause. "Sustainability of modular product families". In: *International conference on engineering design*. Milan, Italy, 2015.

[263] O. L. de Weck, C. Eckert, and P. J. Clarkson. "A Classification of Uncertainty for Early Product and System Design". In: *Proceedings of ICED 2007, the 16th International Conference on Engineering Design*. 2007.

[264] P. J. Rathnow. *Integriertes Variantenmanagement: Bestimmung, Realisierung und Sicherung der optimalen Produktvielfalt*. Vol. Bd. 20. Innovative Unternehmensführung. Göttingen: Vandenhoeck & Ruprecht, 1993. ISBN: 9783525125694.

[265] R. Dellanoi. "Kommunalitäten bei der Entwicklung variantenreicher Produktfamilien". Dissertation. St. Gallen University, 2006.

[266] H. A. ElMaraghy, ed. *Changeable and Reconfigurable Manufacturing Systems*. London: Springer London, 2009. DOI: 10.1007/978-1-84882-067-8.

[267] K. Sinha, S.-.-Y. Han, and E. S. Suh. "Design structure matrix–based modularization approach for complex systems with multiple design constraints". In: *Systems Engineering* 23.2 (2020), pp. 211–220. ISSN: 1098-1241. DOI: 10.1002/sys.21518.

[268] T. Nakane, N. Bold, H. Sun, X. Lu, T. Akashi, and C. Zhang. "Application of evolutionary and swarm optimization in computer vision: A literature survey". In: *IPSJ Transactions on Computer Vision and Applications* 12.1 (2020), p. 394. ISSN: 1882-6695. DOI: 10.1186/s41074-020-00065-9.

[269] A. Slowik and H. Kwasnicka. "Evolutionary algorithms and their applications to engineering problems". In: *Neural Computing and Applications* 32.16 (2020), pp. 12363–12379. ISSN: 0941-0643. DOI: 10.1007/s00521-020-04832-8.

[270] M. Abd Elaziz, A. Dahou, L. Abualigah, L. Yu, M. Alshinwan, A. M. Khasawneh, and S. Lu. "Advanced metaheuristic optimization techniques in applications of deep neural networks: A review". In: *Neural Computing and Applications* 5.4 (2021), p. 115. ISSN: 0941-0643. DOI: 10.1007/s00521-021-05960-5.

[271] N. A. Zolpakar, M. F. Yasak, and S. Pathak. "A review: Use of evolutionary algorithm for optimisation of machining parameters". In: *The International Journal of Advanced Manufacturing Technology* 115.1-2 (2021), pp. 31–47. ISSN: 0268-3768. DOI: 10.1007/s00170-021-07155-7.

[272] M. Mohammadi, M. Nastaran, and A. Sahebgharani. "Development, application, and comparison of hybrid meta-heuristics for urban land-use allocation optimization: Tabu search, genetic, GRASP, and simulated annealing algorithms". In: *Computers, Environment and Urban Systems* 60 (2016), pp. 23–36. ISSN: 01989715. DOI: 10.1016/j.compenvurbsys.2016.07.009.

[273] J. Schwaab, K. Deb, E. Goodman, S. Lautenbach, M. J. van Strien, and A. Grêt-Regamey. "Improving the performance of genetic algorithms for land-use allocation problems". In: *International Journal of Geographical Information Science* 32.5 (2018), pp. 907–930. ISSN: 1365-8816. DOI: 10.1080/13658816.2017.1419249.

[274] G. Atali, İ. Pehlivan, B. Gürevin, and H. İ. Seker. "Chaos in metaheuristic based artificial intelligence algorithms: A short review". In: *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES* (2021), pp. 1354–1367. ISSN: 13036203. DOI: 10.3906/elk-2102-5.

[275] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, 2016. ISBN: 9780262035613.

[276] C. Pritlove, C. Juando-Prats, K. Ala-leppilampi, and J. A. Parsons. "The good, the bad, and the ugly of implicit bias". In: *The Lancet* 393.10171 (2019), pp. 502–504. ISSN: 01406736. DOI: 10.1016/S0140-6736(18)32267-0.

[277] W. Zhang, Y. Zhou, and J. Zhu. "A comprehensive study of feature definitions with solids and voids for topology optimization". In: *Computer Methods in Applied Mechanics and Engineering* 325 (2017), pp. 289–313. ISSN: 00457825. DOI: 10.1016/j.cma.2017.07.004.

[278] L. Wilkinson. "Tree Structured Data Analysis : AID , CHAID and CART". In: *Sawtooth/SYSTAT Joint Software Conference*. Sun Valley, ID, USA, 1992.

[279] K. P. Murphy. *Machine learning: A probabilistic perspective / Kevin P. Murphy*. Adaptive computation and machine learning series. Cambridge, Mass. and London: MIT Press, 2012. ISBN: 9780262018029.

[280] T. G. Dietterich. "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization". In: *Machine Learning* 40.2 (2000), pp. 139–157. ISSN: 08856125. DOI: 10.1023/A:1007607513941.

[281] R. E. Schapire. "The strength of weak learnability". In: *Machine Learning* 5.2 (1990), pp. 197–227. ISSN: 08856125. DOI: 10.1007/BF00116037.

[282] L. Breiman. "Bagging predictors". In: *Machine Learning* 24.2 (1996), pp. 123–140. ISSN: 08856125. DOI: 10.1007/BF00058655.

[283] L. Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 08856125. DOI: 10.1023/A:1010933404324.

[284] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. New York, NY: Springer New York, 2009. ISBN: 978-0-387-84857-0. DOI: 10.1007/b94608.

[285] T. Chen and C. Guestrin. *XGBoost: A Scalable Tree Boosting System*. 2016. DOI: 10.1145/2939672.2939785. URL: http://arxiv.org/pdf/1603.02754v3.

[286] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. "What is the best multi-stage architecture for object recognition?" In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 2146–2153. ISBN: 978-1-4244-4420-5. DOI: 10.1109/ICCV.2009.5459469.

[287] G. E. Hinton, S. Osindero, and Y.-W. Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554. ISSN: 0899-7667. DOI: 10.1162/neco.2006.18.7.1527.

[288] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. "Extracting and composing robust features with denoising autoencoders". In: *Proceedings, Twenty-fifth International Conference on Machine Learning*. Ed. by A. McCallum and S. Roweis. Helsinki, Finland: University of Helsinki], 2008, pp. 1096–1103. ISBN: 978-160558205-4.

[289] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. *A Review on Deep Learning Techniques Applied to Semantic Segmentation*. 2017. URL: http://arxiv.org/pdf/1704.06857v1.

[290] G. Leivas Oliveira. "Encoder-decoder methods for semantic segmentation: efficiency and robustness aspects". Dissertation. 2019. DOI: 10.6094/UNIFR/150065.

[291] D. P. Kingma and M. Welling. *Auto-Encoding Variational Bayes*. 2014. URL: http://arxiv.org/pdf/1312.6114v10.

[292] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. ISSN: 00189219. DOI: 10.1109/5.726791.

[293] M. Peemen. *Convolutional neural network to classify a traffic sign*. 2021. URL: https://developer.nvidia.com/discover/convolutional-neural-network (visited on 06/14/2021).

[294] S. Ruder. *An overview of gradient descent optimization algorithms*. 2016. URL: https://arxiv.org/abs/1609.04747.

[295] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2014. URL: http://arxiv.org/pdf/1412.6980v9.

[296] G. Jurman and C. Furlanello. "A unifying view for performance measures in multi-class prediction". In: *PloS one* 7.8 (2012). DOI: 10.1371/journal.pone.0041882. URL: http://arxiv.org/pdf/1008.2908v1.

[297] R. Delgado and X.-A. Tibau. "Why Cohen's Kappa should be avoided as performance measure in classification". In: *PloS one* 14.9 (2019), e0222916. DOI: 10.1371/journal.pone.0222916.

[298] D. Chicco and G. Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". In: *BMC genomics* 21.1 (2020), p. 6. DOI: 10.1186/s12864-019-6413-7.

[299] U. Maulik and S. Bandyopadhyay. "Performance evaluation of some clustering algorithms and validity indices". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.12 (2002), pp. 1650–1654. ISSN: 01628828. DOI: 10.1109/TPAMI.2002.1114856.

[300] M. Z. Rodriguez, C. H. Comin, D. Casanova, O. M. Bruno, D. R. Amancio, L. d. F. Costa, and F. A. Rodrigues. "Clustering algorithms: A comparative approach". In: *PloS one* 14.1 (2019), e0210236. DOI: 10.1371/journal.pone.0210236.

[301] D. Burdick, M. Calimlim, and J. Gehrke. "MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases". In: *Proceedings of the 17th International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 443–452.

[302] L. Parsons, E. Haque, and H. Liu. "Subspace clustering for high dimensional data". In: *ACM SIGKDD Explorations Newsletter* 6.1 (2004), pp. 90–105. ISSN: 1931-0145. DOI: 10.1145/1007730.1007731.

[303] K. Liu, D. Detwiler, and A. Tovar. "Optimal Design of Nonlinear Multimaterial Structures for Crashworthiness Using Cluster Analysis". In: *Journal of Mechanical Design* 139.10 (2017), p. 101401. ISSN: 1050-0472. DOI: 10.1115/1.4037620.

[304] J. MacQueen. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, CA: University of California Press, pp. 281–297.

[305] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. "Top 10 algorithms in data mining". In: *Knowledge and Information Systems* 14.1 (2008), pp. 1–37. ISSN: 0219-1377. DOI: 10.1007/s10115-007-0114-2.

[306] G. J. Williams and Z. Huang. "Mining the knowledge mine". In: *Australian Joint Conference on Artificial Intelligence*. Springer, 1997, pp. 340–348.

[307] K. Sequeira and M. Zaki. "ADMIT: anomaly-based data mining for intrusions". In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, pp. 386–395.

[308] A. K. Jain. "Data clustering: 50 years beyond K-means". In: *Pattern Recognition Letters* 31.8 (2010), pp. 651–666. ISSN: 01678655. DOI: 10.1016/j.patrec.2009.09.011.

[309] D. Steinley. "K-means clustering: A half-century synthesis". In: *The British journal of mathematical and statistical psychology* 59.Pt 1 (2006), pp. 1–34. ISSN: 0007-1102. DOI: 10.1348/000711005X48266.

[310] R. Redner and H. Walker. "Mixture densities, maximum likelihood and the em algorithm". In: *SIAM Review* 26.6 (1984).

[311] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm." In: *Journal of the Royal Statistical Society Series B* 39.6 (1977).

[312] C. C. Aggarwal and C. K. Reddy. "Data Clustering: Algorithms and Applications". In: 2 (2013). DOI: 10.1201/b15410.

[313] C. Fraley. "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis". In: *The Computer Journal* 41.8 (1998), pp. 578–588. ISSN: 0010-4620. DOI: 10.1093/comjnl/41.8.578.

[314] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. In: *Journal of Intelligent Information Systems* 17.2/3 (2001), pp. 107–145. ISSN: 09259902. DOI: 10.1023/A:1012801612483.

[315] P. Jaccard. "Nouvelles recherches sur la distribution florale". In: *Bulletin de la Sociète Vaudense des Sciences* 44 (1908), pp. 223–270.

[316] L. Hubert and P. Arabie. "Comparing partitions". In: *Journal of Classification* 2.1 (1985), pp. 193–218. ISSN: 0176-4268. DOI: 10.1007/BF01908075.

[317] E. B. Fowlkes and C. L. Mallows. "A Method for Comparing Two Hierarchical Clusterings". In: *Journal of the American Statistical Association* 78.383 (1983), pp. 553–569.

[318] C. Tang, Y. Ling, X. Yang, W. Jin, and C. Zheng. "Multi-View Object Detection Based on Deep Learning". In: *Applied Sciences* 8.9 (2018), p. 1423. ISSN: 2076-3417. DOI: 10.3390/app8091423.

[319] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. URL: http://arxiv.org/pdf/1506.02640v5.

[320] P. J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 03770427. DOI: 10.1016/0377-0427(87)90125-7.

[321] D. J. Ketchen and C. L. Shook. "The application of cluster analysis in strategic management research: An analysis and critique". In: *Strategic Management Journal* 17.6 (1996), pp. 441–458. ISSN: 0143-2095.

[322] R. L. Thorndike. "Who belongs in the family?" In: *Psychometrika* 18.4 (1953), pp. 267–276. ISSN: 0033-3123. DOI: 10.1007/BF02289263.

[323] L. H. Nguyen and S. Holmes. "Ten quick tips for effective dimensionality reduction". In: *PLoS computational biology* 15.6 (2019), e1006907. DOI: 10.1371/journal.pcbi.1006907.

[324] S. Ayesha, M. K. Hanif, and R. Talib. "Overview and comparative study of dimensionality reduction techniques for high dimensional data". In: *Information Fusion* 59.3 (2020), pp. 44–58. ISSN: 15662535. DOI: 10.1016/j.inffus.2020.01.005.

[325] J. P. Cunningham and Z. Ghahramani. "Linear Dimensionality Reduction: Survey, Insights, and Generalizations". In: *Journal of Machine Learning Research* 16 (2015), pp. 2859–2900.

[326] D. Ting and M. I. Jordan. *On Nonlinear Dimensionality Reduction, Linear Smoothing and Autoencoding*. 2018. URL: http://arxiv.org/pdf/1803.02432v1.

[327] S. Wold, K. Esbensen, and P. Geladi. "Principal component analysis". In: *Chemometrics and Intelligent Laboratory Systems* 2.1-3 (1987), pp. 37–52. ISSN: 01697439. DOI: 10.1016/0169-7439(87)80084-9.

[328] J. Shlens. *A Tutorial on Principal Component Analysis*. 2014. URL: http://arxiv.org/pdf/1404.1100v1.

[329] L. van der Maaten and G. Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[330] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. *Generative Adversarial Networks*. 2014. URL: http://arxiv.org/pdf/1406.2661v1.

[331] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha. *StarGAN v2: Diverse Image Synthesis for Multiple Domains*. Apr. 12, 2019. URL: http://arxiv.org/pdf/1912.01865v2.

[332] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. "Generative Adversarial Networks: An Overview". In: *IEEE Signal Processing Magazine* 35.1 (2018), pp. 53–65. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2765202. URL: http://arxiv.org/pdf/1710.07035v1.

[333] A. Kusiak. "Convolutional and generative adversarial neural networks in manufacturing". In: *International Journal of Production Research* 58.5 (2020), pp. 1594–1604. ISSN: 0020-7543. DOI: 10.1080/00207543.2019.1662133.

[334] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. "Image-to-Image Translation with Conditional Adversarial Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5967–5976. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.632.

[335] L. Wang, W. Chen, W. Yang, F. Bi, and F. R. Yu. "A State-of-the-Art Review on Image Synthesis With Generative Adversarial Networks". In: *IEEE Access* 8 (2020), pp. 63514–63537. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2982224.

[336] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2242–2251. ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.244.

[337] M.-Y. Liu, T. Breuel, and J. Kautz. *Unsupervised Image-to-Image Translation Networks*. 2017.

[338] W. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. "Multimodal Unsupervised Image-to-image Translation". In: *Computer Vision - ECCV 2018*. Ed. by V. Ferrari. LNCS sublibrary: SL6 - Image processing, computer vision, pattern recognition, and graphics. Cham, Switzerland: Springer, 2018, pp. 172–189. ISBN: 978-3-030-01246-5.

[339] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang. "DRIT++: Diverse Image-to-Image Translation via Disentangled Representations". In: *International Journal of Computer Vision* 128.10-11 (2020), pp. 2402–2417. ISSN: 0920-5691. DOI: 10.1007/s11263-019-01284-z.

[340] W. Wu, K. Cao, C. Li, C. Qian, and C. C. Loy. "TransGaGa: Geometry-Aware Unsupervised Image-To-Image Translation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 8004–8013. ISBN: 978-1-7281-3293-8. DOI: 10.1109/CVPR.2019.00820.

[341] Y.-J. Lin, P.-W. Wu, C.-H. Chang, E. Chang, and S.-W. Liao. "RelGAN: Multi-Domain Image-to-Image Translation via Relative Attributes". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 5913–5921. ISBN: 978-1-7281-4803-8. DOI: 10.1109/ICCV.2019.00601.

[342] T. L. Sung and H. J. Lee. "Image-to-Image Translation Using Identical-Pair Adversarial Networks". In: *Applied Sciences* 9.13 (2019), p. 2668. ISSN: 2076-3417. DOI: 10.3390/app9132668.

[343] H. Chang, J. Lu, F. Yu, and A. Finkelstein. "PairedCycleGAN: Asymmetric Style Transfer for Applying and Removing Makeup". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 40–48. ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00012.

[344] W. Cho, S. Choi, D. K. Park, I. Shin, and J. Choo. "Image-To-Image Translation via Group-Wise Deep Whitening-And-Coloring Transformation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 10631–10639. ISBN: 978-1-7281-3293-8. DOI: 10.1109/CVPR.2019.01089.

[345] L. Ma, X. Jia, S. Georgoulis, T. Tuytelaars, and L. van Gool. "Exemplar guided unsupervised image-to-image translation with semantic consistency". In: *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*. 2019.

[346] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. "Semantic Image Synthesis With Spatially-Adaptive Normalization". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 2332–2341. ISBN: 978-1-7281-3293-8. DOI: 10.1109/CVPR.2019.00244.

[347] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz. "Few-Shot Unsupervised Image-to-Image Translation". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 10550–10559. ISBN: 978-1-7281-4803-8. DOI: 10.1109/ICCV.2019.01065.

[348] X. Yu, Y. Chen, T. Li, S. Liu, and G. Li. "Multi-mapping Image-to-Image Translation via Learning Disentanglement". In: *Advances in Neural Information Processing Systems 32*. 2019. URL: https://arxiv.org/abs/1909.07877.

[349] Economidis, N., Franke, C., and Golumba, J. *xMCF – A Standard for Describing Connections and Joints in the Automotive Industry v3.0*. Berlin, Germany, 2016.

[350] E. Ahmed, A. Saint, A. E. R. Shabayek, K. Cherenkova, R. Das, G. Gusev, D. Aouada, and B. Ottersten. *A survey on Deep Learning Advances on Different 3D Data Representations*. 2018. URL: http://arxiv.org/pdf/1808.01462v2.

[351] A. S. Gezawa, Y. Zhang, Q. Wang, and L. Yunqi. "A Review on Deep Learning Approaches for 3D Data Representations in Retrieval and Classifications". In: *IEEE Access* 8 (2020), pp. 57566–57593. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2982196.

[352] Y.-P. Xiao, Y.-K. Lai, F.-L. Zhang, C. Li, and L. Gao. "A survey on deep geometry learning: From a representation perspective". In: *Computational Visual Media* 6.2 (2020), pp. 113–133. ISSN: 2096-0433. DOI: 10.1007/s41095-020-0174-8.

[353] W. Guo, J. Wang, and S. Wang. "Deep Multimodal Representation Learning: A Survey". In: *IEEE Access* 7 (2019), pp. 63373–63394. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2916887.

[354] C. Shen, H. R. Roth, H. Oda, M. Oda, Y. Hayashi, K. Misawa, and K. Mori. *On the influence of Dice loss function in multi-class organ segmentation of abdominal CT using 3D fully convolutional networks*. URL: http://arxiv.org/pdf/1801.05912v1.

[355] C. Zhang, C. Liu, X. Zhang, and G. Almpanidis. "An up-to-date comparison of state-of-the-art classification algorithms". In: *Expert Systems with Applications* 82.2–3 (2017), pp. 128–150. ISSN: 09574174. DOI: 10.1016/j.eswa.2017.04.003.

[356] I. K. Kazmi, L. You, and J. J. Zhang. "A Survey of 2D and 3D Shape Descriptors". In: *2013 10th International Conference Computer Graphics, Imaging and Visualization*. IEEE, 2013, pp. 1–10. ISBN: 978-0-7695-5051-0. DOI: 10.1109/CGIV.2013.11.

[357] J. Zhao, X. Xie, X. Xu, and S. Sun. "Multi-view learning overview: Recent progress and new challenges". In: *Information Fusion* 38 (2017), pp. 43–54. ISSN: 15662535. DOI: 10.1016/j.inffus.2017.02.007.

[358] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. "Multi-view Convolutional Neural Networks for 3D Shape Recognition". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 945–953. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.114.

[359] M. Seeland and P. Mäder. "Multi-view classification with convolutional neural networks". In: *PloS one* 16.1 (2021). DOI: 10.1371/journal.pone.0245230.

[360] D. Maturana and S. Scherer. "VoxNet: A 3D Convolutional Neural Network for real-time object recognition". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928. ISBN: 978-1-4799-9994-1. DOI: 10.1109/IROS.2015.7353481.

[361] M. Meng, J. Xia, J. Luo, and Y. He. "Unsupervised co-segmentation for 3D shapes using iterative multi-label optimization". In: *Computer-Aided Design* 45.2 (2013), pp. 312–320. ISSN: 00104485. DOI: 10.1016/j.cad.2012.10.014.

[362] M. Dominguez, F. P. Such, S. Sah, and R. Ptucha. "Towards 3D convolutional neural networks with meshes". In: *IEEE International Conference on Image Processing: Proceedings*. Piscataway, NJ: IEEE, 2017, pp. 3929–3933. DOI: 10.1109/ICIP.2017.8297019.

[363] C. Hane, S. Tulsiani, and J. Malik. "Hierarchical Surface Prediction for 3D Object Reconstruction". In: *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 412–420. ISBN: 978-1-5386-2610-8. DOI: 10.1109/3DV.2017.00054.

[364] A. Sinha, J. Bai, and K. Ramani. "Deep Learning 3D Shape Surfaces Using Geometry Images". In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Cham: Springer International Publishing, 2016, pp. 223–240. ISBN: 978-3-319-46465-7.

[365] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. "Indoor Semantic Segmentation using depth information". In: *First International Conference on Learning Representations (ICLR 2013)*. 2013, pp. 1–8. URL: https://arxiv.org/pdf/1301.3572.pdf.

[366] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. "Multimodal Deep Learning". In: *Proceedings of the 28 th International Conference on Machine Learning*. Bellevue, WA, USA, 2011. URL: https://icml.cc/2011/papers/399_icmlpaper.pdf.

[367] T. Baltrusaitis, C. Ahuja, and L.-P. Morency. "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE transactions on pattern analysis and machine intelligence* 41.2 (2019), pp. 423–443. DOI: 10.1109/TPAMI.2018.2798607.

[368] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli. "Multimodal fusion for multimedia analysis: A survey". In: *Multimedia Systems* 16.6 (2010), pp. 345–379. ISSN: 0942-4962. DOI: 10.1007/s00530-010-0182-0.

[369] E. Morvant, A. Habrard, and S. Ayache. "Majority Vote of Diverse Classifiers for Late Fusion". In: *Structural, Syntactic, and Statistical Pattern Recognition*. Ed. by P. Fränti, G. Brown, M. Loog, F. Escolano, and M. Pelillo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 153–162.

[370] P. Liu and Z. Kang. "Integrated topology optimization of multi-component structures considering connecting interface behavior". In: *Computer Methods in Applied Mechanics and Engineering* 341 (2018), pp. 851–887. ISSN: 00457825. DOI: 10.1016/j.cma.2018.07.001.

[371] A. Walsman, Y. Bisk, S. Gabriel, D. Misra, Y. Artzi, Y. Choi, and D. Fox. "EARLY FUSION for Goal Directed Robotic Vision". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1025–1031. ISBN: 978-1-7281-4004-9. DOI: 10.1109/IROS40897.2019.8968165.

[372] S. Petscharnig, K. Schoffmann, J. Benois-Pineau, S. Chaabouni, and J. Keckstein. "Early and Late Fusion of Temporal Information for Classification of Surgical Actions in Laparoscopic Gynecology". In: *2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 2018, pp. 369–374. ISBN: 978-1-5386-6060-7. DOI: 10.1109/CBMS.2018.00071.

[373] V. A. Sindagi, Y. Zhou, and O. Tuzel. *MVX-Net: Multimodal VoxelNet for 3D Object Detection*. 2019. URL: http://arxiv.org/pdf/1904.01649v1.

[374] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency. "Tensor Fusion Network for Multimodal Sentiment Analysis". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by M. Palmer, R. Hwa, and S. Riedel. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 1103–1114. DOI: 10.18653/v1/D17-1115.

[375] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. "Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by J. Su, K. Duh, and X. Carreras. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 457–468. DOI: 10.18653/v1/D16-1044.

[376] O. Pelka, F. Nensa, and C. M. Friedrich. "Branding - Fusion of Meta Data and Musculoskeletal Radiographs for Multi-Modal Diagnostic Recognition". In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2019, pp. 467–475. ISBN: 978-1-7281-5023-9. DOI: 10.1109/ICCVW.2019.00059.

[377] O. Pelka, F. Nensa, and C. M. Friedrich. "Variations on Branding with Text Occurrence for Optimized Body Parts Classification". In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2019, pp. 890–894. DOI: 10.1109/EMBC.2019.8857478.

[378] M. Soltaninejad, G. Yang, T. Lambrou, N. Allinson, T. L. Jones, T. R. Barrick, F. A. Howe, and X. Ye. "Supervised learning based multimodal MRI brain tumour segmentation using texture features from supervoxels". In: *Computer methods and programs in biomedicine* 157 (2018), pp. 69–84. DOI: 10.1016/j.cmpb.2018.01.003.

[379] D. G. Ullman. *The mechanical design process*. 4th ed. McGraw-Hill series in mechanical engineering. Boston: McGraw-Hill Higher Education, 2010. ISBN: 9780071267960.

[380] P. Burggräf, J. Wagner, and T. Weißer. "Knowledge-based problem solving in physical product development—-A methodological review". In: *Expert Systems with Applications: X* 5.1 (2020). ISSN: 25901885. DOI: 10.1016/j.eswax.2020.100025.

[381] A. Dogan and D. Birant. "Machine learning and data mining in manufacturing". In: *Expert Systems with Applications* 166.2 (2021), p. 114060. ISSN: 09574174. DOI: 10.1016/j.eswa.2020.114060.

[382] D. Machalica and M. Matyjewski. "CAD models clustering with machine learning". In: *Archive of mechanical engineering* 66.2 (2019). DOI: 10.24425/ame.2019.128441.

[383] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: (), pp. 77–85. DOI: 10.1109/CVPR.2017.16.

[384] H. Cao, R. Mo, and N. Wan. "3D modelling of a frame assembly using deep learning and the Chu–Liu–Edmonds Algorithm". In: *Assembly Automation* ahead-of-print.ahead-of-print (2019), p. 26. ISSN: 0144-5154. DOI: 10.1108/AA-08-2018-0121.

[385] B. Alkan, D. Vera, B. Ahmad, and R. Harrison. "A Method to Assess Assembly Complexity of Industrial Products in Early Design Phase". In: *IEEE Access* 6 (2018), pp. 989–999. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2777406.

[386] A. Desai. "Ease of product assembly through a time-based design methodology". In: *Assembly Automation* 39.5 (2019), pp. 881–903. ISSN: 0144-5154. DOI: 10.1108/AA-09-2018-0133.

[387] S. Eilmus, T. Gumpinger, T. Kipp, O. Sankowski, and D. Krause. “Early Phase Estimation of Variety Induced Complexity Cost Effects: A Study on Industrial Cases in Germany”. In: *Transdisciplinary Perspectives on Complex Systems*. Ed. by F.-J. Kahlen, S. Flumerfelt, and A. Alves. Cham: Springer International Publishing, 2017.

[388] A. Etienne, S. Mirdamadi, M. Mohammadi, R. Babaeizadeh Malmiry, J.-F. Antoine, A. Siadat, J.-y. Dantan, R. Tavakkoli, and P. Martin. “Cost engineering for variation management during the product and process development”. In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 11.2 (2017), pp. 289–300. ISSN: 1955-2513. DOI: 10.1007/s12008-016-0318-3.

[389] S. M. Hasan, A. A. Baqai, S. U. Butt, and U. K. q. Zaman. “Product family formation based on complexity for assembly systems”. In: *The International Journal of Advanced Manufacturing Technology* 95.1-4 (2018), pp. 569–585. ISSN: 0268-3768. DOI: 10.1007/s00170-017-1174-4.

[390] G. Kim, Y. Kwon, E. S. Suh, and J. Ahn. “Analysis of Architectural Complexity for Product Family and Platform”. In: *Journal of Mechanical Design* 138.7 (2016), p. 3. ISSN: 1050-0472. DOI: 10.1115/1.4033504.

[391] P. Krus. “Design Space Configuration for Minimizing Design Information Entropy”. In: *ICoRD’15 – Research into Design Across Boundaries Volume 1*. Ed. by A. Chakrabarti. Vol. 1. New Delhi: Springer India, 2015, pp. 51–60. DOI: 10.1007/978-81-322-2232-3{\textunderscore}5.

[392] V. Modrak, S. Bednar, and Z. Soltysova. “Application of Axiomatic Design-based Complexity Measure in Mass Customization”. In: *Procedia CIRP* 50 (2016), pp. 607–612. ISSN: 22128271. DOI: 10.1016/j.procir.2016.04.196.

[393] K. Oh, H. W. Kim, D. Kim, J. Lee, J. Lee, and Y. S. Hong. “Product Interface Design for Complexity Management in Assembly Systems”. In: *IEEE Access* (2020), p. 1. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3045041.

[394] G. Schuh, S. Rudolf, and T. Vogels. “Performance Measurement of Modular Product Platforms”. In: *Procedia CIRP* 17 (2014), pp. 266–271. ISSN: 22128271. DOI: 10.1016/j.procir.2014.02.028.

[395] G. Schuh, C. Dölle, S. Schmitz, J. Koch, M. Höding, and A. Menges. “Data-Based Determination of the Product-Oriented Complexity Degree”. In: *Procedia CIRP* 70 (2018), pp. 144–149. ISSN: 22128271. DOI: 10.1016/j.procir.2018.03.293.

[396] A.-K. Weiser, B. Baasner, M. Hosch, M. Schlueter, and J. Ovtcharova. “Framework of a variant management for modular product architectures”. In: *International Design Conference - Design 2016*. 2016.

[397] T. Sadowski, M. Kneć, and P. Golewski. “Spot Welding–Adhesive Joints: Modelling and Testing”. In: *The Journal of Adhesion* 90.4 (2014), pp. 346–364. ISSN: 0021-8464. DOI: 10.1080/00218464.2013.766599.

[398] A. P. Lehnerd and M. H. Meyer. *The power of product platforms*. Free Press, 2014. ISBN: 9781439137444.

[399] G. Schuh, C. Doelle, J. Koch, and P. Grutza. “Identification of the cost-benefit-optimal product configuration”. In: *Procedia CIRP* 70 (2018), pp. 386–391. ISSN: 22128271. DOI: 10.1016/j.procir.2018.03.294.

[400] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, Q. Chen, N. K. Chowdhury, B. Fang, H. Fu, T. Furuya, H. Li, J. Liu, H. Johan, R. Kosaka, H. Koyanagi, R. Ohbuchi, A. Tatsuma, Y. Wan, C. Zhang, and C. Zou. "A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries". In: *Computer Vision and Image Understanding* 131 (2015), pp. 1–27. ISSN: 10773142. DOI: 10.1016/j.cviu.2014.10.006.

[401] X. W. Wang, Y. P. Shi, R. Yu, and X. S. Gu. "Intelligent Welding Robot Path Planning". In: *Proceedings of the 2015 Chinese Intelligent Automation Conference*. Ed. by Z. Deng and H. Li. Vol. 337. Lecture Notes in Electrical Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 25–32. ISBN: 978-3-662-46462-5. DOI: 10.1007/978-3-662-46463-2{\textunderscore}4.

[402] A. Nibali, Z. He, S. Morgan, and L. Prendergast. *Numerical Coordinate Regression with Convolutional Neural Networks*. 2018. URL: http://arxiv.org/pdf/1801.07372v2.

[403] R. Zbiciak and C. Grabowik. "Feature Recognition Methods Review". In: *Proceedings of the 13th International Scientific Conference*. Ed. by E. Rusiński and D. Pietrusiak. Lecture Notes in Mechanical Engineering. Cham, Switzerland: Springer, 2017, pp. 605–615. ISBN: 978-3-319-50938-9. DOI: 10.1007/978-3-319-50938-9{\textunderscore}63.

[404] J. Romero-Hdz, B. Saha, G. Toledo-Ramirez, and I. Lopez-Juarez. "A Reinforcement Learning Based Approach for Welding Sequence Optimization". In: *Transactions on Intelligent Welding Manufacturing*. Ed. by S. Chen, Y. Zhang, and Z. Feng. Singapore: Springer Singapore, 2018. ISBN: 978-981-10-7042-6.

[405] K. Alexopoulos, N. Nikolakis, and G. Chryssolouris. "Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing". In: *International Journal of Computer Integrated Manufacturing* 33.5 (2020), pp. 429–439. DOI: 10.1080/0951192X.2020.1747642.

[406] K. Wagstaff and S. Rogers. "Constrained K-means Clustering with Background Knowledge". In: *Eighteenth International Conference on Machine Learning*. Ed. by C. E. Brodley and A. P. Danyluk. 2001, pp. 577–584.

[407] T. Gerlach. "Optimierung der Schweißpunkt Vorhersage mit Hilfe von Künstlicher Intelligenz". Master. Reutlingen: Hochschule Reutlingen, 2021.

[408] V. Suppakitpaisarn, A. Ariyarit, and S. Chaidee. "A Voronoi-based method for land-use optimization using semidefinite programming and gradient descent algorithm". In: *International Journal of Geographical Information Science* 35.5 (2021), pp. 999–1031. ISSN: 1365-8816. DOI: 10.1080/13658816.2020.1841203.

[409] C. Dong, C. C. Loy, K. He, and X. Tang. *Image Super-Resolution Using Deep Convolutional Networks*. Dec. 31, 2014.

[410] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. URL: http://arxiv.org/pdf/1505.04597v1.

[411] V. Badrinarayanan, A. Kendall, and R. Cipolla. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. URL: http://arxiv.org/pdf/1511.00561v3.

[412] D. F. Perez Ramirez. "Agile Development of a Machine Learning Framework to Predict Joining Locations Using the Example of Resistance Spot Welding in the Automotive Industry". Master thesis. Munich, Germany: TUM Department of Informatics, 2019.

[413] C. E. Shannon. “Communication in the Presence of Noise”. In: *Proceedings of the IRE* 37.1 (1949), pp. 10–21. ISSN: 0096-8390. DOI: 10.1109/JRPROC.1949.232969.

[414] Y. Wang, D. K. Ramanan, and M. Hebert. “Learning to Model the Tail”. In: *Proceedings of (NeurIPS) Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc, 2017, pp. 7032–7042.

[415] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. J. Snead, I. A. Cree, and N. M. Rajpoot. “Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images”. In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1196–1206. ISSN: 0278-0062. DOI: 10.1109/TMI.2016.2525803.

[416] E. R. Dougherty. *An introduction to morphological image processing*. 1992. ISBN: 0-8194-0845-X.

[417] A. Kanezaki, Y. Matsushita, and Y. Nishida. *RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints*. 2016. DOI: 10.1109/CVPR.2018.00526.

[418] T. Szkodny. “Calculation of the Location Coordinates of an Object Observed by a Camera”. In: *Man-Machine Interactions 3*. Ed. by A. Gruca, T. Czachórski, and S. Kozielski. Advances in intelligent systems and computing. Cham: Springer International Publishing and Imprint: Springer, 2014, pp. 139–151. ISBN: 9783319023090.

[419] S. Xia, Z. Xiong, Y. Luo, WeiXu, and G. Zhang. “Effectiveness of the Euclidean distance in high dimensional spaces”. In: *Optik* 126.24 (2015), pp. 5614–5619. ISSN: 00304026. DOI: 10.1016/j.ijleo.2015.09.093.

[420] L. van der Maaten. “Learning a Parametric Embedding by Preserving”. In: *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*. Ed. by D. van Dyk and M. Welling. Vol. 5. Clearwater Beach, Florida, USA: PMLR, 2009, pp. 384–391.

[421] H. D. Nguyen. “Maschinelles Lernen in der Fertigungstechnik: Mixed-Input-Modelle zur Vorhersage von Schweißpunkten in der Automobilindustrie”. Masterthesis. 2020.

[422] N. K. Rout, M. Atulkar, and M. K. Ahirwal. “A review on content-based image retrieval system: Present trends and future challenges”. In: *International Journal of Computational Vision and Robotics* 11.5 (2021), p. 461. ISSN: 1752-9131. DOI: 10.1504/IJCVR.2021.117578.

[423] D. Bau, J.-Y. Zhu, J. Wulff, W. Peebles, B. Zhou, H. Strobelt, and A. Torralba. “Seeing What a GAN Cannot Generate”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 4501–4510. ISBN: 978-1-7281-4803-8. DOI: 10.1109/ICCV.2019.00460.

[424] A. Shafahi, P. Saadatpanah, C. Zhu, A. Ghiasi, C. Studer, D. Jacobs, and T. Goldstein. “Adversarially robust transfer learning”. In: *8th International Conference on Learning Representations (ICLR 2020) (virtual),* Addis Ababa, Ethiopia, 2020. DOI: 10.3929/ethz-b-000458750.

[425] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. URL: http://arxiv.org/pdf/1409.1556v6.

[426] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. ISSN: 0920-5691. DOI: 10.1007/s11263-015-0816-y.

[427] R. C. Jain, B. G. Schunck, and R. Kasturi. *Machine vision*. McGraw-Hill series in computer science. Artificial intelligence. New York and London: McGraw-Hill, 1995. ISBN: 0-07-032018-7.

[428] S. Prohaska. "Skeleton-based visualization of massive voxel objects with network-like architecture". Dissertation. Potsdam, 2007. URL: http://opus.kobv.de/ubp/volltexte/2007/1488/.

[429] H. W. Kuhn. "The Hungarian Method for the assignment problem". In: *Naval Research Logistics Quarterly* 2 (1955), pp. 83–97.

[430] J. Munkres. "Algorithms for the Assignment and Transportation Problems". In: *Journal of the Society for Industrial and Applied Mathematics* 1.5 (1957), pp. 32–38.

[431] R. Jonker and A. Volgenant. "A shortest augmenting path algorithm for dense and sparse linear assignment problems". In: *Computing* 38.4 (1987), pp. 325–340. DOI: 10.1007/BF02278710.

[432] M. Müller. "Dynamic Time Warping". In: *Information Retrieval for Music and Motion*. Ed. by M. Müller. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[433] M. Kuhn, F. Schaefer, and H. Otten. "Process complexity as a future challenge – a quality management perspective". In: *The TQM Journal* 30.6 (2018), pp. 701–716. ISSN: 1754-2731. DOI: 10.1108/TQM-10-2017-0123.

[434] C. L. Hansen, N. H. Mortensen, L. Hvam, and U. Harlou. "Calculation of Complexity Costs-An Approach for Rationalizing a Product Program". In: *Proceedings of NordDesign 2012, the 9th NordDesign conference*. Ed. by P. K. Hansen, J. Rasmussen, K. A. Jřrgensen, and C. Tollestrup. NordDESIGN. Design Society, 2012.

[435] S. A. Wilson and A. Perumal. *A Fresh Take on Complexity Costs: Six Principles for Waging War on an Intractable Issue*. American Management Association, 2014. URL: https://www.amanet.org/training/articles/a-fresh-take-on-complexity-costs.aspx (visited on 11/14/2018).

[436] H. Schleich, J. Schaffer, and L. F. Scavarda. "Managing complexity in automotive production". In: *19th International Conference on Production Research*. 2007.

[437] D. Robertson and K. Ulrich. "Planning for Product Platforms". In: *Sloan Management Review* 39.4 (1998), pp. 19–31.

[438] L. Fredendall and T. J. Gabriel. "Manufacturing Complexity". In: *POMS Conference*. 2003.

[439] J. Lee Rodgers and W. A. Nicewander. "Thirteen Ways to Look at the Correlation Coefficient". In: *The American Statistician* 42.1 (1988), pp. 59–66. ISSN: 0003-1305. DOI: 10.1080/00031305.1988.10475524.

[440] K. Pearson. "X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50.302 (2009), pp. 157–175. ISSN: 1941-5982. DOI: 10.1080/14786440009463897.

[441] H. Cramér. *Mathematical Methods of Statistics (PMS-9)*. Princeton University Press, 1946. ISBN: 9781400883868. DOI: 10.1515/9781400883868.

[442] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357. ISSN: 1076-9757. DOI: 10.1613/jair.953.

[443] R. Kohavi. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection". In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. Vol. 2. Morgan Kaufmann Publishers Inc, 1995, pp. 1137–1143. DOI: 10.5555/1643031.1643047.

[444] J. M. Hammersley and U. Grenander. *Probability and Statistics: The Harald Cramer Volume*. Vol. 123. 1960. DOI: 10.2307/2342476.

[445] M. Shahrokh Esfahani and E. R. Dougherty. "Effect of separate sampling on classification accuracy". In: *Bioinformatics (Oxford, England)* 30.2 (2014), pp. 242–250. DOI: 10.1093/bioinformatics/btt662.

[446] S. Chen, Y. Zhang, and Z. Feng, eds. *Transactions on Intelligent Welding Manufacturing*. Singapore: Springer Singapore, 2018. ISBN: 978-981-10-7042-6. DOI: 10.1007/978-981-10-7043-3.

[447] J. Long, E. Shelhamer, and T. Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2014. URL: http://arxiv.org/pdf/1411.4038v2.

[448] A. Araujo, W. Norris, and J. Sim. "Computing Receptive Fields of Convolutional Neural Networks". In: *Distill* 4.11 (2019). DOI: 10.23915/distill.00021.

[449] P. Abbeel. *Maximum Likelihood (ML), Expectation Maximization (EM)*. Berkeley, CA, 2013-10-23. URL: https://people.eecs.berkeley.edu/~pabbeel/cs287-fa13/slides/Likelihood_EM_HMM_Kalman.pdf.

[450] M. G. Kendall. "A New Measure of Rank Correlation". In: *Biometrika* 30.1/2 (1938), p. 81. ISSN: 00063444. DOI: 10.2307/2332226.

[451] V. Dhameliya. "Development of A Multi–View Based Supervised Machine Learning Methodology to Predict Joining Locations in the Automotive Industry". Masterthesis. 2021.

[452] A. Essien and C. Giannetti. "A Deep Learning Model for Smart Manufacturing Using Convolutional LSTM Neural Network Autoencoders". In: *IEEE Transactions on Industrial Informatics* 16.9 (2020), pp. 6069–6078. ISSN: 1551-3203. DOI: 10.1109/TII.2020.2967556.

[453] E. Grädel, P. G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Y. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. ISBN: 978-3-540-00428-8. DOI: 10.1007/3-540-68804-8.

[454] B. Alkan, D. A. Vera, M. Ahmad, B. Ahmad, and R. Harrison. "Complexity in manufacturing systems and its measures: A literature review". In: *European J. of Industrial Engineering* 12.1 (2018), pp. 116–150. ISSN: 1751-5254. DOI: 10.1504/EJIE.2018.089883.

[455] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. URL: http://arxiv.org/pdf/1706.02413v1.

[456] X. Shen. *A survey of Object Classification and Detection based on 2D/3D data*. 2019. URL: http://arxiv.org/pdf/1905.12683v1.

[457] R. K. Srivastava and G. Hinton. "Dropout- Simple Way to Prevent Overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958.

[458] R. Wirth and J. Hipp. "Crisp-DM: Towards a standard process modell for data mining". In: *Proceedings of the 4th International Conference on the Practical Application of Knowledge Discovery and Data Mining*. 2000, pp. 29–39.

[459] P. C. Bruce and A. Bruce. *Practical statistics for data scientists: 50 essential concepts / Peter Bruce and Andrew Bruce*. First edition. Sebastopol, CA: O'Reilly, 2017. ISBN: 9781491952962.

[460] Z. S. Harris. "Distributional Structure". In: WORD 10.2-3 (2015), pp. 146–162. ISSN: 0043-7956. DOI: 10.1080/00437956.1954.11659520.

[461] C. Spearman. "The Proof and Measurement of Association between Two Things". In: *The American Journal of Psychology* 15.1 (1904), p. 72. ISSN: 00029556. DOI: 10.2307/1412159.

[462] H. D. Nguyen. *Datenanalyse und Evaluation von Verbindungsszenarien in der Fertigungsindustrie*. Dortmund, 2019.

[463] C. Croux and C. Dehon. "Influence functions of the Spearman and Kendall correlation measures". In: *Statistical Methods & Applications* 19.4 (2010), pp. 497–515. ISSN: 1618-2510. DOI: 10.1007/s10260-010-0142-z.

[464] H. Scheffe. *The analysis of variance*. Wiley Classics Library. New York: John Wiley and Sons, 1999. ISBN: 0471758345.

[465] W. H. Kruskal and W. A. Wallis. "Use of Ranks in One-Criterion Variance Analysis". In: *Journal of the American Statistical Association* 47.260 (1952), pp. 583–621. DOI: 10.1080/01621459.1952.10483441.

[466] G. D. Ruxton and G. Beauchamp. "Time for some a priori thinking about post hoc testing". In: *Behavioral Ecology* 19.3 (2008), pp. 690–693. ISSN: 1465-7279. DOI: 10.1093/beheco/arn020.

[467] O. J. Dunn. "Multiple Comparisons Using Rank Sums". In: *Technometrics* 6.3 (1964), pp. 241–252. ISSN: 0040-1706. DOI: 10.1080/00401706.1964.10490181.

Index