Data-based knowledge extraction and neuro-symbolic learning of parametrisation processes in manufacturing

Richard Nordsieck

Angaben zur Veröffentlichung / Publication details:

Nordsieck, Richard. 2024. “Data-based knowledge extraction and neuro-symbolic learning of parametrisation processes in manufacturing.” Augsburg: Universität Augsburg.

Data-Based Knowledge Extraction and Neuro-Symbolic Learning of Parametrisation Processes in Manufacturing

Richard Nordsieck

Dissertation
zur Erlangung des akademischen Grades Doktor der Naturwissenschaften
der Fakultät für Angewandte Informatik der Universität Augsburg

23. September 2023

Data-Based Knowledge Extraction and Neuro-Symbolic Learning of Parametrisation Processes in Manufacturing

Erstgutachter:	**Prof. Dr. Jörg Hähner**, Lehrstuhl für Organic Computing, Universität Augsburg, Deutschland
Zweitgutachter:	**Prof. Dr. Bernhard Bauer**, Professur für Softwaremethodik für verteilte Systeme, Universität Augsburg, Deutschland
Tag der mündlichen Prüfung:	8. Februar 2024

Abstract

The goal of modern manufacturing is to use machines to produce a product according to economic target criteria, e.g. quality characteristics or production speed. To achieve these target criteria, parametrisation processes, in which skilled operators iteratively adjust parameters of the machinery, are conducted. During parametrisation processes, raw material and energy is used. Furthermore, both personnel and production capacity are required. In contrast to this, external factors such as demographic change with resulting skilled labour shortage and climate change highlight the finite nature of these resources while increasing customisation of products leads to a decrease in production batches. Therefore, reducing the amount of iterations and thereby the duration of parametrisation process needed is in economic as well as ecological interest. If it is not possible to decrease the complexity of these production processes to allow the operators to cope with the challenges of skilled labour shortage and to increase the efficiency of raw material usage significantly, the producing industry in Germany, as it exists today, is unlikely to prevail. This would lead to significant societal challenges due to the high amount of employment by this sector.

Given sensing equipment that is able to measure quality characteristics, the goal of reducing the time and resources spent during parametrisation processes can be achieved by providing operators with suggestions for, or—if the quality of predictions are good enough and the corresponding parameters can be digitally affected—automatic selection of, a suitable parametrisation. Applying learning systems to this setting seems a reasonable choice given the widespread success they enjoyed in recent years. However, even given the move towards interconnecting and digitalising manufacturing through initiatives such as Industry 4.0, an application of learning system is met with several challenges in practice. Firstly, manufacturing in general, but German manufacturing in particular, is characterised by medium-sized businesses that heavily rely on the manufacture or use of special purpose machinery. This leads to limited amounts of data, since machines and the products they produce are often quasi unique, being produced at a very limited volume. Furthermore, these medium-sized businesses often struggle in holistically digitalising their processes and machines due to short-term economic planning which leads to an incomplete data landscape. Secondly, the data available is strongly skewed. This is due to several years of manual optimisation of the production processes in question which leads to relatively few unsatisfactory products when compared to the overall number of produced products.

This thesis contributes to mitigating these practical challenges to manufacturing by proposing a neuro-symbolic fusion of learning systems and the operators' expert knowledge. Its main contributions are twofold: Firstly, the concept of data-based knowledge extraction is introduced to elicit tacit expert knowledge with minimal obstruction. Secondly, two regularisation-based architectures for a neuro-symbolic fusion of the expert knowledge and learning systems are presented. These contributions are developed on requirements gathered from two real-world case studies, of which one is from an industrial manufacturing setting, and evaluated on a synthetic as well as real-world dataset.

Data-Based Knowledge Extraction Extracting tacit knowledge gained by experienced operators through multiple years' worth of conducting parametrisation processes is not trivial. Traditional interview and observation based knowledge extraction methods are encompassed

with several drawbacks, e.g. time & specific knowledge engineering personnel required as well as difficulties to extract quantified knowledge. Based on the analysis and formalisation of parametrisation processes, this thesis develops a data-based knowledge extraction approach. To mitigate the challenges regarding data quantity and quality outlined above, the approach augments process data with additional information interactively given by operators in times when their cognitive load is relatively low. The procedural knowledge resulting from the data-based knowledge extraction approach is represented in knowledge graphs which are usually mostly used for factual knowledge.

Neuro-Symbolic Learning The procedural knowledge contained in knowledge graphs constitutes symbols for neuro-symbolic learning. To directly fuse the procedural knowledge with neural learning systems, a low dimensional vector representation is assumed to be beneficial. To this end, custom embedding aggregation methods, which use established established knowledge graph embedding methods, are developed and evaluated for procedural knowledge. Two regularisation-based neuro-symbolic learning architectures are presented and thoroughly evaluated. These approaches promise the benefits of improved quality of predictions, better generalisability and smaller requirements regarding data quantity. This enables the application of learning systems in challenging settings such as manufacturing, enabling it to cope with the challenges outlined above.

Acknowledgements

First of all, I would like to thank my advisor Jörg Hähner for supporting this undertaking from the beginning and providing me with the freedom I needed. Additionally, thanks belong to Bernhard Bauer for volunteering as second reviewer and keeping the necessary pressure up on the last meters. Furthermore, Ulrich Huggenberger played a pivotal role in the setup of this thesis by having the vision of a research environment at XITASO and by extending the trust and freedom required to build it.

This directly leads to the diverse array of colleagues I had the pleasure of working with and that deserve thanks. Starting with Andreas Angerer who mentored me and volunteered to proofread, Michael Heider who was the closest collaborator in the research projects, Anton Winschel for sparring sessions, Alwin Hoffmann for his love to detail in formalisations and David Pätzel who was more than willing to share his wealth of knowledge about Bayesian statistics. To Sina Busch, who thankfully reminded me not to lose track of the end users' requirements and helped me through one or two valleys of disillusionment as well as Simon Engelmann and Jessica Friedline who were always open to discuss and improve my artistic shortcomings. Furthermore, it was a pleasure to work with a range of highly motivated students such as Dat Le Thanh, Luis Schweigard, André Schweizer and Anton Hummel who had a lasting impact on my academic journey.

Last but not least, I would like to thank friends and family who provided the necessary support network, allowed me to vent spontaneous frustrations and were never shy to offer encouragements or motivation. Especially, thanks go to Arantza who had the unenviable pleasure of accompanying me in the final weeks but was never shy to play out her positive attitude and lift the mood with a joke.

Contents

Part I.

Motivation, Formalisation & Scenarios

Manufacturing is confronted by various challenges from demographic change to human operators interacting with more complex technology. This part outlines these challenges and why this thesis is suited to address them. To this end, parametrisation processes are formalised (see Chapter 2) and case studies presented (see Chapter 3).

1 Introduction

The goal of this work is to contribute to optimising parametrisation processes in manufacturing by providing a methodology for data-based knowledge extraction and neuro-symbolic learning. This helps to address challenges experienced in introducing machine-learning systems in real-world manufacturing scenarios. In the first chapter, the motivation and problem statements are discussed. This is followed by an overview of the research goal and the corresponding research questions as well as contributions of this thesis. Finally, the structure of this thesis is presented.

1.1. Motivation and Goals

Manufacturing is a significant industry which was responsible for 25.79% of Germany's gross domestic product in 2017 [Sta22] and includes both, users and producers of machinery. In fact, 6000 companies are employing one million employees in machinery and plant engineering alone [VDM17]. Manufacturing consists of production processes that aim to utilise 'goods and services [...] to extract raw materials or to produce or manufacture goods and to generate services' [Blo+14]. These processes are facilitated by humans or machines and produce products according to target criteria, e.g. quality characteristics or productions speed. Consider injection moulding as an example in which plastic is heated and injected to a mould to produce a product, e.g. plastic bricks. Often, production processes are sophisticated and require a high degree of specialisation on the operators' part to fulfil the target criteria and mitigate occurring quality defects as evidenced by the fact that humans are one of the main drivers of quality defects [SC17]. To mitigate occurring quality defects, operators adapt parameters of the production process which we refer to as parametrisation or reparametrisation. Another example showcasing the complexity is extrusion, where a production process incorporates an assembly of multiple machines stretching often to up to 100 meters leading to a delay between parametrisation and observable change in the overall quality characteristics [HSB22]. Combined with complex parameter-quality dependencies, this leads to long onboarding times, i.e. employees are only able to solve occurring quality defects independently after multiple years of work experience [TBH19]. Small-batch and lot-size-one production were introduced to answer the market's demand of shorter product life cycles [PRF08], more individualised products [KSH12] and special purpose machinery [Fra16]. They further exacerbate the complexity for machine operators due to the high amount of reparametrisations necessary. Due to a shrinking workforce caused by demographic changes [AB18; Bäc+18; Löh+18; TBH19] and higher fluctuations of the workforce, it is exceedingly hard for manufacturing companies to find, train and retain qualified operators. Consequently, the complexity of parametrisation processes needs to be lowered to achieve lower onboarding times of operators. Assistance systems, which propose parametrisations to the operator promise to be an answer to this challenge [Han+20; HSB22; Apt+18; TBH19]. In the context of Industry 5.0

they strengthen the socio-technical combination of operator and machinery [Xu+21] bridging the gap between fully automated processes that often utilise learning systems and those where human cognition and reasoning remains necessary [FO17]. Compared to assistance systems based on static extracted knowledge the learning component allows the system to be able to react to changes in the environment, e.g. to the production process or to the desired product.

Applying learning systems to live production settings, however, is not straight forward as evidenced by the fact that only 14% of related works in manufacturing use data from running productions [TM22]. We propose that the same characteristics that increase the complexity for human operators, e.g. small-batch production, also increase the complexity for the application of learning systems. Therefore, either a high degree of generalisability of trained models or successful transfer learning strategies become necessary since it is unfeasible to learn from the resulting small amounts of process iterations. Also, even though the degree of connectivity between machines is growing [Mor+16], which necessitates digital sensor measurements, data concerning production processes is still coarse, i.e. missing or imprecise [Bou+19]. The coarsity stems from hardware sensors that are often imprecise, prone to anomalies and only applicable to a subset of a product's quality characteristics. Consequently, proper cleaning and validation is necessary for the sensor data which is not always present in practice. Furthermore, production processes have been manually optimised to a high degree resulting in a strong imbalance of satisfactory products to those showing quality defects. Lastly, the learning systems are required to deal with a high dimensionality (the amount of affectable parameters often ranges into the hundreds).

To ensure the competitiveness of German manufacturers in times of demographic change and increasing requirements regarding workplace flexibility, this thesis sets out to mitigate these practical issues. The overall goal is to optimise parametrisation processes in manufacturing by addressing the parameter prediction problem (see Section 13.1) which is closely related to the predictive quality problem [TM22]. The resulting parametrisations, which mitigate quality defects observed in the previous process iteration, can then be used in intelligent assistance systems [TBH19; Han+20] as recommendations to the operator or automatically applied if their quality is satisfactory. To address the challenges outlined above, specifically generalisability and coarsity, we follow the current trend towards neuro-symbolic learning systems [Hit+22] that combine formalised knowledge with learning systems to increase generalisability and performance on low amounts of data [Hit+22; Sar+21; SPG19]. The expert knowledge which constitutes the basis for our symbolic system is not readily available since in manufacturing knowledge is usually tacit, i.e. experts cannot verbalise it directly, and therefore difficult to extract. Furthermore it has mostly procedural characteristics, i.e. it is knowledge about how to accomplish a goal and which techniques to apply [Kra02]. Since established knowledge extraction methods for procedural knowledge require a large amount of manual effort and often result in unquantified knowledge [HSB22], which does not result in conclusions of actual parametrisations but rather high-level dependencies, the research of data-based knowledge extraction techniques is a further goal. Through our holistic approach towards data-based knowledge extraction and neuro-symbolic learning systems, we provide the foundation for assistance systems for the parametrisation problem in manufacturing. Furthermore, to the best of our knowledge, it constitutes the first successful application of neuro-symbolic learning systems that utilise symbolic knowledge to increase the predictive performance in manufacturing.

1.2. Contribution

This thesis addresses challenges faced by machine learning approaches in industrial settings. To this end, it develops approaches of data-based knowledge extraction and neuro-symbolic methods for the fusion of expert knowledge and learning systems. A majority of the approaches have previously been published or are in the process (see Appendix A). The contributions of this thesis are:

- a **formalisation of the parametrisation process**, which aids the understanding of the process operators undertake and builds the theoretical foundation of our approach to data-based knowledge extraction
- a flexible and extensible **framework for generating synthetic datasets (PDPK)** containing **process data along with corresponding procedural knowledge** which provides a foundation for research with procedural knowledge in parametrisation settings
- a **data-based knowledge extraction method** that allows the extraction of procedural knowledge of operators in a quantified manner without extensively disrupting their day to day business. To this end:
  - identifying a period where the operators' cognitive load is reduced and querying them for **insights** into their parametrisation choices. These insights constitute example based extracted knowledge segments
  - development of a method for the **aggregation of knowledge segments** into **formalised rules** with quantified conclusions that can be used to mitigate quality defects and can serve as a knowledge base out of which recommendations can be generated
- different **representations** of procedural knowledge in Knowledge Graphs (KGs) and an evaluation of established embedding methods on these
- developing a **method to identify and embed the parts of the KG** pertaining to a given quality defect, i.e. all rules that could be applied to this input, into vectors that can be processed by downstream learning systems. To evaluate the quality of these embeddings, we introduce a **metric, matches@k,** which is suited to evaluate the quality of these embeddings.
- the development, adaption and evaluation of **neuro-symbolic learning systems** that utilise formalised expert knowledge to guide learning systems through regularisation.

1.3. Thesis Structure

This thesis is structured in four parts, with Figure 1.1 showing the general outline. Here, arrows indicate dependencies between the respective chapters or parts, giving the reader an orientation about which chapters to read in which order. In Part I the first chapter, *Introduction*, deals with foundations necessary for the motivation for this thesis and the goals it hopes to achieve. Also, it summarises the contributions of this thesis to the state of the art and gives an overview of the thesis structure. The second chapter introduces *a Formalisation of Parametrisation Processes* on which the elicitation methodology (see Chapter 5) is based. The third chapter, *Case Studies*, details

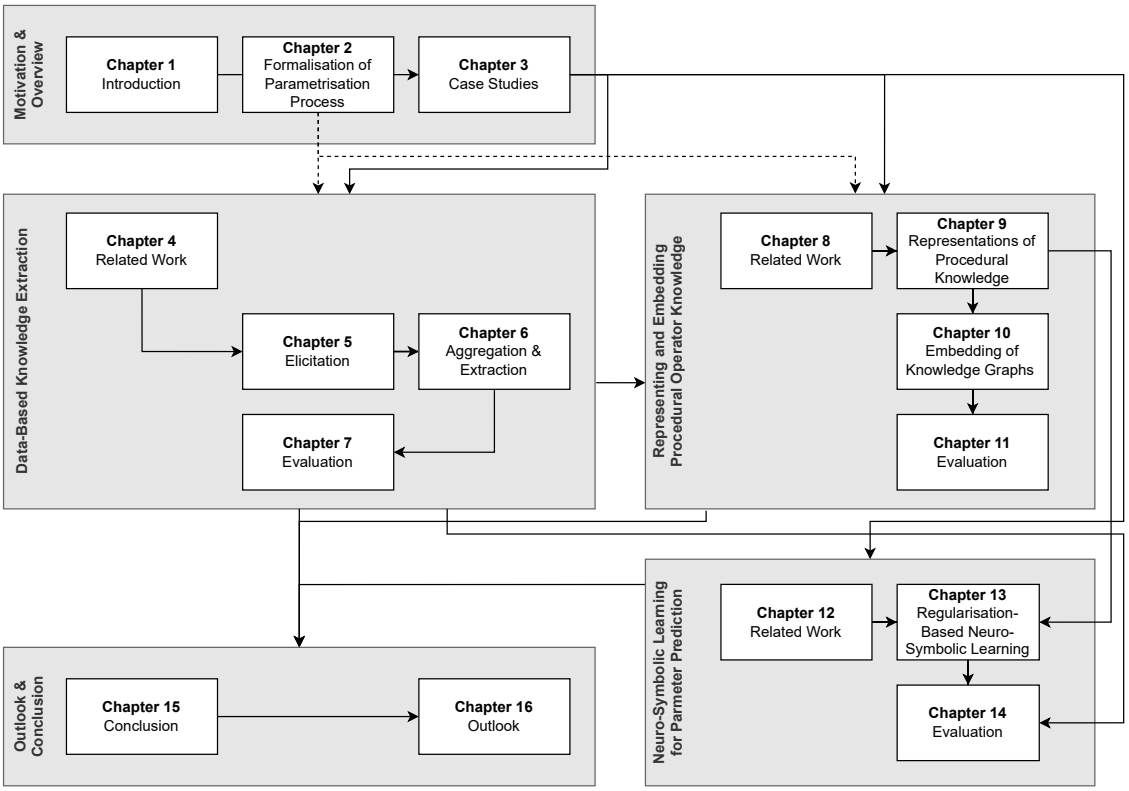


Figure 1.1.: Outline of the thesis with arrows showing dependencies between parts or chapters.

the case studies on which this thesis is built. Namely, we analyse an additive manufacturing process (fused deposition modelling (FDM)) and polymer extrusion as representatives for discrete and continuous production processes, respectively. Furthermore, we present a framework for the generation of synthetic process data with consistent procedural knowledge (PDPK) and the main dataset we generated using PDPK.

The second part constitutes a part of the main body of this thesis and details the concept of *Data-Based Knowledge Extraction*, which we developed to be used in conjunction or independently of more traditional knowledge extraction methods to arrive at quantified procedural knowledge with less aggressive intrusions into the day to day business of operators. We present the state of the art in Chapter 4, *Related Work*, and detail our reasoning behind the process and infrastructure for the *Elicitation* (see Chapter 5) of example based knowledge segments. These are then aggregated into coherent rule bases in Chapter 6. Advantages and limitations of this methodology are presented in Chapter 7, *Evaluation*.

The third part deals with representations and embeddings of procedural operator knowledge, regardless whether it is provided through data-based or traditional knowledge extraction techniques. Again, we first present *Related Work* in Chapter 8 before detailing knowledge graphs *Representations* of procedural knowledge in Chapter 9. Approaches towards embedding parts of the knowledge graphs, that constitute rules pertaining to an input or query, using established embedding methods and sum-based aggregations are presented in Chapter 10. The *Evaluation* presented in Chapter 11 evaluates node embeddings and subgraph aggregations for the presented

representations via a surrogate metric for downstream performance, *matches@k*, which compares distances between graph and embedding space.

The fourth part concludes the main part of the thesis, outlining our understanding of *Neuro-Symbolic Learning Systems* and their uses and limitations in manufacturing scenarios. Based on *Related Work* presented in Chapter 12, we identify regularisation-based neuro-symbolic as a promising research direction to fuse expert knowledge and learning systems. In Chapter 13, we detail its application to manufacturing, developing two architectures: one relying on knowledge graph embeddings, the other directly using the extracted knowledge. Their characteristics are compared to a multi-layer perceptron which does not make use of additional knowledge in Chapter 14. Also, we are able to re-evaluate the effect of different embedding methods and the dataset similarities with this downstream scenario.

The fifth part concludes the thesis by summarising our findings in Chapter 15 and giving an outlook in Chapter 16.

2

A Formalisation of Parametrisation Processes in Manufacturing

The inspection of various manufacturing processes during this thesis has shown that parametrisation processes follow a similar pattern, i.e. the quality of the produced product is influenced by a multitude of factors and the operator tries to mitigate occurring quality defects. This chapter is based on the publications [Nor+21; Nor+22a]—citations of these works are not separately denoted.

On a high level of abstraction, the general parametrisation process encountered in manufacturing and observed in both real-world case studies is shown in Figure 2.1. It consists of a parametrisation, i.e. a concrete instance of process parameters (see Definition 2.1.1), which controls the behaviours of a production process. This production process then produces the desired product. The quality of the produced product is regularly controlled manually or automatically, either at the place of manufacturing or in downstream laboratories. If the product has not reached a sufficient quality or a quality anomaly is detected the operator has to adjust the parametrisation accordingly.

In the following, this chapter formalises the parametrisation process. At first, discrete production processes, which are characterised by the production of a discrete and identifiable product, e.g. a plastic brick, are considered. Their formalisation is then shown to be applicable to continuous production processes, in which a continuous quantity of product—for example edge banding—is produced, by discretising them.

2.1. Discrete Processes

Discrete manufacturing processes are processes that produce a specific quantity of discrete, identifiable products, e.g. plastic bricks, in each process iteration [MO19]. Representatives of discrete manufacturing are, e.g. injection moulding, additive manufacturing techniques or casting processes.

The remainder of this section provides a formalised view of the parametrisation process practised in such discrete processes. Factors that affect the process, and thereby the quality of the produced product, can be divided into factors the machine operator can adjust during production and those that are considered non-adjustable (e.g. environmental conditions). Adjustable parameters are referred to as process parameters $p \in P$, with quantified Parameters $\rho \in \mathrm{P}$ defined as $\mathrm{P} = \{(\nu, p) \in \langle \text{quantifies} \rangle \mid p \in P \text{ and } \nu \in \mathbb{R} \text{ or } \nu \in [\mathbb{R}, \mathbb{R}]\}$, i.e. p is quantified by a real valued number $\nu \in \mathbb{R}$ or a parameter range. $\langle \text{quantifies} \rangle$ describes a relation between a real valued quantification, and another entity, e.g. ν and process parameter p. A parametrisation $\phi_i \in \Phi$ of a product which is produced through iterations of the production process $i \in \mathbb{J}$, is represented as a vector consisting of concrete instances of $n \in \mathbb{N}$ adjustable parameters: $\phi_i = (\rho_0, \ldots, \rho_n)$. Analogously,

Figure 2.1.: Parametrisation process as encountered in manufacturing.

quality characteristics $q \in Q$, which are exhibited by the produced product, are quantified by $m \in \mathbb{N}$ different $o \in \mathrm{O}$ with $\{(\mu, q) \in \langle \text{quantifies} \rangle \mid q \in Q \text{ and } \mu \in [\mathbb{R}, \mathbb{R}]\}$ and described by $\theta_i \in \Theta$ with $\theta_i = (\mathrm{o}_0, \ldots, \mathrm{o}_m)$, which results from the corresponding parametrisation.

Parametrisation choices are made on the basis of non-adjustable factors $\mathcal{A} = \mathcal{Z} \sqcup \mathcal{C} \sqcup \mathcal{T}$, where the multi-dimensional expressions of product or object characteristics $z \in \mathcal{Z}$, target criteria $c \in \mathcal{C}$ and environmental factors $\tau \in \mathcal{T}$ are dependent on the manufacturing process. A production task $\omega \in \Omega$ is defined as a combination of characteristics $\mathcal{Z}$ and target criteria $\mathcal{C}$, $\omega = \mathcal{Z} \sqcup \mathcal{C}$.

Definition 2.1.1 (Parametrisation Process)**.** Based on our observations of parametrisation processes in different manufacturing domains, we propose that operators iteratively choose a parametrisation for each iteration of the production process according to a complex internal mental model [MO19] that is based on prior knowledge, education and experience:

$$\phi_{i+1} = f(\{z, c, \tau\}, \{\phi_0, \ldots, \phi_i\}, \{\theta_0, \ldots, \theta_i\}, \mathcal{M}) \tag{2.1}$$

with:

- object characteristics $z \in \mathcal{Z}$ which characterise the product that should be produced,
- target criteria $c \in \mathcal{C}$ (e.g. optimal quality or minimal cycle times),
- current environmental parameters $\tau \in \mathcal{T}$ that are dependent on the process e.g. environment

temperature, material used or machine health,

- parametrisations $\phi_0, \ldots, \phi_i \in \Phi$ for previous iterations, each consisting of quantified process parameters $\rho \in \mathrm{P}$,
- qualities $\theta_0, \ldots, \theta_i \in \Theta$ achieved at previous iterations for the respective parametrisation,
- tacit declarative, procedural and conceptual knowledge $\mathcal{M}$ available to the machine operator that has been gained through e.g. instruction, training or education.

$\mathcal{M}$ is assumed to be relatively constant during the parametrisation process, since the parametrisation process is of short duration compared to the time the operator has spent learning and building $\mathcal{M}$.

Initially, a production process is parametrised by a standard parametrisation ϕ_s of default values, which is, for example, determined during initial setup of the production process and can be dependant on the produced object. To evaluate $\phi_0 := \phi_s$, the operator executes the manufacturing process resulting in a product with the quality θ_0. The operator then tries to iteratively—the current process iteration is denoted as i—find a (pseudo-) optimal parametrisation fulfilling the constraints given by $\{z, c, \tau\}$. Just as with the initial parametrisation, the current parametrisation ϕ_i is evaluated by executing the manufacturing process and subsequently evaluating the quality of the produced product by quality assurance, leading to a new θ_i. Based on θ_i and additional constraints on the parametrisation process, such as a set budget for parametrisation, the operator decides whether to conclude the iterative process or to continue in hopes of finding a parametrisation that returns a quality better fulfilling the target criteria. In the latter case, $f(\cdot)$ is evaluated anew. Usually, however, the iterative search concludes when a good parametrisation is satisfactory in regard to the target criteria and can not be improved upon within a few iterations. After the iterative search has concluded, selecting the optimal parametrisation $\hat{\phi}^* = \arg\max_{\phi}\{\theta_\phi\}$ for production completes the parametrisation process and allows production to commence.

Underlying the operator's decision process regarding the chosen parameters is the concept of influences α_{i+1}.

Definition 2.1.2 (Influence)**.** Influences α_{i+1} for the choice of the parametrisation ϕ_{i+1} is given by:

$$\begin{aligned}\alpha_{i+1} = \{x \,|\, & x \in \mathcal{A} \sqcup \mathcal{P} \sqcup \mathcal{Q} \text{ and} \\ & x \text{ influences the choice of values for } \phi_{i+1}\},\end{aligned}$$

where $\mathcal{P}$ and $\mathcal{Q}$ are the power set of all parameters, P, and qualities, Q, respectively.

To illustrate the parametrisation process and its formalisation we refer to Figure 2.1 and consider the following simplified examples of two parametrisation processes as described later in detail in Chapter 3:

Example 2.1.1. An operator is tasked to manufacture an object *a* that has not been produced previously with optimal achievable quality. The operator starts by using the default parametrisation $\phi_s \in \Phi$ with $\phi_s = ((p_1, 4), (p_2, 5), (p_3, 6))$ with $p_1, p_2, p_3 \in P$ for the first iteration: $\phi_i^a := \phi_s$ for $i = 0$. Assume, this results in quality $\theta_0^a \in \Theta$ with $\theta_0^a = ((q_1, 1), (q_2, 0), (q_3, 2))$ and $q_1, q_2, q_3 \in Q$. Since we measure quality defects in this example, a quality of 0 in each dimension would

be considered optimal. To improve the third quality aspect $\theta^a_{i,3} = \mathrm{o}^a_3 = (q_3, 2)$ with $\mathrm{o}^a_3 \in \mathrm{O}$, which then constitutes an influence α_1 for the next iteration, $i = 1$, the operator adjusts the parametrisation to $\phi^a_1 = ((p_1, 10), (p_2, 5), (p_3, 5))$, which results in $\theta^a_1 = ((q_1, 1), (q_2, 0), (q_3, 0))$. Note, that the tuples constituting ϕ and θ are interchanged for readability, e.g. instead of $(p_1, 4)$ in ϕ_s, $(4, p_1)$ is conforming to its definition.

Underlying our definition of the parametrisation process is the following simplification: we rely on the set values as opposed to measurements taken during the parametrisation process. This is founded on two observation: firstly, in a well calibrated manufacturing process the fluctuation of measured values centre around the set value with their mean approaching the set value. Secondly, in practice, multiple parameters can affect one measured value. In this case, intended changes of a measured value can usually be traced to a specific parameter, e.g. in FDM printing the bed temperature could be changed after the first layer.

2.2. Continuous Processes

Continuous production processes, e.g. extrusion, are characterised by the production of undifferentiated products. Also, in contrast to discrete processes, the production process cannot be easily stopped during production which leads to more complex solution strategies [MO19]. To apply our formalisation of discrete production processes, we show how continuous processes can be discretised. This allows the produced products to be used in the context of digital product twins (DPTs) [EFG22], which combine physical products with data gathered during their production or use. A requirement for the successful discretisation, however, is that knowledge about which parameters or qualities correspond to which piece of the produced product is present, i.e. the time at which the parameters affected the product or the product exhibited the qualities, respectively. This knowledge can be gained by analysing the physical location in the production line where the parameter affects the produced product, or the quality is measured. Furthermore, knowledge about the individual time required until a setpoint p^{set} of a parameter p is reached, $\Delta_a(p^{\text{set}}, p^{\text{act}})$, is necessary. Since parameters are often controlled by closed control loops, reaching the setpoint is defined by a parameter specific equality function, $=_p$, that takes the variance around the setpoint introduced by the controller into account. If no actual value for a parameter, p^{act}, is measurable, the duration can be approximated by relying on domain knowledge. Finally, a quantity that constitutes the sampling rate, and thereby the size of the discretised product, has to be defined depending on the specific domain, e.g. a certain length, such as twenty centimetres, in the extrusion of edge banding or a certain volume, e.g. a barrel, in the production of liquids. Based on this knowledge about the production process along with the speed at which the production line operates, the DPT can be calculated by aggregating the values of the respective parameters and qualities that occurred within the bounds set by the size of the discretised product. The resulting product twins can be viewed as corresponding to process iterations of discrete production processes.

To illustrate the compatibility with the parametrisation process for discrete production processes presented in Section 2.1, Figure 2.2 shows the interplay of actions undertaken to mitigate a quality defect and DPTs. Here, in process iteration 3 quality characteristic q is recognised as defective. After a delay, Δ_o, in which the operator constructs a parametrisation ϕ_{i+1} that mitigates the defective q by adjusting the setpoint of parameter p^{set} of parameter p. The production process strives to control the actual value p^{act} to equal this setpoint but usually this is associated with a further delay Δ_a until $p^{\text{act}} =_p p^{\text{set}}$ in process iteration 5. While the respective sub process that is

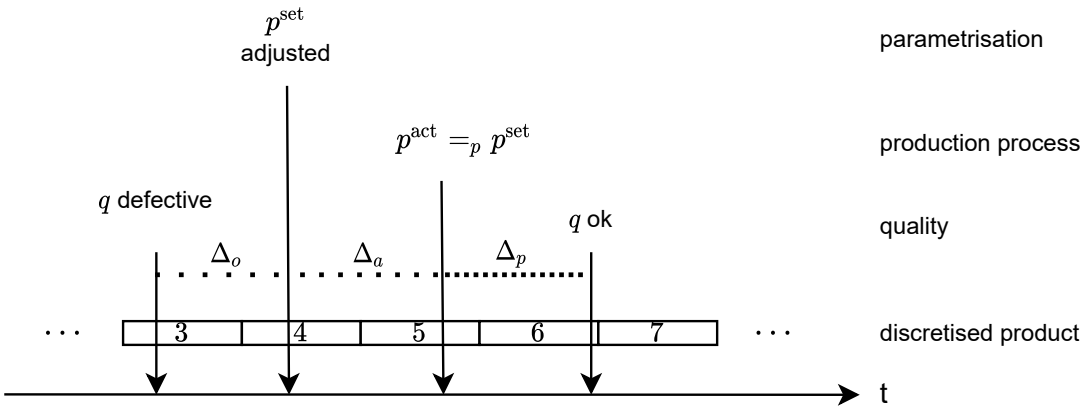


Figure 2.2.: Discretised continuous production process with a parametrisation process adjusting p because of a quality defect q. Arrows denote events that occurred during the continuous production.

affected by p has now adapted to the new parametrisation, the resulting quality q can only be measured after a further delay Δ_p after which the product passed the respective sensor or quality assurance station.

For discrete and continuous production processes, a parametrisation process starts either with an initial parametrisation or a reparametrisation to deal with an occurring quality defect. For discrete production processes, however, an iteration of the parametrisation process corresponds to exactly one iteration of the production process. In continuous production processes, however, this is not necessarily the case. Here, an iteration of a parametrisation process can encompass multiple iterations of the discretised continuous production process, i.e. multiple DPTs. Theoretically, the opposite could be true if the amount of product described by a DPT is excessively large. In practice, however, this should be avoided since otherwise no clear statement regarding the achieved quality for this DPT can be made if strong fluctuations occur during its production. Therefore, we will only address the case of one parametrisation process encompassing one or more DPTs. Since DPTs contain aggregates of the parameters and quality characteristics that were set or observed during their manufacture, we disregard any DPTs for which the DPT is not sufficiently stable according to $=_p$ or its analogously defined counterpart $=_q$, i.e. the variance for parameters or quality characteristics exceeds their respective limits. In the case of an initial parametrisation process, the first DPT created for this task constitutes the beginning of the iteration of the parametrisation process. The iteration can be concluded by two options: (1), with the DPT for which the quality characteristics are within their target range; (2) with the beginning of the next iteration of the parametrisation process as described below. In the case of a parametrisation process conducted to mitigate a quality defect, the beginning of the process is more flexible, e.g. in the example shown in Figure 2.2 DPT 4 constitutes the first iteration of the production process for which the aggregate of q is defective. Its conclusion is governed by the same conditions as the initial parametrisation process, i.e. corresponding to DPT 7, in which q is again stable within the target range which constitutes a conclusion according to the first case.

3
Case Studies

This section describes the case studies which are used throughout this thesis. The first two case studies, a fused deposition modelling (FDM) (see Section 3.1) and a polymer extrusion process (see Section 3.2), are used to validate transferability of the formalisation of the parametrisation process across varying parametrisation processes. To be able to evaluate the methods building on the real-world case studies in edge cases, we developed a framework for the generation of synthetic process data with consistent synthetic procedural knowledge (see Section 3.3).

3.1. Fused Deposition Modelling

A description of this case study has previously been published in [Nor+19] and [Nor+21] – citations of these works are not separately denoted.

Additive manufacturing has become increasingly popular since its commercial introduction in the late 1980s and early 1990s and is mostly used for rapid prototyping and manufacturing of small quantities of items, where moulding, casting or other conventional techniques would be too expensive [WG14]. Another benefit of additive manufacturing is the ability to produce complex forms, which are hard to manufacture using machining or moulding, but which can be found in medical applications or if weight is of concern. We decided to study fused deposition modelling (FDM) as an affordable and easily controllable representative for discrete production processes which contains a multitude of influencing factors and configurable parameters. FDM is an additive manufacturing technique in which a given object, usually represented through a computer aided design (CAD) model, is constructed layer-wise on a *print bed* [GRS15]. To this end, the model is pre-processed in a so-called *slicer* which breaks down the CAD model into a path that governs which areas materials should be extruded on and controls the parameters of the FDM printer (e.g. print speed or extrusion temperature). Based on these settings, material is heated until the extrusion temperature is reached and it is of sufficient viscosity to be extruded through a nozzle, which then follows the pre-planned path to construct the object.

The quality of a FDM printed object is dependent on multiple factors with differing degrees of influence [MMB15] as shown in Figure 3.1. As we focus on developing an approach for optimising parameters that are adjustable by software (e.g. printer temperature or print speed), some machine parameters (e.g. nozzle diameter or filament width) are treated as fixed. Also, since we focus on parametrisation processes rather than predicting maintenance intervals, the printers are regularly inspected and maintained to compensate for wear and tear of mechanical parts. There are several environmental factors that affect the quality of a produced product. In the following, we will provide an overview of the environmental factors most relevant to FDM printing as well as exemplary parameters used to mitigate their effects:

- Humidity has a significant impact on material quality. However, due to the time it requires

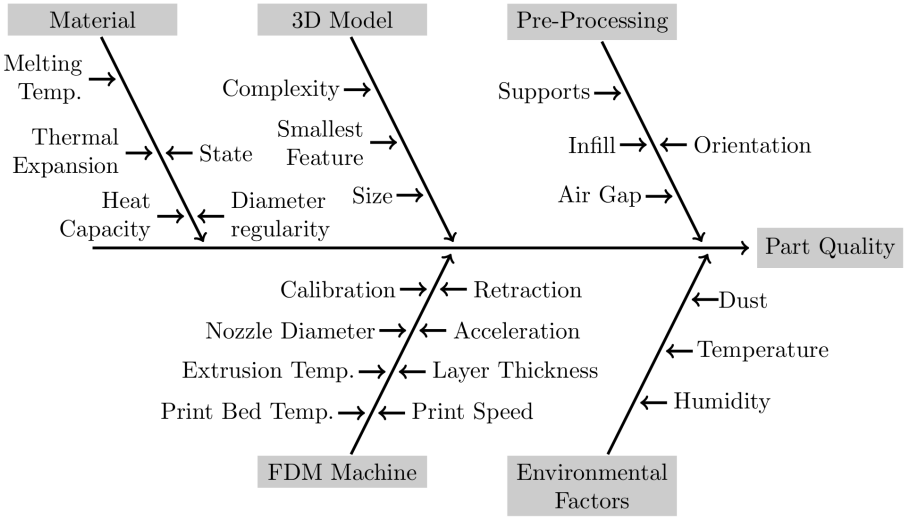


Figure 3.1.: Cause and effect (Ishikawa) diagram of the most important influence factors on product quality in an FDM production process. Previously published in [Nor+19].

to infuse the filament it is more relevant during storage of filaments than during the printing process.

- Temperature is another factor, which, in contrast to humidity, directly affects print quality. When molten polymers cool down they solidify and contract, at individual rates. In the case of FDM printers, this can lead to warping (i.e. parts of the printed object deform leading to geometric deviations and in the worst case loss of cohesion to the build plate). It typically occurs due to temperature differences between lower and upper layers. Therefore, many FDM printers have heated print beds and cooling fans to heat the lower and cool the upper layers, allowing the mitigation of this issue through parametrisation.
- Another relevant factor that often varies under real-word conditions are raw-material quality and composition, which influence print quality due to different characteristics (e.g. regarding viscosity or melting point).

The main FDM parameters modifiable through software are either related to temperature, movement or pre-processing (i.e. the generation of support structures and slicing). They are presented along with affected quality characteristics in the following:

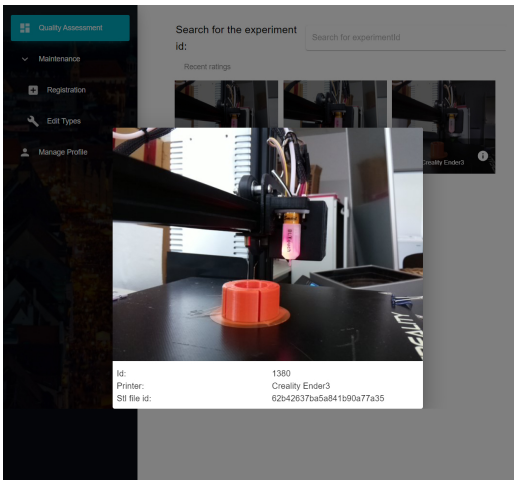
- Printing bed and extrusion temperature, for example, can offset low environmental temperatures.
- Layer thickness not only influences the optical appearance of a produced product but also the material properties during and after printing.
- Bridging occurs when the printer tries to extrude material in an area, where no (immediate) lower layer is present, thus forming a bridge between at least two supporting pillars.
- Bridging becomes harder to accomplish if the extruded filament cools down too fast or too slow. Low print speed can also result in bridging failing. While in some cases quality issues

Figure 3.2.: Creality Ender-3 with a KUKA YouBot to automatically extract finished products. Next to the printer the blue Raspberry Pi that controls the printer and measures environmental factors, i.e. temperature and humidity, is visible. Previously published in [Wie+21].

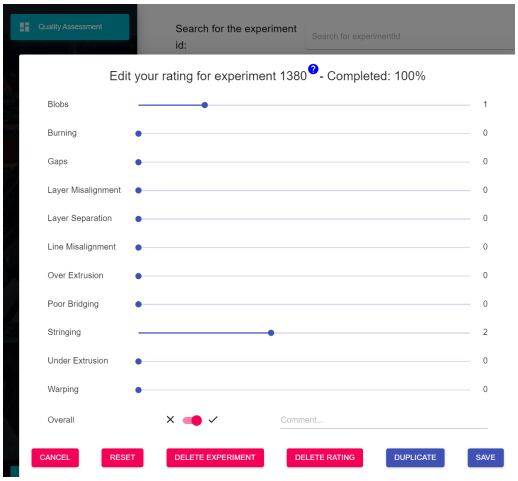
could be prevented by simple print orientation changes, e.g. bottom up [KK17], we focus on quality defect mitigation through parametrisation since we aim for a process agnostic solution and rotating the object is not also feasible.

- Print speed is another process parameter that affects quality. Slower printing allows the extruded material to better set in place, rather than getting dragged along. Therefore, adjusting print speed and acceleration quickly becomes a trade-off between quantity and quality of produced products.

Four consumer grade FDM printers—two Creality Ender-3 (pictured in Figure 3.2), one Anycubic Kossel and one Prusa i3 MK3—were used in this case study. Three of which were cartesian and one (Anycubic Kossel) was a delta printer. These were chosen due to enduring popularity of Ender-3 and Prusa as well as the different operating principle of the delta printer which, in theory, allows to study domain transfer. Also, they do not have an enclosure which allows a more direct study of environmental effects on the printing process. In practice, however, results obtainable by the Anycubic Kossel were unsatisfactory and highly dependent on mechanical adjustments which is why it was retired ahead of schedule. The resulting low amount of data that was gathered on it does not form a part for this thesis. Materials used were PLA, ABS and PETG of different suppliers, each of which has individual characteristics (e.g. a unique melting point)



(a) Detail view of the last produced product to allow the operator to ensure the rating is given for the correct product.



(b) Overview of the rating interface.

Figure 3.3.: Screenshots of the web-frontend used in the FDM case study for rating the quality of produced products. The operator is able to view the produced object in detail and show tooltips containing exemplary images for the respective quality characteristic to arrive at consolidated ratings.

and consequently necessitates different parametrisations. Furthermore, they can be differentiated by their difficulty to achieve flawless prints, with PLA being the easiest material and ABS the hardest. The slicing software's parameters define the process parameters of the FDM printers within the physical limits of the respective printer. Due to general similarity of the cartesian FDM printers and the low amount of data gathered for the only delta printer, we do not differentiate between the printers and their slightly different process parameters.

As a slicing software that acts as our HMI, we use Cura[1] which offers approx. 540 parameters for the printers we use, of which approx. 50 are regularly adjusted by operators, e.g. *print_bed_temperature* and *material_print_temperature*. We treat the respective printers' default configuration in Cura as the default parametrisation ϕ_s, as these are usually derived from at least some experience and constitute a good starting point for operators. Objects and their characteristics $\mathcal{Z}$ consist of all 3D model files, with dimensions fitting the printers' capacity. The product quality θ is defined by thirteen visually detectable quality metrics, e.g. warping, bridging or stringing. The operator performs quality assurance duties once after each part is produced and rates eleven quality characteristics per product on a discrete scale of 0 (flawless) to 5 (worst) using a web-frontend (see Figure 3.3). To arrive at consistent ratings the operators are onboarded accordingly and have sample images of the respective error in the software. While a number of target criteria can be formulated, for this case study, we limit them to the optimal quality in regards to metrics contained in $\mathcal{Q}$. Since c is constant, we can simplify $f(\cdot)$ accordingly. While environmental factors $\mathcal{T}$ can be plentiful, for this case study, we narrow them down to: humidity, temperature, printer and the raw material, separated by producer, type and colour. We do not

[1] https://ultimaker.com/software/ultimaker-cura

treat $\mathcal{P}$ as an influential factor since dependencies between process parameters are automatically calculated by Cura.

Our experimental evaluation is therefore based on

$$\phi_{i+1} = f^{\text{FDM}}(\{z, \tau\}, \{\phi_0, \ldots, \phi_i\}, \{\theta_0, \ldots, \theta_i\}, \mathcal{M}). \tag{3.1}$$

and

$$\alpha_{i+1}^{\text{FDM}} = \{x \mid x \in \mathcal{A}^{\text{FDM}} \sqcup \mathcal{Q} \text{ and } x \text{ influences the choice of values for } \phi_{i+1}\},$$

with non-adjustable factors $\mathcal{A}^{\text{FDM}} = \mathcal{Z} \sqcup \mathcal{T}$. We equipped the printers with humidity and temperature sensors and extended Cura's user interface via a plugin that directly uploads the object to be produced to the dataset. To produce the respective products, we rely on OctoPi[2] hosted on a Raspberry Pi 2 Model B[3] to control the respective FDM printer via the firmwares Klipper[4] and Marlin[5] depending on the printer.

Dataset Over the course of about two years, the operators in this case study performed a total of about 1100 iterations, of which 436 provide additional information (see Chapter 5) about the *operator reasoning* on which the data-based knowledge extraction method is based. The iterations containing additional information were performed by nine operators of differing training and process experience levels. The amount of process iterations concur with other experimental manufacturing settings [TM22].

While the FDM process offers over 500 adjustable software parameters, according to our experience as well as the data we gathered, operators tend to only vary a smaller number of these. In our case study, the operators adjusted a total of 58 parameters while 34 of these were adjusted in more than 5% of iterations and 8 were adjusted in more than 25% of instances. Individual printers, as well as differing materials and, most importantly, object geometries impact the chosen parameters, which leads to frequent reconfiguration for optimal quality.

The 436 examples are distributed onto 205 parametrisation processes with the longest requiring 20 iterations, while 144 processes saw no reconfiguration after the initial parametrisation. When considering the 61 parametrisation processes with more than one iteration, the mean length was 4.79 with a standard deviation of 3.89 iterations. While we did provide a default configuration—individually for each printer—operators rarely relied on it for the initial parametrisation. In fact, only 31 parametrisation processes started with the default, of which 19 were immediately successful, while the others needed additional operator readjustments. This illustrates the need for experienced operators and, until operators have gained this experience, the need to assist them in decision making. Operators usually varied 4.32 parameters on average per iteration. Interestingly, operators seemed not to be able to discern whether producer, chemical mixture (and thus the colour) or the raw material, was important and tended to ascribe changes to all three. Typically, one to three influences from a previous rating, describing the critical shortcomings of the product, were deemed influential. Operator confidence for parameter choices varied during

[2] https://octoprint.org/
[3] https://www.raspberrypi.com/products/raspberry-pi-2-model-b/
[4] https://www.klipper3d.org/
[5] https://marlinfw.org/

the process. Both 0% and 100% were named. Interestingly, in one instance a 100% confidence in the identified influential factors was still followed by four additional parametrisation tests with decreasing confidences (75%, 50%, 25% and 25%, respectively). We assume that there could be two possible explanations for such a pattern: (1) initial overconfidence which was lowered after the parametrisation did not have the desired effect or (2) after achieving a certain quality, the operator tries out different parametrisations to see if the quality can be boosted further in a trial-and-error manner. In one instance, where the initial parametrisation had a confidence of 0%, the operator showed 100% confidence in his attribution of influences for the last iteration and was able to conclude the process. For properties of the resulting rule base represented as knowledge graphs, we refer to Section 9.3.

3.2. Polymer Extrusion

The second case study is a representative of a continuous extrusion process, with Figure 3.4 and Figure 3.5 showcasing a schematic and a section of a real extrusion line, respectively. In industrial edge banding production, polymer granulate is melted and extruded through a tool by the extruder (shown on the right hand side of Figure 3.4). It then travels up to 100m through up to 20 machines each effecting one or more refining process steps (e.g. cooling, adhesive application, primer & colour application) on the product [HSB22]. The finished edge banding can later be applied to exposed sides of materials, e.g. particle board, by downstream manufacturers to give the illusion of a solid material. Contrary to the discrete case study, the continuous is characterised by the operators' 'supervision of a large, distributed system by monitoring data from different sources' [MO19] which, combined with the element of time, necessitates higher requirements on the operators and introduces higher cognitive load. Product anomalies, i.e. deviations from a process standard, which can occur either abruptly or gradually affect quality characteristics, can lead to negative economic consequences, e.g. due to recalls [Ven+03]. If such an anomaly is detected the goal of the operator is to 'bring the process back to a normal [...] operating state' [Ven+03]. In continuous manufacturing processes anomalies tend to be more individual and require more individual mitigations than in discrete processes [MO19].

One of the most frequent sources for quality defects is the printing of decor on the edge banding via several gravure printing process steps which are analogously controlled. In this case, detection of quality defects also requires detailed inspections, since small colour and print pattern deviations are hard to detect. Furthermore, it is hard to automatically detect, which is otherwise often the case for quality defects which can be directly measured [MO19]. This is due to the high amount of variance and indeterministic nature of decor between different products and parts of a product, respectively, which limit the applicability of machine learning solutions. Also continuous manual quality assurance is unlikely to be present due to the operators' cognitive load. Another example is the control of mass pressure of the extruder which leads to geometric defects [Gar+19] if incorrect. Depending on the parameters used to control the mass pressure mechanical defects, e.g. the extruder screw braking, can occur which highlights the importance of the interplay of different parameters. In contrast to the decor example outlined above, its parameters are available digitally and its result can be directly digitally measured via the width of the edge banding, which is why it was chosen as a use-case in the extrusion case study. Similarly to the FDM case study, we can express this case study in the formalism presented in Chapter 2 by discretising it as described in Section 2.2. As in the FDM case study, the number of target criteria is also limited to metrics

Figure 3.4.: Schematic of a continuous production line, consisting of several individual machines, of an extrusion process, starting with an extruder on the right. It is closely related to the polymer extrusion process constituting this case study. Adapted from Hörner, Schamberger and Bodendorf [HSB22].

contained in $\mathcal{Q}$. In contrast to the FDM case study, however, we do not possess measurements of environmental conditions, which limits the influential factors to $\mathcal{Q}$. Therefore,

$$\phi_{i+1} = f^{\text{EXT}}(\{z\}, \{\phi_0, \ldots, \phi_i\}, \{\theta_0, \ldots, \theta_i\}, \mathcal{M}). \tag{3.2}$$

and

$$\alpha_{i+1}^{\text{EXT}} = \{x \,|\, x \in \mathcal{A}^{\text{EXT}} \sqcup \mathcal{Q} \text{ and } x \text{ influences the choice of values for } \phi_{i+1}\}.$$

with non-adjustable factors $\mathcal{A}^{\text{EXT}} = \mathcal{Z} \sqcup \mathcal{P}$.

However, since we do not have access to a consolidated dataset, this case study is only used from a theoretical perspective to provide the practical foundations and validation for the formalisation (see Chapter 2) and considerations regarding the interface for data-based knowledge extraction (see Chapter 5).

3.3. Synthetic Dataset for Parametrisation Processes

Material in this section has previously been published in [Nor+23b] – citations of this work are not separately denoted.

To the best of our knowledge, apart from the FDM case study there are no datasets available, which contain process data and corresponding procedural knowledge that could be used to evaluate the methods developed in this thesis. To address this issue, we present a framework, PDPK[6], for generating synthetic datasets of process data and accompanying procedural knowledge in manufacturing contexts which is

1. highly configurable to suit a multitude of manufacturing scenarios,
2. modular to ensure adaptability to unforeseen scenarios, and
3. synthesises process data according to underlying rules for which representations in procedural knowledge graphs are generated.

The framework is designed to reflect the case studies presented in Sections 3.1 and 3.2. With this framework, datasets can be created that can be used to evaluate predictive quality systems, approaches of process knowledge-guided or neuro-symbolic learning and data-based knowledge extraction as well as embeddings of procedural knowledge.

[6] http://purl.org/pdpk

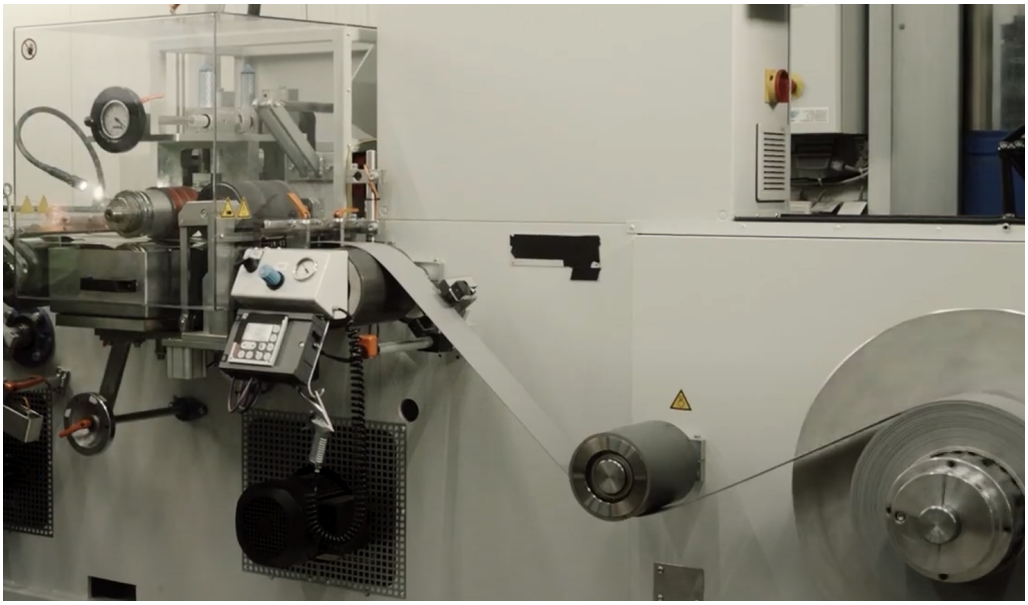


Figure 3.5.: Illustration of an inline quality assurance station situated before the final component, a haul-off that ensures that the edge banding is consistently pulled through the extrusion line. Here, the machine operator ascertains the overall quality of the produced product. Extracted from [REH23].

While synthetic industrial datasets are prevalent for image data [NHF22], they are relatively rare for process data or knowledge graphs. Industrial process data is simulated by [HT07] and [Jes+05]. Hoag and Thompson [HT07] present a framework for generating industrial size datasets using a XML specification and additional constraints. Jeske et al. [Jes+05] describe a system that 'generates data using statistical and rule-based algorithms and [. . .] semantic graphs that [contains] interdependencies between attributes'. Regarding knowledge graphs, Linjordet and Balog [LB20] discuss the use of templates for knowledge graph generation and associated problems due to leakage across data splits. Libes, Lechevalier and Jain [LLJ17] explore challenges and suggest desirable features for the generation of synthetic data for manufacturing scenarios, which we address in the next section.

3.3.1. PDPK: A Framework to Synthesise Datasets for Manufacturing

The FAIR [Wil+16] framework (see [Nor+23b] for details) for the generation of the dataset, PDPK, consists of a simulated production process, a parametrisation process as well as components to facilitate the extraction of the underlying procedural knowledge and splitting the data into train and test sets. The parametrisation process according to Chapter 2 consists of observing the quality characteristics of the previous iteration and iteratively adjusting process parameters until the target criterium is met. In our case, the target criteria consists of a subset of quality characteristics to optimise, $Q_{opt} \subseteq Q$ and a threshold $t \in [0, 1]$, for a score function

$$v(i, Q_{opt}) = \frac{\sum_{q_j \in Q_{opt}} \frac{\mu_{i,q_j} - \min(d_{q_j})}{|d_{q_j}|}}{|Q_{opt}|},$$

with process iteration i, range of the respective quality $d \in [\mathbb{R}, \mathbb{R}]$ and quality quantification μ for the respective quality.

Of the features suggested by Libes, Lechevalier and Jain [LLJ17], we implement the features relevant to our specific scenario. In particular, we address:

- **data hiding** since our data is predominantly intended to be used for training and validating machine learning algorithms, we adhere to *walk-forward-testing* and generate a test set that is separate and can be considered hidden from the training set. The generated process data is *filtered* during the creation of contiguous process iterations. Also PDPK can be viewed as a black box generator from the perspective of the algorithms that initiate it and consume its results.
- **data quality** by modelling *unreliable* physical systems with the option to introduce noise into the synthesised production process.
- **data types** by focusing on generating data relating to product quality.
- **repeatability** by allowing the seeding of the complete generation pipeline. Furthermore, by conforming to the FAIR principles a replicable versioning of the framework code is achieved.
- **reproducibility** by providing the exact versions of libraries required, the framework ensures that the generated data is reproducible, regardless of the specific setup
- **model type** since PDPK can be viewed as a *special-purpose model* for the synthetisation of artificial production processes and their parametrisation processes.
- **integration, interoperability & standards** by using widely used formats and standard, e.g. *.csv for process data and *.ttl for Resource Description Framework (RDF) representations of the rule base we allow their integration and interoperability with consumers.

3.3.1.1. Production Process

To simulate the production process on which the parametrisation process is conducted, we rely on separating the $P \times Q$ space in disjunct parts. The parameter $(P \times Q)_c$ defines which percentage of (p, q)-pairs represent causal dependencies, i.e. have functions that generate coherent values according to the selected configuration, e.g. linear, quadratic or logarithmic. Causal dependencies are further divided by the parameter $(P \times Q)_k$, which governs the percentage of causal combinations that are treated as known to the experts, i.e. the underlying causal dependencies are available for exploitative behaviour during parametrisation processes and form the bulk of the procedural knowledge. In practice, however, expert knowledge does not perfectly reflect reality since it can contain concepts of which the expert is convinced that they are true but are actually false. To compensate for this, the parameter $(P \times Q)_{k_n}$ describes the percentage of noise, i.e. erroneous $(P \times Q)$ tuples, in the set which results from applying $(P \times Q)_k$. In the following, we will use the syntax previously used for the percentages to denote the resulting sets to arrive at a more concise notation. Furthermore, the causal dependencies between ps and qs are constrained by the parameters $(p - q)_{\text{min}}$ and $(p - q)_{\text{max}}$, which denote the range from which the number of quality characteristics a given parameter affects is chosen. Unknown causal

dependencies govern the behaviour of the production process but have to be discovered by the simulated operator by exploration of the $P \times Q$ space. In any case, values for ps and qs are bound to ranges, $d \in [\mathbb{R}, \mathbb{R}]$, representing their respective limits, just as real-world parameters are typically bound to a range.

As previously noted, each causal dependency is underpinned by a function $f_{m,j} : d_{p_m} \to d_{q_j}$. For a process iteration i, the quantification μ_{i,q_j} of the resulting quality

$$\mu_{i,q_j} = \frac{\sum_{p_m \in P_{q_j}} f_{m,j}(\nu_{i,p_m})}{\left|P_{q_j}\right|}$$

is calculated by averaging the influences of all quantifications ν for all $p_m \in P_{q_j}$ that affect q_j.

3.3.1.2. Parametrisation Process

A dataset is comprised of a set of multiple parametrisation processes. Each of these consists of a series of parametrisations and their resulting quality characteristics. A randomly selected subset of quality characteristics is initially erroneous and consequently optimised over a non-deterministic number of iterations. To mimic different degrees of operator knowledge, a certain percentage of the dataset's parametrisation processes consists of exploitative behaviour while the rest consists of explorative behaviour. If the mitigation is attempted exploitatively, the process parameters are adjusted by subtracting the Δ calculated according to the following function from the previous value of p:

$$\Delta\nu_{i,p_m} = \frac{\sum_{q_j \in Q_{adj,p_m}} \Delta\nu^j_{i,p_m}}{|Q_{adj,p_m}|},$$

with $Q_{adj,p_m} = \{q_j | q_j \in Q_{opt} \wedge (p_m, q_j) \in (P \times Q)_k\}$ and

$$\Delta\nu^j_{i,p_m} = \begin{cases} 0, & \text{if } \frac{\mu_{i,q_j} - \min(d_{q_j})}{\min(d_{q_j})} \leq t \\ (f^{-1}_{m,j})'(\mu_{i-1,q_j}), & \text{otherwise} \end{cases}.$$

For parametrisation processes that should be conducted according to the explorative behaviour, process parameters are adjusted independently by $\Delta\nu_{i,p_m} = -0.1 \cdot |d_{p_m}| \cdot \lambda$, with λ chosen from $\{-1, 1\}$ representing the direction, i.e. increase or decrease, in which the process parameter is adjusted. If the score improves, the previously adjusted parameter is further decreased until the threshold t is reached. If the score decreases, the parameter is adjusted in the opposite direction and if the score does not change, another process parameter is adjusted in the next iteration. Mapped to the formalism presented in Chapter 2, a parametrisation of the synthetic case study is defined as,

$$\phi_{i+1} = f^{\text{SYN}}(\{z\}, \{\phi_s, \ldots, \phi_i\}, \{\theta_s, \ldots, \theta_i\}, \mathcal{M}). \tag{3.3}$$

and

$$\alpha^{\text{SYN}}_{i+1} = \{x \,|\, x \in \mathcal{Q} \text{ and } x \text{ influences the choice of values for } \phi_{i+1}\}.$$

3.3.1.3. Knowledge Generation

Based on the underlying causal dependencies of (p, q) pairs, $(P \times Q)_k$, rules (see Chapter 5) are generated, which can be represented as knowledge graphs depending on different representation patterns, presented in Section 9.2. The subject, q, and object, p, of the rule are given by the respective (p, q) pair. For quantified conclusions and conditions, i.e. parameters and quality characteristics, the underlying functions have to be sampled to arrive at quantified parameters or quality characteristics, respectively. Aggregated parameter quantifications, $\hat{\nu}$, for a relation between p_m and q_j are determined by

$$\hat{\nu}^{l,u}_{q_j,p_m} = \frac{\sum_{s=l}^{u-1} f^{-1}_{m,j}(s) - f^{-1}_{m,j}(s+1)}{\left|d_{q_j}\right| - 1}, \tag{3.4}$$

with $l = \min\left(d_{q_j}\right)$ and $u = \max\left(d_{q_j}\right)$, the lower and upper bounds of the range of the parameter, respectively. Conditions, i.e. quality characteristics, are quantified by determining ranges of q for which different parameters should be applied, e.g. by utilising estimators, such as L2 [FD81], or providing a domain specific amount of ranges. For each range, quantified parameters are calculated as described in Equation (3.4) with l and u the lower and upper limit of the respective range. An exemplary rule would be: *If q within [1,3] adjust p by 2.4* .

3.3.2. Dataset

In this section, we describe a dataset that we propose to serve as a benchmark to increase the comparability of the performance of evaluated methods even if different datasets or domain specific adaptions to PDPK have been made. Consequently, it is intended to be used for all possible application scenarios of PDPK, i.e. data-based knowledge extraction methods, embedding methods on procedural knowledge and for downstream tasks utilizing the procedural knowledge. We provide insights into the PDPK parameters that form the basis for the synthetisation and, consequently, characteristics of the resulting dataset. They are chosen based on the real-world case studies presented in Sections 3.1 and 3.2.

3.3.2.1. Parameters

Using the parameters shown in Sections 3.3.1.1 and 3.3.1.2 as well as those discussed in the following, we can replicate characteristics of datasets from different manufacturing domains. To arrive at benchmark data with which comparability between different approaches can be ensured even if a different PDPK parametrisation is used, we propose a parametrisation that is inspired by observations of manufacturing processes in both fused-deposition-modelling as well as polymer extrusion. While the complexity in regards to amount of parameters and quality characteristics is strongly influenced by the FDM case study, the goal is not to perfectly replicate one of these processes. In both cases, a significant fraction of possible (p, q) combinations are not causally related since usually a p or q is only influenced by a small number of qs or ps, respectively. E.g. while there might be 20 different adjustable parameters, the surface finish quality characteristic is only dependent on material and air temperature but independent from the speed at which material is fed into the extruder. Therefore, we set the number of (p, q) combinations with causal dependencies, $(P \times Q)_c$, to 10%. Also, unknown influences, e.g. of an environmental nature, are present in both scenarios. To address this observation, we set $(P \times Q)_k = 75\%$ to

leave ample room for exploitative behaviour. Since we want to mimic operators with different levels of experience, we introduce 25% of erroneous information to the expert knowledge via $(P \times Q)_{k_n} = 25\%$. The number of parameters, $|P|$, is set to 46, while the number of qualities, $|Q|$, is set to 16. $(p-q)_{\min}$ and $(p-q)_{max}$ are set to 1 and 14, respectively. Also the maximum length, i.e. the number of process iterations, of a exploitative parametrisation process is set to 15 as in all but the most extreme cases the synthesised parametrisation processes conclude within 15 process iterations. All modules' pseudo-random generators are seeded with 42.

3.3.2.2. Properties

The resulting dataset, which in size and structural complexity is comparable to the FDM dataset, contains 500 process iterations, and 56 parametrisation processes, with the average parametrisation process requiring an average of 8.93 process iterations. In each process iteration an average of 2.64 parameters were adjusted. While the number of process iterations might appear small, it is realistic in domains that produce a large variety of products at limited batch sizes as well as experimental settings often found in related work [TM22]. For properties of the resulting rule base represented as knowledge graphs, we refer to Section 9.3.

Part II.

Data-Based Knowledge Extraction

To counteract the effects of demographic change, i.e. loss of workforce and process knowledge, on manufacturing companies, this part proposes a data-based method to extract procedural operator knowledge pertaining to the parametrisation process encountered in manufacturing. Furthermore, the extracted knowledge can be utilised in downstream applications, e.g. assistance systems through approaches shown in Parts III and IV. Chapter 4 provides an overview over different kinds of knowledge and the state of the art of knowledge extraction using data-based as well as interview-based approaches. In Chapter 5 we detail the elicitation process which yields example-based knowledge segments which are aggregated into a coherent knowledge base from which rules are extracted in Chapter 6. Lastly, Chapter 7 evaluates the proposed method on real-world as well as synthetic data. This part draws on work previously published in [Nor+21; Nor+22a; Nor+23b; Nor+23a] and as such uses parts of these publications. The corresponding passages are not explicitly cited again for the sake of readability.

4
Related Work

This chapter presents related work of knowledge acquisition and extraction as well as foundations of the concepts introduced in the following chapters. It starts with an overview of different kinds of knowledge before detailing different approaches to knowledge elicitation, extraction and acquisition. Lastly, foundations of knowledge graphs which we use as a data structure for the knowledge extraction are introduced.

4.1. Definition of Knowledge

In the field of epistemology, the theory of knowledge, countless definitions of knowledge have been proposed. While a detailed overview of these is out of the scope of this thesis, this section will introduce the concept of knowledge to be able to differentiate between different kinds of knowledge relevant for this thesis and encountered in the relevant literature.

One such definition can be found in the field of education, where Krathwohl [Kra02] presents a revised version of Bloom's [BK56] taxonomy of educational objectives including the types of knowledge included in these. He differentiates between factual, conceptual, procedural and meta-cognitive knowledge. Firstly, *factual knowledge* describes the basic elements that have to be known to be acquainted with a domain and are therefore a prerequisite to solve problems. It consists of a knowledge of terminology and knowledge of specific details and elements. Secondly, *conceptual knowledge* describes the interrelationships among the elements classified under factual knowledge that allows them to 'function together' [Kra02]. It is further divided into knowledge of classifications and categories, knowledge of principles and generalisations as well as knowledge of theories, models and structures. Thirdly, *procedural knowledge* describes knowledge about how to do something, i.e. skills, when to apply which skill and how to inquire to gain knowledge of new skills. It is further divided into knowledge of subject-specific skills and algorithms, knowledge of subject-specific techniques and methods and knowledge of criteria for determining when to use appropriate procedures. Lastly, *metacognitive knowledge* which describes knowledge related self-awareness and knowledge of (self-)cognition. It is further divided into strategic knowledge, knowledge about cognitive tasks, which includes contextual and conditional knowledge as well as self-knowledge.

In the context of knowledge acquisition (see Section 4.2) Shadbolt and Smart [SS15] differentiate the experts' knowledge into domain, inference, task and strategic knowledge. Firstly, *domain knowledge* is 'knowledge that describes the concepts and elements in the domain and relations between them' [SS15]. As such, it can also be referred to as declarative knowledge, and is strongly related to Krathwohl's definition of factual and conceptual knowledge. Secondly, *inference knowledge* refers to a high level description of expert behaviour by describing the 'type of inferences that will be made [by the expert] and what role knowledge' plays in them [SS15]. As

such, it describes how the components of expertise are used in the overall system. Thirdly, *task knowledge* describes the goal and which tasks need to be accomplished in which order to reach that goal. This conforms to Krathwohl's definition of procedural knowledge but places a stronger emphasis on the sequential order of the execution of tasks to reach a goal, which is only implicitly mentioned in Krathwohl [Kra02]. Lastly, *strategic knowledge* 'that monitors and controls the overall problem solving process' [SS15], which also forms a subcategory in Krathwohl's definition. They highlight that knowledge within any of these categories might be implicit—also referred to as tacit—or explicit.

An orthogonal taxonomy with partly overlapping terms is proposed by Gruber [Gru89], who differentiates between substantive knowledge, i.e. analytical knowledge about what is perceived in the given domain and whether it is valid, and strategic or control knowledge, i.e. knowledge about what action to perform next. While substantive knowledge can be seen as related to factual knowledge in Krathwohl's taxonomy, Gruber's strategic knowledge is best covered by procedural knowledge.

In contrast, Paiva, Roth and Fensterseifer [PRF08] consider 'know-what', i.e. knowledge about where to find required information, and 'know-how', i.e. knowledge required to deal with challenges. The first could be classified as factual while the second is closer related to procedural knowledge in Kratwohl's taxonomy.

The knowledge most relevant for our scenario is procedural knowledge as outlined by Krathwohl. In particular, the order in which tasks are executed, as explicitly modelled in Shadbolt and Smart [SS15], is not of significance in parametrisation processes of the case studies we consider. Therefore, we will limit our use of the term procedural knowledge to knowledge that describes how to solve tasks.

4.2. Knowledge Acquisition

Knowledge Acquisition describes the process of 'gaining understanding about the processes underlying the observable behaviour of an entity' [LA16]. These entities can be categorised in natural, man-made and human activity [LA16]. Since we focus on extracting expert knowledge about a man-made system through analysing human interaction with this system, we limit the overview of knowledge acquisition approaches to the last category. The knowledge acquisition process can be divided into three main stages [Boo89; Coo94; CKH06]. While the literature has minor inconsistencies in terminology and understanding of what constitutes the respective steps, they can be summarised by elicitation, analysis and representation [LA16]. Firstly, knowledge is *elicited* by a knowledge engineer or elicitor from either experts or other sources, e.g. procedure manuals or task documentations [Coo94]. Secondly, the elicited knowledge is then *analysed* or explicated to discover structures, findings and meaning [CKH06]. Lastly, *representation* or formalisation describes the modelling into a computational model [Coo94; Boo89] and externalisation and presentation [CKH06] of the gained knowledge. In the following, we will use the term knowledge extraction as a synonym to knowledge acquisition.

Knowledge acquisition methods are widely categorised into the following four categories: informal techniques, i.e. observation and interviews, process tracing techniques, conceptual techniques and formal methods [Coo94; WS04; CKH06; YF11; Sha+15]. To achieve a better overall result, knowledge acquisition methods of different categories are frequently combined [Bai79; Coo94; YF11; HSB22]. In the following, we present those categories, however, a more detailed

understanding of the categories, e.g. subcategories or examples of interview or observation methods, is beyond the scope of this thesis and already well covered in literature [Coo94; LA16; BP19]. *Informal techniques* rely on observing or interviewing the expert in order to elicit, often tacit, knowledge about how to perform the task in question. As such, they are highly verbal and descriptive, necessitating a knowledge engineer who is well versed in the application domain. Also, they can lead to copious amounts of unstructured data and are prone to subjectivity of the knowledge engineers. On the other hand, they are well suited for the elicitation stage, provide the opportunity to explore a topic to the desired depth and require less advance preparation with the task than the following methods. Gavrilova and Andreeva [GA12] provide an analysis of common informal methods for their suitability for the extraction of tacit knowledge and a classification according to the role of the knowledge engineer. *Process tracing techniques* consist of think-aloud protocols that are conducted during or after the execution of a task. They are therefore example-based, which could provide more specific knowledge and exhibit a higher degree of structure compared to informal methods. Depending on the specific method, they are well suited for the elicitation as well as analysis stage. Process tracing techniques can be split into five subcategories: verbal reports, non-verbal reports, protocol analysis, decision analysis and cognitive walk-through [LA16; Coo94]. Through their example-based nature and the 'quantitative analysis of decision points within a task' constituting decision analysis [Coo94], they are strongly related to our proposed data-based knowledge extraction approach. *Conceptual techniques*, however, can use multiple sources of expertise, e.g. multiple experts or procedure manuals to 'produce structured, interrelated representations of relevant concepts within a problem domain' [LA16]. Therefore, they are well suited for the representation stage. *Formal methods* use assumptions of the application domain and the human mind and cognition to computationally simulate human task-solving behaviours. As these assumptions are unlikely to yield satisfactory simulations in complex manufacturing processes, where physical processes and tacit domain knowledge have to be simulated at a high level of detail, we will not further detail these approaches.

Leu and Abbass [LA16] introduce an orthogonal classification, consisting of human, human-inspired and machine knowledge acquisition agents. *Human agents*, i.e. humans as elicitors, conduct the methods categorised according to the three techniques, i.e. informal, process tracing or conceptual, as described above. *Human-inspired agents* contain automated variants of the methods used by human agents. Additionally, they include the category of formal methods. *Machine agents* utilise methods from the field of knowledge discovery and data mining (KDDM). This field encompasses 'the entire knowledge extraction process, including how the data is stored and accessed, how to develop efficient and scalable algorithms that can be used to analyse massive datasets, how to interpret and visualise the results, and how to model and support the interaction between human and machine' [KM06]. Also KDDM extracts 'both the structure and the causal relations existing in artefactual data' [LA16]. As such it can be considered as part of knowledge acquisition. While there are some discrepancies regarding the stages of the KDDM process [KM06] they can be divided into three stages, i.e. data pre-processing, data mining and knowledge interpretation [LA16].

Differences between the categories described above arise in the amount of data that they are able to process which is limited by the amount of elicitors and experts, known as the 'knowledge-engineering bottleneck' [KN11], in the case of human agents and limited by the experts and to a lesser degree the elicitators in human-inspired agents. For machine agents the amount of data is only limited by processing power and data storage, however, they require a high amount of data

to function properly. The richness of the analysis achievable is comparable for these categories, however, the amount of insights that can be provided is limited by the amount of data that they can process, leading to a preference of machine agents if they are applicable. The elicitation phase of our approach can bee seen as a human-inspired agent while the aggregation and interpretation is done via a machine agent.

Leu and Abbass [LA16] note, that a significant part of machine agents for knowledge acquisition are based on multi agent systems. For these, the multifaceted concept of *trust* as defined in the field of *Organic Computing* deals with complex heterogeneous systems similar to our socio-technical system use case [Ste+10]. One important aspect of which is *credibility*, which assesses good faith participation and competence of partners. While we currently handle this issue implicitly during aggregation, dedicated methods to establish trust might improve future results. Our scenario can be classified as a socio-technical self-adaptive system, with the operator adapting to observations of the manufacturing process. According to Ramirez, Jensen and Cheng [RJC12] the most relevant sources of run-time uncertainties are related to interactions of the system and its context. To address these, the application of subjective logic to aggregate run-time observations into actionable knowledge has been proposed by Petrovska et al. [Pet+20]. However, since their methodology is centred on cyber-physical systems and ours on human operators it is not directly transferable.

4.2.1. Knowledge Extraction Methods

In the following, we will present knowledge extraction methods for human, human-inspired and machine agents with a focus on industry.

Human and Human-inspired Agents In the context of manufacturing scenarios, Deslanders et al. explored structured interviews to extract production rules for parameters that directly influence part quality [DP97]. Combinations of the techniques mentioned above—selected according to their time and personal efforts—have been successfully applied to the extraction of tacit procedural knowledge as mixed method approaches in extrusion and additive manufacturing [HSB20; HSB22; Gra+17]. Furthermore, eliciting tacit knowledge in visual inspection use cases is addressed using informal methods by [Joh+19]. Guerra-Zubiaga [Gue17] focusses on the elicitation of tacit motion sequences in manufacturing. However, as mentioned above, these approaches are time intensive for operators and knowledge engineers especially if quantified knowledge should be extracted.

Seymoens et al. describe a different approach that aims to extract tacit knowledge by integrating domain experts during algorithm creation [Sey+19]. However, the experts' knowledge is only implicitly contained in the resulting algorithms. Therefore, it is not available explicitly for onboarding or other downstream purposes. The same is true for the approach of Wilson et al. [Wil+22], that employ *Quality Function Deployment*, i.e. ranking the 'influence of process attributes on desired process outcomes [...] in order of influence [for] the purpose of defining features for a machine learning model' [Wil+22], and graphical highlighting to increase the weight of the respective features for further processing by a support vector machine.

Gebus et al. propose a factory-wide knowledge-based decision support system for remedying faults. They argue for an approach limiting the knowledge engineer's interactions with domain experts, instead enabling experts through systems that interview them on a case-by-case basis [GL07]. However, they noted difficulties with heterogeneous levels of expertise of and acceptance by the experts. Kharlamov et al. proposed ontologies to represent industrial information models in

manufacturing scenarios. In addition, they presented a tool which allows engineers to build these ontologies, without deep knowledge on semantic technologies or ontology creation [Kha+16]. We follow their approach to move ontology or knowledge formalisation towards people that are closer to the process as opposed to external ontology engineers. However, our approach requires no previous knowledge of semantic technologies and is more suited for operators, which usually have a lower level of education compared to engineers. Also, while Kharlamov et al. analyse information models at a level of abstraction suited to integrate well with a manufacturing execution system, we focus on parametrisation processes which require a lesser degree of abstraction.

Aerts, Deryck and Vennekens [ADV22] report of difficulties in applying informal methods for design problems even at a high degree of abstraction in the elicited knowledge and consequently transitioned to a more structured *Decision Model and Notation* (DMN) approach. To narrow down the solution space for engineers in a decision support system, they employed first order logic to encode design selections with a low degree of abstraction. Since this proved 'significantly more challenging' [ADV22] than extracting the more abstract knowledge they chose to validate it with the extracted abstract knowledge. Even if during the narrowing of the solution space quantified conditions were used for some rules, it still leaves significant freedom for the engineers to make decisions on a case-by-case basis. While we also strive towards quantified conclusions to cater to the needs of our application domain, we have stronger constraints regarding the time required by the approach which renders this approach unfeasible in our case.

Nicholson et al. [Nic+20] use a social process based on Delphi [LT+75] for the creation of Bayesian networks that represent domain knowledge. Through the use of a Delphi based process, cognitive biases that affect individual experts are mitigated [OHa19]. Another crowdsourcing approach to knowledge extraction and decentralised knowledge graph construction is pursued by Wang et al. [Wan+19]. By voting on proposed triples, co-workers create a knowledge graph of techniques and rate a worker's perceived technical skill for a certain technique. While we also pursue a crowdsourcing approach, we are focused on tacit procedural and conceptual knowledge rather than factual knowledge.

Machine Agents Logical analysis of data can be used to extract if-then rules as shown for parameters and influences in a fault diagnosis scenario by Bai et al. [BTW18]. However, they are very compute intensive and have not been validated in the context of procedural knowledge.

In addition, several natural language processing [Ban+18; Sch+12] based approaches exist for various domains, such as oil and gas or health care [Asi+18]. A different approach is described by Katti, which allows ontologies to be generated from annotated source code of manufacturing execution systems [Kat20]. However, both kinds of approaches are not directly applicable to our scenario since they are limited to factual knowledge. An approach that is specific to procedural knowledge is presented by Pareti et al. [Par+14]. However, it still requires textual descriptions of the operators' thought processes during parametrisation which rarely exists for manufacturing processes [Rin19], possibly due to the knowledge engineering paradox, which states that the more competent the experts, the less they are able to describe their solutions [GL07].

Yet another approach is presented by Sun et al. [Sun+14] that follow an example-based approach based on a recommender system that operates on point cloud data and expert decisions. If the recommended manufacturing plan differs from the experts' choice, informal techniques are conducted to elicit the underlying rules. These are represented in Nested Ripple Down Rules, i.e. cascading if-then rules, which can be used by workers to improve their understanding of the

process and to improve the recommender system.

In an additive manufacturing scenario a purely data-based approach that uses the Classification And Regression Tree (CART) algorithm, a type of Decision Tree (DT), to be able to predict surface roughness based on part location and orientation and return the resulting rules in a knowledge graph has been developed [Ko+19; Ko+21]. However, these works only treat a very limited subset of the possible parameter space and lack a quantitative evaluation.

Approaches that automatically analyse task documentations compiled by human experts have also gained traction [Ber21; Sch+22]. They allow knowledge extraction without further consultation of the expert. However, their application is limited to domains in which the expert is obliged to document the executed task in a structured manner.

In human-in-the-loop or interactive learning systems, an expert gives feedback to the suitability of a learning system's prediction, see for example [Sun+18]. This, like our approach, constitutes a data-based knowledge extraction. However, this knowledge is usually fed directly into the learning system and therefore remains implicit as opposed to our approach which makes it explicit. Additionally, in our case, the externalised knowledge originates from the same operator that defined the specific parametrisation, leading us to assume that it is of higher quality. Furthermore, interactive systems are at risk of influencing the expert by the data that is shown to them [Dae+18], which is not the case in our approach. While automated prompt generation was shown to outperform the human prompts regarding the quality of knowledge extracted from language models [Shi+20], it is questionable whether it is suited to compete with more traditional elicitation methods since language models can only contain knowledge that was previously elicited and part of their training data. We included this work since even though it has so far not been validated in industry, it is a step towards active knowledge acquisition systems that 'actively and autonomously seek for and fill the agreed ontological construct' [LA16].

Another related field of research are rule mining and explainable knowledge graph clustering approaches, e.g. AIME+ [Gal+15], AnyBurl [Mei+19] or ExCut [Gad+20]. However, they often require ontological constructs, e.g. knowledge graphs, to operate on and are therefore not suited for the initial elicitation. Also, 'rule learning with searching over the graph always suffers from efficiency due to huge search space' [Zha+19b]. Still, they could be viewed as alternatives to our aggregation methods—differing, however, in required amount of data and purpose. Rule mining is also conducted on raw data, e.g. by Khan and Parkinson [KP18b] that mine server logs to arrive at association rules that help in identifying mitigation approaches for security relevant misconfigurations. However, the amount of data present is orders of magnitude larger than in our case studies. Thus it was not deemed plausible to directly transfer to our scenario and consequently not investigated more closely.

4.3. Knowledge Graphs and Ontologies

Knowledge graphs have been 'considered the coming of age of the integration of knowledge and data at large scale with heterogeneous formats' [GS21], which is why we decided to use them as a basis to represent the extracted rules. They are a graph-based representation of knowledge with nodes or vertices representing 'entities of interest and whose edges represent potentially different relations between these entities.' [Hog+21]. While there exist some exceptions, the data model underlying knowledge graphs can be classified in three different types: directed edge labelled, heterogeneous or property graphs [Hog+21]. Firstly, *directed edge labelled (del) graphs* or

multi-relational graphs use directed labelled edges, e.g. president of, to connect different entities, e.g. Joe Biden and USA. The corresponding piece of information *(Joe Biden, president of, USA)* is treated as a triple of head entity, relation and tail entity and can be stored accordingly. They are standardised by the *resource description framework (RDF)* [Cyg+14]. Secondly, *heterogeneous graphs* or heterogeneous information networks are akin to directed edge labelled graphs with the difference that type annotations of the vertices form a structural part of the graph. To express that USA is a country, one would have a vertex *USA:Country* in a heterogeneous graph whereas it would have to be expressed as an extra triple in del graphs, i.e. *(USA, is a, Country)*. Lastly, *property graphs* allow the storage of property-value pairs for entities or relations, which provides a greater flexibility when modelling data [Hog+21]. However, they are not yet standardised which leads to less literature and libraries building on them. For this reason, we use property graphs only as a data structure for the data-based knowledge extraction, and represent them as del graphs in Part III to be able to use the wealth of embedding models available in literature. Since knowledge graphs are a topic that relates to multiple parts of this thesis some related works are placed in parts where the respective aspects have a higher relevance: related work for industrial applications of knowledge graphs are discussed in Section 8.1; knowledge graph embedding methods that transform the graph into numerical features are reviewed in Section 8.3 as their understanding is not necessary for this part.

To increase the interoperability of knowledge graphs, *ontologies*, that 'are a concrete, formal representation—a convention—on what terms mean within the scope in which they are used', are adopted [Hog+21]. In the following, we give an overview of ontologies in use in manufacturing and the FDM domain in particular. We do not conform to them since they provide certain limitations in regards to the information we want to represent and if necessary they can be easily integrated to our representations for interoperability purposes. While for FDM, ontologies are focused more on capabilities of the printer and less on actual parameters [DR17; HS19; Ko+19], process ontologies, e.g. CDM-Core [Maz+16] and ADACOR [BL07] contain concepts for machine *setup* or *launching* operations. However, since parameter adjustments are also executed during production we do not utilise these vocabularies. Concerning products, MCCO [Usm+11] contains the concept of rejected parts. Related to that, CDM-Core introduces geometric flaws of different, fixed severities. Quality characteristics as described in this thesis, however, are not limited to geometric flaws and need to be expressed in a finer granularity. Also, due to the abstraction level of the generated data, the concept of a product, the produced quantity or the quality of raw materials as present in ONTO-PDM [PDT12] are not modelled in our case and are therefore not used. Furthermore, Braun et al. [Bra+07] make the argument that knowledge elicitation and formalisation is a 'learning process in which the involved individuals deepen their understanding of the real world and of an (appropriate) vocabulary to describe it'. Consequently, integrating the discovered knowledge segments in an ontology seems only suitable after gathering significant knowledge in a domain. Additionally, depending on the specific application scenario different ontologies could be prevalent which lowers the value of integrating our concepts with one of these ontologies in advance. Therefore, the ontology integration is considered to be out of the scope of this thesis.

Knowledge Graph Completion Our aggregation approach relates to the problem of knowledge graph completion for predicting relations between pairs of known nodes in a graph, as candidate triples have to be checked whether to be included in the knowledge graph or not.

Consequently, work done on filtering candidate triples by Borrego et al. [Bor+19] is conceptually related. Weller et al. [Wel+20] is a further related work, who predict missing relations for a given head entity without having knowledge about the tail entity based on a combination of association rule learning and community detection. However, the main difference between these approaches and ours is that in knowledge graph completion candidate triples are usually generated based on information contained in the knowledge graph, while in our case they are provided by process data and operator information gained in a manufacturing process. Also, in KG completion an existing graph is further refined, whereas in our case the initial graph has to be created, which leads to inherent differences in filtering methodology.

5 Elicitation

The task of (re-) parametrisation, i.e. finding a parametrisation for which the manufacturing process produces satisfactory results, is currently mostly done by experienced operators following an iterative time and resource intensive process, which is similar among different manufacturing processes (see Chapter 2 and the grey part of Figure 5.1). To recapitulate, we give a brief overview of a parametrisation process: at first, the manufacturing process, which is influenced by environmental influences, is executed with a standard parametrisation. Then, the quality of the produced product is evaluated in regard to specific target criteria by the operator or specific quality assurance personnel. Based on the quality, the operator decides whether to accept the current parametrisation or continue the parametrisation cycle. If the operator decides to continue the parametrisation process, a new parametrisation is chosen, which is believed to be suitable to address quality defects of the result of the previous process iteration. This process is repeated until a successful parametrisation is found or other underlying hindrances, such as identified hardware problems, have to be addressed. In the latter case, the parametrisation process needs to be restarted after the underlying hindrances have been mitigated.

To extract tacit and procedural operator knowledge required for successful parametrisation processes, knowledge extraction processes (see Section 4.2), which are usually conducted by knowledge engineers and experts, need to be performed. Here, knowledge extraction of tacit knowledge constitutes the knowledge acquisition bottleneck [SG87], since, even with modern methods, it remains a time intensive process [HSB22]. Also, difficulties appearing during knowledge extraction have to be addressed, such as the knowledge engineering paradox [GL07]. Furthermore, automatically extracting knowledge from documents is not feasible for the procedural knowledge in parametrisation processes since, often, it is not available in documents in industrial domains [Rin19].

In this chapter, we present an approach to gather semantic segments of procedural knowledge by extending human-machine-interfaces (HMIs) of the machine or production line controls, giving the experts the possibility to provide insights into their reasoning as to why a certain parametrisation is chosen (visualised in Figure 5.1 by the arrow from relevant influences an operator thinks of, to the knowledge extraction module). This has the benefit that vertices and edges in the knowledge graph automatically have a consistent syntax and terminology as the production data, facilitating its application and integration. Our approach can be considered to be a human-inspired knowledge extraction agent according to the classification shown in Leu and Abbass [LA16] automating interview processes that would otherwise be conducted by an elicitor in process tracing techniques. The approach has two benefits over existing work in knowledge extraction: (1) we assume that it is less intrusive than separate interviews, which are often a necessary part of expert knowledge extraction [HSB22], leading to fewer requirements of specially trained personnel and (2) it is more time efficient since fewer personnel is needed and

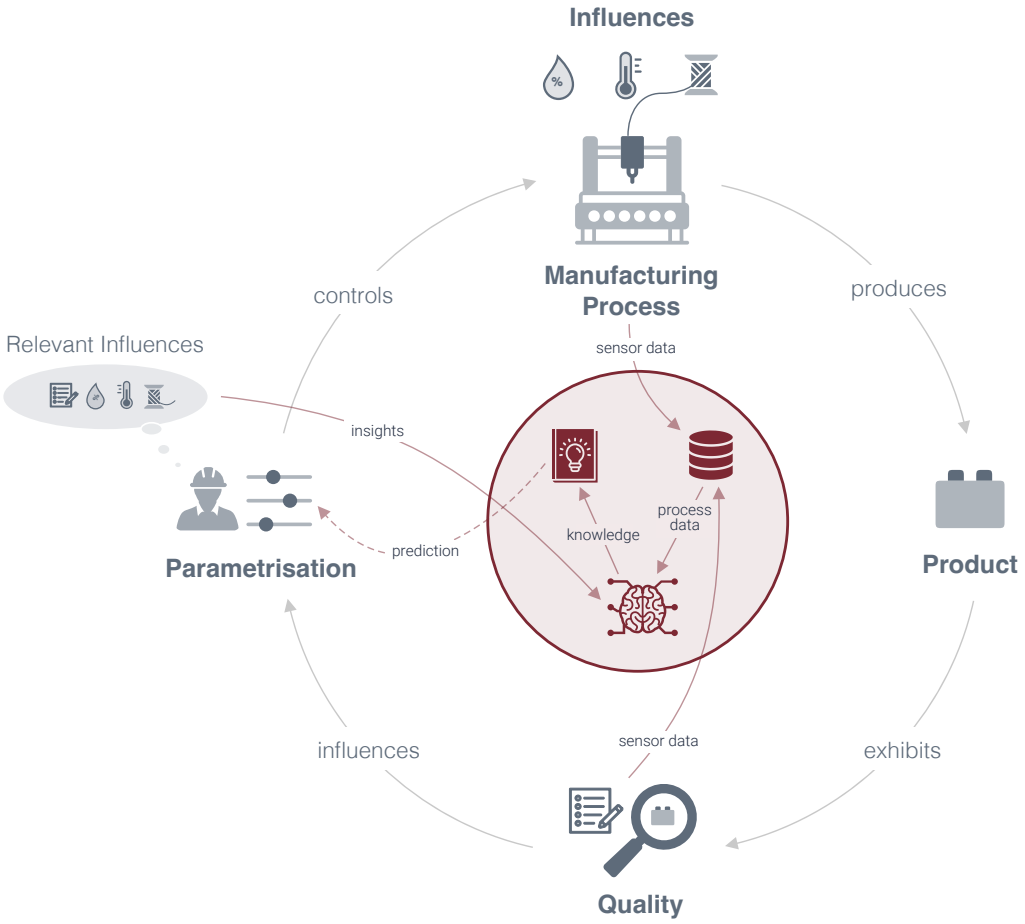


Figure 5.1.: Currently practiced parametrisation process (grey) consisting of a process that is controlled by a parametrisation, which is chosen iteratively by an operator to minimise quality defects of the produced part in regards to target criteria. The proposed integration of the data-based knowledge extraction into the process is shown in red.

experts often have a lower workload between launching or adjusting the process execution and evaluating the product. This period can be used by the experts to share their reasoning with our approach. Furthermore, our approach addresses the knowledge acquisition bottleneck through automation and, since it is example-based, does not suffer from group dynamics that can occur if several experts are interviewed together [Nic+20]. Additionally, it is in principle able to function without a knowledge engineer, whose input is only needed if the extracted knowledge should be integrated with other available domain knowledge.

To extract tacit procedural as well as conceptual knowledge of the operators' mental model $\mathcal{M}$, we propose a system based on recording individual parametrisations, ϕ, returned by $f(\cdot)$ (see Chapter 2) and the relevant influences for the parametrisation. For each evaluation of their respective function $f_{\mathcal{M}}(\cdot)$, i.e. for each iteration of the parametrisation, operators are asked to

provide a domain specific subset (see Chapter 3) of

$$\alpha_i = \{x \,|\, x \in \mathcal{A} \sqcup \mathcal{P} \sqcup \mathcal{Q} \text{ and } x \text{ influences the choice of values for } \phi_i\}$$

of the most influential values from $f_{\mathcal{M}}$'s inputs for the parametrisation ϕ_i. These influences are selectable in an extension of their familiar human machine interface (HMI). Furthermore, they optionally have they have the possibility to provide their confidence $\zeta_i \in [0,\, 1]$, that the chosen influences are correctly identified. Based on these influences, knowledge segments are generated and represented in knowledge graphs with quantifications of the entities represented as properties. Note, that α_i refers to factors observable in the current process iteration i. If qualities are a part of α_i, the corresponding quantified qualities are selected from the quality achieved in the previous iteration, θ_{i-1}, since these are the values the operator is presented with. Since we assume that unadjusted parameters respective to the initial parametrisation ϕ_s or the previous iteration ϕ_{i-1} are irrelevant, we only generate vertices and edges for those parameters that were adjusted. This leads to a considerably more compact representation of the relevant segments of the operators' knowledge. We chose knowledge graphs as a data structure for the information in question since it facilitates the prospective combination of multiple knowledge graphs describing different aspects of the manufacturing process [WBD17].

Depending on the domain and the respective situation of the operator, α_i can be provided either for all influences at once, i.e. in a combined, or in a sequential manner for each influence separately. In the following, we present two methods to generate the respective knowledge graphs.

5.1. Combined Influence Elicitation

In the combined case, a knowledge graph, k_i, representing the knowledge segment provided by the operator in a process iteration i, is constructed. It is defined by vertices $v_i = \delta_i \cup \alpha_i$ and edges $e_i = \delta_i \times \alpha_i$, where $\delta_i = \{p \in \phi_i \,|\, p \text{ is adjusted during the parametrisation process}\}$ denotes all adjusted parameters. Note, that since the experts are not asked to establish direct links between individual influences and parameters, we are not able to differentiate which influences α resulted in which parameters p if multiple quality defects are addressed at the same time. Therefore, we have to treat all edges as candidates for influenced-by relations which can lead to a more complex graph. However, this decreases the task complexity for operators considerably, since they do not have to decide which parameter relations actually exist and save time and effort to denote them accordingly.

An example of an HMI in which the operator is able to highlight all influences at once in a a combined manner can be seen in Figure 5.2. Here, we show the extended Cura user interface as used in the FDM case study (see Section 3.1). On the right, we can see the parametrisation pane where the process parameters can be adjusted. As soon as the operator confirms the parametrisation he is presented with a pop-up on the left in which he can denote environmental parameters and quality characteristics of previously produced parts that influenced his parametrisation. Additionally, they are able to share insights via a text field and give an indication of their confidence for the chosen parametrisation. The information in the text field, however, was not used for knowledge elicitation purposes since it was not frequently used by the operators. In theory, this would provide operators the option to provide more detailed insights into their reasoning, however using it

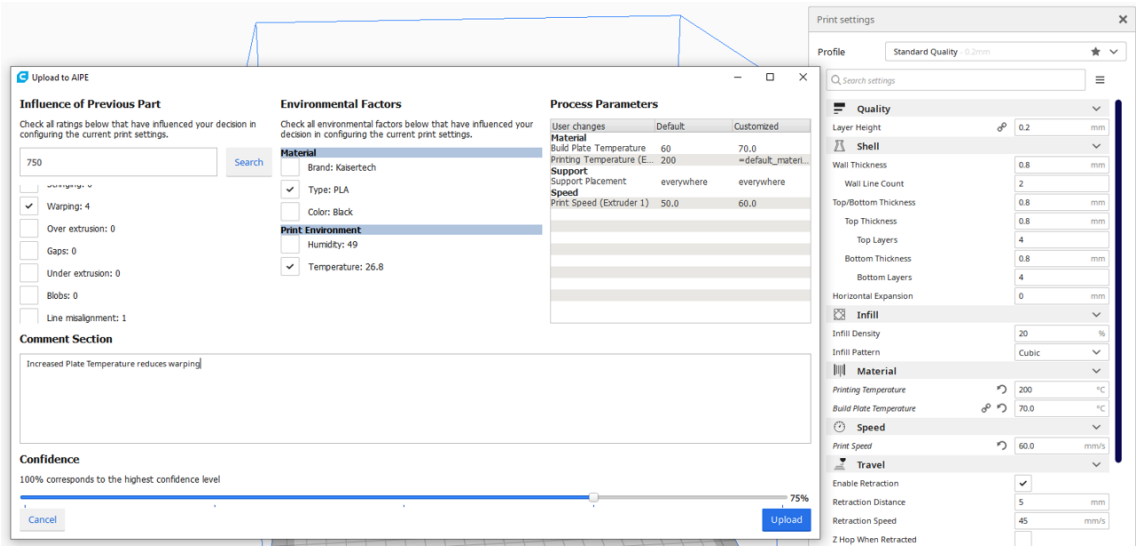


Figure 5.2.: Extended Cura HMI for the FDM case study. First, a parametrisation is defined in the *Print settings* dialog on the right. Then, influential quality characteristics and environmental factors can be highlighted (centre left).

increases their cognitive load and would probably necessitate specifically tailored incentivisation.

5.2. Sequential Influence Elicitation

In contrast to the combined case, individual influences a which constitute α_i have to be independently considered for the sequential influence elicitation. Here, a knowledge segment in form of a knowledge graph k_i^a is created for each $a \in \alpha_i$ for a process iteration i. To this end, vertices and edges are defined by $v_i^a = \iota \cup a$ and $e_i^a = \iota \times a$, respectively, for all process parameters p influenced by a, which are referred to as $\iota = \{p \mid p \in \delta_i \text{ and influenced by } a\}$. The sequential case does not exhibit the limitation regarding unclear cause-effect relations of the provided information, as is the case in the combined case if multiple influences are present, since a separate knowledge graph is generated for each influence. However, the cognitive and operational load for the operator is higher due to the increased interaction and meta-cognitive reflection required.

To showcase the flow the operator has to navigate in the sequential case, we refer to Figure 5.3 which is based on the HMI extension of the polymer extrusion case study (see Section 3.2). To capture contiguous parameter changes we need to collect parameter changes over a timespan due to the continuous nature of the use case. In particular, we collect parameter changes starting with an initial parameter change until a domain specific timeout has been reached. After the timeout is exceeded, the operator is presented with a dialog, as shown in Figure 5.5a, with influences on the left and a summary of the parameter changes on the right. In contrast to the combined case (see Figure 5.2), the user experience is increased by preselecting all parameters which decreases the amount of elements the operator has to select. If the operator only mitigated one quality characteristic, he selects the corresponding influence and shares the data by clicking *Confirm & Close* (see Figure 5.5b). In this case, the resulting knowledge segment of the sequential elicitation equals that of the combined elicitation. If, however, multiple quality defects were mitigated, he

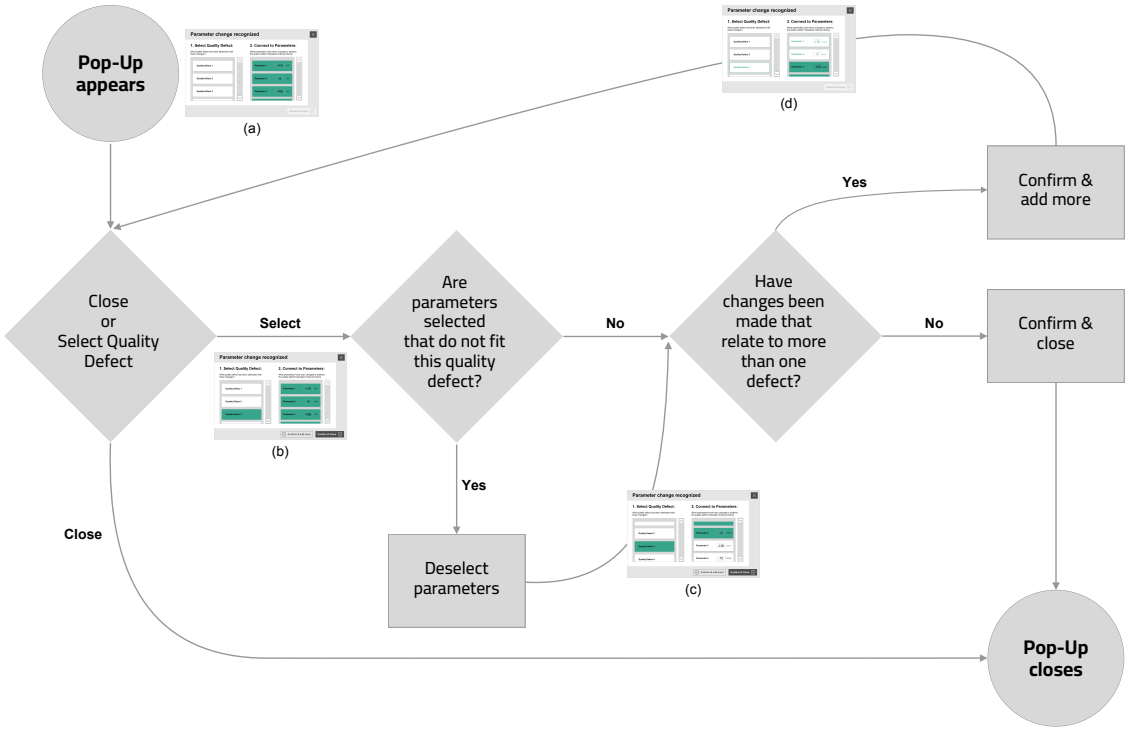
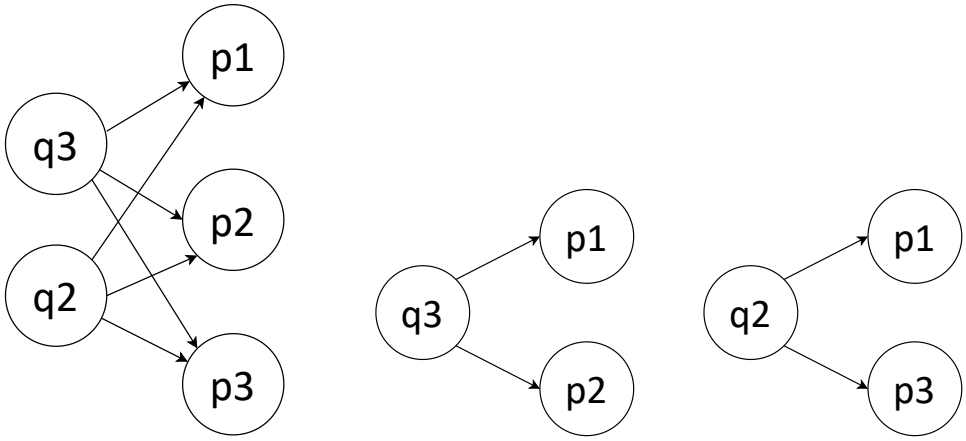


Figure 5.3.: Extension of the extrusion line's HMI embedded in a control flow to facilitate 1-to-n relations between influences and affected parameters. Close-ups of the individual screens are shown in Figure 5.5.

selects one mitigated quality characteristic followed by the deselection of those parameters that were not adjusted for this quality characteristic (see Figure 5.5c). The resulting information is shared through clicking *Confirm & add more*. The next screen, shown in Figure 5.5d, highlights the already shared parameters and quality characteristics and preselects the remaining parameters. Note, that selecting a previously selected parameter remains possible, allowing multiple influences to reference the same parameter. This loop is continued until all parameters have been shared or the dialog is closed manually. An alternative to the proposed workflow would be to rely on the operator to actively open the dialog after finishing a parametrisation. Since in our continuous case study complex parametrisations involving multiple parameter changes for one or multiple adjustments of quality defects are encountered seldomly, we chose the first approach since we believe the benefits of actively prompting the operator outweigh the drawbacks of a slightly higher cognitive load if the timeout is inadequately chosen.

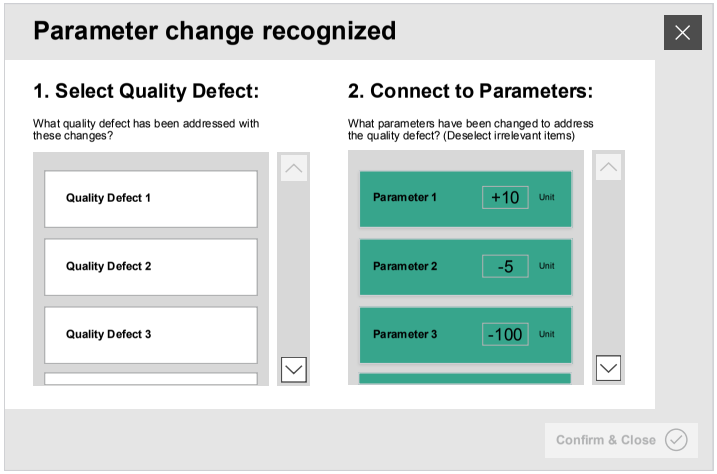
Figure 5.4 allows a graphical comparison between the knowledge segments elicited with sequential or combined knowledge elicitation. In both cases, the operator adjusted process parameters 1 and 2 to mitigate quality defect 3 and process parameters 1 and 3 to mitigate quality defect 2. In contrast to the sequential elicitation, it is no longer discernible which parameter was adjusted to mitigate which quality defect for the knowledge segment extracted via combined elicitation.



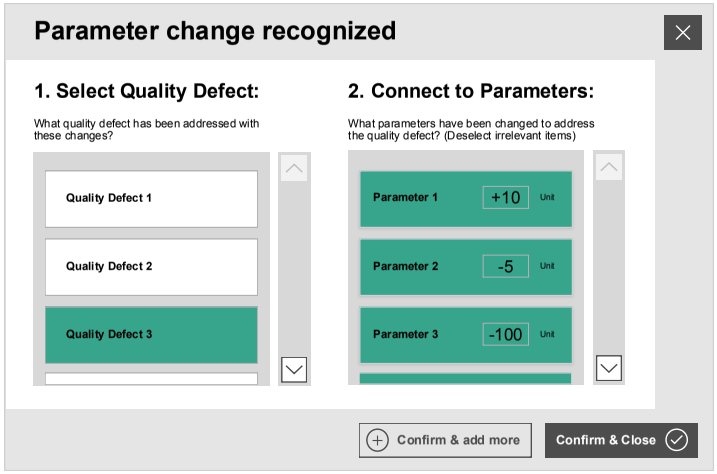
(a) Knowledge segment resulting from combined influence elicitation.

(b) Knowledge segments resulting from sequential influence elicitation.

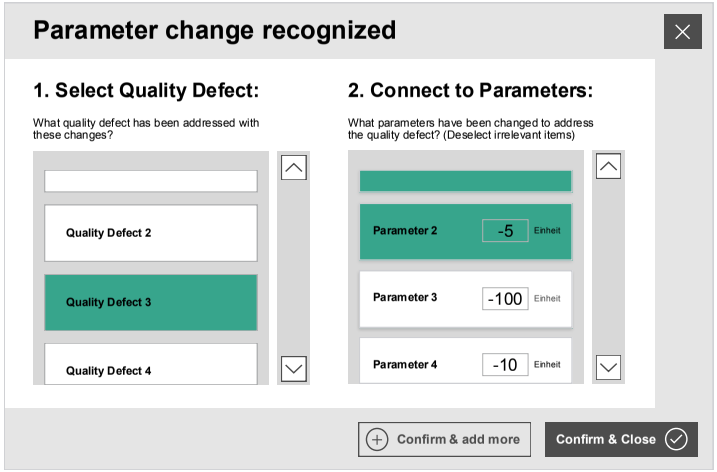
Figure 5.4.: Examples of knowledge segments elicited by applying the combined and sequential approach for the same operator reasoning.



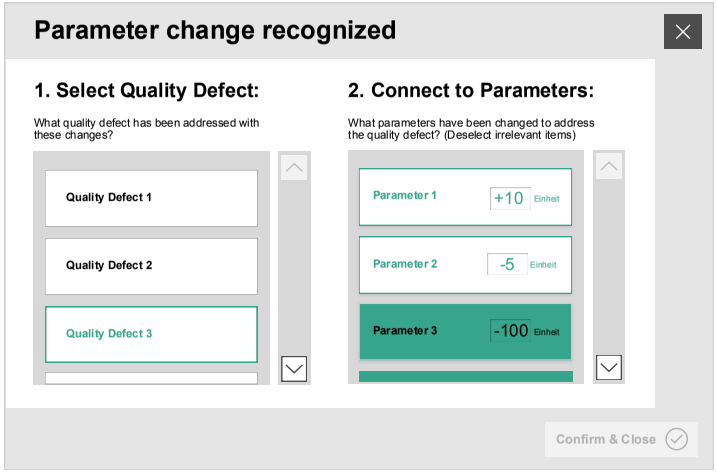
(a) Initial prompt to highlight a quality defect.



(b) Prompt to select corresponding parameters.



(c) Decision whether multiple quality defects were adressed.



(d) Second iteration of the elicitation process for another quality defect.

Figure 5.5.: Individual screens of the modal dialog used in the sequential elicitation flow as shown in Figure 5.3

6
Aggregation & Extraction

To create a knowledge graph from which a succinct rule base—containing reliable information on how machine parametrisation influences a product's quality—can be extracted, the knowledge segments elicited by the methods presented in Chapter 5 need to be aggregated. In this chapter, we introduce the formalisation which forms the basis for the description of the aggregation and extraction methods. Additionally, naive as well as weighting and filtering-based aggregation methods and the extraction of rules from the knowledge graph are presented. Naive aggregation methods treat every knowledge segment equally. This leads to problems with heterogeneous knowledge sources, i.e. operators with varying level of experience and cooperativeness, and consequently to erroneous relations in the knowledge graph, since the information the relations are based on could be erroneous or highlight non-causal influence–parameter change relationships. Reasons for erroneous information could be influences with atomicity greater than one, i.e. unclear cause-effect relations that can occur because multiple influences were highlighted for multiple parameters adapted in the case of combined elicitation (see Section 5.1), inexperienced operators, untrustworthy operators or operators that tend to have a higher trust in their experience than warranted [KD99] and consequently provide knowledge segments which they think are truthful, but in reality are not. These shortcomings are addressed by introducing a data-based weighting of the operators' information, which allows us to control for varying degrees of worker reliability and other individual biases, increasing the quality of the aggregated knowledge. Finally, we show how to automatically extract rules from the aggregated knowledge graph which are (1) quantified and (2) human-readable in natural language so that they can be directly used in a passive assistance system in case of quality defects.

6.1. Formalisation

This section builds on the formalisation presented in Chapter 2 and introduces the foundations for knowledge aggregation and rule extraction methods. This is illustrated by extending Example 2.1.1, showing how information contained therein can be used to aggregate knowledge graphs containing relations, which in turn can be used to extract rules. While influences can be quality characteristics, environmental conditions or parameters, see Definition 2.1.2, we limit them to quality characteristics in the following to increase clarity.

Definition 6.1.1 (Relation)**.** A relation $\eta_{q,p}$ is defined as the tuple (q, p) between a quality $q \in Q$ and a parameter $p \in P$.

Definition 6.1.2 (Rule with Quantified Conclusion)**.** A rule with quantified conclusion $r_{q,\hat{\rho}}$ is defined as a tuple $r_{q,\hat{\rho}} = (\eta_{q,p}, \nu) = (q, p, \nu)$ with a quality characteristic that will be improved upon $q \in Q$, i.e. the condition, and the quantified conclusion, i.e. the quantified parameter. Note,

that while $\hat{\rho}$ denotes aggregated quantified conclusions, we refer to q instead of o since it is not quantified. The aggregated quantified parameter is quantified by ν. We differentiate three different kinds of quantification:

1. $\nu_\Delta \in \mathbb{R}$ with $\nu_\Delta = \Delta(\phi_{i,p}, \phi_{i+1,p}) \rightarrow \phi_{i+1,p} - \phi_{i,p}$ describing the relative amount by which the parameter should be adjusted, where $\phi_{i,p}$ selects the value of the $\rho \in \mathrm{P}$ corresponding to p at the process iteration i.

2. $\nu_a \in \mathbb{R}$ with $\nu_a = \phi_{i+1,p}$, i.e. the absolute value that the process parameter should be set to in the next iteration.

3. $\nu_{[]} = [x, y]$, with $x, y \in \mathbb{R}$ the interval of absolute values for p out of which the operator can freely choose for the next parametrisation.

Based on the sign of the parameter change a ν_Δ quantified conclusion of the rule can be defined as increasing if the quantification is positive and decreasing if the quantification is negative. Depending on the information available—relative values are of a finer granularity than intervals, absolute values are suited for categorical parameters—a corresponding quantification is chosen.

Example 6.1.1. This example builds on Example 2.1.1, i.e. $\phi_0^a = ((p_1, 4), (p_2, 5), (p_3, 6))$ leads to $\theta_0^a = ((q_1, 1), (q_2, 0), (q_3, 2))$ which is improved to $\theta_1^a = ((q_1, 1), (q_2, 0), (q_3, 0))$ by $\phi_1^a = ((p_1, 10), (p_2, 5), (p_3, 5))$. Assuming combined influence elicitation and a ν_Δ conclusion quantification, we conclude a set containing two rules with quantified conclusions, $R_a = \{r^a_{q_3,p_1} = (q_3, p_1, 6), r^a_{q_3,p_3} = (q_3, p_3, -1)\}$, since it is not clear which specific parameter change affected the quality characteristic q_3 at process iteration 1, also expressed as $\theta_{1,3}$. Note, that the parameter-quality relations have an atomicity of two since an operator cannot highlight relationships between parameters and quality characteristics on an atomic level.

For a second task, i.e. manufacturing object b, the operator also starts with the default parametrisation, i.e. $\phi_0^b = ((p_1, 4), (p_2, 5), (p_3, 6))$. However, due to different object characteristics this leads to quality $\theta_0^b = ((q_1, 3), (q_2, 1), (q_3, 1))$. Now, the operator choses $\phi_1^b = ((p_1, 11), (p_2, 5), (p_3, 6))$, i.e. increasing $\phi^b_{1,1} = \rho_1$ by 7, which leads to quality $\theta_1^b = ((q_1, 0), (q_2, 0), (q_3, 0))$. The resulting rule set is $R_b = \{r^b_{q_1,p_1} = (q_1, p_1, 7), r^b_{q_2,p_1} = (q_2, p_1, 7), r^b_{q_3,p_1} = (q_3, p_1, 7)\}$.

Definition 6.1.3 (Rule with Quantified Condition)**.** A rule with quantified condition $r_{\hat{\mathrm{o}},\hat{\rho}}$ is defined as a tuple $r_{\hat{\mathrm{o}},\hat{\rho}} = (r_{q,\hat{\rho}}, \mu) = (q, p, \nu, \mu)$, with p, q and ν defined as in Definition 6.1.2 and a quantification of the condition, $\mu = [x, y]$ with $x, y \in \mathbb{R}$, which is defined as the interval for which the rule is valid. A rule with quantified condition always has a quantified parameter as well.

To be able to compare rules, e.g. for comparing them to a ground truth, we define three equality operators:

Definition 6.1.4 (High-level equality $=_h$)**.** Influence, i.e. quality characteristic, and parameter have to concur for both rules:

$$r_{x,y} =_h r_{m,n} \iff \begin{cases} x = m \wedge p^{r_{x,y}} = p^{r_{m,n}}, & \text{if } x, m \in Q \text{ and } y, n \in \mathrm{P} \\ q^{r_{x,y}} = q^{r_{m,n}} \wedge p^{r_{x,y}} = p^{r_{m,n}}, & \text{if } x, m \in \mathrm{O} \text{ and } y, n \in \mathrm{P} \end{cases}$$

for quantified conclusion and quantified condition rules with q^r, and p^r selecting the quality or parameter of the respective rules.

Definition 6.1.5 (Mid-level equality $=_m$)**.** Is only defined for rules with relative or absolute parameter quantifications, ν_Δ and ν_a, respectively. In addition to fulfilling $=_h$, both rules need to be of the same rule type, e.g. quantified conclusions as in Definition 6.1.2 or conditions as in Definition 6.1.3. Additionally:

$$\begin{aligned} r_{x,y} =_m r_{m,n} \iff \quad & r_{x,y} =_h r_{m,n} \\ & \wedge \operatorname{sgn}(\nu^{r_{x,y}}) = \operatorname{sgn}(\nu^{r_{m,n}}), \end{aligned}$$

for quality characteristics $x, m \in Q$ or $x, m \in \mathrm{O}$ and quantified parameters $y, n \in \mathrm{P}$ for rules with quantified conclusions and quantified conditions, respectively. While mid level equality in the case of relative parameter quantifications, ν_Δ, signifies that the parameter has to be adjusted in the same direction, in the case of absolute parameter quantification ν_a it only ascertains that both quantifications are either positive or negative. While the latter has no practical validity, it's used in definitions building on this definition.

Definition 6.1.6 (Quantification equality $=^\nu$, $=^\mu$)**.** To compare quantified rules, different equality measures are applied depending on the type of their quantification $\nu \in \{\nu_\Delta, \nu_a, \nu_{[]}\}$ or μ, which is treated the same as $\nu_{[]}$, since they are both intervals. Additionally to fulfilling $=_m$, the following definitions hold depending on the respective quantifications:

- for quantification equality of two rules quantified by intervals, i.e. either $\nu_{[]}$ or μ, the intersection of the intervals over the base interval has to be greater than a threshold. For rules with quantified conditions μ and threshold t_μ:

  $$\begin{aligned} r_{x,y} =^\mu r_{m,n} \iff \quad & r_{x,y} =_h r_{m,n} \\ & \wedge \frac{|\mu^{r_{x,y}}| \cap |\mu^{r_{m,n}}|}{|\mu^{r_{m,n}}|} \geq t_\mu. \end{aligned}$$

  It is analogously defined for $=^{\nu_{[]}}$.

- for rules quantified by ν_a or ν_Δ, the suggested parameter change needs to be similarly quantified, i.e. within a threshold t_{ν_a} or t_{ν_Δ}, respectively. For rules with conclusions quantified by ν_a:

  $$\begin{aligned} r_{x,y} =^{\nu_\Delta} r_{m,n} \iff \quad & r_{x,y} =_m r_{m,n} \\ & \wedge d(\nu_\Delta^{r_{m,n}}, \nu_\Delta^{r_{x,y}}) \leq t_{\nu_\Delta}, \end{aligned}$$

  where $d(a, b)$ is the relative distance between two values $a, b \in \mathbb{R}$

  $$d(a, b) = 1 - \frac{\min(|a|, |b|)}{\max(|a|, |b|)}.$$

  $=^{\nu_a}$ is analogously defined. Note, that due to the check for mid-level equality a and b are either both positive or negative in both cases.

- to compare rules with different conclusion quantification types, e.g. $r_{x,y}$ quantified by $\nu_{[]}$

and $r_{m,n}$ quantified by ν_a, it is defined by

$$\begin{aligned} r_{x,y} =^{\nu_a,\nu_{[]}} r_{m,n} \iff & \quad r_{x,y} =_h r_{m,n} \\ & \wedge \big(\quad \nu_a^{r_{x,y}} \in \nu_{[]}^{r_{m,n}} \\ & \qquad \vee \ \min(d(\nu_a^{r_{x,y}}, \min(\nu_{[]}^{r_{m,n}})), \\ & \qquad\qquad d(\nu_a^{r_{x,y}}, \max(\nu_{[]}^{r_{m,n}}))) \leq 0.5 \cdot t_{\nu_a} \big) \end{aligned}$$

If ν_a is not part of $\nu_{[]}$, we reuse the constant t_{ν_a} to check whether the lowest or highest value of $\nu_{[]}$ is close to ν_a. However, in this case we multiply t_{ν_a} by 0.5 to account for the fact that the margin could extend to both sides of the interval.

Definition 6.1.7 (Low-level equality $=_l$)**.** Low-level equality is defined depending on the quantification and rule type. For rules with quantified conclusions $=_l^\nu$ is the low-level equality for the respective quantification type using $=^\nu$ as defined in Definition 6.1.6. For rules with quantified conditions and conclusions, a composite of $=^\nu$ and $=^\mu$, the analogous low-level equality of condition quantifications, describes the low-level equality, $=_l^{\nu,\mu}$:

$$\begin{aligned} r_{x,y} =_l^{\nu,\mu} r_{m,n} \iff & \quad r_{x,y} =^\mu r_{m,n} \\ & \wedge r_{x,y} =^\nu r_{m,n}. \end{aligned}$$

Applied to the rules of Example 6.1.1 this leads to $r_{3,1}^a =_l^\nu r_{3,1}^b$ for $t_{\nu_\Delta} = 0.3$.

The main problem this chapter addresses is finding an aggregation function that yields a minimal set of correct rules, $R_=$, regarding a given level of abstraction $= \in \{=_h, =_m, =_l\}$, of which parameters to apply to mitigate all quality defects that can occur in a given (re-)parametrisation process that conforms to the process described in Chapter 2. For rules with quantified conclusions quantified by absolute or relative values

$$R_{=^\nu} = \{r \,|\, r \in Q \times P \times \mathbb{R}\}_{=\nu},$$

while for rules with quantified conditions and conclusions

$$R_{=^\mu} = \{r \,|\, r \in Q \times P \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}\}_{=\mu}.$$

In the case of conclusions quantified by a range, another $\mathbb{R}$ is added.

Example 6.1.2. Continuing Example 6.1.1 with rule sets R_a and R_b, a possible candidate for one such rule set could be $R_{=_l^{\nu_\Delta}} = R^a \cap R^b$, where $\cap$ is defined for equality under $=_l^{\nu_\Delta}$ and each ν_Δ in $R_{=_l^{\nu_\Delta}}$ is calculated as the mean of the corresponding quantifications from R^a and R^b. Consequently, $R_{=_l^{\nu_\Delta}} = \{r_{q_1,p_1} = (q_1, p_1, 7), r_{q_2,p_1} = (q_2, p_1, 7), r_{q_3,p_1} = (q_3, p_1, 6.5), r_{q_3,p_3} = (q_3, p_3, -1)\}$.

To arrive at the optimal rule set, we utilise a knowledge graph as a data structure which is aggregated by the methods presented in Section 6.2. The quality of the knowledge graph is measured by comparing extracted rules to an expert validated ground truth.

6.2. Aggregation

For influences elicited in a combined manner, the knowledge graph containing all knowledge segments can be naively aggregated by $k = \bigcup_{i \in \mathcal{I}} k_i$, with k_i a knowledge segment for a process iteration i (see Section 5.1 for the definition). Since for sequentially elicited influences, each process iteration can contain multiple knowledge segments, these need to be aggregated as well. Consequently, $k = \bigcup_{i \in \mathcal{I}} \bigcup_{a \in \alpha_i} k_i^a$ for sequentially elicited knowledge. To keep the definitions of the following aggregation methods succinct, we will only define them for knowledge elicited in a combined manner. The definitions for sequentially elicited knowledge are analogously to that seen for the naive aggregation.

In practice, and illustrated by Examples 2.1.1, 6.1.1 and 6.1.2, operators can adjust more than one process parameter at once, for example if they want to limit the amount of iterations needed during the parametrisation process. This leads to influences with atomicity greater than one, i.e. multiple influences that are in relation with multiple process parameters. Also, process parameters and influences are multi-dimensional with over 500 process parameters and 13 quality characteristics in the case of the FDM case study, see Section 3.1. This high dimensional space can be reduced to 58 and 13, respectively, by relying only on parameters that were actively adjusted during the parametrisation process. Factors such as high dimensionality, few examples, the unsupervised nature of the problem, as well as operator and environmental biases, illustrate that identifying the process parameter(s) influenced by a singular influence and vice versa is a non-trivial task that is best addressed with heuristics. Consequently, we design methods to ascertain the reliability of information provided by the operators with the goal of filtering out erroneous information. The underlying idea is to first weight the relations and then filter them with an adaptive threshold.

We investigate methods that are applied on quantified parameters or qualities highlighted by influences α before or after graph aggregation. For those that are applied before graph aggregation, we focus on operator confidence, frequency of the relation, as well as quality improvements. The *operators' confidence* ζ is directly contained in the knowledge segment of an iteration and does not need computation. *Relation frequency* is established by summing the occurrences of a relation in the knowledge segments over all process iterations and dividing this by the number of unique relations. This is then assigned to the weight of the respective relations:

$$w_{F_{q,p}} = \frac{\sum_{i \in \mathcal{I}} \eta_{q,p} \in k_i}{|\{\eta_{q,p} \in k \mid p \in P, q \in Q\}|}$$

Quality improvement is given by calculating $\Delta(q_i, q_{i+1})$ or short Δq. These three techniques can be combined at will. Weights for relation frequency and quality improvement are multiplied if they are combined. Operator confidence, if combined, is factored in by multiplying it with the weight obtained by relation frequency F or quality improvement Δq, respectively.

Clustering-based weighting methods, denoted by C, are utilised to detect outliers and are applied to subsets of the iterations' attributes. Based on different combinations of subsets s_h with $h \in \{p, q, \alpha, \Delta p, \Delta q, \Delta\alpha\}$, OPTICS [Ank+99] is used for density-based clustering. In this context we use α to denote quantified parameters or qualities highlighted by α. We utilise the Minkowski distance and set the minimum amount of neighbours constituting a cluster to two. The resulting clusters are then used to weight the relations of the knowledge graph by counting the number of iterations contributing to the respective relation that were successfully assigned to a cluster, thereby discounting the outlying iterations that were not successfully assigned a cluster. This is

then normalised by the amount of iterations and used as the relation weight:

$$w_{C_{s_h},\eta} = \frac{|\{\eta(i) \mid i \in \mathcal{I}, \text{ and } i \in \text{OPTICS}(s_h)\}|}{|\{\eta(i) \mid i \in \mathcal{I}\}|},$$

where OPTICS generates sets of clustered iterations and unclustered iterations are not included in any sets and $\eta(i)$ determines if a specific relation $\eta_{q,p}$ can be extracted from the given iteration i.

Another clustering-based approach, denoted by CC is based on the assumption that similar parametrisations should lead to similar qualities. Therefore, iterations that are successfully clustered in parameter space should also be clustered in quality space. We calculate weights similarly to the clustering methodology described above for absolute and relative (defined analogously) qualities and parameters:

$$w_{CC_{p,q,\eta}} = \frac{|\{\eta(i) \mid i \in \mathcal{I} \text{ and } i \in \text{OPTICS}(p) \cap \text{OPTICS}(q)\}|}{|\{\eta(i) \mid i \in \mathcal{I}\}|}$$

In contrast to the previous methods, the *influence validity* method, denoted by IV, directly tries to quantify the correctness of the operator's notion that the changed process parameters are improving the highlighted quality characteristics. To achieve this, the sum of the highlighted quality characteristics is divided by the overall quality for each iteration contributing to the relation. The resulting fractions are then summed and normed by the amount of iterations that contributed to this relation and used as its weight:

$$w_{IV_\eta} = \frac{\sum_{\{\eta(i)|i\in\mathcal{I}\}} \frac{\sum_y \theta_{i,y}}{\sum_y \theta_{i+1,y}}}{|\{\eta(i) \mid i \in \mathcal{I}\}|}$$

The adaptive threshold is computed by taking the mean $\bar{x}$, median $\tilde{x}$ or elements smaller than the first quartile Q_1 over all relations. Based on this, relations whose weight is smaller than the threshold are removed from the knowledge graph.

6.3. Rule Extraction

Based on the central artefact of Section 6.2, a knowledge graph that contains all relevant data, this section details how actionable rules with quantified conclusions and conditions can be extracted from the aggregated knowledge. The resulting rule base has two purposes:

1. it can be utilised as part of a decision support system which allows the operator to solve occurring quality defects. For example, if confronted with a specific problem regarding a defective quality characteristic of the current process iteration $q_i \in Q$, the operator is able to choose fitting rules pertaining to the same quality characteristic from the rule base. By applying them to the next parametrisation, he is able to remedy this specific quality defect.

2. it can serve as the initial rule base which is further refined by operators through a downstream process.

6.3.1. Quantified Conclusions

To extract rules with quantified conclusions, we rely on the aggregated knowledge graph to identify high-level relations between influences, i.e. quality characteristics in our case, and adapted parameters. Since the quantified rules rely on the aggregated knowledge, they represent aggregations of iterations and are therefore independent of decisions at a specific point in time. In the following, we rely on these aggregated rules. Quantification of aggregations can be quantified by analysing p and Δp over all iterations contributing to the relation that exhibited the respective high-level relation. Based on this, we are able to discern both the conclusion (i.e. increase, decrease, set) as well as the parameter quantification of the conclusion of the rules. A parameter describing the conclusion can be quantified in three ways: relative and absolute amount as well as interval of absolutes as introduced in Definition 6.1.2. Intervals $\nu_{[]}$ is determined by $\mathbb{E}(\Delta\phi_{i,p}) \pm \sigma(\Delta\phi_{i,p})$, of all process iterations $i \in \mathcal{I}$, where $\mathbb{E}$ is the expected value. Relative parameter adjustments, ν_{Δ} is determined by $\mathbb{E}(\Delta\phi_{i,p})$, of all process iterations $i \in \mathcal{I}$. Absolute values in the case of categorical parameters are quantified by value with the highest occurrence. While learning specific parameter-quality functions would be preferable, initial experiments were unsuccessful due to the low amount of samples. However, since functions in manufacturing are usually monotonic [Kur+21], using the expected value seems like a reasonable approximation, that could be refined if sufficient data is available.

6.3.2. Quantified Conditions

Rules with quantified conclusion have the drawback that one quantification covers the whole range of the condition, i.e. the defective quality characteristic. In practice, however, a more fine granular condition would allow, e.g. a stronger adjustment of the parameter if the quality characteristic is more defective. To achieve this, this section describes an approach to extract rules conditions quantified by ranges.

The approach consists of four stages:

1. *Clustering similar quality characteristics* by determining the number of clusters or providing a domain specific number, e.g. ratings of quality characteristics between 0 and 5 which constitute six clusters, as is the case in our FDM case study (see Section 3.1). Examples for estimators are L2 [FD81], which is relatively robust against outliers or Doane's formula [Doa76], which is frequently used for non-normal distributed data. In the following, we will use Doane's formula since it achieved the most consistent results which we validated with our domain knowledge on the FDM dataset. Based on the number of clusters, the size of the clusters can be directly calculated if they are assumed to be distributed evenly.

2. *Extracting a rule for each cluster*, using its lower and upper limits as condition quantification. The conclusion quantification is computed for each rule as outlined in Section 6.3.1 based on the knowledge segments that have quality characteristics lying in the respective cluster. Contiguous clusters could lead to rules with similar conclusion quantification which increases the size of the rule base without adding additional information.

3. Consequently, the rule base can be compacted. To *identify rule candidates for merging*, Stage 1 is repeated on the resulting conclusion quantifications. All rules which have adjacent condition quantifications and a conclusion quantification in the same cluster are considered to be mergeable without loss of information.

4. Finally, the compacted rule base is created by *merging* the quantified conditions by $\mu = \bigcup_{r \in R_c} \mu^r$ with R_c the set of candidate rules for each cluster. As in Step 2, the conclusion quantification is computed for each rule as outlined in Section 6.3.1 based on the knowledge segments that contributed to the candidate rules.

7 Evaluation

To evaluate the effect of influences elicited and aggregated with the methodology presented in Chapters 5 and 6, we conduct a series of experiments to evaluate its performance on the FDM (see Section 3.1) and synthetic case study (see Section 3.3). While both can be evaluated against a ground truth, the latter allows the analysis of the effect of different complexities in underlying data. We evaluate the combination of elicitation, aggregation and extraction and address the following research questions via the quantitative evaluation conducted in Section 7.1:

- RQ1: Is there a benefit of weighting and filtering-based aggregation methods compared to naive aggregation?
- RQ2: Does operator feedback provide a benefit compared to purely data-based knowledge elicitation?
- RQ3: How does complexity affect the performance of knowledge extraction?

RQ1 is mainly addressed in Section 7.1.2 but also referenced in Sections 7.1.3 and 7.1.4. While RQ2 is addressed in Section 7.1.3, Section 7.1.4 deals with RQ3. Section 7.2 provides a small-scale evaluation of the quality of the resulting rules in the FDM case study.

7.1. Evaluation Against Ground Truth

This section evaluates the aggregation and extraction methods against a ground truth for the FDM and synthetic case study. It introduces the methodology used to arrive at the ground truth for FDM data and studies the effect of influences, and of the complexity in underlying data on the performance and suitability of the aggregation methods.

7.1.1. Experimental Setup

This section provides an overview of the experimental setup, detailing ground truth and evaluation set creation as well as the selection of the shown aggregation methods.

7.1.1.1. Ground Truth Creation

While the extraction of a ground truth for the synthetic case study—the knowledge generation is described in more detail in Section 3.3.1.3—is trivial, i.e. the generated knowledge is extracted as ground truth before noise is added, it is not as straightforward for the FDM use case since well defined ground truths are not available for the FDM domain at the required granularity. Also,

unstructured knowledge bases e.g. Simplify3D's troubleshooting guide[1], from which they could be created are unsuited since the methodology underlying their creation is unknown. Therefore, we describe a methodology to obtain a ground truth against which the proposed approach can be compared against on the FDM case study.

To arrive at a list of rule candidates that can be judged by a group of experts, the naive aggregation (see Section 6.2) followed by the extraction of quantified conclusion rules (see Section 6.3.1 was applied to the FDM dataset when it comprised 411 produced products for which influences were elicited. Three FDM experts were independently tasked with classifying whether the candidate rule are correct on a high level of abstraction as defined by the high level equality $=_h$ (see Section 6.1), i.e. whether there exists a relation between parameter and condition. If that was the case they proceed to evaluate the rule at a medium and, lastly, a low level of equality. As described in Section 6.1, medium equality, $=_m$ is achieved if the rules' action, e.g. increase or decrease, is suitable, whereas low equality, $=_l^\nu$, is achieved if the quantification—the amount the parameter should be adjusted—is within 30% of the experts' opinion, $t_{\nu_\Delta} = t_{\nu_a} := 0.3$. The concrete threshold was established through conversations with operators in the FDM scenario about applicable intervals and investigating the data to align operator/dataset generator behaviour. If a rule is unequal at low or medium level the experts adjust it according to their knowledge. However, the resulting ground truth can only be used for quantified conclusion rules, since these form the basis of the candidate rule set. In principle, the same method can be conducted on a candidate rule set created by quantified condition rule extraction. However, this was not done due to the large amount of quantified condition rules created by the naive aggregation method and the corresponding required effort to rate them accordingly.

The resulting individual rule sets are merged by averaging—mean for numerical, most frequent value for categorical—quantifications and assigning an action accordingly if at least two experts validated a rule for a given relation. The now aggregated rule set is suitable for use as ground truth since multiple experts evaluated the given rules in a detailed manner, providing corrections where necessary. Especially, due to the experts' corrections the ground truth is not a subset of the baseline which would aggravate the comparison between naive and weighting and filtering-based aggregation methods. Still, it is likely to focus on a certain subset of the experts' knowledge. However, since this subset is informed by parameter choices encountered in practice, it can be argued that it focusses on the most practically relevant area of expertise. The experts provided a mean of 41 rules each with a standard deviation of 2.16 rules. All experts agreed on 20 rules, 21 different rules were confirmed by two experts, whereas another 21 rules were only approved by a single expert. This highlights the individuality of experts that gained their experience on different printers, materials and objects and highlights the importance of our approach to accept only rules that are agreed upon by at least two experts. As the number of involved experts and the fractural nature of their knowledge is similar to industrial settings, we assume that the process of ground truth creation is applicable to other industrial domains. In practice, the ground truth creation process could also be used as an editorial step to increase the quality of knowledge extracted with weighting and filtering-based aggregation methods before it is used in an assistance system.

[1] https://www.simplify3d.com/support/print-quality-troubleshooting/

7.1.1.2. Evaluation Set Creation

To evaluate overlap between rules contained in the aggregated knowledge graph and the ground truth, we utilise the three equality operators defined in Section 6.1, which are suited to evaluate rules on differing degrees of abstraction. To be able to evaluate metrics on rule sets, e.g. ground truth and prediction, of differing size and ordering an evaluation set is needed. An evaluation set of two rule sets is created by comparing each given rule with the given equality operator. If the rules are equal according to the equality operator, a substitution is carried out, if it is unequal it is treated as a unique rule. Substituted rules, as well as unique rules of both rule sets are concatenated to form the evaluation set.

Example 7.1.1. Referring back to Example 6.1.1, i.e. rule sets $R_a = \{r^a_{q_3,p_1} = (q_3, p_1, 6), r^a_{q_3,p_3} = (q_3, p_3, -1)\}$ and $R_b = \{r^b_{q_1,p_1} = (q_1, p_1, 7), r^b_{q_2,p_1} = (q_2, p_1, 7), r^b_{q_3,p_1} = (q_3, p_1, 7)\}$, $R^{a,b} = \{r^{a,b}_{q_3,p_1}, r^a_{q_3,p_3}, r^b_{q_2,p_1}, r^b_{q_1,p_1}\}_{=^{\nu_\Delta}_l}$ would be the evaluation set for R^a and R^b for $=^{\nu_\Delta}_l$ with $t_{\nu_\Delta} = 0.3$ since $r^a_{q_3,p_1} =^{\nu_\Delta}_l r^b_{q_3,p_1}$.

Both rule sets are compared against the evaluation set resulting in binary sets that can be directly compared. We evaluate the similarity between a prediction and ground truth with precision, recall and F1 score (values in [0,1], higher is better). Also, the number of rules resulting from using a given aggregation method is reported.

7.1.1.3. Method Selection

We apply the aggregation methods described in Section 6.2 and evaluate them for the three levels of abstraction. Here, the low abstraction level is the one with the greatest relevance for industrial use cases, since it provides data-based quantifications, which are often not available using traditional knowledge extraction methods. Consequently, we focus the evaluation on it. To select a subset of methods that are shown in the experiments in Sections 7.1.3, 7.1.4.1 and 7.1.4.2, the best two methods for each experiment, dataset, i.e. FDM or synthetic, abstraction level, i.e. high, medium and low, and rule type, i.e. quantified conclusions and quantified conditions, were selected. For the synthetic dataset ten runs were executed with different random seeds to account for variations in the randomly chosen functions. In the resulting graphics, the mean is reported along with standard deviation.

7.1.2. Performance on Default Datasets

This experiment compares the selected aggregation methods against the naive aggregation, highlighting improvements to be gained on the default dataset for the FDM and synthetic dataset as described in Sections 3.1 and 3.3, respectively. In Section 7.1.4, the synthetic dataset generation will be used to synthesise datasets with characteristics differing from the default. Table 7.1 shows the best performing aggregation methods for each abstraction level (see Section 6.1) for the FDM and synthetic dataset for combined influence elicitation. As explained in Section 7.1.1.1, a ground truth for the FDM dataset is only available for quantified conclusion rules. For the synthetic dataset, results for quantified conclusion PDPK_ρ and quantified condition PDPK_o rules are reported separately. For datasets FDM and PDPK_o, we can observe that precision, recall and F1 increase for all methods with increasing level of abstraction. This is intuitively explainable by considering that it is much harder to forecast the exact parameter quantifications compared

to conclusion or even the pure existence of relations. The number of rules stays constant over equality levels for all datasets and rule types since the level of abstraction used during evaluation is independent of the aggregation method. However, for rules with quantified conditions, the amount of rules is higher for all aggregation methods than for rules with quantified parameters due to the division of one unquantified into multiple quantified conditions.

For the FDM dataset, the best performing methods vary between the different abstraction levels. Here, the aggregation method $\zeta\, F\, \Delta q\, \#\, \bar{x}$ performs best on the high abstraction level with an absolute increase over the naive baseline of 0.21 which corresponds to 48.74% for F1. However, this aggregation method is not able to compete on lower abstraction levels, where $F\, \#\, Q_1$ achieves a relative increase in F1 of 46.49% (0.15 absolute) over the naive baseline on the medium abstraction level and $C\, p\, q\, \#\, Q_1$, which achieves an increase of 24.54% (0.05 absolute) on the low abstraction level. Both $F\, \#\, Q_1$ and $\zeta\, F\, \Delta q\, \#\, \bar{x}$ are not able to compete on the low abstraction level since they reduce the rule set too aggressively by 75.51% and 74.15% to 36 and 38 rules, respectively, leading us to believe that due to the aggressive filtering data points necessary for correctly calculating the quantifications are discarded. In contrast to this, $C\, p\, q\, \#\, Q_1$ reduces the rule set only by 30.61% to 102 rules. In general, the naive baseline has a comparatively high recall with 0.97 for high, 0.73 for medium and 0.46 for low abstraction levels. However, since all knowledge segments are included regardless of their validity the precision is relatively low with 0.27, 0.20 and 0.12, respectively. By removing a part of the erroneous knowledge segments, the weighting and filtering-based aggregation methods increase precision—usually, however, at a small cost in regards to recall—which leads to an overall improvement in F1.

For the synthetic dataset generated via PDPK, $F\, \#\, Q_1$ achieves consistently achieves the best results over abstraction methods and rule types, i.e. quantified conditions, PDPK_ρ, and quantified conclusions, PDPK_o. For PDPK_ρ it achieves an increase over the naive baseline in F1 of 21.75% (0.16 absolute) by increasing precision to 1.0 and a decrease in of the rule set by 50.81% which translates to 40.6 rules. Note, that these results do not change for different abstraction levels because of the linear nature of the underlying functions in the default synthetic dataset. For PDPK_o, it achieves an increase for high, medium and low abstraction levels of 12.92% (0.09 absolute), 7.20% (0.06 absolute) and 12.83% (0.05 absolute) while decreasing the rule set by 42.17% (61.4 rules). While the performance of PDPK_ρ and PDPK_o for high and medium abstraction levels is relatively close—within 3.91% and 8.78%, respectively—we see a significant performance drop of for the low abstraction level with 44.70%. Due to the linear nature of the default synthetic dataset and the corresponding static optimal conclusion quantification, quantified conditions can not provide a benefit in this case. To investigate this more closely, they are compared on more complex functions in Section 7.1.4.3.

Overall, we conclude that the weighting and filtering-based aggregation methods provide a significant benefit since they consistently outperform the naive aggregation method while also reducing the rule set. The reduced rule set facilitates their use in a passive assistance system, since the rule application is, intuitively, easier if there are fewer rules to choose from. However, since there is no single best aggregation method discernible for all datasets and abstraction levels, we propose that they are evaluated on the respective scenario via a suitable scenario specific ground truth that can be extracted as described in Section 7.1.1.1. Due to the low overall results achieved for the low abstraction level, the resulting rule base should be evaluated before it is directly applied in an assistance system via an explicit validation step including the domain experts.

In the following, we modify the default datasets to evaluate different characteristics and answer

RQ2 and RQ3. We focus the evaluation on the lowest abstraction level, since quantified rules are harder to extract and of greater interest in industry.

Table 7.1.: Results for selected aggregation methods compared to the ground truth (see Section 7.1.1.1) for FDM and synthetic datasets. For synthetic, rules with quantified parameters, PDPK_ρ, and conclusions, PDPK_o, are listed separately. For the synthetic datasets, only the mean is reported for space reasons, standard deviation can be seen in Figures 7.1b and 7.1c. Best performing methods (according to F1 score) per level are highlighted in bold. For the pattern underlying the composite method names refer to Chapter 6.

		FDM				PDPK_ρ				PDPK_o			
Level	Method	Precision	Recall	F1	# rules	Precision	Recall	F1	# rules	Precision	Recall	F1	# rules
high	$C\,p\,q\,\#\,Q_1$	0.36	0.90	0.52	102	0.67	0.84	0.74	60.2	0.65	0.84	0.74	126.6
	$F\,\#\,Q_1$	0.67	0.59	0.62	36	1.00	0.82	**0.90**	39.3	1.00	0.76	**0.86**	84.2
	$\zeta\,F\,\Delta q\,\#\,\bar{x}$	0.66	0.61	**0.63**	38	0.72	0.24	0.21	19.8	0.74	0.23	0.21	44.6
	naive	0.27	0.98	0.43	147	0.59	0.98	0.74	79.9	0.63	0.97	0.76	145.6
mid	$C\,p\,q\,\#\,Q_1$	0.27	0.68	0.39	102	0.67	0.84	0.74	60.2	0.65	0.84	0.74	126.6
	$F\,\#\,Q_1$	0.50	0.44	**0.47**	36	1.00	0.82	**0.90**	39.3	0.91	0.74	**0.82**	84.2
	$\zeta\,F\,\Delta q\,\#\,\bar{x}$	0.37	0.34	0.35	38	0.72	0.24	0.21	19.8	0.63	0.22	0.18	44.6
	naive	0.20	0.73	0.32	147	0.59	0.98	0.74	79.9	0.63	0.97	0.76	145.6
low	$C\,p\,q\,\#\,Q_1$	0.18	0.44	**0.25**	102	0.67	0.84	0.74	60.2	0.40	0.34	0.37	126.6
	$F\,\#\,Q_1$	0.19	0.17	0.18	36	1.00	0.82	**0.90**	39.3	0.60	0.33	**0.43**	84.2
	$\zeta\,F\,\Delta q\,\#\,\bar{x}$	0.13	0.12	0.13	38	0.72	0.24	0.21	19.8	0.46	0.11	0.12	44.6
	naive	0.13	0.46	0.20	147	0.59	0.98	0.74	79.9	0.39	0.37	0.38	145.6

7.1.3. Performance on Elicited vs. Purely Data-Based Knowledge Segments

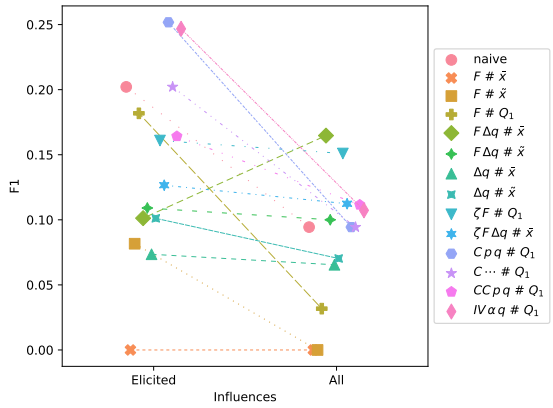
Building on Section 7.1.2, this experiment contrasts the performance on default datasets, i.e. where knowledge segments are elicited according to the methodology presented in Section 5.1, to knowledge segments extracted via a purely data-based approach, i.e. knowledge segments constructed by the cross product of all quality defects and adapted parameters. We assume that the dimensionality reduction achieved by knowledge segments elicited via influences provided by operator feedback results in a rule base that outperforms rules extracted based on all process iterations. Therefore, we expect an increased F1 score for a lower amount of rules for elicited influences compared to that of all influences present in the data. Note, that the performance on the default datasets has already been analysed in Section 7.1.2.

Figure 7.1 shows the results in F1 of the selected aggregation methods for FDM and synthetic datasets for the low abstraction level. In the latter case, results for rules with quantified parameters and conditions, PDPK_ρ and PDPK_o are reported separately. For FDM (see Figure 7.1a) the selected aggregation methods perform an average of 34.22% (0.05 absolute) worse on all influences compared to those elicited. For the best performing method for elicited knowledge, $C\,p\,q\,\#\,Q_1$, this decrease is especially stark with 62.50%. The exception to this trend is $F\,\Delta q\,\#\,\bar{x}$ which improves by 62.65% to become the best performing method on the purely data-based setting. Interesting in that regard, is that $F\,\Delta q\,\#\,\bar{x}$ relies on the improvement of quality observed between process iterations. The fact that its importance grows significantly for purely data-based knowledge segments could be an indication towards the quality of the operator feedback. The amount of rules extracted increases by an average of 69.28%. However, it depends heavily on the extraction method, since frequency-based methods, e.g. $F\,\#\,Q_1$, lead to a small decrease in numbers which still translates to 25.00-38.89% due to the extremely small initial rule bases. Especially noteworthy is that the improvement achieved by weighting and filtering-based aggregation methods over the naive baseline increases by 49.94 percentage points from 24.55% to 74.49%, which is an indication of their capability to effectively filter erroneous data points.

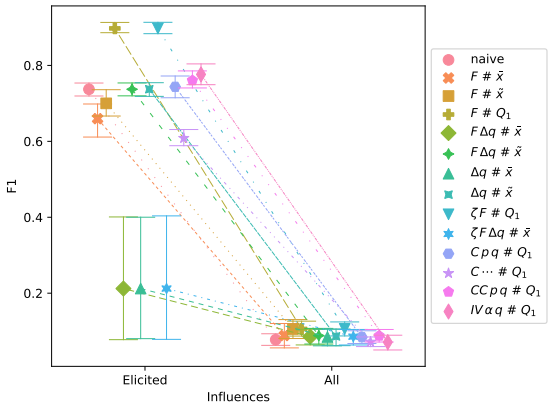
For the synthetic dataset with quantified parameters PDPK_ρ, the results are shown in Figure 7.1b. We can see the same basic trends observed for the FDM dataset, i.e. decreasing F1 score and increasing amount of rules, between elicited influences and those extracted in a purely data-based manner. However, these are much more pronounced with 82.09% (0.54 absolute) on F1 and 841.55% (311.63 rules) on the amount of rules. We assume, that these stark differences are due to a non-uniform distribution in occurring quality defects in the FDM scenario. The best performing method, $F\,\#\,Q_1$, remains the same with an improvement over the naive aggregation method of 15.02 percentage points from 21.75% to 36.77%. Interestingly, $F\,\Delta q\,\#\,\bar{x}$, $\Delta q\,\#\,\bar{x}$ and $\Delta q\,\#\,\tilde{x}$ underperform consistently when contrasted with their performance on the FDM dataset. Also, $F\,\Delta q\,\#\,\tilde{x}$ performs relatively well, highlighting the importance of different thresholds.

Rules with quantified conditions on the synthetic dataset, PDPK_o, shown in Figure 7.1b, follow the trends outlined for the those with quantified parameters in great detail, albeit in a different value range since the achievable scores in F1 are lower and amount of rules extracted higher. While the average F1 score decreases by 68.00%, the amount of rules extracted increases 8.47 fold.

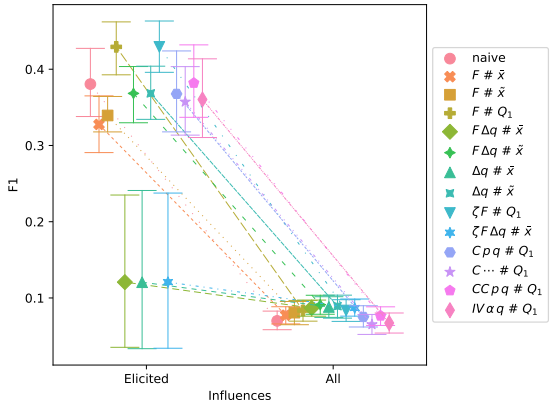
Based on the results observed for the different datasets and rule types, we conclude that our hypotheses were accurate and that operator feedback does provide a significant benefit when compared to purely data-based knowledge elicitation.



(a) Rules with quantified conclusions on the FDM dataset.



(b) Rules with quantified conclusions on the synthetic dataset PDPK_{ρ}.



(c) Rules with quantified conditions on the synthetic dataset PDPK_{o}.

Figure 7.1.: Results of Section 7.1.3 investigating elicited influences vs. those gained by a purely data-based approach.

7.1.4. Degree of Complexity in Underlying Data

To be able to answer RQ3, how complexity affects knowledge extraction, we analyse two uncertainties in underlying data that can be simulated on the synthetic dataset. While Section 7.1.4.1 details the effect of noise in the expert knowledge and consequently the knowledge segments, Section 7.1.4.2 deals with different degrees of atomicity, i.e. multiple quality characteristics present in a knowledge segments which increases the difficulty to aggregate it adequately. Another source of complexity is introduced by the complexity of functions underlying the quality-parameter relationships and analysed in Section 7.1.4.3. All experiments are conducted on synthetic datasets that allow us to change the underlying characteristics. Our hypotheses is that with growing complexity of the dataset the overall performance in F1 decreases and the amount of rules increases which leads to an increase in benefit of weighting and filtering-based aggregation methods.

7.1.4.1. Degree of Noise in Expert Knowledge

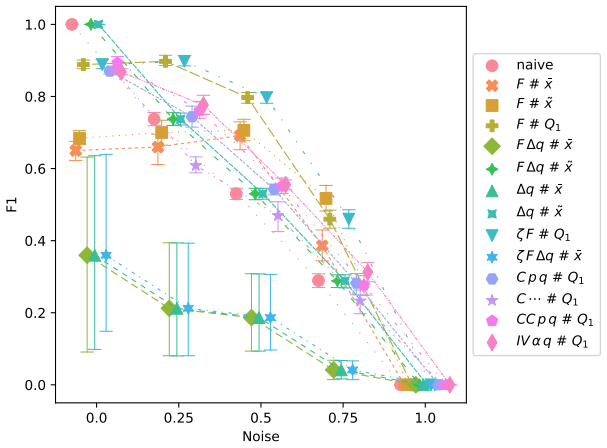
Possible reasons for noise in expert knowledge in real-world settings could be inexperienced operators or experts who are convinced their knowledge is true while, in fact, it is not. To investigate the performance of the proposed aggregation methods in this regard, this experiment varies the degree of noise present in the expert knowledge. In specific, the degree of noise is evaluated at $(P \times Q)_{k_n} \in [0, 0.25, 0.5, 0.75, 1]$, where 0, corresponding to 0%, denotes no noise, i.e. 100% accurate expert knowledge, while 1, corresponding to 100% denotes random expert "knowledge". Note, that the x-axis is treated as categorical and values are placed within the respective bins to avoid overlap. $(P \times Q)_{k_n} = 0.25$ equals the standard configuration used in Section 7.1.3 for the results achieved on knowledge segments elicited from operators. We assume that with growing degree of noise the complexity of the dataset increases and consequently that the hypotheses formulated in Section 7.1.4 hold.

In Figure 7.2, we can see that the hypothesis that F1 decreases with increasing noise is accurate for synthetic datasets with quantified parameters, PDPK_ρ, and quantified conditions, PDPK_o. For PDPK_ρ (see Figure 7.2a), F1 decreases by an average over all aggregation methods of 19.01% (0.13 absolute), 17.37% (0.12 absolute), 51.27% (0.24 absolute) and 0.28 absolute, respectively, for all degrees of noise. For the respective best performing aggregation method for the different degrees of noise, F1 decreases by 10.25% (0.10 absolute), 11.19% (0.10 absolute), 35.12% (0.28 absolute) and 0.52 absolute, respectively. However, different aggregation methods perform best for different amounts of knowledge. For 0% noise, the naive method along with $F\,\Delta q \,\#\, \tilde{x}$ and $\Delta q \,\#\, \tilde{x}$ performs best, while for 25% and 50% $F \,\#\, Q_1$ performs best with an improvement over the naive aggregation method of 21.75% (0.16 absolute) and 50.38% (0.27 absolute), respectively. The good performance of the naive aggregation method on 0% noise is not surprising, since if there is no noise present in the dataset at all, there are no erroneous knowledge segments that can be filtered out to achieve a benefit. For 75% noise $F \,\#\, \tilde{x}$, outperforms $F \,\#\, Q_1$ by 12.49%, which corresponds to an improvement over the naive aggregation method of 78.78% (0.23 absolute). The achieved decrease of the respective best method over naive aggregation in terms of rules (see Figure 7.2b) is 50.81% (40.6 rules), 52.09% (58.6 rules) and 70.66% (109.8 rules) for 25%, 50% and 75% of noise, respectively. For 100% noise no method is able to extract rules as illustrated by the F1 score of 0.

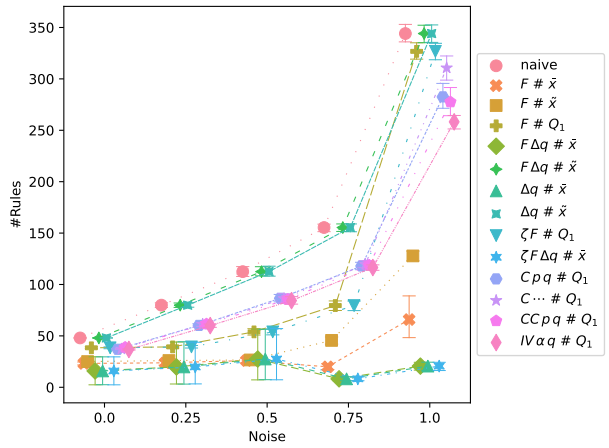
In general, PDPK_o (see Figures 7.2c and 7.2d) behaves analogously to PDPK_ρ. Differences being worse overall results, and the naive method being the sole aggregation method to achieve the best performance for 0% noise. In both PDPK_ρ and PDPK_o, $F\,\Delta q \,\#\, \bar{x}$, $\Delta q \,\#\, \bar{x}$ and $\Delta q \,\#\, \tilde{x}$ perform

consistently worse than the other aggregation methods.

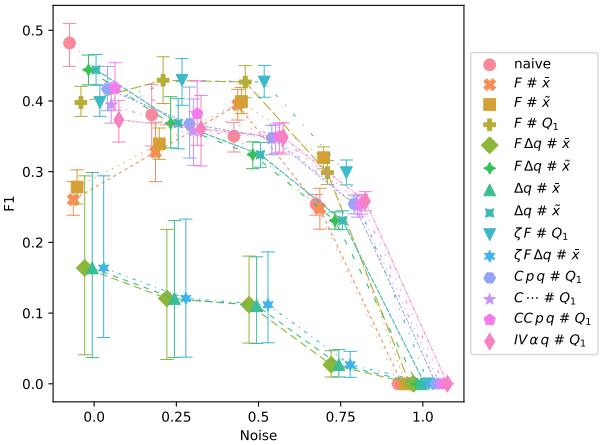
We conclude, that for varying degrees of noise in expert knowledge the hypotheses hold and increasing complexity of this kind increases the importance of weighting and filtering-based aggregation methods up to the degree where satisfactory rules cannot be extracted any more.



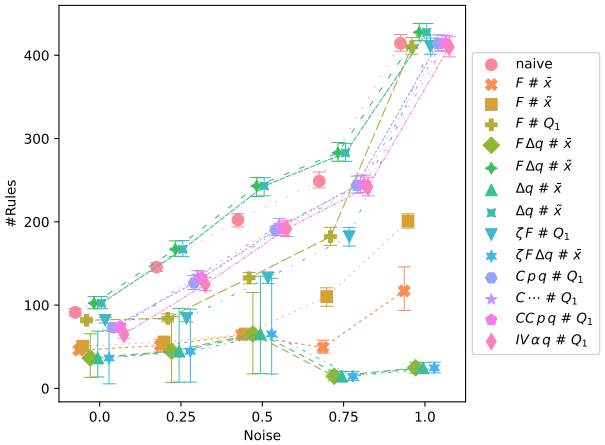
(a) F1 for PDPK$_\rho$.



(b) Amount of rules for PDPK$_\rho$.



(c) F1 for PDPK$_o$.



(d) Amount of rules for PDPK$_o$.

Figure 7.2.: Results of Section 7.1.4.1 investigating varying degrees of noise in expert knowledge

7.1.4.2. Degree of Atomicity

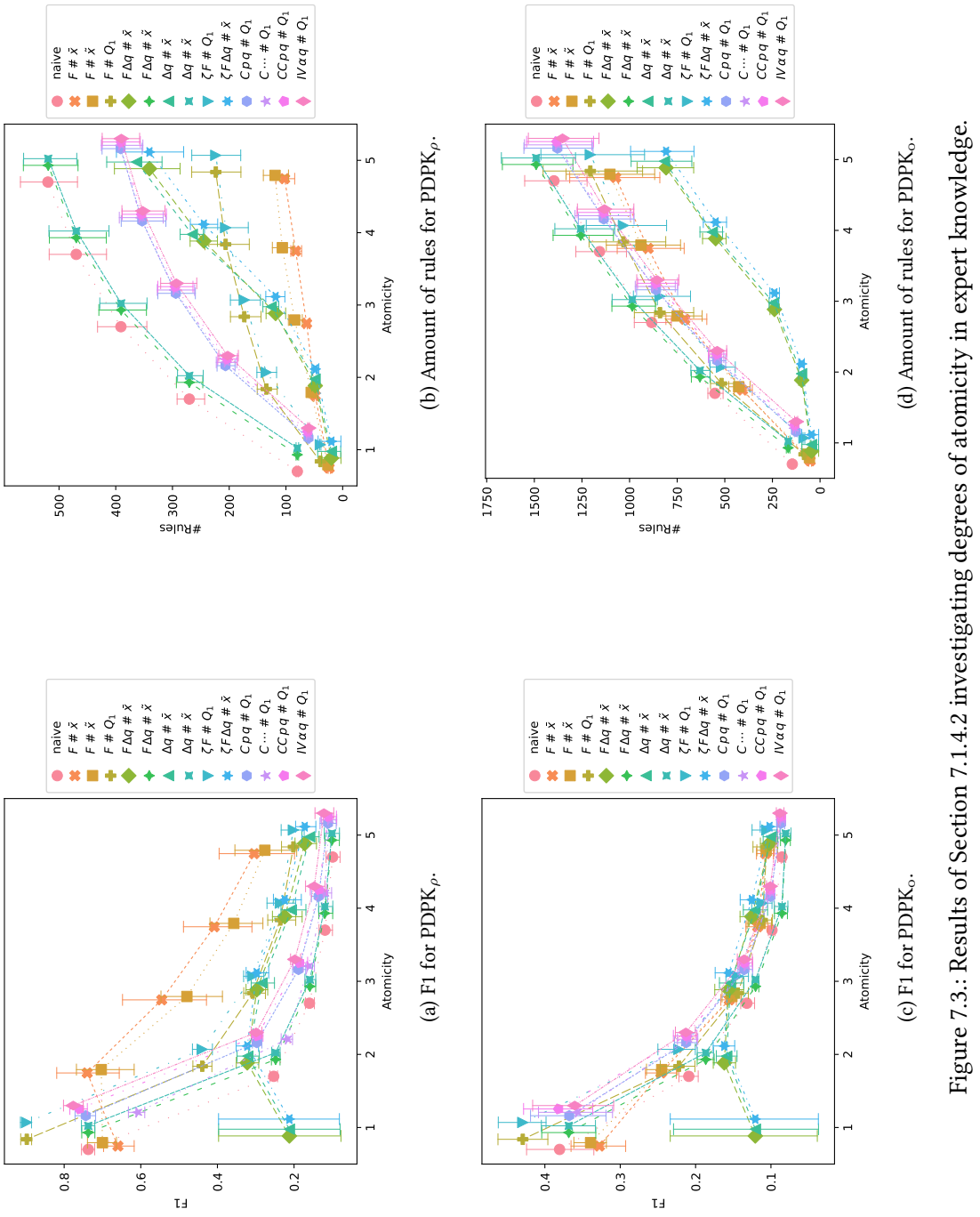
If operators mitigate multiple quality characteristics at once by modifying multiple parameters, the resulting knowledge segments exhibit an atomicity greater than 1, if they are elicited with the methodology presented in Section 5.1. The notion behind the concept of atomicity is related to that common in chemistry, i.e. the amount of atoms, i.e. in our case quality characteristics, which constitute a molecule. To allow us to quantify the effects of atomicity on the complexity of the resulting data, this experiment varies the degree of atomicity present in the expert knowledge. This enables us to contrast the characteristics of data with atomicity of 1, as achieved by the sequential influence elicitation method (see Section 5.2) to data with possibly higher atomicity resulting from the combined influence elicitation method. This experiment is structured like the experiment on the degree of noise in expert knowledge (see Section 7.1.4.1). As such, atomicity is evaluated for $a \in [1, 2, 3, 4, 5]$, with 1 and 5 resulting from one and five quality characteristics remedied simultaneously, respectively, with $a = 1$ corresponding to the degree used in the default dataset analysed in Sections 7.1.3 and 7.1.4.1. As in Section 7.1.4.1, the x-axis is treated as categorical. Again, we assume, that with growing atomicity the complexity of the dataset increases and consequently that the hypotheses formulated in Section 7.1.4 hold.

For rules with quantified conclusions on the synthetic dataset, PDPK_ρ (see Figure 7.3a), we observe that F1 decreases by an average over all aggregation methods of 27.13% (0.26 absolute), 27.65% (0.10 absolute), 25.36% (0.07 absolute), 18.44% (0.04 absolute), respectively, for each atomicity increment. For the best performing methods, $F \# Q_1$ for $a = 1$ and $F \# \bar{x}$ for $a \in \{2, 3, 4, 5\}$, this change occurs more evenly with 17.57% (0.15 absolute), 26.28% (0.19 absolute), 25.17% (0.14 absolute), 25.54% (0.10 absolute), respectively. The similarities in regard to the best methods for the noise experiment could indicate that the kinds of complexity introduced by changes in atomicity are related to those introduced by noise. However, the deviation of $F \# \bar{x}$ over different iterations of the synthetic dataset generation is relatively high when compared to other aggregation methods. The naive baseline is consistently outperformed for each degree of atomicity by the best aggregation method by 21.75% (0.16 absolute), 192.35% (0.49 absolute), 241.20% (0.39 absolute), 242.33% (0.29 absolute), 206.25% (0.20 absolute), respectively. The best aggregation methods leads to a decrease in the number of rules (see Figure 7.3b) when contrasted to the naive aggregation method of 50.81% (40.6 rules), 82.13% (222 rules), 83.76% (327.4 rules), 82.37% (386.9 rules) and 80.33% (417.4 rules) for all degrees of atomicity. In fact, the rule base of $F \# \bar{x}$ only increased by 329.41% (78.4 rules) between $a = 1$ and $a = 5$ compared to the naive aggregation method which increased by 550.31% (439.7 rules).

Like in the previous experiments, the behaviour of rules with quantified conditions, present in PDPK_o, (see Figure 7.3c) is closely related to that of rules with quantified parameters present in PDPK_ρ. The differences are a lower F1 score overall as well as a stronger decrease in performance between the first and second atomicity increment of 42.71% (0.18 absolute) compared with 17.57% on PDPK_ρ. Also, $F \# Q_1$ is the best aggregation method on $a = 1$, being outperformed by $F \# \bar{x}$ on $a \in \{2, 5\}$ which again is outperformed by $F\,\Delta q \# \bar{x}$ on $a \in \{3, 4\}$. The naive aggregation method is outperformed by the respective best aggregation method by 12.83% (0.05 absolute), 17.66% (0.04 absolute), 18.37% (0.02 absolute), 27.09% (0.03 absolute), 25.66% (0.02 absolute) for each atomicity level.

As in Section 7.1.4.1, we conclude that for varying degrees of atomicity the hypotheses hold and increasing complexity of this kind increases the importance of weighting and filtering-based aggregation methods. The decision whether combined or sequential influence elicitation is used

can be made dependent on the intended application domain to best fit the respective production and parametrisation process, e.g. how many quality defects are usually addressed simultaneously, and expected cognitive load of the operators.



(a) F1 for PDPK$_\rho$.

(b) Amount of rules for PDPK$_\rho$.

(c) F1 for PDPK$_o$.

(d) Amount of rules for PDPK$_o$.

Figure 7.3.: Results of Section 7.1.4.2 investigating degrees of atomicity in expert knowledge.

7.1.4.3. Degree of Complexity in Parameter-Quality Functions

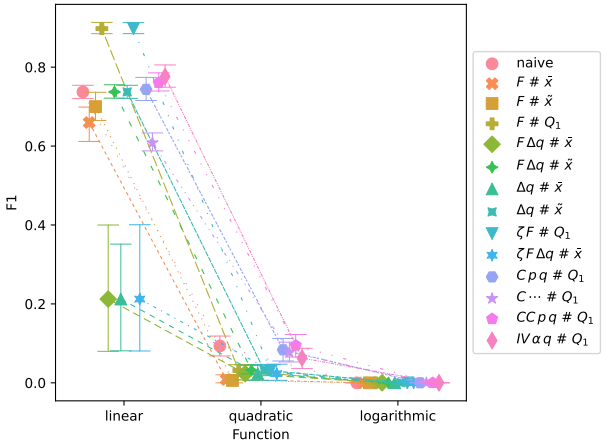
Depending on the specific characteristics of parameter-quality-relationships, the underlying functions can be of different complexities. This experiment evaluates the effect of different function complexities, i.e. linear, quadratic and logarithmic for rules with quantified conclusions and conditions, PDPK_ρ and PDPK_o, respectively, on the synthetic dataset. As such, it provides a quantitative evaluation into a possible benefit of quantified conditions on function types other than linear functions as seen in the previous datasets. Since the default dataset is based on linear functions, we do not analyse the corresponding results in detail since this has already been done in Sections 7.1.3, 7.1.4.1 and 7.1.4.2. We assume, that quadratic and logarithmic functions are more complex than linear and consequently expect worse results but a greater effect of quantified conditions and weighting and filtering-based aggregation methods for them. Also, we expect that—in contrast to the previous experiments—quantified conclusions prove beneficial for more complex function types.

For rules with quantified conclusions as present in PDPK_ρ (see Figure 7.4a), we observe that F1 sharply decreases for quadratic functions by an average margin of 0.59, which corresponds to 92.31%. Between the respective best aggregation methods $F\ \#\ Q_1$ and $CC\ p\ q\ \#\ Q_1$ there is a decrease of 0.80, which corresponds to 89.45%. For quadratic functions, $F\ \#\ Q_1$ provides only a slight benefit over the naive aggregation method of 1.33%. In contrast to this, the amount of rules (see Figure 7.4b) remains relatively constant overall, with a slight decrease for a majority of aggregation methods when compared to linear functions. The exception to this trend are $F\ \Delta q\ \#\ \bar{x}$, $\Delta q\ \#\ \bar{x}$ and $\zeta\ F\ \Delta q\ \#\ \bar{x}$, which increase for quadratic and logarithmic functions as well as $F\ \#\ Q_1$ and $\zeta\ F\ \#\ Q_1$ which increase for quadratic functions. In contrast to the naive aggregation method, the amount of rules was reduced by 24.47% (18.5 rules) for the best aggregation method for quadratic functions.

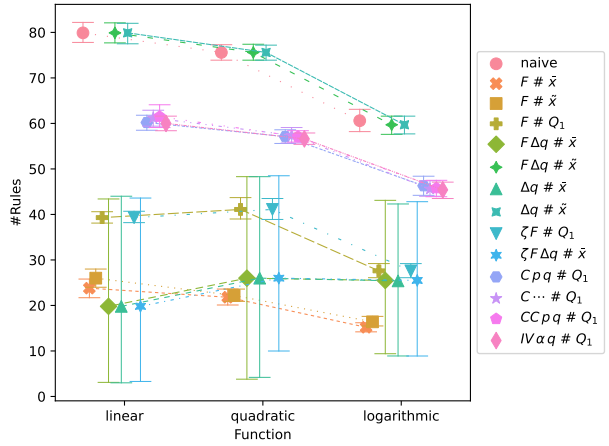
For rules with quantified conditions as present in PDPK_o (see Figure 7.4c), we observe that linear functions perform worse than on PDPK_ρ. However, within PDPK_o an increase in F1 between linear and quadratic functions is visible. Over all aggregation methods quadratic functions outperform linear ones by 31.38% (0.09 absolute). $F\ \#\ Q_1$ is the best performing aggregation method on linear and quadratic functions with an increase in F1 by 0.09, which translates to 21.67%. For logarithmic functions, we observe an average decrease of 0.17 which translates to 52.26% when compared to linear functions. Between the best methods, $F\ \#\ Q_1$ and $C\ \cdots\ \#\ Q_1$, this decrease is quantified by 0.23 which translates to 53.70%. While for quadratic functions an improvement for $F\ \#\ Q_1$ over the naive aggregation method of 12.83% (0.05) can be reported, for logarithmic functions practically no benefit of weighting and filtering-based aggregation methods is visible with a difference of only 0.5%. The amount of extracted rules varies depending on the method. For $F\ \#\ Q_1$ an increase by 89.4 rules corresponding to 106.18% can be observed between linear and quadratic functions. For logarithmic functions the amount of rules decreases by 77.32% when compared to linear functions for the respective best aggregation methods in regards to F1.

We conclude that the hypothesis that the increased complexity of quadratic and logarithmic functions compared to linear functions leads to worse F1 results holds. However, our second hypothesis that for these more complex functions the benefit of using weighting and filtering-based aggregation methods is more pronounced turned out to be false. A possible explanation for this behaviour could be that the aggregation and extraction methods were optimised on the FDM dataset which does not exhibit quadratic functions and consequently better suited to aggregate quantifications by averaging individual values. Fitting local functions would be an

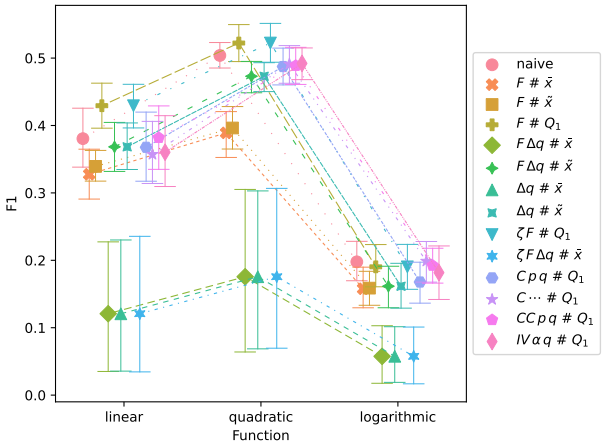
alternative approach that does not suffer from this effect but achieved only unsatisfactory results in initial experiments due to the low sample size. Contrasting the results of rules with quantified conclusions to those with quantified conditions, we conclude that while for linear functions the extraction of quantified conditions is detrimental, it provides a benefit for quadratic and logarithmic rules as evidenced by an absolute increase of 0.43, which translates to 451.15%, and 0.20, respectively.



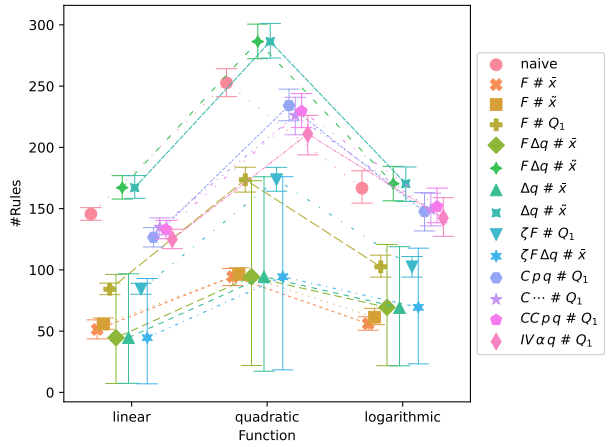
(a) F1 for PDPK$_\rho$.



(b) Amount of rules for PDPK$_\rho$.



(c) F1 for PDPK$_o$.



(d) Amount of rules for PDPK$_o$.

Figure 7.4.: Results of Section 7.1.4 investigating different function types.

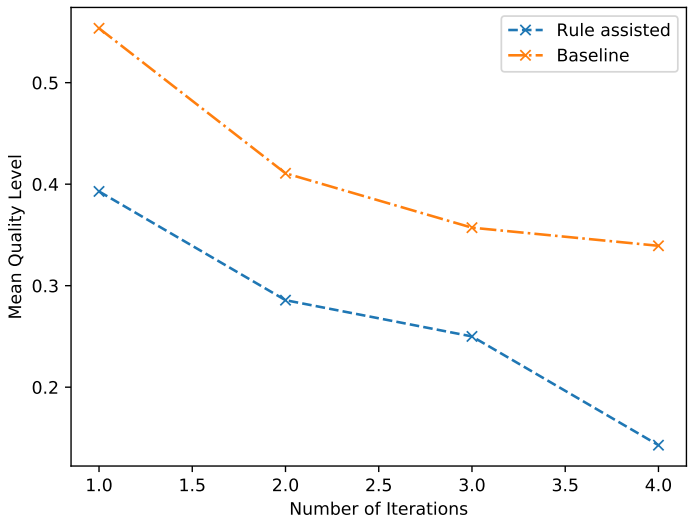


Figure 7.5.: Mean quality level (lower is better) per iterations as achieved by the baseline as well as rule assisted group. For comparability the figure is truncated at iteration 4.

7.2. Applicability in Practice

To study the applicability of the filtered aggregated rule set in practice, e.g. as a passive assistance system, we utilised the FDM case study (see Section 3.1). We selected six participants of varying experience, that did not previously participate in data collection or ground truth creation. They were presented with a printing task designed to challenge several quality characteristics. The target criteria was to optimise both quality and iterations required. To ensure a comparable environment, each participant utilised the same printer with the same material and the same initial parametrisation.

The participants were divided in two groups of three. The baseline was created by the control group that completed the task solely relying on their previous experience and process knowledge. The second group was given the rule set aggregated by $C\,p\,q\,\#\,\bar{x}$—the best aggregation method for the 411 experiments with corresponding knowledge segments that were available when this experiment was conducted.

To resemble the operator decision capability prevalent in manufacturing scenarios, the operators in the second group could apply as many rules pertaining to observed quality defects as they saw fit. However, their degree of FDM relevant expertise was significantly lower than that of the baseline group. The uneven split regarding operator experience was chosen to evaluate to which degree novices (with rules) are able to compete with experienced operators (without rules). The control group took a mean of 5.33 iterations with a standard deviation of 0.47, whereas the rule assisted group took an average of 5.0 iterations with a standard deviation of 1.4. The high standard deviation in the rule assisted group is explained by one operator that required seven iterations compared to the four iterations required by the other operators.

A comparison of the achieved quality can be seen in Figure 7.5, in which the achieved mean quality (consisting of four defective quality characteristics) of both groups is compared for each

process iteration. The figure is truncated at four iterations since otherwise the comparability is not ensured because of the different numbers of iterations required by the participants. While the environmental temperature was higher for the rule assisted group than the control group, experienced operators should be able to compensate environmental temperature based on their experience. Therefore, this does not explain the general offset in performance. After iteration four, the achieved quality of the baseline improved while one operator of the group utilising rules decreased in achieved quality until finding a suitable parametrisation at iteration seven.

For eight of the observed 13 occurring quality defects in the FDM dataset, rules could be extracted. However, for some quality characteristics they were erroneous which lead all participants in the rule assisted group to be unable to significantly improve upon these quality characteristics. All the more interesting is the fact that for the first four iterations the application of the rules achieves a better quality faster than the control group, which consisted of more experienced operators. Apart from the general offset, this is underpinned by the fact that the difference between achieved mean quality at iteration four is larger than that at iteration one. Also, in the rule assisted case different rules were applied. Due to the inexperienced operators, we assume that this hints at a high uncertainty while selecting rules. Aiding the operators through programmatic rule selection could further increase the applicability of the extracted rules in practice. Overall, the applicability of the extracted rules in practice can be assessed as positive since they lead to quicker parametrisations which attain a better quality faster than the control group. However, because of the small study size and environmental factors, which are hard to control, the results of this experiments only serve as an indication of the quality of the extracted rules.

Part III.

Representing and Embedding Procedural Operator Knowledge

To facilitate the use of procedural operator knowledge, extracted e.g. by the method proposed in Part II, in neural network based predictive systems, it is beneficial to transform it into a vector representation. Chapter 8 details the state of the art for both representations as well as embeddings. To facilitate interchangeability with other data and knowledge sources, we investigate the use of knowledge graph representations of procedural knowledge as a machine readable, standardised information model (IM) (see Chapter 9). We then investigate ways to embed these knowledge graphs into low dimensional vector representations (see Chapter 10). Chapter 11 evaluates both representations and embedding approaches using a surrogate metric for downstream tasks devised in this thesis, *matches@k*. This part draws on work previously published in [Nor+22c; Nor+22b; Nor+23b] and as such uses parts of these publications. The corresponding passages are not explicitly cited again for the sake of readability.

8
Related Work

This chapter introduces concepts and related work pertaining to the representation and embedding of procedural knowledge.

8.1. Knowledge Graphs as Industrial Information Models

Knowledge graphs (KGs) are a popular data structure to integrate knowledge of multiple heterogeneous sources in industry [Buc+21; Noy+19]. Combined with approaches that allow reasoning over knowledge graphs, e.g. for link prediction to facilitate knowledge graph completion, they are a logical choice for semantic industrial information models [GLu20; Kal+20; Rin+17]. However, the knowledge typically represented in these industrial information models is of mostly factual and conceptual nature, reaching the level of 'knowledge of principles and generalizations' of Krathwohl [Kra02]'s taxonomy. It concerns equipment [GLu20; Rin+17], material [GLu20], process segments [GLu20], parts [Noy+19], products [Noy+19; GLu20], events [Rin+17], underlying measurements [RLL16] and geospatial information [Pet+17] as well as relations interconnecting these entities. Based on this information, use cases addressed range from anomaly detection [Kal+20] over process monitoring [Hub+18; Kal+20] to locating parts and equipment in plants [Pet+17] and risk management [Hub+18]. To quantify the complexity of knowledge bases, [DV22] presents objective metrics. To summarise, industrial knowledge graphs are gaining traction in practice but are frequently dealing only with factual knowledge and binary relations [Buc+21]. In Section 9.2 we address this limitation by providing modelling patterns for procedural knowledge.

8.2. Representing Relations of Higher Arities

Knowledge Graphs are by definition limited to triples [Fat+20]. To represent relations of a higher arity in knowledge graphs, the concept of reification known for statements in RDF is applied to n-ary relations [Noy+06]. It transforms them to triples by introducing auxiliary entities [Fat+20; Noy+06]. The W3C differentiates between reification and representing n-ary relations as binary relations since here 'the intent is to talk about instances of a relation, not about statements about such instances' as is done in RDF [Noy+06]. However, since in practice this semantic distinction is not made in literature [Fat+20] we refer to it as reification. Even though reification is established as a technique to convert non-binary relations to binary ones, standard embedding methods do not work well on the resulting knowledge graphs as shown by [Fat+20]. Therefore, we introduce the related but separate concept of chained-binary relations in Section 9.2 and analyse their performance in Chapter 11.

8.3. Graph Embedding Methods

Low dimensional vector representations of data are referred to as embeddings [Mik+13; Wan+17; Ham20]. Consequently, in the field of graph representation learning [Ham20] these embeddings are computed by embedding methods which operate on graphs. In the following, we will differentiate between embedding methods that were developed for refinements of the knowledge graph and those that strive to embed the graph for a data-mining task often focussing on a downstream usage. While examples for knowledge graph refinement are knowledge graph completion and knowledge graph alignment [Wan+17; Ros+21; Dai+20], examples for the data-mining perspective are information retrieval, recommender systems or predicting variables outside of the knowledge graph's scope by utilizing the knowledge graph as its data [Coc+17; PHP22]. This distinction is in accordance with the literature, that independently developed embedding methods from these different perspectives [PHP22].

8.3.1. Embedding Methods for Knowledge Graph Refinement

Knowledge graph embedding methods for refinements operating directly on the knowledge graph, e.g. knowledge graph completion entity classification, triple classification and entity resolution, have been extensively researched [Wan+17; Ros+21; Dai+20; Ji+21]. While a complete overview of all embedding methods is beyond the scope of this thesis (see e.g. Wang et al. [Wan+17] and Ji et al. [Ji+21]), we limit this section to those relevant for this thesis and refer to the respective publication for more details. Embedding methods rely on different principles, e.g. translation (e.g. TransE [Bor+13], TransH [Wan+14], RotatE [Sun+19]), bilinearity (e.g. RESCAL [NTK11], DistMult [Yan+14], ComplEx [Tro+16], TuckER [BAH19], CP [Hit27] and SimplE [KP18a]), and region-based (e.g. BoxE [Abb+20]) and exhibit different strengths and weaknesses [Wan+17; Abb+20]. Schlichtkrull et al. [Sch+18] extend GCNs to knowledge graphs and present a relational GCN (R-GCN) for entity classification in knowledge graphs. Furthermore, they show how it can be combined with a DistMult decoder to function in a link prediction setting. Another approach of adapting convolutional networks for link prediction can be found in ConvE [Det+18]. Also, most of these embedding methods cannot capture literals that further refine the entities in a knowledge graph. However literals could be used to represent quantifications included in procedural knowledge. As such, methods explicitly catering for literals as analysed in [Ges+21] are of special interest for this thesis. However, whether direct transfer of the results achieved using these methods to graphs containing procedural knowledge is possible needs to be validated.

Higher Arity While knowledge graphs represent binary knowledge triples, the underlying knowledge bases that are represented in knowledge graphs can also contain non-binary relations of higher arity [Abb+20]. To directly model non-binary relations the concept of hypergraphs is introduced by [Fat+20]. Therefore, several extensions to binary embedding methods such as m-TransH [Wen+16], HSimplE, m-DistMult and m-CP [Fat+20] for TransH, SimplE [KP18a], DistMult and CP, respectively, have been proposed. According to Abboud et al. [Abb+20], conceptual and practical challenges limit the generalisability of binary embedding methods. Liu, Yao and Li [LYL20] proposed m-Tucker and GETD as generalisations of TuckER, however, they are only applicable if the knowledge base or hypergraph contains only relations of one specific arity. For varying arities, HypE [Fat+20], a convolutional embedding method; ReAIE [Fat+21], a method

that is able to represent relational algebra operations and BoxE [Abb+20] a region-based method have been proposed.

Evaluating Knowledge Graph Refinement Embeddings Link prediction is the standard scenario to evaluate embedding methods for refinement of knowledge graphs as evidenced by most major publications relying on them, e.g. [Bor+13; Cha+20; Tro+16; Sun+19; Zha+19b; Yan+21; Sch+18; Lin+15; Wan+14]. Link prediction is the task of predicting either head h or tail t of a triple (h, r, t) [Wan+17]. However, doubt has been cast both on biases [Ros+21; MPT20; TC19] in the widely used benchmark datasets, e.g. FB15k and FB15k-237 [Tou+15] derived from Freebase [Bol+08] or WN18 and WN18RR [Det+18] derived from WordNet [Mil+90], as well as on the more general capability of KG embeddings to capture semantics [Jai+21]. Therefore, behavioural testing of embedding methodologies is gaining attention [Ben+21] and standardised benchmarks are being proposed [PP22a]. Also, more complex datasets e.g. YAGO3-10 [Det+18] and DBpedia50k/DBpedia500k [SW18] have been proposed.

Contrary to the procedural knowledge forming the focus of this thesis, the datasets and knowledge graphs used to evaluate embedding methods usually contain mostly factual or conceptual knowledge. Since we are aiming at using embeddings not only for link prediction but to improve the performance of learning systems, an evaluation of the embeddings' encoding of the required semantic information is necessary in our case. To this end, we outline a surrogate metric for a downstream scenario *matches@k* in Section 11.1.1.

8.3.2. Embedding Methods for Data-Mining

Cochez et al. [Coc+17] provide an overview of embedding methods of knowledge graphs stemming from a data-mining background and consequently tailored primarily for out-of knowledge graph applications and downstream tasks, e.g. question answering, recommender systems and predicting variables [PHP22]. At the intersection of embedding methods for downstream tasks and those for knowledge refinement, Portisch and Paulheim [PP22b] extend RDF2Vec [RP16], an embedding method that conducts random walks in the knowledge graph and uses word2vec [Mik+13] to embed the resulting walks, for the link prediction problem. Furthermore, they provide a more thorough comparison of link prediction embedding methods to those encountered in data mining than done by Cochez et al. [Coc+17].

They compare the performance of both strands of research on the respectively other problem highlighting different capabilities of the respective methods.

Apart from the world of knowledge graph specific methods, there exist a several methods for homogeneous graphs such as node2vec [GL16] and DeepWalk [PAS14]. Graph Neural Networks (GNNs) are a further widespread embedding method for homogeneous graphs. Examples include spectral Graph Convolutional Networks (GCNs) [KW16; DBV16], spatial GNNs [HYL17a] and attention enabled GNNs [Vel+18; BAY21]. For further detail we refer to Hamilton [Ham20]. However, since homogeneous graphs allow only one relation type and could be undirected, transforming knowledge graphs to homogeneous graphs to be able to apply these methods comes with a definite loss of semantics.

Embedding methods for heterogeneous graphs, also referred to as heterogeneous information networks (HIN), which include directed and heterogeneous relations have also been proposed, e.g. [Fu+20] that extend GNNs for heterogeneous graphs. Heterogeneous information networks, traditionally conform to underlying schemas, which can be exploited by embedding methods

[Shi+16]. However, these schemas are what differentiates them from knowledge graphs, that are usually schema less [Shi+16]. Another difference is that the type of a node is 'part of the graph model itself rather than being expressed by a special relation' [Hog+21] as in the case of knowledge graphs. Interestingly, HIN embedding methods—even though their roots lie in data-mining and therefore downstream evaluation heavy scenarios—are frequently evaluated on settings similar to knowledge graph refinement, e.g. link prediction. Still, since they stem from different areas of research they are evaluated on different datasets and to the best of our knowledge not directly benchmarked against each other.

In general, however, embedding methods for data mining and downstream tasks are usually indirectly evaluated by the performance of the downstream task. In contrast to embedding methods for knowledge graph refinement, they are usually trained (semi-) supervised.

8.3.3. Rules & Embeddings

While rules have, to the best of our knowledge, not been directly represented in knowledge graphs used as industrial information models, they have been frequently used in different semantic contexts.

Rule representation in graphs Representations of rules in graphs have been addressed by [CM09]. They explored how bi-coloured graphs can be used to encode condition and conclusion of rules on both a general level as well as, optionally, for specific entities encoded via attributes. However, their approach does not offer a way to provide quantifications to either conditions or conclusions. Also, established embedding methods are not directly applicable to bi-coloured graphs, which limits the usability of their approach for e.g. knowledge-infused learning.

Assisted Embeddings More often than being directly represented in graphs, logical rules are used as auxiliary information for knowledge graph embeddings [Guo+16; Guo+18]. Here, rules are either provided by algorithm designers or domain experts to capture common sense knowledge or automatically mined from knowledge graphs [Guo+18]. Zhang et al. [Zha+19b] create relations based on rules that lead to an increase in embedding performance. Ringsquandl et al. [Rin+18] conclude that the performance of KG completion can be increased by utilizing embeddings of events, i.e. time-series data [Rin+17] during KG embedding. However, in contrast to the events or logical rules employed in these approaches, rules based on extracted operator knowledge frequently contain quantifications of conditions or conclusions which makes them more complex and unsuited to the described approaches.

Rule Representations in Embeddings Gutiérrez-Basulto and Schockaert [GS18] mention that previous approaches are not suited to embed rules, which are an integral part of representing procedural knowledge. They provide theoretical considerations how this could be alleviated and how 'existential rules can be exactly represented using convex regions of knowledge graph embeddings' [GS18]. While a methodology that provides exact representation of general rules in embeddings would prove greatly helpful in evaluating the suitability of different methodological choices of rule representation in knowledge graphs, we cannot rely on Gutierrez's methodology since the rules containing the experts' extracted knowledge are more complex than the existential rules considered. Furthermore, representing rules in knowledge graphs as opposed to embeddings,

provides a significant benefit for information models as the representation is more direct and can be directly accessed and interpreted. In contrast to this, Abboud et al. [Abb+20] present an implemented approach, BoxE, with experimental evaluation, that is able to represent existential rules and deal with higher arities. Similarly suited to graph containing higher arities, Fatemi et al. [Fat+21] presented ReAIE that 'can represent a large subset of primitive relational algebra operations'.

9

Representations of Procedural Knowledge in Industrial Information Models

This chapter discusses the importance of standardised, semantic industrial information models, e.g. knowledge graphs, and presents contributions of this thesis regarding alternatives to represent procedural knowledge in knowledge graphs. As such, it draws on work previously published in [Nor+22c; Nor+22b; Nor+23b].

9.1. Importance of Procedural Knowledge in Industrial Information Models

On the one hand, standardised semantic information models (IMs) and standards for their hosting, such as the Industry 4.0 asset administration shell, are gaining traction in the industrial internet of things where they can be used to facilitate interoperability and data interchange between different companies, production plants, lines or machines [Pla+19; BCB21; Pil+20]. On the other hand, industrial knowledge graphs are becoming prevalent for the integration of heterogeneous data as shown in Section 8.1. We are not aware of the use of industrial IMs containing extracted expert knowledge, which belongs to the more advanced conceptual, i.e. 'knowledge of models, theories, structure' [Kra02], and procedural categories of knowledge. This knowledge is of heuristics-like nature and usually obtained through multiple years of experience. A more detailed explanation of expert knowledge and procedural knowledge for the manufacturing scenario investigated here is given in Section 4.1.

Since expert knowledge is playing a crucial role in many industrial day to day processes—from the design of components to reparametrisation processes necessary to deal with quality defects during production—and only available to a limited number of people, representing it in a standardised way is of great interest for the industry. Therefore, we propose that including procedural knowledge would enhance the applicability of IMs in several ways:

1. a semantic integration between the *what* of given machinery and the *how* to operate it

2. suitability for predictive quality use cases, e.g. by utilizing the resulting knowledge graphs, which would contain quantified knowledge, to increase the performance of learning systems. This could lead to an increase in the ability to generalise and cope with low amounts of data—both frequent challenges in industrial contexts

3. a standardised representation of procedural knowledge which would (1) enable the creation of digital process twins that could be supplied alongside machinery for operating and training purposes, thereby reducing the impact of changes in a production line, (2) provide

a standardised way to combine it with varying kinds of knowledge from different sources, e.g. physical limits of machinery provided by process engineering and (3) enable the fusion of knowledge extracted by traditional [HSB20] as well as data-based (see Part II) methods.

9.2. Representations of Procedural Knowledge

In manufacturing use cases, a high proportion of expert knowledge is tacit operator knowledge which pertains to parametrisation of machinery. As such, it contains knowledge about both conceptual relationships between process parameters and quality characteristics as well as procedural behaviours that lead to the achievement of goals, i.e. how and in what order to adapt process parameters to mitigate occurring quality defects and achieve a perfect parametrisation. In this thesis, we focus on the aspect of how parameters are adjusted. This tacit operator knowledge can be extracted by various methods as shown in Part II leading to rules at different levels of abstraction. Also, information concerning the same problem might be available from alternative sources such as process engineering documents or handbooks which could be combined or contrasted with knowledge operators gained in practice. As such, adapting and building on definitions for operator knowledge presented in [Nor+22a] (see Section 6.1), we inspect several modelling patterns for representing the underlying knowledge in knowledge graphs, where different abstraction levels can be combined at will. Compared to formal logic based representations, we chose the representation in knowledge graphs since it provides the benefit that end-users, e.g. process engineers, do not have to 'familiarise themselves with the rigour and details of formal logic semantics to communicate with the system' [SRG23]. From the above mentioned modelling patterns, an appropriate degree of abstraction can be chosen to either reflect the kind of knowledge that is available, e.g. unquantified relations yielded by traditional knowledge extraction methods, or of the information models' specific domain. This is relevant since we expect that the higher complexity, i.e. through hierarchies, provides challenges for embedding approaches. Avoiding unneeded complexity in the representation is therefore likely to achieve better results. As such, we recommend choosing the highest abstraction level, that is able to encode all present information. Note, that the representations of higher abstraction levels can be easily integrated with representations of lower abstraction levels. Therefore, including operator knowledge of a different source which utilises a different abstraction level is possible.

In the following, we will extend the definitions presented in Section 6.1 and align our terminology with manufacturing scenarios to increase the readability of examples.

9.2.1. Unquantified Rules

A rule at the highest level of abstraction, i.e. an unquantified rule, could be verbalised as *If quality characteristic q is unsatisfactory then adjust process parameter p.* It can be viewed as an *implies* relation between the condition *quality characteristic q* and the conclusion *parameter p.* This corresponds to the triple notation of (head, relation, tail) common in knowledge graphs. According to the definitions of Sections 2.1 and 6.1, $p \in P$, $q \in Q$, and $\eta_{q,p}$ denote parameters, quality characteristics and a relation between a parameter and a quality characteristic, respectively. Unquantified rules are defined with the triple $r_{\eta_{q,p}} = (q, \langle \text{implies} \rangle, p)$ that specifies η as an *implies* relation between p and q (alternatively written as $(q, p) \in \langle implies \rangle$). In the following, we will omit the indices p, q for readability. A modelling alternative would be a *is a* relation between q and

the semantic meaning of quality characteristics. However, this semantic information is already encoded by the directed relation. Making it explicit would only increase syntactic complexity and hierarchy.

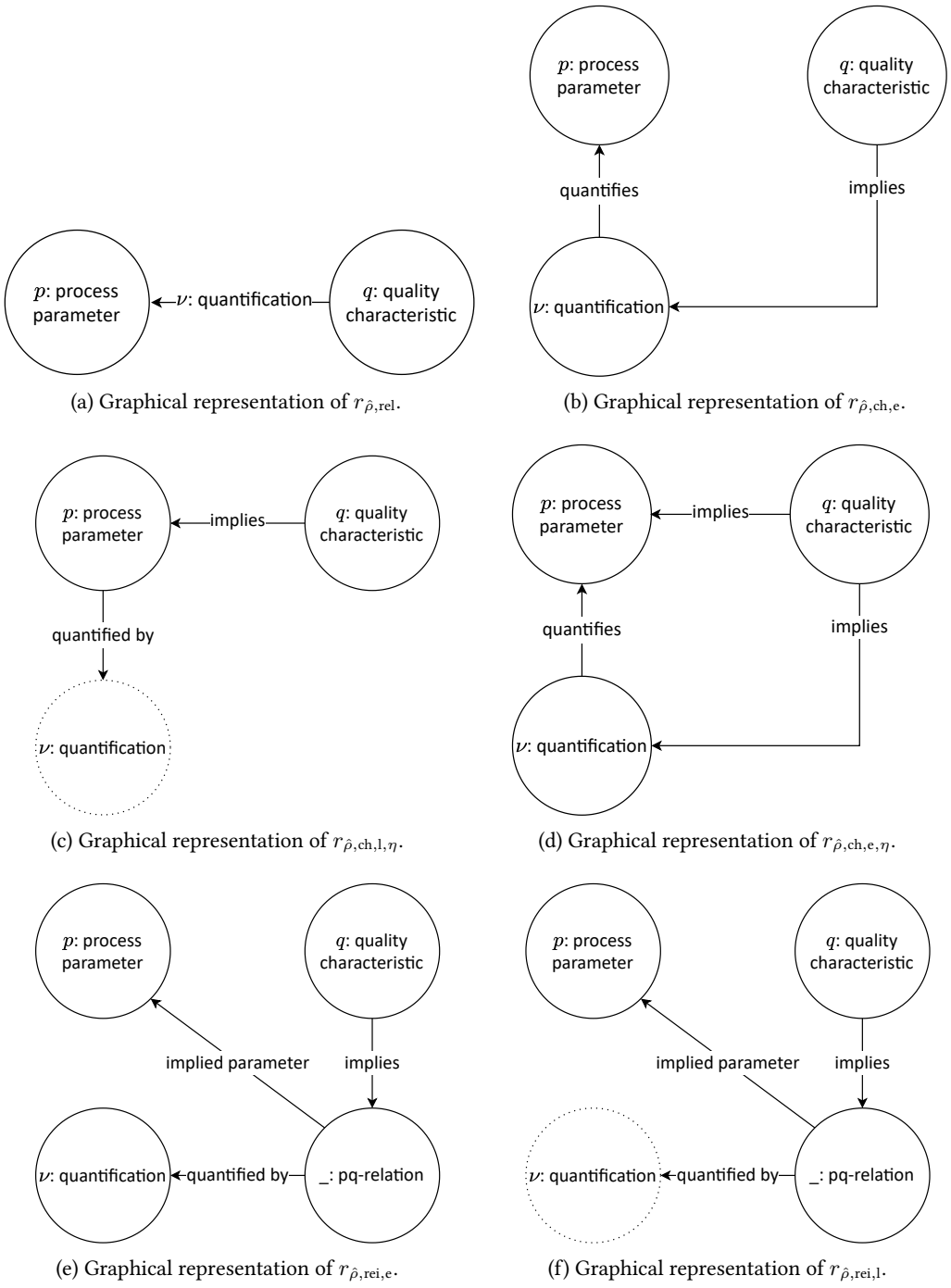
9.2.2. Quantified Conclusions

Rules with quantified conclusions, i.e. parameters, $r_{\hat{\rho}}$ can be verbalised as *If quality characteristic q is unsatisfactory then adjust process parameter p by* ν with $\nu \in \mathbb{R}$ if they refer to relative parameter adjustments or if the parameter is *set to* an absolute value (see Definition 6.1.2). Since these adjustments are the most prevalent, we limit the following descriptions to them. If quantifications in the form of ranges are required, we refer to Section 9.2.3 that outlines how these are handled for quantified conditions. This yields the quadruple, $r_{\hat{\rho}} = (q, \langle\text{implies}\rangle, p, \nu)$ which can be simplified as a tuple building on unquantified rules, r_η, as used in Section 9.2.1, yielding $r_{\hat{\rho}} = (r_\eta, \nu)$. While on an implementation level the relations used are differentiated by quantification type, i.e. absolute and relative, we will refer to them in the undifferentiated version to increase readability. The fact that the the relation $r_{\hat{\rho}}$ is ternary, limits the applicability of established embedding methods, since usually they expect knowledge graphs with binary relations. Consequently, $r_{\hat{\rho}}$ has to be transformed to binary relations if we want to make use of existing embedding methods. To be able to investigate alternatives to the established reification concept, used to reduce arity and introduced in Chapter 8, we present two representation alternatives i.e. relation-based and chained-binary. For comparison, we also transform the relations according to the reification concept. As each of these modelling patterns has unique advantages and drawbacks, it is unclear which representation leads to the best results. Therefore, we refer to Chapter 11 for experimental results. Rules of this degree of abstraction are applicable if the quality cannot be quantified but the actions to mitigate it are usually the same. In the following, we denote the rules of quantified conclusions as $r_{\hat{\rho}}$ since several observations are aggregated into the quantification.

Relation-Based Representation of Quantified Conclusions Generally, we want to keep the representation as succinct as possible. Therefore, the representation of unquantified rules (r_η) is extended to conceptually use the numeric literal ν as a weight of the *implies* relation (see Figure 9.1a). However, since common knowledge graph embedding methods do not support literals of relations, the quantifications were modelled as separate relations. This incurs a loss of semantics which is especially disadvantageous if knowledge from multiple sources or of multiple levels of detail is represented in the graph, as is usually the case in practice. The quantified parameter $\rho \in \mathrm{P}$, where $\mathrm{P} = \{(\nu, \langle\text{quantifies}\rangle, p) \mid p \in P \text{ and } \nu \in \mathbb{R}\}$ is not modelled explicitly but able to be reconstructed through the relation connecting p and q and its weight ν. In the following, we will refer to this representation as $r_{\hat{\rho},\text{rel}}$.

Chained-Binary Representation of Quantified Conclusions To address the issue exhibited by $r_{\hat{\rho},\text{rel}}$, we rely on the fact that the ternary relation can be written as a chain of two binary relations, i.e. $r_{\hat{\rho},\text{ch}} = (q, \langle\text{implies}\rangle, (\nu, \langle\text{quantifies}\rangle, p))$, which can then be directly represented in the graph as can be seen in Figure 9.1b. However, this indirection could be challenging for embedding methods to pick up since it severs the direct semantic link between p and q.

At the quantification level, we're confronted with two modelling alternatives: treating quantifications as entities, $r_{\hat{\rho},\text{ch,e}}$, which diminishes the semantic value of the respective nodes or explicitly modelling them as literals, $r_{\hat{\rho},\text{ch,l}}$, which keeps their semantics largely intact but provides



(a) Graphical representation of $r_{\hat{\rho},\mathrm{rel}}$.

(b) Graphical representation of $r_{\hat{\rho},\mathrm{ch,e}}$.

(c) Graphical representation of $r_{\hat{\rho},\mathrm{ch,l},\eta}$.

(d) Graphical representation of $r_{\hat{\rho},\mathrm{ch,e},\eta}$.

(e) Graphical representation of $r_{\hat{\rho},\mathrm{rei,e}}$.

(f) Graphical representation of $r_{\hat{\rho},\mathrm{rei,l}}$.

Figure 9.1.: Graphical representation of rules with quantified conclusions $r_{\hat{\rho}}$ (see Section 9.2.2).

further challenges for established knowledge graph embedding methods, that lack explicit support for literals [Ges+21]. The second alternative brings with it the need to keep the high-level *implies* between p and q entities as in the unquantified case since otherwise there is no direct

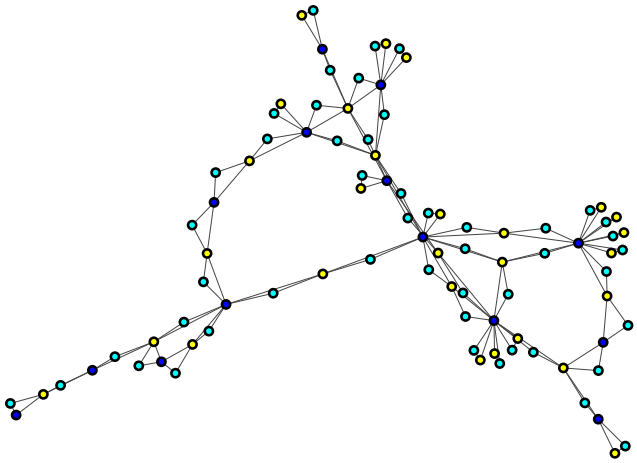


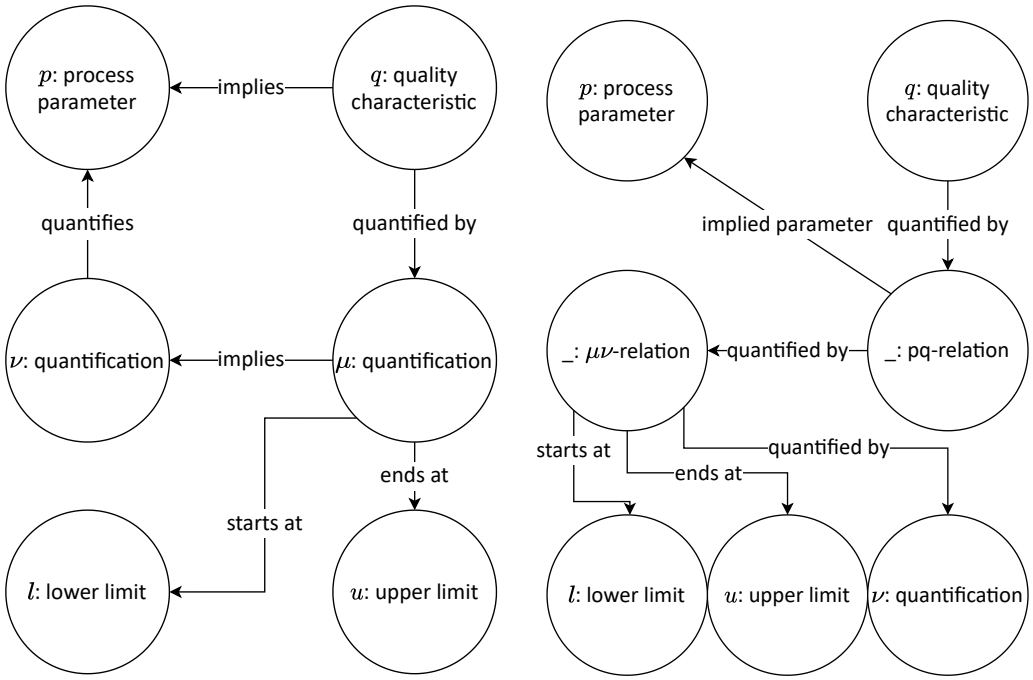
Figure 9.2.: Resulting knowledge graph of representing the expert knowledge of the synthetic dataset PDPK$_\rho$ with $r_{\hat{\rho},\text{ch,e},\eta}$. Parameters are shown in yellow, qualities in blue and quantified conclusions in teal.

connection between them in the graph. Also, the parameter needs to be connected to its quantification by a *quantified by* relation. We denote this literal-based representation as $r_{\hat{\rho},\text{ch,l},\eta}$. This modelling decision leads to this representation exhibiting a compositional property since $(p, \nu) \in \langle\text{implies}\rangle \wedge (\nu, q) \in \langle\text{quantifies}\rangle \implies (p, q) \in \langle\text{implies}\rangle$. For consistency, we introduce a modelling alternative with the same characteristic to the entity-based modelling alternative, resulting in $r_{\hat{\rho},\text{ch,e},\eta}$. While Figures 9.1b to 9.1d allows a graphical comparison of representatives of the previously described modelling alternatives, Figure 9.2, shows the resulting knowledge graph of representing the expert knowledge of the synthetic dataset PDPK$_\rho$ with $r_{\hat{\rho},\text{ch,e},\eta}$.

Reification-based Representation of Quantified Conclusions Another approach to represent n-ary relations is presented in a Noy et al. [Noy+06]. It consists of first introducing an auxiliary anonymous entity describing the relation instance. Then the subject, i.e. the head, of the relation is linked to this auxiliary entity. Also, relations between the auxiliary entity and all additional attributes of this instance of the relation are added. Figures 9.1e and 9.1f provide a graphical visualisation of these representations.

9.2.3. Quantified Conditions

In addition to quantified parameters that serve as actionable recommendations for operators, the conditions, i.e. quality characteristics, can also be quantified to arrive at more descriptive rules. With quantified conditions, it is possible to represent more advanced concepts, e.g. for more pronounced defects of a quality characteristic, parameters need to be adjusted more substantially. While in theory quantified quality characteristics could be used without quantified parameters, it is not beneficial in practice since the conclusion of the rule would remain the same. As such, we consider quantified parameters as a prerequisite of quantified quality characteristics. Rules with aggregated quantified quality characteristics and parameters $r_{\hat{\text{o}},\hat{\rho}}$ can therefore be verbalised as *If quality characteristic q is within μ, then adjust process parameter p by ν*, where μ defines the interval $[l, u]$. This results in the 5-tuple $r_{\hat{\text{o}},\hat{\rho}} = (q, \mu, \langle\text{implies}\rangle, p, \nu)$. Note, that, analogous to ρ



(a) Graphical representation of $r_{\hat{o},\hat{\rho},\mathrm{ch,e},\eta}$.

(b) Graphical representation of $r_{\hat{o},\hat{\rho},\mathrm{rei,e}}$.

Figure 9.3.: Exemplary graphical representations of modelling patterns for rules with quantified conditions $r_{\hat{o},\hat{\rho}}$ (see Section 9.2.3) with reification and chained-binary representation patterns.

for quantified conclusions, i.e. parameters, we utilise $o \in O$, with $O = \{(\mu, q) \in \langle \text{quantifies} \rangle \mid q \in Q \text{ and } \mu \in \mathbb{R}\}$ to denote specific quantified conditions, i.e. quality characteristics.

If the 5-ary relation—5-ary since μ consists of an upper and lower bound—is modelled as chained-binary relations, it has to be modelled explicitly. Strictly speaking this leads to a further indirection, since the *implies* relation now connects the actual value of quantified parameter and quality characteristic, which are decoupled from their semantic interpretation by a *quantifies* relation. Transformed into hierarchical triples this yields:

$$r_{\hat{o},\hat{\rho}} = ((\mu, \langle \text{quantifies} \rangle, q), \langle \text{implies} \rangle, (\nu, \langle \text{quantifies} \rangle, p))$$

The representations for rules with quantified conditions are structured similarly as those of quantified parameters (see Section 9.2.2). As quantification of conditions only provide a benefit if conclusions are also quantified the presented representations (see Figure 9.3) show the combination of both. They are provided with variations of entities, $r_{\hat{o},\hat{\rho},\mathrm{ch,e}}$, literals, $r_{\hat{o},\hat{\rho},\mathrm{ch,l}}$, and the inclusion of high-level relations η, ($r_{\hat{o},\hat{\rho},\mathrm{ch,e},\eta}$ and $r_{\hat{o},\hat{\rho},\mathrm{ch,l},\eta}$ for entity and literal-based quantification of the quality, respectively). For the sake of brevity, only the representation with high-level relations is shown in Figure 9.3a. Also, since embedding methods are unable to deal with complex semantic entities or literals such as ranges [Ges21], the range of μ is modelled with separate entities or literals, in the case of $r_{\hat{o},\hat{\rho},\mathrm{ch,l}}$. Furthermore representations with the reification pattern are presented for entities and literals, $r_{\hat{o},\hat{\rho},\mathrm{rei,e}}$ (see Figure 9.3b) and $r_{\hat{o},\hat{\rho},\mathrm{rei,l}}$, respectively. They contain

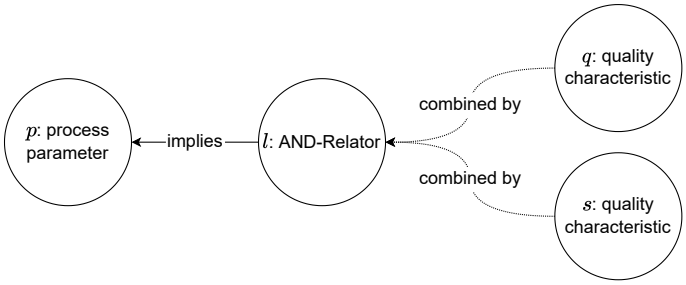


Figure 9.4.: Graphical representation of modelling pattern r_{q^n}, i.e. an *implies* relation indirectly spanning multiple conditions, that are connected by a relator vertex.

a second auxiliary relation to allow the definition of multiple quantifications per quality.

9.2.4. Multiple Conditions

Sometimes a specific parametrisation is only relevant if multiple conditions align. These rules, r_{q^n}, could be verbalised as *If quality characteristic x and quality characteristic z are unsatisfactory, then adjust process parameter p.* For this modelling pattern, we omitted quantifications for the sake of brevity. r_{q^n} could be trivially encoded by having multiple separate rules for each quality characteristic influencing the same process parameter in the IM, e.g. $r_{\eta_{q,p}}$ and $r_{\eta_{s,p}}$, with $q, s \in Q$ and $p \in P$. However, in this case it would not be clear whether the rules are related according to a logical *AND*, *OR*, or a different operator altogether. As such, we propose the introduction of a relator vertex representing the logical operator required by the r_{q^n} in question. For example an *AND-relator* as shown in Figure 9.4 this yields $r_{q^n} = \{(q, \langle\text{implies}\rangle, p), (s, \langle\text{implies}\rangle, p)\}_{\text{AND}}$, where $\{\}_{\text{AND}}$ denotes the set of all relations combined by the respective *AND-relator* l. This can be expanded to the following hierarchical triple:

$$r_{q^n} = \{((q, \langle\text{combined by}\rangle, l), \langle\text{implies}\rangle, p), ((s, \langle\text{combined by}\rangle, l), \langle\text{implies}\rangle, p)\}_{\text{AND}}$$

In theory, there is no limit to the number of relations being combined by a relator, the definitions here are based on the example in Figure 9.4.

The case of one condition influencing several conclusions can be represented by adding a separate rule for each of the conclusions, which proved sufficiently detailed for the case studies considered in this thesis. Therefore, this case does not require a special relator vertex.

9.2.5. Inclusion of Process Data

Noy et al. [Noy+19] make the point that 'it is critical not to lose the linkage between the relationships stored in the graph and where those relationships come from' [Noy+19]. While they refer to the discovery process, we assume that capturing semantics of operator knowledge in IMs could be aided by including process knowledge, especially since Ringsquandl et al. [Rin+18] have achieved promising results in knowledge graph completion by considering event embeddings. In manufacturing, orders Ω, which can be viewed as compositions of process iterations I, are produced for certain amounts of time. We propose to explicitly model this process data as $(i, \langle\text{belongs to}\rangle, \omega)$, where $i \in I$ and $\omega \in \Omega$. The process iterations can be connected with the resulting quantified parameters, $(\rho_i, \langle\text{chosen in}\rangle, i)$, where $\rho \in \text{P}$, quantified quality characteristics

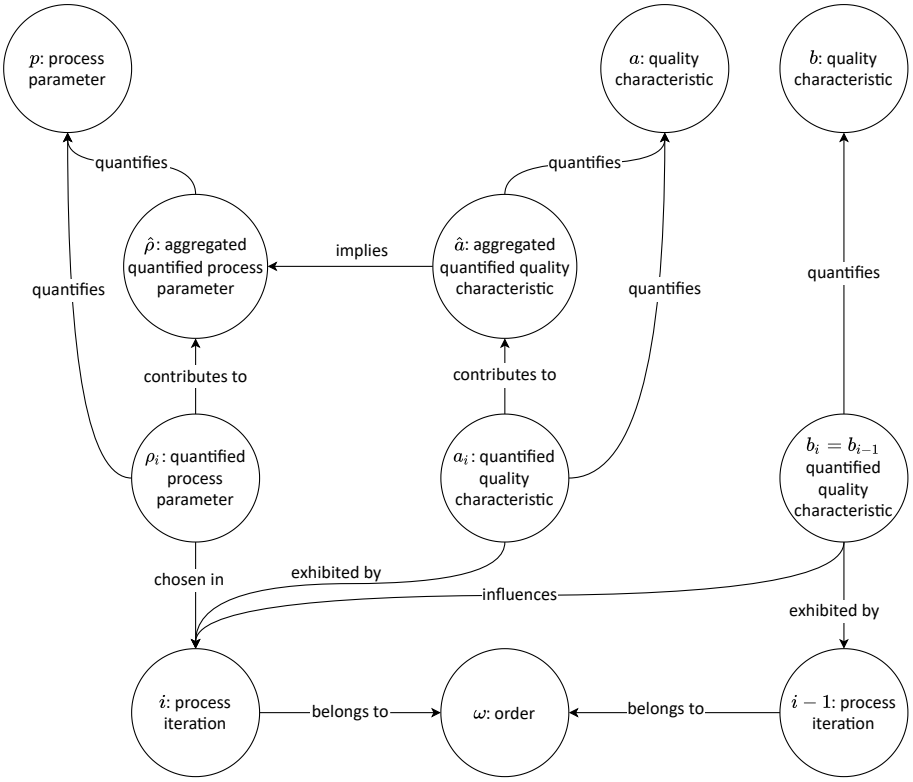


Figure 9.5.: Information Model for process data, i.e. parametrisation processes and their iterations, as well as a rule. The rule's condition and conclusion is quantified relying on aggregations of quantified process parameters and quality characteristics that result from the iteration processes.

$(a_i, \langle\text{is exhibited after}\rangle, i)$, and quantified influences $(b_{i-1}, \langle\text{influences}\rangle, i)$, where $a, b \in \mathrm{O}$ and $(b_{i-1}, \langle\text{is exhibited after}\rangle, i-1)$. P and O are defined in the modelling patterns for *quantified parameters* and *quantified conditions*, respectively. We note, that process iteration i is preceded by $i-1$ in order ω, i.e. the quality characteristic b_{i-1} is exhibited after the conclusion of the preceding process iteration.

Connecting the process data with the modelling patterns described above is beneficial, especially in the case of data-based knowledge extraction, since it allows explanation by example. The definition is analogous for parameters p. Using this, we can denote a relationship between process data and aggregate as $(a_i, \langle\text{contributes to}\rangle, \hat{a})$, where $\hat{a}$ is an aggregation of quantified quality characteristic expressed in a specific rule. Both, a_i and $\hat{a}$, are quantified quality characteristics and as such are equivalent in regards to the level of indirection. The only difference is that the value of a_i is sampled from the real world process whereas $\hat{a}$ is calculated based on a set of observations, e.g. through averaging them.

If all process data is included in the IM, a *supports* relation could also be defined following the quality metric used in rule mining [Ma+19; Ho+18] to denote which observed quantifications support which rule. However, since data-based rules rely on aggregates, a direct comparison between $\hat{q}_j$ and q_i, as well as $\hat{p}_j$ and p_i would fail and need to be relaxed by an interval in which they are considered equal. Also, since the rules are split between parameters and quality

characteristics there would have to be two separate relations. These support relations would increase the information content of the knowledge graph but would be aimed at a different use-case, i.e. validation of the quality of the rules (see Part II). However, since the following parts of this thesis are limited to knowledge graphs containing procedural knowledge, we omit these relations.

9.3. Properties of Representations

Here, we will give an overview of the properties shown by different representations. Firstly, with increasing expressiveness of and information contained in the representation its complexity is increasing. This increase of abstraction can be seen in the increase in hierarchy for each additional quantification or multiple conditions. Secondly, if process data is included in the IM it is likely to lead to a strong imbalance, since process data is much more readily available than extracted operator knowledge. Both aspects highlight the challenges embedding methodologies for IMs with operator knowledge have to address. As most of the relations described in Section 9.2 are not symmetric it would be easy to generate inverse relations, e.g. *implied by* for *implies*, which could be beneficial in knowledge graph completion settings.

In the following, we limit the analysis of the representations to r_η, $r_{\hat{\rho}}$ and $r_{\hat{\text{o}},\hat{\rho}}$, since multiple conditions and inclusions of process data do not introduce significant new complexities and $r_{\hat{\text{o}},\hat{\rho}}$ is sufficiently close to data encountered in practice in the scope of our case studies. An overview of properties exhibited by at least one of the representations can be seen in Table 9.1. While the properties composition, indirection and literals have been previously described, asymmetry directly results from the modelling decisions made, i.e. if $(p, q) \in \langle\text{implies}\rangle \implies (q, p) \notin \langle\text{implies}\rangle$. Since process parameters, quality characteristics and quantifications belong to different classes, all representations exhibit heterogeneous entities. Heterogeneous relations are exhibited by all representations apart from r_η that only includes the *implies* relation.

The properties of the knowledge graphs resulting from applying the representations used in the following sections to the FDM and synthetic dataset PDPK_ρ are shown in Table 9.2. Compared to other industrial knowledge graphs the number of edges, vertices and relations is relatively small [Buc+21; Noy+19]. While this could be addressed by adjusting the dimensions of the $P \times Q$ space accordingly, it is a characteristic of procedural knowledge that it is only available in limited quantities, especially compared to knowledge graphs containing information about products or production lines. Also, the amount of edges and vertices decrease for representations that explicitly represent values as literals since these (and the relations to them) are counted separately. In general, however, the more detailed a representation, the greater the amount of edges and vertices. The average closeness centrality of each vertex is given by

$$C(x) = \frac{|V| - 1}{\sum_y d(y, x)},$$

with the number of vertices denoted by $|V|$ and the distance d between x and y, is very small and further decreasing for the reification-based representations and the representation including conditions without high-level relations, $r_{\hat{\rho},\text{rei},\cdot}$ and $r_{\hat{\rho},\text{ch,e}}$, respectively. The average degree

Table 9.1.: Overview of properties exhibited by representations which need to be addressed by embedding methods. For the pattern underlying representation names refer to Section 9.2.

Representation	Asymmetry	Composition	Literals	Indirection	Arity	Heterogen.
r_{η}	✓	✗	✗	✗	2	✗
$r_{\hat{\rho},\text{rel}}$	✓	✗	✗	✗	2	✓
$r_{\hat{\rho},\text{ch,e}}$	✓	✗	✗	✓	3	✓
$r_{\hat{\rho},\text{ch,e},\eta}$	✓	✓	✗	✓	3	✓
$r_{\hat{\rho},\text{ch,l},\eta}$	✓	✓	✓	✓	3	✓
$r_{\hat{\rho},\text{rei,e}}$	✓	✗	✗	✓	3	✓
$r_{\hat{\rho},\text{rei,l}}$	✓	✗	✓	✓	3	✓
$r_{\hat{\text{o}},\hat{\rho},\text{ch,e}}$	✓	✗	✗	✓	5	✓
$r_{\hat{\text{o}},\hat{\rho},\text{ch,e},\eta}$	✓	✓	✗	✓	5	✓
$r_{\hat{\text{o}},\hat{\rho},\text{ch,l},\eta}$	✓	✓	✓	✓	5	✓
$r_{\hat{\text{o}},\hat{\rho},\text{rei,e}}$	✓	✗	✗	✓	5	✓
$r_{\hat{\text{o}},\hat{\rho},\text{rei,l}}$	✓	✗	✓	✓	5	✓

centrality for each vertex,

$$C_D(x) = \frac{|V| - 1}{\deg(x)},$$

with $\deg(x)$ the degree, i.e. the amount of in- and outgoing relations of x, is relatively low for all representations but further decreasing for representations $r_{\hat{\rho},\text{rei},\cdot}$ and $r_{\hat{\rho},\text{ch,e}}$. The average neighbour degree of each vertex, given by

$$\frac{1}{|N(i)|} \sum_{j \in N(i)} \deg(j),$$

where $N(i)$ are the neighbours of vertex i, is at its highest for $r_{\hat{\rho},\text{ch,e},\eta}$ which includes the superfluous η relation, which leads to a more interconnected graph with the degree of quality characteristics being higher which leads to the higher deviation. $r_{\hat{\rho},\text{rei},\cdot}$ show a relatively low average neighbour degree, especially on the FDM dataset. The average degree shows a similar behaviour although less pronounced.

Table 9.2.: Properties of the knowledge graphs resulting from applying the representations used in the following sections to the FDM and synthetic dataset PDPK$_\rho$. #E, #V, and #R are abbreviations for number of edges, vertices and relations.

DS.	Rep.	#E.	#V.	#R..	Close. Cent.	Deg. Cent.	Avg. Neigh. Deg.	Avg. Deg.
PDPK$_\rho$	r_η	48	42	1	0.03±0.03	0.06±0.04	3.16±2.73	2.29±1.55
	$r_{\hat{\rho},\text{rel}}$	48	42	48	0.03±0.03	0.06±0.04	3.16±2.73	2.29±1.55
	$r_{\hat{\rho},\text{ch,e}}$	96	90	2	0.01±0.01	0.02±0.01	1.67±0.97	2.13±1.06
	$r_{\hat{\rho},\text{ch,e},\eta}$	144	90	2	0.02±0.02	0.04±0.03	4.11±2.41	3.2±2.47
	$r_{\hat{\rho},\text{ch,l},\eta}$	48	42	1	0.03±0.03	0.06±0.04	3.16±2.73	2.29±1.55
	$r_{\hat{\rho},\text{rei,e}}$	144	138	3	0.01±0.01	0.02±0.01	1.83±0.47	2.09±1.2
	$r_{\hat{\rho},\text{rei,l}}$	96	90	2	0.01±0.01	0.02±0.01	1.67±0.97	2.13±1.06
FDM	r_η	34	27	1	0.05±0.06	0.1±0.1	5.92±4.34	2.52±2.64
	$r_{\hat{\rho},\text{rel}}$	34	27	30	0.05±0.06	0.1±0.1	5.92±4.34	2.52±2.64
	$r_{\hat{\rho},\text{ch,e}}$	68	57	2	0.02±0.03	0.04±0.03	2.46±1.72	2.28±1.78
	$r_{\hat{\rho},\text{ch,e},\eta}$	102	57	2	0.03±0.04	0.06±0.07	6.82±3.82	3.47±3.79
	$r_{\hat{\rho},\text{ch,l},\eta}$	34	27	1	0.05±0.06	0.1±0.1	5.92±4.34	2.52±2.64
	$r_{\hat{\rho},\text{rei,e}}$	102	91	3	0.02±0.01	0.02±0.02	2.18±0.73	2.24±1.66
	$r_{\hat{\rho},\text{rei,l}}$	68	61	2	0.02±0.02	0.04±0.03	2.51±1.86	2.23±1.75

10 Embedding of Knowledge Graphs Containing Procedural Knowledge

In this chapter, we present a methodology of how to embed knowledge graphs containing extracted operator knowledge that is available as rules into low dimensional vectors. Situated in the context of an envisioned predictive system (see Part IV), our approach aims to embed subgraphs relevant to a given problem. Our approach is threefold and consists of the following stages: Firstly, individual node embeddings are calculated by established embedding methods (see Section 10.1). Secondly, subgraphs relevant to the specific input are selected (see Section 10.2). Thirdly, the node embeddings pertaining to the subgraphs are aggregated as presented in Section 10.3. The overall approach is shown in Figure 10.1.

10.1. An Overview of Node Embedding Methods and Their Properties

Individual node embeddings are calculated by established knowledge graph embedding methods. Since Portisch, Heist and Paulheim [PHP22] concluded that embeddings for link prediction

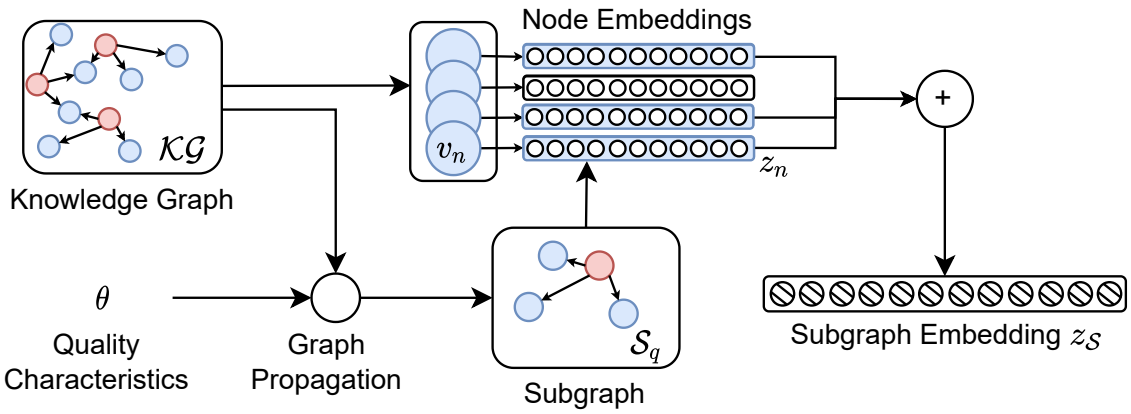


Figure 10.1.: *Illustration of the embedding methodology.* Node embeddings are calculated for the knowledge graph. The achieved qualities θ determine the head node (red) to propagate from. Node embeddings of nodes present in the resulting subgraph (highlighted in blue) are aggregated to one subgraph embedding $z_{\mathcal{S}}$. In this illustration, $z_{\mathcal{S}}$ corresponds to $z_{\mathcal{S}}^{\bar{h}}$ since the head node is not included.

Table 10.1.: Overview of properties which embedding methods address on a theoretical level. For the pattern underlying representation names refer to Section 9.2. All examined embedding methods are capable of representing heterogeneous relations which is why it was omitted from this table.

		Asymmetry	Composition	Literals	Indirection
Embedding	TransE	✓	✓	✗	✗
	ComplEx	✓	✗	✗	✗
	ComplEx-LiteralE	✓	✗	✓	✗
	DistMult	✗	✗	✗	✗
	DistMult-LiteralE$_g$	✗	✗	✓	✗
	RotatE	✓	✓	✗	✗
	BoxE	✓	✗	✗	✗
	RDF2Vec	✓	✓	✗	✓

are, to a certain extent, also suited for downstream tasks, we chose both, methods initially intended for embeddings for link prediction as well as downstream tasks in our investigation to determine which is suited best for embedding procedural knowledge. We decided on a set of knowledge graph embedding methods to investigate, namely TransE, ComplEx, ComplEx-LiteralE, DistMult-LiteralE$_g$, DistMult, RotatE, BoxE and RDF2Vec (see Section 8.3), which includes embedding methods of different complexity that satisfy one or more of the properties (see Table 9.1) exhibited by the representations described in Section 9.2. Table 10.1 shows an overview of which properties are addressed by the respective embedding method.

RDF2Vec seems to be the best suited method from a theoretical perspective. Due to its execution of random walks on the graph, it is likely able to capture relational properties [PP21]. Also, the indirections introduced by modelling the quantification as a chained-binary relations, a property with which other embedding methods struggle, can be embedded if the walks are executed to a sufficient depth. In addition, link prediction methods outperform RDF2Vec only on well curated datasets [PHP22]. Since industrial datasets are notoriously skewed, biased and incomplete this could be a further benefit on real world data.

Several works mention that the embedding methods' ability to capture composition in the graph could have a detrimental effect on the embedding quality [Zha+19a; Abb+20]. To investigate this in our scenario, we selected a number of embedding methods that encode composition and a number which do not.

Also, while there exists a broad spectrum of embedding methods that are able to embed entities, there are only relatively few which explicitly take literals into consideration [Ges+21]. As such, we added LiteralE [Kri+19] to our evaluation that builds on ComplEx and gated DistMult for which Gesese et al. [Ges+21] report good results on numeric literals. The other properties are shared with the respective underlying embedding method.

As we can see, there is not a single embedding method that is a perfect fit for the properties exhibited by the representations. Therefore, we rely on the experimental evaluation in Chapter 11 for indications of the importance of different properties.

Table 10.2.: Overview of propagation depth for each representation. For the pattern underlying representation names refer to Section 9.2.

	Representation	Propagation Depth (d)
	r_{η}	1
	$r_{\hat{\rho},\text{rel}}$	1
	$r_{\hat{\rho},\text{ch,e}}$	2
	$r_{\hat{\rho},\text{ch,e},\eta}$	2
	$r_{\hat{\rho},\text{ch,l},\eta}$	1
	$r_{\hat{\rho},\text{rei,e}}$	2
	$r_{\hat{\rho},\text{rei,l}}$	2
	$r_{\hat{\text{o}},\hat{\rho},\text{ch,e}}$	3
	$r_{\hat{\text{o}},\hat{\rho},\text{ch,e},\eta}$	3
	$r_{\hat{\text{o}},\hat{\rho},\text{ch,l},\eta}$	1
	$r_{\hat{\text{o}},\hat{\rho},\text{rei,e}}$	3
	$r_{\hat{\text{o}},\hat{\rho},\text{rei,l}}$	2

10.2. Selecting Relevant Subgraphs

To select the most relevant nodes that can be aggregated for a given input, a subgraph $\mathcal{S}_q$ is generated by propagating from the node corresponding to the input. In our case, the input consists of quantifications of quality characteristics achieved in the previous process iteration. Selecting the quality characteristic exhibiting the most pronounced error yields the node most strongly corresponding to the input. The depth until which the propagation is executed is limited by d, which is chosen depending on the respective representation. The propagation ignores edges direction and stops in parameter vertices. Combined with the precondition that the knowledge graph contains only procedural knowledge in the selected representations, this leads to an algorithm that dependably captures all relevant nodes and vertices of n-ary relations.

Table 10.2 shows an overview of the propagation depth for the respective representations. For unquantified conclusions, r_{η}, and quantified conclusions modelled through relations, $r_{\hat{\rho},\text{rel}}$, $d = 1$ as in both cases all relevant nodes are connected be either the high-level *implies* relation, η, or the respective relation for the quantification. For entity-based representations of quantified conclusions $r_{\hat{\rho},\text{ch,e}}$ and conditions $r_{\hat{\text{o}},\hat{\rho},\text{ch,e}}$ that model the ternary relation through chained-binary relations, i.e. a combination of two relations, $d = 2$. The same is true for the respective representations containing the additional high-level relation η. For the literal-based representation with chained-binary relations, $r_{\hat{\rho},\text{ch,l},\eta}$ and $r_{\hat{\text{o}},\hat{\rho},\text{ch,l},\eta}$, $d = 1$, since no node embedding for the literal is computed as it only influences the embedding of the nodes connected to it. Since this means that propagating from q to p is unfeasible for literal-based representations not containing the high-level relation η these representations are not suited for a downstream predictive setting and will not be further investigated.

For representations building on reification $d = 2$ for quantified conclusions, regardless of entity or literal-based quantifications. For entity-based quantified conclusions $r_{\hat{\text{o}},\hat{\rho},\text{rei,e}}$ $d = 3$, while for literal-based quantified conclusions $r_{\hat{\text{o}},\hat{\rho},\text{rei,l}}$ $d = 2$.

10.3. Sum-Based Embedding Aggregation

Individual node embeddings z of nodes v, with $v \in \mathcal{S}_q$, where $\mathcal{S}_q$ is the subgraph corresponding to the input θ, form the basis of the subgraph's embedding and are aggregated following a sum-based [HYL17b] approach. Our embedding aggregation approach (cf. Figure 10.1) is loosely based on the methodology outlined in Kursuncu, Gaur and Sheth [KGS20] that presented an approach towards embeddings for learning systems addressing classification in an natural language processing (NLP) setting, rather than KG completion or predictive quality scenarios. Main differences stem from a different subgraph structure that necessitates an adapted aggregation method. Since it is not certain in which case inclusion of the head node does provide additional semantic information and consequently whether it should be considered, we provide two alternatives. If the head node is not included, the subgraph embedding is aggregated as follows:

$$z_{\mathcal{S}}^{\bar{h}} = \sum_{(v_i,v_j)\in\mathcal{S}} z_j \otimes D(z_i, z_j),$$

where v_i, v_j are the pairs of head and tail nodes resulting from the graph propagation and $D(z_i, z_j)$ is a distance measure between the node embeddings of v_i and v_j. However, the head node z_0 can also be included in the subgraph embedding via:

$$z_{\mathcal{S}}^{h} = z_0 + \sum_{(v_i,v_j)\in\mathcal{S}} z_j \otimes D(z_i, z_j).$$

Due to the pattern-based nature of knowledge graphs in our scenario, we expect the relations to be the same independent of the starting node for most representations. Also, they have previously been considered by the knowledge graph embedding methods. Therefore, we do not consider the relations' embedding in the aggregation.

Since it is not clear whether the head node contains semantic information and should consequently be included in the computation of the subgraph embedding $z_{\mathcal{S}}$, we treat it as a parameter along with the distance measure and conduct an experiment in Chapter 11 to evaluate its effect.

11 Evaluation

While node embedding methods are mostly evaluated on link prediction tasks [PHP22], this is not necessarily indicative of their performance on downstream scenarios, since the task can be completely unrelated as in our case (see Section 13.1). Therefore, we introduce a surrogate metric for downstream scenarios, *matches@k*, and use it to evaluate embedding methods and aggregations that correspond to rules or specific parts of knowledge graphs. This allows us to get an understanding of the capability of our proposed approach to deal with the challenges posed by the selected representations of procedural knowledge (see Section 9.3).

11.1. Experimental Setup

We investigate a set of well established knowledge graph embedding methods, namely TransE [Bor+13], ComplEx [Tro+16], ComplEx-LiteralE, DistMult-LiteralE$_g$ [Kri+19], DistMult [Yan+14], RotatE [Sun+19], BoxE [Abb+20], as well as RDF2Vec [RP16], an embedding method optimised for downstream tasks. These methods are evaluated on selected knowledge graph representations (see Chapter 9) of rules forming the groundtruth of the FDM and synthetic PDPK$_\rho$ dataset (see Sections 3.1 and 3.3). In particular, we limit ourselves to analysing representations with quantified conclusions.

We chose to train 46-dimensional embeddings, since this dimension allows its direct use in our knowledge-guided architectures (see Part IV). The training was conducted using PyKEEN [Ali+21] with an Adam optimiser with learning rate 4×10^{-4} and weight decay 1×10^{-5} for the link prediction methods and pyRDF2Vec [Van+22] for RDF2Vec as well as the default parametrisations of the embedding methods. The number of epochs—TransE: 400, ComplEx: 1000, ComplEx-LiteralE: 650, RotatE: 700, DistMult: 800, DistMult-LiteralE$_g$: 200, BoxE: 1500 and RDF2Vec: 1000—was chosen individually for each embedding method by inspecting its convergence on the train set.

Null hypothesis significance tests (NHST) frequently used to assure significance of results are flawed for method comparison due to multiple reasons, e.g. the fact that with 'enough data arbitrarily small effects' can be classified as significant [Ben+17]. Therefore, we employ Bayesian performance analysis by using Calvo models [Cal+19; CCL18] implemented in the cmpbayes library[1] to arrive at statistically grounded conclusions that are of more practical relevance. Calvo models estimate the 'probability of each algorithm being the best along with a notion [...] of the uncertainty about this estimation' [Cal+19].

[1] https://github.com/dpaetzel/cmpbayes

11.1.1. Matches@k

Metrics commonly used to evaluate embeddings in knowledge graph completion settings, e.g. AMRI and *hits@k*, are unsuited to establish the quality of embeddings of operator knowledge in respect to their performance on downstream tasks. Instead, we propose that the fundamental behaviour of rule embeddings in predictive quality scenarios should be that the parameters adjusted for similar quality characteristics in the (sub-) graph, should be as equal as possible to those in the embedding space. Therefore, we define a metric in analogy to *hits@k*, *matches@k*, based on the amount of overlap between the k closest quality characteristics in embedding and graph space. To be able to establish the amount of overlap, a set of the k closest quality characteristics is prepared by ordering them descendingly according to their similarity—amount of overlapping parameters adjusted (higher is better) and Euclidean distance (lower is better) for graph and embedding space, respectively. Then, the number of matches between the respective sets is calculated for each quality characteristic. Divided by k it results in a value in $[0, 1]$, where higher values indicate a better result. By conducting the comparison on a set, we ensure that small differences in similarity are not unduly exaggerated in the overall metric since the order is not important to determine a match.

For *matches@k*, k has to be chosen according to the respective dataset. It decreases in expressiveness with increasing k since the unordered nature of the comparison leads to the number of matches equalling the number of quality characteristics $|Q|$ if $k = |Q|$, which would correspond to an overlap of $100\,\%$. Therefore, inspecting the actual similarities between quality characteristics in the graph space is necessary to determine the k which is representative for real world similarity. This can either be done by relying on domain knowledge or by determining the point at which the similarities are abruptly decreasing. In the following experiment, we use $k = 3$ since we established experimentally that for greater k the similarities between the quality characteristics are rapidly increasing on the FDM dataset due to the underlying distribution in occurring quality defects. $k = 3$ is also suitable for the synthetic dataset.

In the following, we use *matches@k* to answer the research questions outlined above on the FDM as well as synthetic (see Section 3.3.2) datasets.

11.2. Subgraph Embedding Aggregations—Similarity Between Graph and Embedding Space

Using *matches@k* as a surrogate of the performance on downstream scenarios, we answer the following research questions aimed at arriving at a conclusion which combination of representation (see Section 9.2), node embedding (see Section 10.1) and embedding aggregation (see Section 10.3) to use in Part IV:

- RQ 1: What is the impact of representing quantified values as literals or entities?
- RQ 2: How do different representation treating ternary relations compare against each other?
- RQ 3: Which embedding methods are able to deal with the indirections introduced through more detailed representations?

- RQ 4: Which combination of embedding method and embedding aggregations, i.e. head node inclusion or distance measure, performs best?

Of the experimental setup described in Section 11.1, 30 iterations were conducted on the FDM and synthetic dataset to mitigate effects that could occur due to the random initialisation. Since in the proposed neuro-symbolic system generalisation of the learnt embeddings to previously unseen knowledge is not required as domain knowledge is unlikely to change, *matches@k* is evaluated in-sample.

Table 11.1.: Subgraph Embedding Aggregation—Results for the FDM dataset as measured with *matches@3*. CE-LitE and DM-LitE$_g$ are abbreviations for ComplEx-LiteralE and DistMult-LiteralE$_g$, respectively. Best results per representation and aggregation method are highlighted in bold.

R.	H.	Dist	TransE	ComplEx	CE-LitE	RotatE	DistMult	DM-LitE$_g$	BoxE	RDF2Vec
r_{η}	h	euc	0.48±0.04	0.61±0.06	0.69±0.06	0.78±0.05	0.79±0.07	0.51±0.05	**0.81±0.07**	0.4±0.1
		jac	0.55±0.09	0.53±0.07	0.67±0.08	0.75±0.07	**0.78±0.08**	0.54±0.08	0.7±0.07	0.41±0.07
	$\bar{h}$	euc	0.52±0.03	0.5±0.04	0.55±0.06	0.61±0.05	0.73±0.09	0.52±0.05	0.72±0.06	**0.83±0.06**
		jac	0.53±0.05	0.51±0.04	0.56±0.07	0.63±0.06	0.69±0.07	0.53±0.07	**0.73±0.06**	0.5±0.03
$r_{\hat{\rho},\text{rel}}$	h	euc	0.48±0.06	0.62±0.07	0.56±0.06	0.56±0.06	0.53±0.08	0.5±0.08	**0.69±0.06**	0.52±0.1
		jac	0.55±0.07	0.51±0.07	0.51±0.06	**0.56±0.06**	0.56±0.08	0.54±0.06	0.5±0.08	0.51±0.1
	$\bar{h}$	euc	0.51±0.05	0.5±0.03	0.51±0.04	0.52±0.05	0.54±0.06	0.5±0.04	0.59±0.05	**0.69±0.08**
		jac	0.52±0.05	0.51±0.04	0.49±0.04	0.53±0.05	0.53±0.06	0.52±0.05	**0.63±0.04**	0.49±0.03
$r_{\hat{\rho},\text{ch,e}}$	h	euc	0.36±0.05	0.38±0.05	0.44±0.07	0.41±0.03	0.47±0.08	0.36±0.05	0.51±0.05	**0.66±0.04**
		jac	0.38±0.07	0.36±0.07	0.43±0.08	0.39±0.07	**0.49±0.08**	0.36±0.06	0.45±0.06	0.47±0.05
	$\bar{h}$	euc	0.38±0.04	0.34±0.04	0.43±0.05	0.4±0.04	0.47±0.07	0.36±0.05	0.49±0.06	**0.73±0.08**
		jac	0.4±0.05	0.34±0.04	0.42±0.06	0.38±0.07	0.5±0.06	0.35±0.07	0.49±0.07	**0.56±0.07**
$r_{\hat{\rho},\text{ch,e},\eta}$	h	euc	0.44±0.05	0.47±0.05	0.51±0.06	0.55±0.05	0.6±0.07	0.45±0.06	**0.67±0.06**	0.61±0.08
		jac	0.49±0.07	0.43±0.06	0.49±0.09	0.53±0.08	**0.62±0.09**	0.45±0.07	0.62±0.06	0.52±0.07
	$\bar{h}$	euc	0.45±0.04	0.43±0.04	0.55±0.06	0.52±0.06	0.61±0.06	0.46±0.05	**0.63±0.06**	0.58±0.08
		jac	0.47±0.05	0.42±0.06	0.53±0.08	0.53±0.06	**0.61±0.06**	0.46±0.06	0.61±0.06	0.55±0.08
$r_{\hat{\rho},\text{ch,l},\eta}$	h	euc			**0.72±0.08**			0.49±0.05		
		jac			**0.71±0.09**			0.56±0.07		
	$\bar{h}$	euc			**0.58±0.07**			0.51±0.04		
		jac			**0.57±0.08**			0.51±0.05		
$r_{\hat{\rho},\text{rel,e}}$	h	euc	0.3±0.06	0.28±0.05	0.4±0.1	0.27±0.05	0.37±0.07	0.3±0.06	0.43±0.07	**0.67±0.07**
		jac	0.32±0.07	0.29±0.05	0.4±0.09	0.3±0.07	0.4±0.08	0.32±0.06	**0.41±0.06**	0.37±0.05
	$\bar{h}$	euc	0.3±0.06	0.29±0.04	0.39±0.09	0.27±0.05	0.37±0.06	0.3±0.05	0.41±0.07	**0.7±0.05**
		jac	0.33±0.07	0.3±0.05	0.38±0.09	0.3±0.06	0.39±0.08	0.31±0.06	**0.45±0.05**	0.43±0.07
$r_{\hat{\rho},\text{rel,l}}$	h	euc			**0.45±0.08**			0.35±0.06		
		jac			**0.45±0.09**			0.37±0.06		
	$\bar{h}$	euc			**0.44±0.08**			0.36±0.06		
		jac			**0.43±0.08**			0.37±0.05		

The performance of the embedding aggregations for rules with quantified conclusions extracted from the ground truth of the the FDM dataset are shown in Table 11.1 while those for the synthetic dataset are shown in Table 11.2. Here, embedding aggregations are evaluated with or without included head nodes, h and $\bar{h}$, respectively, and for Euclidean (euc) and Jaccard (jac) distances for each representation. In the following, we address the research questions based on these results.

Table 11.2.: Subgraph Embedding Aggregation—Results, for the synthetic dataset, $PDPK_\rho$, as measured with *matches@3*. CE-LitE and DM-LitE_g are abbreviations for ComplEx-LiteralE and DistMult-LiteralE_g, respectively. Best results per representation and aggregation method are highlighted in bold.

R.	H.	Dist	TransE	ComplEx	CE-LitE	RotatE	DistMult	DM-LitE_g	BoxE	RDF2Vec
r_η	h	euc	0.36±0.05	0.4±0.05	0.48±0.06	0.46±0.03	0.5±0.04	0.36±0.06	**0.51±0.04**	0.37±0.07
		jac	0.37±0.06	0.37±0.05	0.46±0.06	0.46±0.05	**0.49±0.06**	0.37±0.07	0.49±0.06	0.36±0.07
	$\bar{h}$	euc	0.34±0.04	0.31±0.04	0.42±0.05	0.38±0.04	**0.47±0.04**	0.31±0.05	0.47±0.05	0.38±0.05
		jac	0.31±0.05	0.32±0.05	0.4±0.05	0.39±0.05	0.45±0.05	0.31±0.06	**0.48±0.04**	0.3±0.04
$r_{\hat{\rho},\text{rel}}$	h	euc	0.33±0.04	0.4±0.04	0.45±0.06	0.33±0.05	0.34±0.05	0.33±0.05	**0.49±0.05**	0.29±0.05
		jac	0.36±0.06	0.36±0.05	0.38±0.05	0.33±0.06	0.36±0.06	0.34±0.07	**0.42±0.07**	0.29±0.05
	$\bar{h}$	euc	0.3±0.04	0.31±0.03	0.38±0.06	0.28±0.04	0.3±0.04	0.32±0.05	**0.43±0.05**	0.36±0.08
		jac	0.31±0.05	0.31±0.04	0.34±0.05	0.29±0.04	0.31±0.05	0.3±0.05	**0.46±0.05**	0.3±0.04
$r_{\hat{\rho},\text{ch,e}}$	h	euc	0.19±0.05	0.2±0.05	0.28±0.08	0.26±0.06	0.32±0.05	0.19±0.04	0.33±0.06	**0.4±0.05**
		jac	0.24±0.06	0.2±0.05	0.28±0.07	0.27±0.06	**0.32±0.07**	0.23±0.06	0.29±0.05	0.28±0.06
	$\bar{h}$	euc	0.2±0.04	0.17±0.04	0.26±0.07	0.21±0.06	0.29±0.06	0.19±0.04	0.29±0.06	**0.4±0.05**
		jac	0.22±0.05	0.17±0.03	0.25±0.06	0.24±0.06	**0.29±0.06**	0.2±0.05	0.29±0.06	**0.29±0.05**
$r_{\hat{\rho},\text{ch,e},\eta}$	h	euc	0.31±0.06	0.31±0.04	0.38±0.07	0.36±0.05	0.45±0.05	0.3±0.05	0.43±0.04	**0.51±0.06**
		jac	0.32±0.06	0.29±0.06	0.36±0.07	0.39±0.06	0.43±0.06	0.33±0.06	**0.44±0.06**	0.36±0.06
	$\bar{h}$	euc	0.28±0.05	0.26±0.05	0.34±0.07	0.31±0.04	0.41±0.04	0.28±0.05	0.41±0.04	**0.51±0.06**
		jac	0.27±0.05	0.25±0.06	0.32±0.07	0.33±0.06	0.4±0.07	0.28±0.05	**0.42±0.05**	0.37±0.06
$r_{\hat{\rho},\text{ch,l},\eta}$	h	euc			**0.45±0.05**			0.3±0.06		
		jac			**0.44±0.06**			0.35±0.07		
	$\bar{h}$	euc			**0.4±0.05**			0.32±0.05		
		jac			**0.38±0.05**			0.3±0.07		
$r_{\hat{\rho},\text{rei,e}}$	h	euc	0.16±0.04	0.13±0.04	0.21±0.07	0.16±0.05	0.23±0.06	0.15±0.05	0.25±0.05	**0.32±0.05**
		jac	0.17±0.05	0.14±0.04	0.2±0.06	0.18±0.06	**0.24±0.06**	0.17±0.07	0.22±0.04	0.23±0.04
	$\bar{h}$	euc	0.17±0.04	0.12±0.03	0.2±0.05	0.15±0.04	0.23±0.06	0.15±0.04	0.23±0.05	**0.31±0.05**
		jac	0.17±0.05	0.13±0.04	0.19±0.05	0.17±0.04	0.23±0.06	0.16±0.07	0.23±0.05	**0.25±0.05**
$r_{\hat{\rho},\text{rei,l}}$	h	euc			**0.28±0.08**			0.18±0.04		
		jac			**0.27±0.08**			0.23±0.05		
	$\bar{h}$	euc			**0.25±0.09**			0.21±0.05		
		jac			**0.25±0.08**			0.21±0.04		

11.2.1. Impact of Literals

While the explicit representation of quantified values as literals allows a richer semantic expressiveness it severely limits the choice of embedding methods that can be used to calculate the node embeddings. Also, since it is a relatively new area of research and not yet widely adopted [Ges+21], it is questionable if the literal enabled outperform traditional embedding methods. Since the literal enabled methods proposed in Gesese et al. [Ges+21] are based on older embedding methods, which could be outperformed by newer ones we hypothesise that representing quantified values as literals does not outperform entity-based representations.

To validate this hypothesis, we analyse the difference between the best overall result achieved for entity-based representations, i.e. $r_{\hat{\rho},\text{ch,e},\eta}$ and $r_{\hat{\rho},\text{rei,e}}$, that are, apart from the quantified conclusion representation, equivalent to the corresponding literal containing representation, i.e. $r_{\hat{\rho},\text{ch,l},\eta}$ and $r_{\hat{\rho},\text{rei,l}}$. For the synthetic dataset, ComplEx-LiteralE (h, euc) outperforms DistMult-LiteralE_g for

$r_{\hat{\rho},\text{ch,l},\eta}$ and $r_{\hat{\rho},\text{rei,l}}$. The best results for $r_{\hat{\rho},\text{ch,e},\eta}$ and $r_{\hat{\rho},\text{rei,e}}$ are achieved by RDF2Vec by including the head node and using Euclidean distance. For both, the chained-binary and the reification-based representations, RDF2Vec (h, euc) outperforms ComplEx-LiteralE (h, euc) by 13.33% and 14.29%, respectively. For the FDM dataset, DistMult-LiteralE$_g$ is again outperformed by ComplEx-LiteralE (h, euc) in both cases. The best results for $r_{\hat{\rho},\text{ch,e},\eta}$ and $r_{\hat{\rho},\text{rei,e}}$, however, are achieved by BoxE (h, euc) and RDF2Vec ($\bar{h}$, euc), respectively. For the chained-binary-based representations, the results achieved by BoxE are 6.94% beneath those of Complex-LiteralE (0.67 vs. 0.72). In contrast, for the reification-based representations, RDF2Vec outperforms ComplEx-LiteralE by 55.56% (0.7 vs. 0.45).

To answer RQ1, we observe that overall the results seem to indicate that representing parameter quantifications as entities is beneficial in 75% of the analysed cases. However, apart from RDF2Vec on $r_{\hat{\rho},\text{rei,e}}$ the benefits are marginal if standard deviations are taken into account. For $r_{\hat{\rho},\text{ch},\cdot}$ on FDM, where the literal representation performed best, the benefits if standard deviations are taken into account are also marginal. This leads us to conclude that we do not have a clear result but rather a tendency towards using entity-based representations. Consequently, we will use entity-based representations since the respective embedding methods are more established.

11.2.2. Representing Ternary Relations

We expect that ternary relations introduce complexity which leads to worse results when compared to high-level binary relations. With $r_{\hat{\rho},\text{rel}}$, $r_{\hat{\rho},ch,\cdot}$ and $r_{\hat{\rho},\text{rei},\cdot}$ we investigate three alternative methodologies that represent ternary relations. While reifications based representations, $r_{\hat{\rho},\text{rei},\cdot}$, are frequently used in literature, they introduce additional auxiliary entities that do not include semantic information and consequently increase the complexity of the resulting knowledge graph. A possible drawback of $r_{\hat{\rho},\text{rel}}$ is the weighted relation that is unlikely to positively influence weight agnostic embedding methods. This leads us to the hypothesis, that $r_{\hat{\rho},\text{rel}}$ performs worst followed by $r_{\hat{\rho},\text{rei},\cdot}$. We expect the lower amount of vertices to allow representations based on chained-binary relations, $r_{\hat{\rho},ch,\cdot}$, to perform best.

To answer RQ2, we compare the respective best embedding method on the best variant of relation, chained-binary and reification-based representations. For the synthetic dataset BoxE (h, euc) achieves the best results for $r_{\hat{\rho},\text{rel}}$, RDF2Vec ($h/\bar{h}$, euc) for $r_{\hat{\rho},\text{ch,e},\eta}$ and RDF2Vec (h, euc) for $r_{\hat{\rho},\text{rei,e}}$. Of these, the best results are achieved on $r_{\hat{\rho},\text{ch,e},\eta}$, which outperforms those achieved on $r_{\hat{\rho},\text{rel}}$ and $r_{\hat{\rho},\text{rei,e}}$ by 4.08% (0.02 absolute) and 59.38% (0.19 absolute) respectively. Moreover, the results achieved on $r_{\hat{\rho},\text{rei,e}}$ are 34.70% (0.17 absolute) worse than those on $r_{\hat{\rho},\text{rel}}$.

For the FDM dataset, BoxE (h, euc) achieves the best results for $r_{\hat{\rho},\text{rel}}$, RDF2Vec ($\bar{h}$, euc) for $r_{\hat{\rho},\text{ch,e}}$ and RDF2Vec ($\bar{h}$, euc) for $r_{\hat{\rho},\text{rei,e}}$. Compared to the results on the synthetic datasets, the differences are smaller with $r_{\hat{\rho},\text{ch,e}}$ and $r_{\hat{\rho},\text{rei,e}}$ outperforming $r_{\hat{\rho},\text{rel}}$ by 0.04 (5.80%) and 0.01 (1.45%), respectively. Consequently, the best results are achieved on $r_{\hat{\rho},\text{ch,e}}$ which outperforms $r_{\hat{\rho},\text{rei,e}}$ by 4.29% (0.03 absolute).

In contrast to our hypothesis, $r_{\hat{\rho},\text{rei},\cdot}$ is not the clear second best representation with $r_{\hat{\rho},\text{rel}}$ performing significantly better on the synthetic dataset. Considering that for the FDM dataset the best results on $r_{\hat{\rho},\text{rei},\cdot}$ were achieved by RDF2Vec by a relatively large margin over the other embedding methods, we assume that its overall performance is worse than is visible by taking only the respective best method into consideration, which reinforces this observation. In contrast to this, surprisingly good results were achieved on $r_{\hat{\rho},\text{rel}}$, which indicates that the numerous relations introduced do not necessarily reflect negative in *matches@k*. While the overall differences between

the respective representations are rather small when standard deviations are considered, we can view both $r_{\hat{\rho},ch,\cdot}$ and $r_{\hat{\rho},\text{rel}}$ as good candidates. Taking additional characteristics, e.g. semantic expressiveness and conciseness, into account we conclude that the methods perform best on a chained-binary representation, $r_{\hat{\rho},ch,\cdot}$, on both datasets, confirming the respective part of our hypothesis.

11.2.3. Dealing With Indirections

Reification-based representations, $r_{\hat{\rho},\text{rei,e}}$, as well as the representation containing only chained-binary relations, $r_{\hat{\rho},\text{ch,e}}$, introduce an indirection between head node and the respective quantified values of rules. We assume, that this is an hindrance to achieving embeddings of good quality for the majority of embedding methods. The exception being RDF2Vec which flattens the nodes and edges via random walks before embedding them. Also, we assume that representations with indirections perform worse than representations representing the quantification as a separate relation, $r_{\hat{\rho},\text{rel}}$, or representations adding the relation η, $r_{\hat{\rho},\text{ch,e},\eta}$.

To validate the hypothesis regarding the effect of RDF2Vec, we compare the best result achieved for RDF2Vec to the best result achieved using other embedding methods, on $r_{\hat{\rho},\text{ch,e}}$ and $r_{\hat{\rho},\text{rei,e}}$. On the synthetic dataset and $r_{\hat{\rho},\text{ch,e}}$, the performance of the second best embedding method, BoxE (h, euc), is 17.5% (0.07 absolute) beneath that of RDF2Vec (h, euc). On $r_{\hat{\rho},\text{rei,e}}$, the performance of BoxE ($\bar{h}$, euc) is 21.88% (0.07 absolute) beneath that of RDF2Vec ($\bar{h}$, euc). On the FDM dataset and $r_{\hat{\rho},\text{ch,e}}$, the performance of the second best embedding method, BoxE (h, euc) is 30.14% (0.22 absolute) beneath that of RDF2Vec ($\bar{h}$, euc). On $r_{\hat{\rho},\text{rei,e}}$, the performance of BoxE ($\bar{h}$, jac) is 35.71% (0.25 absolute) beneath that of RDF2Vec ($\bar{h}$, euc). This confirms our hypothesis, that RDF2Vec is better suited to deal with indirections.

To quantify the effect of indirections on embedding methods apart from RDF2Vec, we compare results for the respective representations, i.e. $r_{\hat{\rho},\text{ch,e}}$ and $r_{\hat{\rho},\text{rei,e}}$ to $r_{\hat{\rho},\text{rel}}$ and $r_{\hat{\rho},\text{ch,e},\eta}$. For the FDM dataset, BoxE (h, euc) on $r_{\hat{\rho},\text{ch,e}}$ is outperformed by BoxE (h, euc) on $r_{\hat{\rho},\text{rel}}$ and BoxE (h, euc) $r_{\hat{\rho},\text{ch,e},\eta}$ by 35.30% (0.18 absolute) and 31.37% (0.16 absolute), respectively. BoxE ($\bar{h}$, jac) on $r_{\hat{\rho},\text{rei,e}}$ is outperformed by $r_{\hat{\rho},\text{rel}}$ and $r_{\hat{\rho},\text{ch,e},\eta}$ by 53.33% (0.24 absolute) and 48.89% (0.22 absolute). For the synthetic dataset, BoxE (h, euc) on $r_{\hat{\rho},\text{ch,e}}$ is outperformed by BoxE (h, euc) on $r_{\hat{\rho},\text{rel}}$ and DistMult (h, euc) on $r_{\hat{\rho},\text{ch,e},\eta}$ by 48.48% (0.16 absolute) and 36.36% (0.11 absolute), respectively. BoxE (h, euc) on $r_{\hat{\rho},\text{rei,e}}$ is outperformed by $r_{\hat{\rho},\text{rel}}$ and $r_{\hat{\rho},\text{ch,e},\eta}$ by 96.00% (0.24 absolute) and 80.00% (0.20 absolute).

To summarise our findings and answer RQ3, we conclude that RDF2Vec is the best embedding method if representations with indirections are chosen. For the other investigated embedding methods, the performance deficit is significant when compared to other representations with the same information content but mitigated or non-existent indirections. This deficit can be quantified by an average of 65.21% $\pm$ 23.86% and 42.22% $\pm$ 3.13% on the synthetic and FDM datasets, respectively.

11.2.4. Best Overall Combination of Embedding Method and Aggregation

Based on the theoretical comparison of embedding methods' capabilities to the needs of procedural knowledge, we expect RDF2Vec to be the best embedding method for representations containing quantified values. Also, we expect that including head nodes in the aggregation provides a stronger sense of context which is beneficial for the overall results.

To arrive at the overall best combination of embedding method and aggregation, we compare the best results per representation. For the synthetic dataset, RDF2Vec performs best in ∼43%, i.e. on $r_{\hat{\rho},ch,e}$, $r_{\hat{\rho},ch,e,\eta}$ and $r_{\hat{\rho},\text{rei,e}}$, whereas BoxE and ComplEx-LiteralE perform best in ∼28.5% each, i.e. on r_η and $r_{\hat{\rho},\text{rel}}$ as well as $r_{\hat{\rho},\text{ch,l},\eta}$ and $r_{\hat{\rho},\text{rei,l}}$. For each representation, the best results are achieved using Euclidean distance. Including the head node achieves better results in ∼71% of the cases, with the remaining ∼29% a draw. Both aggregation options, however, are dependent on the embedding method. E.g. while not including the head node is beneficial in only 40% of the cases for RDF2Vec, it achieves better or equal results in all cases for the other embedding methods. The Euclidean distance performs better or equal than Jaccard in 100% of cases for RDF2Vec, 60% for BoxE but only 20% for RotatE.

For the FDM dataset, RDF2Vec performs best in ∼43%, i.e. on r_η, $r_{\hat{\rho},ch,e}$ and $r_{\hat{\rho},\text{rei,e}}$, while BoxE achieves the best performance in ∼14% on $r_{\hat{\rho},\text{ch,e},\eta}$, with a draw, albeit with smaller variance for BoxE, with RDF2Vec on $r_{\hat{\rho},\text{rel}}$. As on the synthetic dataset, ComplEx-LiteralE performs best on the representations containing literals. Again, the best distance measure per representation is Euclidean in most of the cases, with a draw on $r_{\hat{\rho},\text{rei,l}}$. However, regarding head node inclusion the results are more indecisive with included head node, h, outperforming the alternative, $\bar{h}$, in ∼43% of the cases, i.e. $r_{\hat{\rho},\text{ch,l},\eta}$, $r_{\hat{\rho},\text{ch,e},\eta}$, and $r_{\hat{\rho},\text{rei,l}}$, and a draw on $r_{\hat{\rho},\text{rel}}$. Again, the results differ between embeddings, with head node inclusion performing best for RDF2Vec in only 20% of the cases as opposed to 80% and ∼86% for BoxE and ComplEx-LiteralE, respectively.

We answer RQ4 by concluding that the theoretical analysis of the embedding methods' properties is reflected in the results with RDF2Vec being one of the two best performing methods on both datasets. While for the best performing embedding methods Euclidean seems to be a good overall distance measure, the assumption that head node inclusion is beneficial in all cases, does not to hold. Nevertheless, a tendency towards the benefits of including head nodes can be observed. On both datasets, however, the results differ by relatively small margins if standard error is taken into consideration. This implies, that results can only serve as an indication.

In the following, we use $r_{\hat{\rho},\text{ch,e},\eta}$, since it is a representation that includes quantified parameters, i.e. the minimal abstraction level of knowledge present in both datasets. We actively decide against using r_η, which achieves very competitive results, since it does not contain quantified parameters and is therefore easier to embed. Also, it contains less information that can be used by the neuro-symbolic learning system proposed in Part IV. The selected representation, $r_{\hat{\rho},\text{ch,e},\eta}$, does not exhibit indirections, which should be beneficial from theoretical and empirical considerations. To inform our decision, which combination of embedding and aggregation method to use, we investigate the model for method comparison proposed by Calvo et al. [CCL18; Cal+19]—that yields the probability of the combination performing best—for $r_{\hat{\rho},\text{ch,e},\eta}$ of the FDM and PDPK$_\rho$ dataset. On FDM, BoxE (h, euc) performs best in 17.98% of cases (see Figure 11.2). While this is not a probability that lets us conclude that it is the clear best, it increases to 44.69% if all aggregation methods for BoxE are combined. This indicates that BoxE is the embedding method that achieves the best. On PDPK$_\rho$, RDF2Vec ($\bar{h}$, euc) performs best in 22.66% of cases (see Figure 11.1). If the probabilities of all aggregation methods for RDF2Vec are combined, it performs best in 44.75% of cases. While RDF2Vec with RDF2Vec ($\bar{h}$, euc) achieved slightly better results than RDF2Vec (h, euc), we will use the latter in the following experiments to use a common aggregation method. This decision is guided by the overall slightly better performance of including the head node h. Also, it increases the amount of information available in the subgraphs.

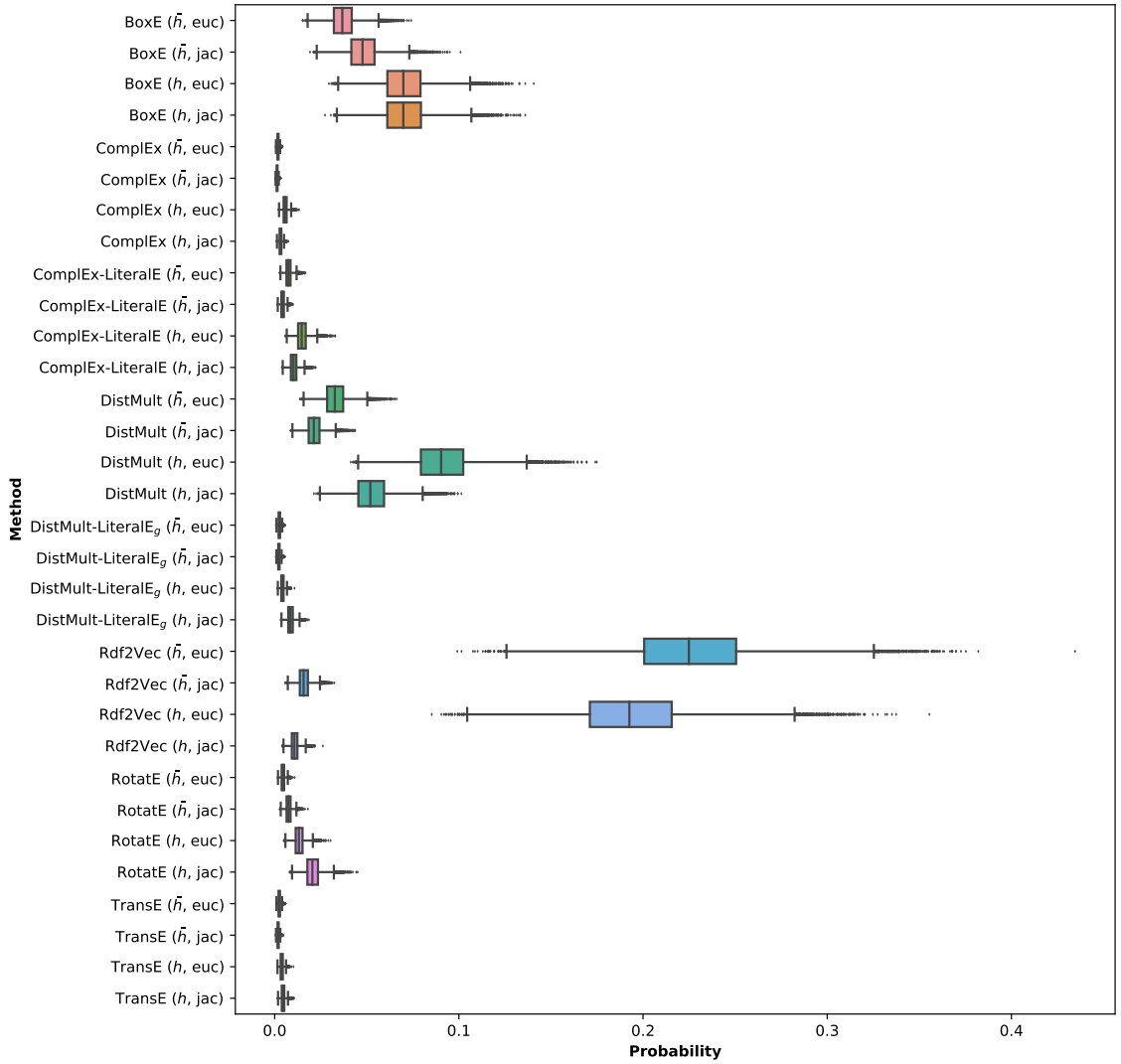


Figure 11.1.: Boxplot of probabilities for the best embedding method and aggregation combination for $r_{\hat{\rho},\mathrm{ch},\mathrm{e},\eta}$ on the synthetic dataset PDPK_{ρ}.

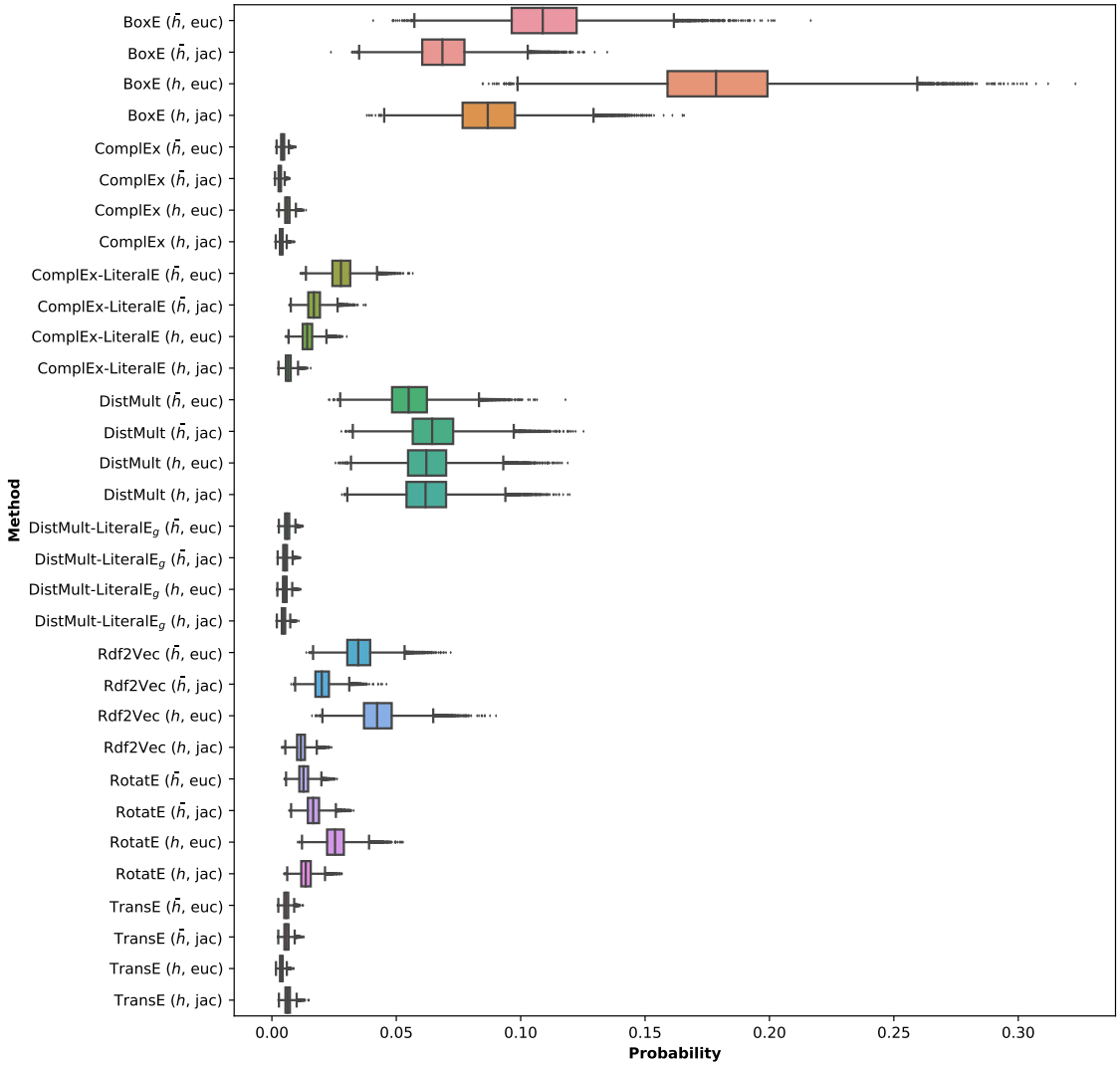


Figure 11.2.: Boxplot of probabilities for the best embedding method and aggregation combination for $r_{\hat{\rho},\mathrm{ch,e},\eta}$ on the FDM dataset.

Part IV.

Neuro-Symbolic Learning for Parameter Prediction

A prerequisite to assistance systems in manufacturing are predictions of sufficient quality. Due to several challenges, e.g. low-amount of data, skewed data and high generalisability, these predictions are not easily achieved. Neuro-symbolic approaches that infuse expert knowledge into neural learning systems, promise to improve the predictions. After providing an overview of the state of the art of combining symbolic knowledge with learning systems in Chapter 12, we present two neuro-symbolic architectures (see Chapter 13) and investigate whether these can utilise the procedural expert knowledge and the corresponding embeddings as presented in Parts II and III to achieve an improvement over a neural baseline (see Chapter 14).

12
Related Work

In this section, we present the state of the art of neuro-symbolic and knowledge-guided learning systems. Also, we highlight work of these fields that is related to manufacturing.

12.1. Neuro-Symbolic Learning: Knowledge, Reasoning & Learning Systems

While, historically, the first wave of AI systems relied heavily on expert systems [ST21], the second wave viewed the 'mind as an emergent property of the brain' [Bes+17] and consequently favoured connectionist methods, i.e. statistical machine learning and deep learning, specifically. So far, however, obstacles such as adaptability, generalisability, common sense and causal reasoning have not been solved relying solely on connectionist AI. Therefore, the community searches for inspiration in cognitive theories that explain human decision making processes [Boo+21], as evidenced by an increased number of publication in top tier conferences since 2017 [Sar+21]. Kahneman formulates one such theory, that divides human decision making into two systems, System 1—fast thinking—that arrives at decisions intuitively and unconsciously, and System 2—slow thinking—that comprises logical and rational thinking to handle more complex situations [Kah11]. Neuro-symbolic learning systems, i.e. the integration of 'well-founded knowledge representation and reasoning [...] with deep learning' [dL23], exhibit the same characteristics, with neural networks focussing on perception being equated with System 1, while reasoning, that 'supports human-like cognition using knowledge structures (e.g., knowledge graphs)' [SRG23] represents System 2 [dL23]. As such, they are viewed as the third wave of AI that is supposed to address the 'increasingly prominent role of trust, safety, interpretability and accountability' [dL23], e.g. via 'enabling traceability and auditing' [SRG23] by mapping perception to knowledge. Further goals are increased out of distribution handling and thereby better generalisability [Sar+21], which has been identified as a shortcoming of current neural networks [Del+22], the capability to solve more complex problems and 'explainability through background knowledge' [Hit+22]. Additionally, neuro-symbolic learning promises increased test performance and the capability to operate on smaller training sets [Bre+23; Sar+21]. While, traditionally, the symbolic part has been defined as 'sound symbolic reasoning' [dAv+15], it is now interpreted more loosely to include, for example, background knowledge in the form of knowledge graph embeddings [Sar+21; Bre+23; Hit+22; SRG23].

Several approaches towards categorising neuro-symbolic systems have been proposed. One of the first by Bader and Hitzler [BH05], which was recently reused in Sarker et al. [Sar+21], consists of eight dimensions grouped into the aspects of interrelation, language and usage. In the following, we reproduce Sarker et al. [Sar+21]'s summary. Firstly, the dimensions describing *interrelation* are:

- *integrated vs. hybrid*, i.e. whether the neuro- and symbolic parts of the system are tightly integrated
- *neuronal vs. connectionist*, i.e. whether the approach mimics biological neuronal networks or the connectionist networks common during the second wave of AI
- *local vs. distributed*, whether relatively few dedicated neurons process the symbolic information or whether it is combined in a highly distributed way
- *standard vs. non-standard*, i.e. whether widely used architectures are used.

Secondly, *language* is described by the dimensions *symbolic vs. logical*, that describe whether the symbolic part of the system is based on structured datatypes, e.g. knowledge graphs or formal logic, and *propositional vs. first-order*, that further defines the logic dimension. Lastly, *usage* is described by *extraction vs. representation*, i.e. whether symbolic information is extracted from a neural network or whether the information flow relies on neural representation of symbolic knowledge, and *learning vs. reasoning*, i.e. whether the overall system focusses on the learning or reasoning part. Our approaches presented in Chapter 13, can be classified as loosely integrated, since learnt information is not fed back to the symbolic part, connectionist, distributed, standard, symbolic, since we rely on knowledge graphs, representation and learning. This follows the trend of current interest in the field as described by Sarker et al. [Sar+21] with the deviation that most reviewed works focus on reasoning.

An orthogonal classification that is frequently used [Sar+21; dL23; Lam+20] was presented by Henry Kautz in the AAAI 2020 Robert S. Engelmore Memorial Award Lecture[1] and refines the interrelation part of the previous classification. It consists of five categories:

1. [symbolic Neuro symbolic], i.e. connectionist deep learning where in- and output are considered symbols
2. [Symbolic[Neuro]], i.e. neural networks for pattern recognition within a symbolic reasoner (e.g. [Sil+16])
3. [Neuro $\cup$ compile(Symbolic)], i.e. 'symbolic knowledge is compiled into the training set of a neural network' [dL23] (e.g. [LC19; ASA18])
4. [Neuro $\rightarrow$ Symbolic], i.e. the outputs of a neural network are cascaded into a symbolic reasoner (e.g. [Mao+19; Man+18])
5. [Neuro[Symbolic]], which corresponds to Kahneman's two systems. In particular, symbolic reasoning within a neural network communicating via decoding concepts to symbolic entities in an attention schema, i.e. a model of the system's attention [Gra19]. As soon as a goal has been identified in the attention schema the reasoner is initiated.

d'Avila Garcez and Lamb [dL23] extend [Neuro $\cup$ compile(Symbolic)] of this classification by including symbolic information 'translated into the initial architecture [(e.g. [Tow93])] and [...] weights of a neural network' [dL23] as well as Logical Neural Networks that create 1-to-1 relations between neurons and logical formulas (e.g. [Rie+20]). Furthermore, they introduce an additional category, Type 5, in which a 'symbolic logic rule is mapped onto an embedding which acts as

[1] https://youtu.be/_cQITY0SPiw?t=1723

a soft-constraint (a regularisation term) on the network's loss function' [dL23] (e.g. via Logic Tensor Networks [SG16]). The architecture, based on compositional combination of black boxes of symbolic and neural modules presented in Tsamoura, Hospedales and Michael [THM21], could be categorised as [Neuro[Symbolic]] due to the high amount of flexibility with which reasoning can be combined with learning. However, their approach does not rely on attention mechanisms. While our approach falls into Type 5 of this taxonomy, the notion of using attention schemas as present in [Neuro[Symbolic]] is related to our task specific knowledge selection mechanism presented in Section 10.2. However, in our case the attention is not learnt but rather encoded knowledge about the schemas underlying the knowledge graph that is used to select fitting domain knowledge.

Sheth, Roy and Gaur [SRG23] provide another categorisation, the main categories consisting of (1) the compression of symbolic knowledge and its integration with neural networks where the reasoning happens implicitly in the neural network and (2) approaches that extract information from neural patterns to allow for subsequent symbolic reasoning. Consequently, it contains [Symbolic[Neuro]], [Neuro $\rightarrow$ Symbolic] and [Neuro[Symbolic]] of Kautz's categorisation. The first category is further divided based on the information compressed into knowledge graph representations, (1a), and formal logic-based representations, (1b). The second category is further divided based on the degree of integration between the neuro- and symbolic parts into decoupled, (2a), and intertwined, (2b), systems. Comparing compressed knowledge graphs and compressed formal logics, they note that the first are at an advantage in regards to scalability and continuous learning after the model is running in production. Conversely, they observe that explainability of the second category is higher than of the first. Depending on the specific architecture our approach can be categorised into 1a or 1b.

The combination of symbolic approaches with machine learning has also been reviewed with a specific focus on semantic web technologies [Bre+23]. They extend the boxology presented in van Harmelen and Teije [vT19] that describes the information flow, i.e. the integration, between symbolic and machine learning components based on patterns, e.g. atomic patterns, i.e. one algorithmic module consuming one input, or fusion patterns, i.e. one algorithmic module consuming multiple inputs. While, similarly to our approach, a majority of reviewed works use knowledge graphs as semantic resources, they reported that, in contrast to our approach, two thirds of the reviewed papers have a symbolic output indicating an unrelated use case. Similarly to our approach, however, a majority of approaches consists of only one fusion of symbolic and machine learning approaches. They conclude that the combination of machine learning and semantic web technologies is still an emerging field based on the observation that 'only a small amount of systems that qualified for medium (4.8%) and high (1.7%) maturity' [Bre+23] with the rest 'describing proof of concepts and initial prototypes' [Bre+23]. Interestingly, in this context, improving performance for low amounts of data is not part of the goals [dAm20]. To summarise: while our approach follows some established processes and methodologies, e.g. the amount of fusion steps and selection of knowledge, and is therefore classifiable in the reviewed taxonomies it is unique in its representation of knowledge and focus on the neural part.

12.2. Knowledge-Guided Learning

Several approaches into combining knowledge with learning system, not limited to neural networks, have been proposed under terms such as *knowledge-infused* [SPG19], *knowledge-injected*,

knowledge-induced and *knowledge-informed* [Rue+21] learning. In the following, we will use the term knowledge-guided learning to summarise those closely related concepts. The goals identified, i.e. less train data, knowledge conformity, better test performance, interpretability [Rue+21], are a subset of those of neuro-symbolic learning.

Rueden et al. [Rue+21] provide a taxonomy for such systems. They differentiate between *sources* of knowledge—in particular *scientific*, *world* and *expert knowledge*—*representations* of knowledge, e.g. *algebraic equations*, *logic rules*, *knowledge graphs* and *human feedback*, as well as different *integrations* into the machine learning system, i.e. as *training data*, *hypothesis set* which is comprised of e.g. network architecture and model structure and *learning algorithm*, e.g. via regularisation terms. Our approach (see Chapter 13), uses expert knowledge represented as knowledge graphs or informal logic rules and is integrated in the learning algorithm. This deviates from the trend of reviewed work in Rueden et al. [Rue+21]. They reported, that while most, i.e. nine, approaches using knowledge graphs integrate them in the hypothesis set, only three approaches integrate them via the learning algorithm. Also, our choice of knowledge graphs and logic rules seems unique as none of the reviewed works explore these representations for expert knowledge. We assume that this is due to a lack of established approaches for representing procedural knowledge in knowledge graphs which we address in Section 9.2. One benefit of this is the ability to seamlessly integrate it with heterogeneous factual data. Instead, expert knowledge is represented as probabilistic relations or human feedback and to a lesser degree algebraic equations such as monotonicity constraints [Mur+18; Kur+21]. Probabilistic relations, which are closest related to the knowledge we use (see Chapter 9), are then integrated mostly via learning algorithms or hypothesis sets via Bayesian networks (see [HGC95; Yet+14; RD03]).

An alternative taxonomy that distinguishes the integration into shallow, semi-deep and deep infusion is provided in Sheth, Padhee and Gyrard [SPG19]. In the following, we present that taxonomy and use it to categorise remaining related work.

Shallow Infusion In *shallow infusion*, additional knowledge is introduced as pretrained models or weight vectors that are directly used as input. Therefore, the learning system does not have to be significantly adapted to ingest the additional knowledge. Examples of shallow infusion are word embeddings, that provide additional knowledge of the context in which words are frequently used [PSM14; Mik+13]; enriched word embeddings [Boj+17; Far+14; Mrk+16; Cas+18], in which additional semantic information is used to refine the embeddings; and deep neural language models that capture the context in which words are used [Pet+18; Dev+18].

Semi-Deep Infusion *Semi-deep infusion* models evaluate the performance of a learning system and add structural or symbolic knowledge if necessary [SPG19]. Examples range from teacher forcing [WZ89; KGS19], neural attention [Sun+17; Vo+17; Gau+19], to knowledge based models [YM17; Yi+18; She+18]. Closely related to our work is Garcia, Renoust and Nakashima [GRN20] that uses embeddings of nodes of artistic knowledge graphs and a regularisation term inferred by an encoder based on the difference of the latent representation of the CNN and the KG embedding in a multi task classification scenario. They show that this approach improves clustering in the latent space of the CNN and improves predictive performance. To justify the classification as semi-deep infusion, we highlight, that the use of regularisation can be viewed as an implicit evaluation of the performance and consequently addition of symbolic knowledge. Chiatti and Daga [CD22] extend this approach by additionally taking word embeddings into consideration to

identify features according to which the training space can be organised to mitigate data sparsity. Our architecture EBR is strongly inspired by Garcia, Renoust and Nakashima [GRN20] with the difference that we work on a regression scenario in a different domain and aggregate node embeddings relevant for the given input. Furthermore, Monka et al. [Mon+21] present a similar architecture which uses a contrastive knowledge graph embedding loss that 'contrasts images [...] against the class label representation' [Mon+21] of the knowledge graph embedding.

Deep Knowledge Infusion In contrast, *deep knowledge infusion* is defined as 'a paradigm that couples the latent representation learned by deep neural networks with the KGs, exploiting the semantic relationships between entities' [SPG19]. Sheth, Padhee and Gyrard [SPG19] outline several open research questions detailing what properties deep knowledge-infusion models could have, i.e. deciding what knowledge is relevant, being able to decide what kind of knowledge to infuse at what latent layer of the learning system and how to merge the knowledge representation with the latent representation of the learning system. [KGS20] present a concept for fusing knowledge contained in knowledge graphs with information contained in latent layers through a differential knowledge engine designed to select relevant knowledge and a knowledge-infusion layer with a separate optimization loop. In the domain of image description, Mogadala et al. [Mog+18] presented a deep infusion approach which increases the generalisability to samples unseen in the training. They employ a separate layer to align and merge the separate feature spaces of textual, semantic and visual representations before feeding the result to an output layer. Gaur, Faldu and Sheth [GFS21] propose a concept of using deep knowledge-infusion of academic knowledge graphs and student's previous learning history at every layer of the pipeline to discover weak concepts that explain the prediction. While this setting is transferable to the manufacturing scenarios considered in this thesis, an experimental evaluation is missing. Furthermore, process knowledge infused AI, has been proposed [She+22; Roy+22b; Roy+22a] as a system for natural language question generation. It combines NLP techniques to identify concepts, decision trees representing process knowledge that are used to analyse the found concepts, a set of constraints applied to the output and finally, a model that generates responses in natural language. Generated questions have been shown to be safer, follow clinical process guidelines and exhibit an increased explainability [Roy+22a]. Again, this approach would be transferable to manufacturing. However, a model that generates parametrisations, i.e. the topic of this thesis, is missing. As such it can be viewed as orthogonal to our approach and a combination could constitute as a possible future research direction.

To summarise: as for the closely related neuro-symbolic taxonomies, our approach is well classifiable in the reviewed knowledge-guided learning taxonomies. While we incorporate established components we combine them in a unique way.

12.3. Related Work in Manufacturing

The application of neuro-symbolic learning in manufacturing, at least in concert with semantic web technologies, is still in its infancy as evidenced by the fact that only 1.5% of the papers analysed by Breit et al. [Bre+23] in their exhaustive literature review belong to the domain *production of goods*, which apart from manufacturing also includes transport and logistics among others. One approach, falling into the [Neuro→ Symbolic] category, combines NLP techniques with a reasoner to allow the performance of maintenance procedures and requesting of information

from manuals on a digital twin [SVJ23]. Ringsquandl et al. [Rin+15], propose to reduce the feature space of a car assembly line through the incorporation of expert knowledge, formulated via the feature ontology in Graph Lasso methods. In the context of bearing fault analysis, Radtke and Bock [RB22] use logic tensor networks (LTNs) to enable 'reasoning in the loss function [to] mimic the decision process of an expert analysing the input data'. To this end, they derive a similarity function for vibration signals which is incorporated into the LTNs. Also in quality control, but based on image data, [Mül+22], present a neuro-symbolic system, that falls into the [Neuro $\rightarrow$ Symbolic] category and uses a CNN based pre-processing of the images. The results are then translated to logical facts and used to train horn clauses that represent defect rules via inductive logic programming (ILP). These defect rules are then used to classify the input image and can be used to generate explanations understandable for domain experts.

For the use-case of parametrisation prediction, Kurnatowski et al. use monotonicity constraints to represent procedural expert knowledge and an adaptive algorithm for monotonic regression for 'laser glass bending and press hardening of sheet metal' [Kur+21]. However, their use-cases are significantly less complex with two and four process parameters, respectively. Additionally, they do not predict parameters but aim to learn the functions describing the parameter-quality relationships. Wirth, Schmid and Voget [WSV22] present an abstract vision paper that intends to use human feedback, e.g. collected via active learning, to increase a parametrisation systems explainability in the automotive sector. Wilson et al. [Wil+22] use extracted expert knowledge to weight features of an SVM that solves regression problems for predicting positions of welds in linear friction welding of bladed disks. Interestingly, they only included the expert knowledge if there was no data available for the concerned entities. They reported an increase in R^2 of 4% for 60 data points when using the expert knowledge compared to a standard SVM. However, for 20 data points the performance was worse and for 40 and 80 data points comparable. Also, due to the manual adjustment of the features' weights it is limited to low level features which negates the feature computation capability of deeper machine learning approaches. In contrast to their approach, ours tries to mitigate low data quantity by guiding the learning system through expert knowledge pertaining to digitally available entities. Furthermore, we rely on the learning system's feature computation capabilities, comparing transformed higher order features and predictions to the available expert knowledge which leads to more consistent improvements.

For a comprehensive review of works concerning the related use-case of predictive quality, i.e. the prediction of a product's quality based on process data, e.g. process parameters, we refer to Tercan and Meisen [TM22]. While the review does not investigate knowledge-guided or neuro-symbolic approaches it still presents insights into a closely related setting. They observed that neural network models seem to be to most prominent with MLP, CNN and RNN being used in 38%, 30% and 5%, respectively, of the cases. Additionally, they provide insights into the quantity of data samples usually encountered in manufacturing settings. If real data is generated experimentally, which is the case for a majority of the reviewed publications, a majority of the publications validate their approaches with less than 1000 samples with a peak around 100 samples. In running production, dataset sizes are larger with a majority of publications using between 5000 and 50000 samples. This corresponds to the dataset sizes used in this thesis for the experimental FDM case study. Using the synthetic dataset generation framework we also generate datasets which lie in the sample range described for industrial publications.

13
Regularisation-Based Neuro-Symbolic Learning for Parameter Prediction

In this chapter, we present the definition of the prediction problem and two approaches to neuro-symbolic learning in manufacturing scenarios based on the regularisation of neural networks. The high-level information flow of data, knowledge and predictions and its possible integration into the parametrisation process via an intelligent assistance system is visualised in Figure 13.1. Both approaches rely on rules representing procedural expert knowledge as described in Section 6.1. While the first approach uses knowledge graph representations (see Chapter 9) of these rules which are embedded (see Chapter 10), the second approach directly uses predictions gained by applying corresponding rules to the input. To ensure comparability between the different approaches, we decided to use the same underlying multilayer perceptron (MLP), i.e. fully connected feed-forward neural network [Ama67]. Since the amount of data in our real world scenario is limited, we chose a relatively shallow MLP architecture with two fully connected hidden layers. An MLP, with the same characteristics, i.e. number and depth of layers, activation function and optimisation algorithm is used as a baseline network (MLP) to evaluate the performance impacts of guiding the learning process via expert knowledge.

13.1. Problem Definition: Parameter Prediction

An assistance system that aids the operator with parametrisation suggestions for the next process iteration, $i + 1$ (see Chapter 2) requires a learning system that provides predictions based on observed quality characteristics, θ_i, and the parametrisation, ϕ_i, of the current iteration, i, of the production process. More formally, we try to find an optimal function, f, for the following regression problem:

$$f(\phi_i, \theta_i) \rightarrow \phi_{i+1}.$$

Here, models are used that predict all process parameters of ϕ_{i+1} at once, since training an individual model for each individual parameter would exceed the computational resources given the amount of parameters present in complex industrial scenarios (see Chapter 3). As an alternative to predicting the following parametrisation, ϕ_{i+1} one could try to directly predict the optimal parametrisation ϕ^* that leads to the best achievable quality. However, in our case studies the data quantity is insufficient to directly predict ϕ^* with sufficient quality. Also, we assume that iterative adjustments allow operators to better grasp the effect of the suggested parametrisation, which is beneficial for validating the suggested parametrisation as well as teaching inexperienced operators how to iteratively parametrise the process. The problem addressed in this chapter is closely related to predictive quality [TM22] scenarios, i.e. the prediction of the expected quality

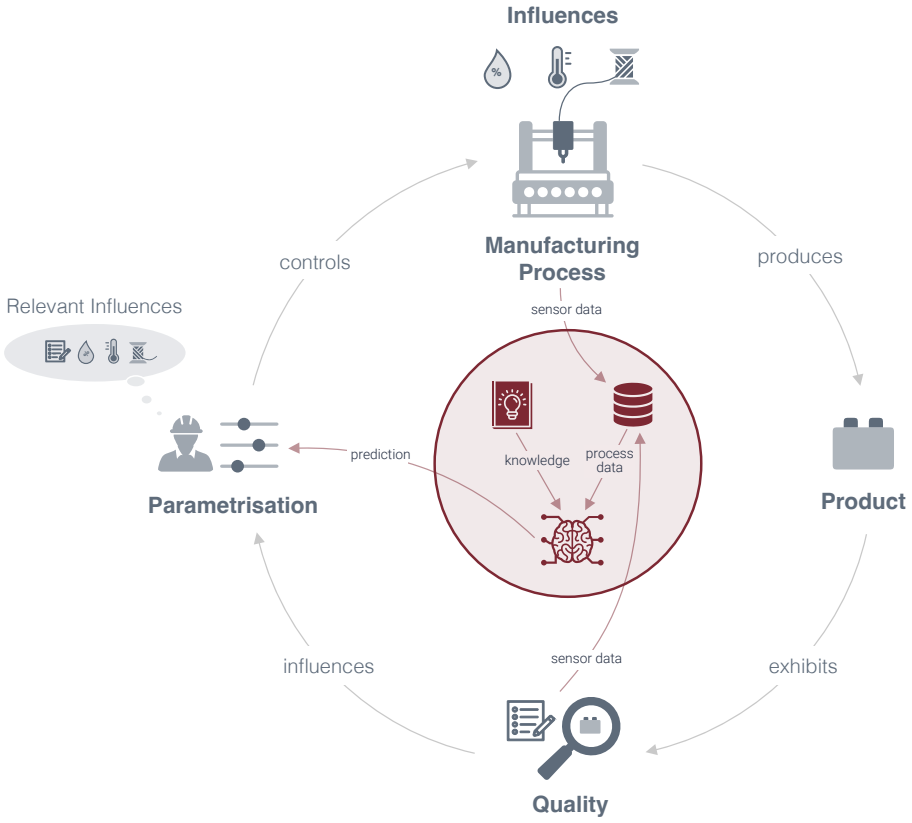


Figure 13.1.: Currently practised parametrisation process (grey) consisting of a process that is controlled by a parametrisation, which is chosen iteratively by an operator to minimise quality defects of the produced part in regards to target criteria. The future integration of neuro-symbolic learning to provide the operator with parametrisation suggestions via an assistance system is shown in red.

of a product given the current process parameters. While the wording in this section focuses on discrete production processes, where an iteration directly corresponds to a produced product, it is also applicable to discretised continuous production processes (see Section 2.2).

13.2. Embedding-Based Regularisation

The embedding based regularisation (EBR) architecture is inspired by ContextNet [GRN20], which uses the output of an encoder that transforms the latent representation h of the network to the embedding space as an regularisation term. In contrast to their architecture, we target a regression problem instead of multitask classification and use an MLP to compute the latent representation of the input compared to their use of ResNet50 [He+16]. Additionally, our contextual embeddings consist of node embeddings aggregated for input dependant subgraphs (see Section 10.3) that represent procedural rules instead of static node embeddings of the knowledge graph used in Garcia, Renoust and Nakashima [GRN20].

Figure 13.2 illustrates the EBR architecture. As described in Section 13.1, the input x is a

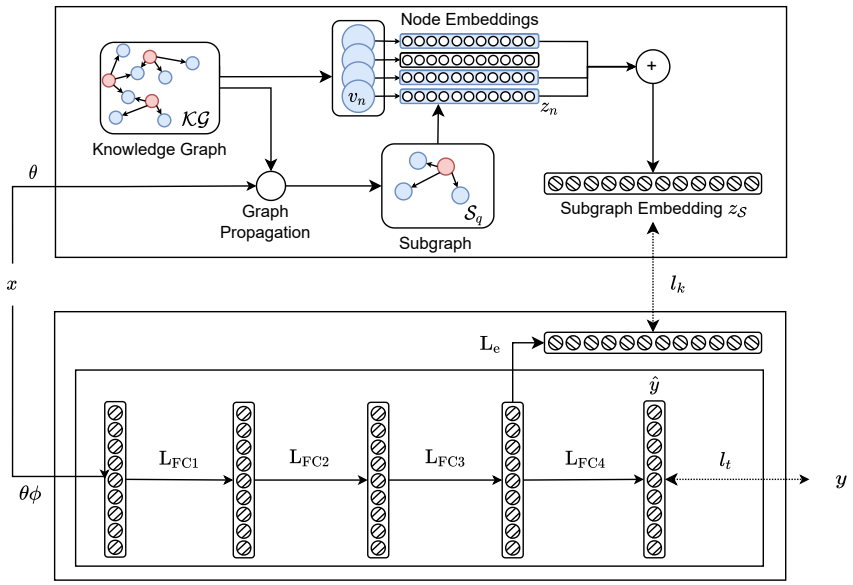


Figure 13.2.: The input or process iteration specific subgraph embedding $z_\mathcal{S}$ calculated in the upper part of the figure (see Sections 10.1 and 10.3) is combined with the MLP in the lower through the regularisation term l_k. To compute this term, an encoder, L_e is necessary to transform the latent representation of the MLP into a compatible semantic space (visualised by the rotation of the vector's values) as the subgraph embedding.

concatenation of the current parametrisation ϕ_i and the quality achieved by it θ_i. In the upper part, the quality θ_i, is used to generate a fitting subgraph and subgraph embedding $z_\mathcal{S}$ by propagating from the worst quality defect and aggregating the node embeddings as described in Section 10.3. In the lower part, we can see the same architecture as used in the baseline, i.e. the MLP with two hidden layers. It takes quality and parametrisation and produces the prediction $\hat{y}$ of the parametrisation of the next iteration, ϕ_{i+1}. Note, that in contrast to prevalent visualisations we focus on the outputs of layers (L). In contrast to the baseline, however, an additional encoder, L_e, tries to transform the latent representation of the network, i.e. the output of the last hidden layer, towards the semantic space of the subgraph embedding $z_\mathcal{S}$. This is intended to achieve a semantically compatible representation of subgraph embedding and latent representation, which is necessary to compute the regularisation term, l_k using smooth L1 loss. Based on the overall loss $\mathcal{L}$, the gradient updates are calculated for both the encoder as well as the MLP's layers. The node embeddings, in contrast, are treated as fixed since the knowledge graph is not updated during training. The overall loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = \lambda_t \sum_{j=1}^{N} l_t(y_j, \hat{y}_j) + \lambda_k \sum_{j=1}^{N} l_k(h_j, z_{\mathcal{S}_j}),$$

with j the j-th training sample, N the number of training samples, and λ_t and λ_k the weights for the mean squared error (MSE), l_t, and smooth L1 loss, l_k, for the prediction and embedding target,

respectively. While l_t is defined as

$$l_t(y_j, \hat{y}_j) = \frac{1}{n}\sum_{o=1}^{n}(y_o - \hat{y}_o)^2,$$

with n the number of predicted process parameters in $\hat{y}$, the smooth L1 loss of l_k is defined as:

$$l_k(h_j, z_{\mathcal{S}_j}) = \frac{1}{n}\sum_{m}^{n}\delta_{j,m},$$

with

$$\delta_{j,m} = \begin{cases} \frac{1}{2}(h_{jm} - z_{\mathcal{S}_{jm}})^2, & \text{if } \left|h_{jm} - z_{\mathcal{S}_{jm}}\right| \leq 1 \\ \left|h_{jm} - z_{\mathcal{S}_{jm}}\right| - \frac{1}{2}, & \text{otherwise} \end{cases},$$

where h_{jm} and $z_{\mathcal{S}_{jm}}$ are the m-th elements for the latent representation h and subgraph embedding $z_{\mathcal{S}}$ of the j-th training sample [GRN20]. λ_t and λ_k govern the influence of the respective loss in the overall loss. We follow Garcia, Renoust and Nakashima [GRN20] in having them sum up to one, i.e. $\lambda_t = 1 - \lambda_k$. The intention behind adding the regularisation term is to achieve a better separation of the latent representation space of the neural network as has been shown for image classification [GRN20]. Since in our scenario knowledge is not guaranteed to be present for every quality characteristic, we implemented a fallback which returns a 0-vector. For the validation and test phase we disregard the subgraph loss l_k and directly use the target MSE l_t since it is a direct measure of the predictive performance and the information that could be gained from the subgraph embedding has been incorporated into the MLP during training.

13.3. Rule-Based Regularisation (RBR)

The rule based regularisation (RBR) approach, shown in Figure 13.3, is closely related to EBR with the difference that the rules represented in the knowledge graph are directly applied, yielding a separate prediction $\hat{y}_{\mathcal{KG}}$. To this end, the parameter quantifications of the rules relevant for the most defective quality of the input are reconstructed by computing the corresponding subgraph $\mathcal{S}$ (see Section 10.3). Note, however, that the lossless reconstruction is only taken for implementational reasons. Conceptually, RBR is closer related to directly applying a rule set to an input problem and therefore the formal logic-based representation. The input specific subgraph, $\mathcal{S}_q$, is then further filtered to only contain quantified parameter adjustments $\hat{\mathrm{P}} \subseteq \mathrm{P}$ each of which contains a specific quantification ν relevant for this input (see Section 6.1). These are then added to the parametrisation of the previous process iteration if they are relative, ν_Δ, or replace the previous quantification if they are absolute, ν_a, which results in the rule based prediction $\hat{y}_{\mathcal{KG}}$. As $\hat{y}_{\mathcal{KG}}$ is directly compatible with the prediction $\hat{y}$, i.e. in the same semantic space, no additional encoder module is necessary and the output of the network $\hat{y}$ can be directly used for the loss computation. However, in contrast to EBR, this leads to the loss, l_k being computed with the prediction of the network compared to the latent representation, i.e. the information is more

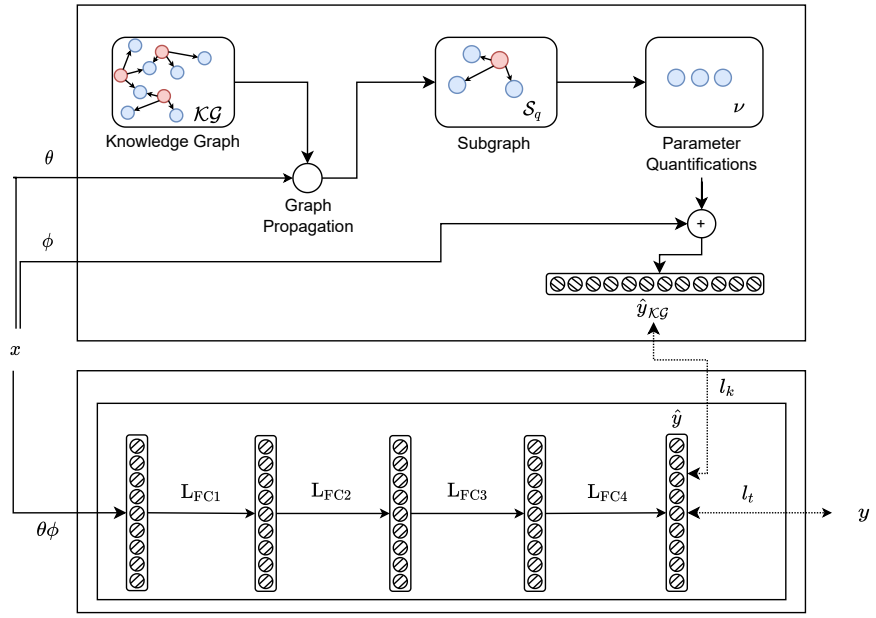


Figure 13.3.: In the upper section, quantifications of parameters pertaining to the most prominent quality characteristic of the input are selected from the knowledge graph and added to the process parameters of the input. It is combined with the MLP shown in the lower section by a regularisation term, l_k, that is calculated for the rule-based prediction and prediction of the network.

processed. This is reflected in $\mathcal{L}$, which is computed by:

$$\mathcal{L} = \lambda_t \sum_{j=1}^{N} l_t(y_j, \hat{y}_j) + \lambda_k \sum_{j=1}^{N} l_k(\hat{y}, \hat{y}_{\mathcal{KG}}),$$

with l_t and l_k as defined in Section 13.2.

14 Evaluation

In this section, we evaluate the two regularisation-based neuro-symbolic learning approaches presented in Chapter 13 by comparing them against a multi-layer perceptron (MLP) with the same properties as the MLPs used in the neuro-symbolic approaches.

14.1. Experimental Setup

We compare embedding (EBR) and rule based (RBR) regularisation approaches against an MLP on a parametrisation prediction problem (see Section 13.1) for the FDM and synthetic, PDPK$_\rho$, scenarios (see Sections 3.1 and 3.3). As in Chapter 7, we will modify the datasets to be able to investigate the characteristics of the approaches. In contrast to Section 7.1, we do not generate the synthetic dataset multiple times with different initialisations of the random generator due to computational constraints. However, in Section 14.2.2, we evaluate the models on varying datasets showing a relatively modest impact on performance when generalising to unseen data. We use rules forming the groundtruth of the expert knowledge data since it represents the maximum quality of knowledge that is achievable if the extracted expert knowledge is validated as described in Section 7.1.1.1. To represent it in a knowledge graph, we apply the representation $r_{\hat{\rho},ch,e,\eta}$ (see Chapter 9) with the selected embedding and aggregation methods, i.e. BoxE and RDF2Vec for FDM and synthetic as well as included head nodes and Euclidean distance for both (see Section 11.2.4).

For the (underlying in the case of EBR and RBR) MLP, rectified linear units [NH10] are chosen as activation functions and optimisation is done by stochastic gradient descent (SGD) [Rud16]. To combat overfitting, dropout [Sri+14] is added to each layer. The models are evaluated by the mean squared error (MSE), lower is better, averaged over all predicted parameters. Since we average over 46 normalised parameters, a MSE of 0.022 corresponds to the maximum error for one parameter if the remaining 45 parameters are predicted perfectly. To arrive at optimal hyperparameters, we conduct five-fold hyperparameter tuning using Bayesian Optimization Hyper Band (BOHB) [FKH18] for the architectures MLP, EBR, and RBR on both default datasets with a fixed budget of 200 samples each. Epochs were not specifically tuned due to computation constraints and convergence being achieved at 100 epochs. The resulting hyperparameters are listed in Table 14.1. All experiments were conducted for 30 iterations to mitigate effects that could occur due to the random initialisation. To interpret our results we analyse their performance and calculate the probabilities of each method to perform best following the Calvo model [Cal+19], described in more detail in Section 11.1.

Table 14.1.: Results of the hyperparameter tuning for MLP, EBR and RBR for FDM and synthetic, $PDPK_\rho$, datasets. Unabbreviated column names are data set, architecture, momentum, weight decay, learning rate and dropout for layers one to four.

DS	Arch.	Moment.	W. Decay	L. Rate	D. L1	D. L2	D. L3	D. L4	λ_t
FDM	MLP	0.6598	0.0280	0.0335	0.6574	0.0885	0.4593	0.0081	
	EBR	0.8515	0.0007	0.0074	0.4774	0.2283	0.3938	0.1016	0.9625
	RBR	0.5708	0.0019	0.0622	0.0207	0.1286	0.2151	0.0160	0.6568
$PDPK_\rho$	MLP	0.7326	0.0026	0.0059	0.4777	0.8184	0.7699	0.0531	
	EBR	0.4463	0.0012	0.0550	0.1623	0.6072	0.5617	0.0642	0.3677
	RBR	0.4823	0.0057	0.0304	0.7777	0.7452	0.5126	0.0152	0.9437

14.2. Performance of Regularisation-Based Neuro-Symbolic Learning

We investigate the following research questions (RQs) to evaluate properties of the different regularisation-based neuro-symbolic architectures (described in Chapter 13) against those of a purely neural baseline:

- RQ1: Does neuro-symbolic learning provide a benefit on low amounts of data?
- RQ2: Does neuro-symbolic learning improve the generalisation ability?
- RQ3: Does representing rules in knowledge graphs and embedding those provide a benefit compared to directly applying them?

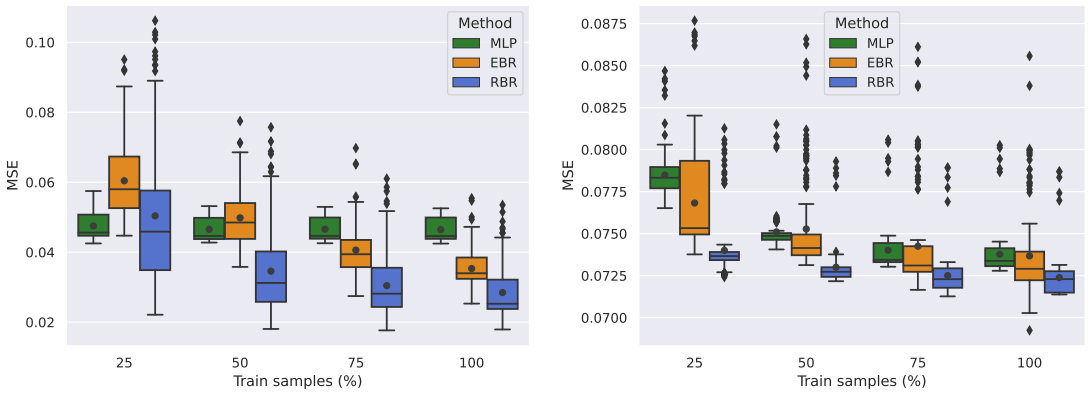
While RQ1 and RQ2 are answered by individual experiments, RQ3 is analysed in each of those.

14.2.1. Data Quantity

To answer RQ1 and establish whether the presented neuro-symbolic approaches provide a benefit on scenarios with low amounts of data, we conduct two experiments that manipulate the size of the dataset, thereby limiting the amount of samples being used. For both experiments, we assume that the performance of all approaches decreases with decreasing dataset size. Also, in accordance with [SRG23; Rue+21; GRN20], we expect the benefit of neuro-symbolic approaches to increase with decreasing amounts of data.

14.2.1.1. Reduction of the Default Dataset

The first experiment uses the FDM and synthetic $PDPK_\rho$ datasets described in detail in Sections 3.1 and 3.2 and discards a varying percentage of the train data, i.e. it uses 25%, 50%, 75% and 100% of the originally available train data. However, the knowledge that is used to guide the network remains the same. Due to the five-fold cross validation, we end up with five unique draws for each percentage. We assume that this is sufficient to ensure that no significant variance is introduced by selecting the percentages. Results, are shown in Figure 14.1. In both cases, the models have reached a convergence between the averaged train and test splits of the five folds, indicating that



(a) FDM

(b) PDPK$_\rho$.

Figure 14.1.: Results of Section 14.2.1.1 evaluating MLP, EBR and RBR on varying percentages of train data of the original FDM and synthetic datasets. The mean is indicated by a filled circle.

the 100 epochs are sufficient and the generalisability to the test set is as good as possible for the available data.

For the FDM dataset, we can see that the first hypothesis clearly holds for the neuro-symbolic methods, with the performance decreasing with decreasing dataset size. While the mean MSE performance of EBR decreases by an average of 19.64% (0.0084 absolute), this behaviour is slightly more pronounced for RBR, which decreases by an average of 22.09% (0.0073 absolute). For RBR the decrease is especially stark between 50% and 25% of the dataset with 45.73% (0.016 absolute). In contrast to this, the performance of the MLP remains relatively constant, decreasing only slightly between 50% and 25% by 2.07% (0.00097). The second hypothesis holds only partly, i.e. there is a benefit, with the benefit of neuro-symbolic approaches over the MLP decreasing with decreasing sample size until 50% and 25%, respectively. While EBR provides an average benefit of 18.41% (0.0086 absolute) over MLP on 75% and 100% of the train dataset, the situation is reversed for 25% and 50% with EBR performing an average of 17.09% (0.0081) worse than MLP. EBR's best result is achieved at 100% EBR with a performance increase of 23.97% (0.011 absolute) over MLP. RBR on the other hand provides an average benefit of 33.03% (0.015 absolute) over MLP on 50%, 75% and 100% of the train dataset. While the greatest benefit is achieved by RBR at 100% of the train data with 38.73% (0.018 absolute), the MLP performs better at 25% by 6.04% (0.0029 absolute). We observe that, while providing a significant benefit for larger amounts of data, the higher variance of the neuro-symbolic methods needs to be considered when applying these models in real-world environments. Over all train data proportions, RBR outperforms EBR by an average of 22.91% (0.01 absolute).

For the synthetic dataset PDPK$_\rho$, the validation results are not as far apart as for the FDM scenario as observable by the scale on the y-axes of Figures 14.1a and 14.1b. The first hypotheses holds, with the mean MSE of all methods decreasing with decreasing train data. While the average decrease is the most pronounced for MLP with 2.10% (0.0016 absolute) and EBR with 1.28% (0.00096 absolute), RBR decreases the least with 0.74% (0.00054). The second hypotheses holds for RBR that continually increases its benefit from 1.88% (0.0014 absolute) at 100% train data to 5.70% (0.0045

absolute) at 25%. Due to several negative outliers, the mean MSE of EBR only provides a benefit of 2.11% (0.0016 absolute) for 25% of the train data. If the median MSE is considered, however, its behaviour is consistent with the hypothesis, increasing from 0.39% (0.00028 absolute) to 3.83% (0.003 absolute). Over all train data proportions, RBR outperforms EBR by an average of 2.79% (0.0021 absolute) and 1.58% (0.0011 absolute) for mean and median MSE, respectively.

Overall, the results are not entirely conclusive. While both hypotheses hold on the synthetic dataset, the high stability of MLP on FDM in regards to data quantity leads to hypotheses one and two holding only partly. However, the neuro-symbolic approaches provide significant benefits on both datasets with 38.73% and 5.7% on FDM at 100% and PDPK$_\rho$ at 25% train data, respectively. If aggregated over the datasets using the Calvo model (see Figure 14.2), we can see that even for 25% of the train data, where the performance of RBR was worst on FDM, RBR still performs best with a probability of 68.68%. For each of the other percentages its probability is above 80%. In regards to RQ3, RBR outperforms EBR by a large margin for each percentage—even for the best probability achieved for EBR, i.e. 12.26% at 100%.

14.2.1.2. Synthetic Datasets with Varying Sizes

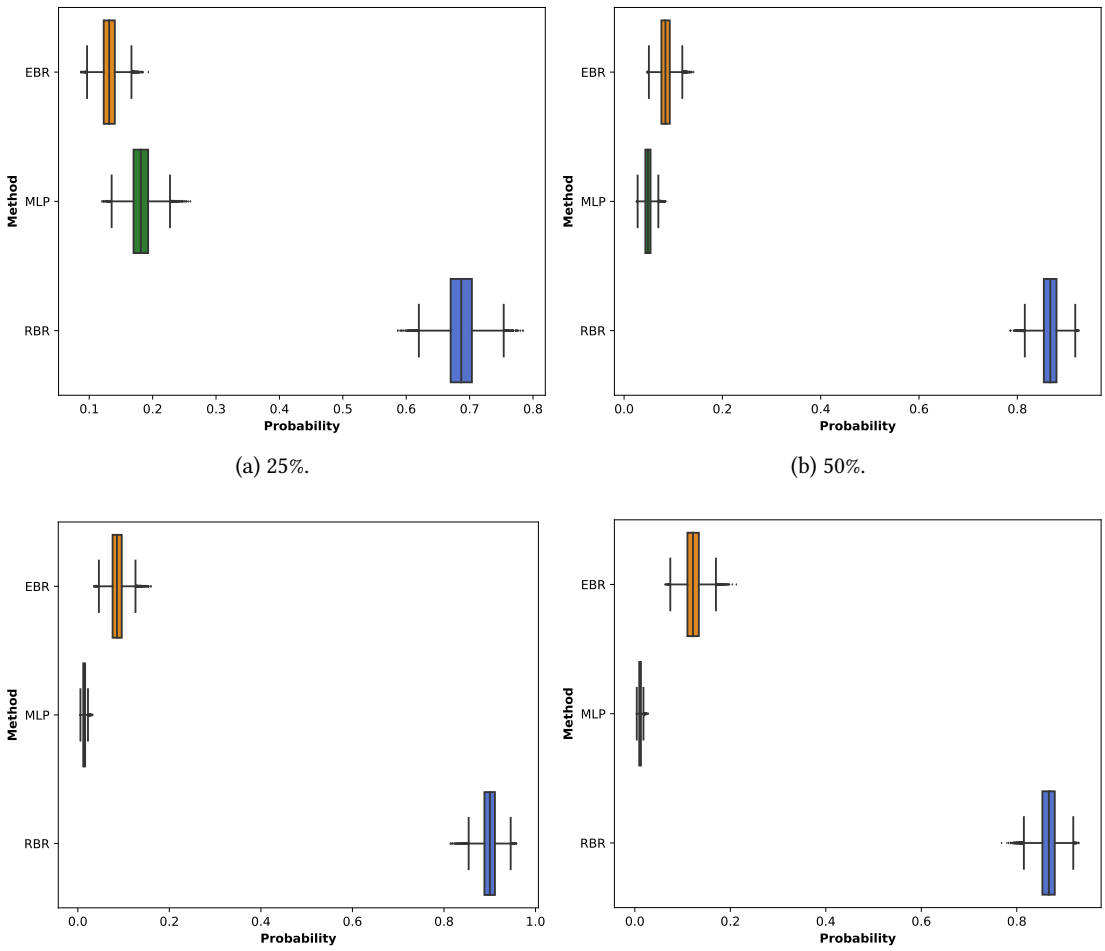
We use PDPK to generate datasets that rely on the same configuration, parameter-quality relations and underlying functions but different sample sizes i.e. 100, 500, 1000, 5000 and 10000. The results, are shown in Figure 14.3. First of, analysing the individual performance shown in Figure 14.3a, we observe that the performance between the different sizes converges around 5000 samples. Consequently, 10000 samples were excessive, since the methods changed only by 0.3%, 0.5% and 0.4% for MLP, EBR and RBR, respectively. As for the synthetic dataset in Section 14.2.1.1, the first hypothesis, i.e. decreasing performance with decreasing data quantity holds for all methods. While the average decrease is the most pronounced for MLP with 10.89% (0.0079 absolute) and EBR with 9.61% (0.0069 absolute), RBR decreases the least with 8.75% (0.0062). While the decrease in performance is not strictly monotonous (the exception lying between 1000 and 500 samples for all models), it is especially stark between 500 and 100 samples with 31.79%, 25.92% and 22.92%, for MLP, EBR and RBR, respectively.

While such a clear trend is not visible for the second hypothesis, i.e. increasing improvement of neuro-symbolic methods over MLP for decreasing amount of data, the lowest amount of data profits significantly more from the neuro-symbolic approaches than the highest. For EBR the improvement changes from 0.87% (0.00057 absolute) at 10000 samples to 4.57% (0.0044 absolute) at 100 samples. The improvement for RBR changes from 2.19% (0.0014 absolute) to 8.49% (0.0083 absolute).

The address RQ3 for this experiment, we observe that over all dataset sizes, RBR outperforms EBR by an average of 2.05% (0.0016 absolute) indicating that it is the superior method. This is reinforced by the corresponding plot of the Calvo method (see Figure 14.3b) ascertaining that it is the best method with a probability of 96.55%.

14.2.2. Generalisability

To answer RQ2, whether regularisation-based neuro-symbolic learning approaches increase generalisability, we use the synthetic dataset generator to create variations of a dataset and evaluate them on models pretrained as per the experimental setup. In contrast to PDPK$_\rho$, the dataset of this experiment does not contain noise in the expert knowledge, since this removes



(a) 25%.

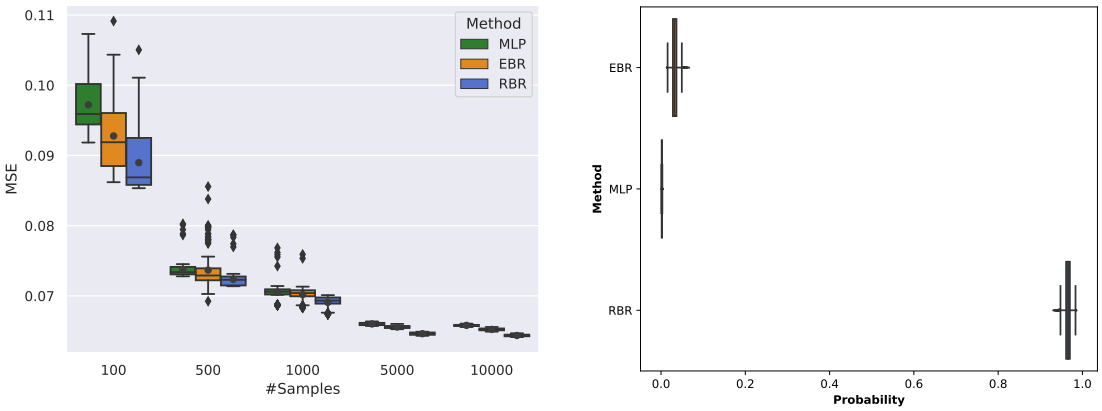
(b) 50%.

(c) 75%.

(d) 100%.

Figure 14.2.: Results of Section 14.2.1.1 aggregated over FDM and PDPK$_\rho$ showing the probabilities for EBR, MLP and RBR being the best method for each train data percentage.

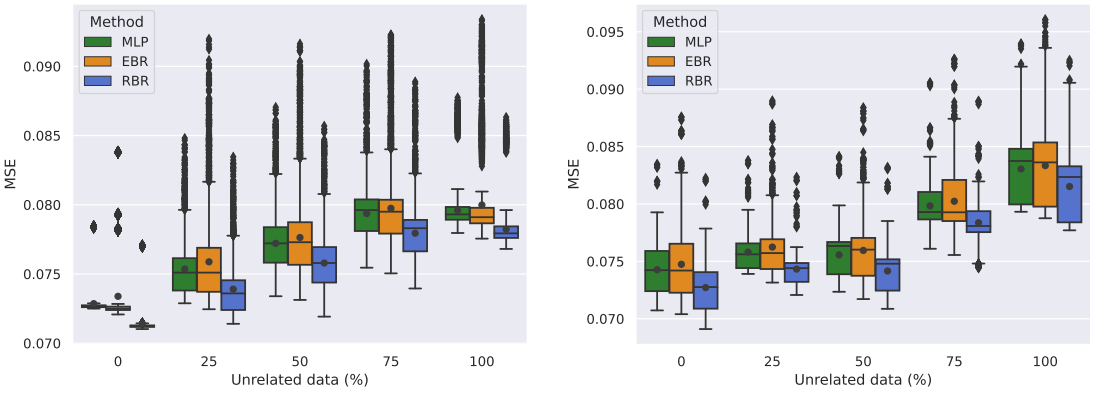
a source of variance at the point of knowledge generation. The rest of the parametrisation is equivalent to that of PDPK$_\rho$. The variations on which generalisability is evaluated are created by randomly replacing 0%, 25%, 50%, 75% and 100% of the 25% parameter-quality, $(p\text{-}q)$, tuples of the dataset that are not part of the the expert knowledge which constitutes the remaining 75% ($(P \times Q)_k = 75\%$, see Section 3.3.2). Therefore, the expert knowledge available to EBR and RBR remains constant over the variations. To mitigate the variances introduced by the sampling, we execute two experimental setups: (1) we repeat the random selection of $(p\text{-}q)$ tuples 30 times (results in Figure 14.4a) or (2) select the $(p\text{-}q)$ tuples once from five previously unseen datasets that consist of similar underlying functions, i.e. the sign of the function coefficients remains the same (results in Figure 14.4b). In contrast to the experimental setup of the previous experiments, where the evaluation was done on the averaged results of the k-fold validation splits, this experiment is evaluated on the overall dataset. While this uses samples as test data that have already been seen in training, we assume that its effect is limited since the models are trained to convergence



(a) Performance on the individual datasets. The mean is indicated by a filled circle.

(b) Probabilities of being the best method according to the Calvo model for the overall experiment, i.e. aggregated over the datasets with varying number of samples.

Figure 14.3.: Results of Section 14.2.1.1 evaluating MLP, EBR and RBR on synthetic datasets with varying sizes but apart from that identical configuration.



(a) $(p\text{-}q)$ selection repeated on the default dataset.

(b) $(p\text{-}q)$ selection repeated on different datasets.

Figure 14.4.: Results of Section 14.2.2 evaluating MLP, EBR and RBR on synthetic datasets with varying amounts of $(p\text{-}q)$ tuples replaced.

between train and test performance. Also, it ensures that the replaced samples are evaluated in combination with unreplaced ones.

We hypothesise, that with increasing $(p\text{-}q)$ tuples replaced, the performance of the approaches decreases. Furthermore, we expect the generalisability to increase, i.e. the benefit of neuro-symbolic models over the MLP to increase, with increasing $(p\text{-}q)$ tuples replacement. The second hypothesis is grounded in the observation of Garcia, Renoust and Nakashima [GRN20] that neuro-symbolic regularisation leads to better segmented latent representations, which we assume could have a positive effect on unseen data.

The results for multiple draws on the default dataset are shown in Figure 14.4a. We observe a similar behaviour as in Section 14.2.1.1 at 100% data, with RBR performing best, followed by MLP and EBR, with EBR suffering from outliers which affect its mean. Also, the introduction of unrelated $(p\text{-}q)$ tuples increases the number of outliers. The first hypothesis mostly holds, with an average decrease of 2.15% (0.0016 absolute), 2.22 (0.0017 absolute) and 2.29 (0.0017) for EBR, MLP and RBR, respectively. However, the decrease between 75% and 100% of replaced $(p\text{-}q)$ tuples is marginal with an average of 0.02% over all methods. The second hypothesis, however, does not hold. Comparing EBR and MLP, we cannot detect a reliable trend if the median is considered. In case of observing the mean, EBR performs consistently worse. RBR performs exactly opposite to our assumption with the benefit over MLP continuously decreasing with increasing $(p\text{-}q)$ tuple replacement from 1.97% to 1.70%.

Comparing these results with the ones seen in Figure 14.4b, that show the $(p\text{-}q)$ selection for different datasets, we observe that while the overall performance is slightly lower for each percentage apart from 50%, the general trends remain the same. The worse performance is intuitively explained by considering that the amount of generalising the models have to accomplish is significantly higher in this case since not only a certain percentage of $(p\text{-}q)$ tuples is replaced but also coefficients of the remaining $(p\text{-}q)$ tuples can change. While the average relative decrease in performance between different datasets and multiple draws on the same is quantified by 0.99% (0.0008 absolute), the average relative benefit of RBR over MLP is 3.78% (0.0014 absolute) higher. This difference could indicate a slight increase in generalisability to unseen datasets using RBR. However, since the decrease in performance is very small, no significant conclusions can be drawn from it.

Overall, our assumption that a better separation in the latent space would lead to increased generalisability is evidently not true. One possible explanation is that our experimental setup does not contain expert knowledge for unrelated $(p\text{-}q)$ tuples, instead relying on the better separation of the latent space for improvements. However, even if the experimental setup would contain expert knowledge for unrelated $(p\text{-}q)$ tuples, they would not play an excessive role during training due to the selection of relevant knowledge taking place in the subgraph embeddings (see Section 13.2) or rule-based predictions (see Section 13.3). As such, the only influence of it would be of indirect nature, i.e. during the node embedding, which we assume would not have a major effect due to the unimpressive performance of EBR. We assume that the generalisability of the approaches presented in this thesis is additionally hampered by the fact that during prediction expert knowledge is not considered, which makes it impossible to consider knowledge pertaining to the unrelated data. Consequently, we interpret our results as an indication that the promise of increased generalisability of neuro-symbolic methods can only be realised if knowledge pertaining to the unrelated data is present and if it is actively being used either during the formation of the embeddings or in the prediction phase. Since a selection of relevant expert knowledge is prevalent in neuro-symbolic approaches [KGS20; SRG23], we assume that benefits could be had if the knowledge selection is learnt as opposed to being selected based on domain knowledge. However, this increases demands on the data quantity. Additionally, an explicit transfer learning step seems to be necessary to transfer between the knowledge used during training and that used during prediction.

To answer RQ3, we observe that RBR outperforms EBR by an average of 2.42% and 1.82% for mean and median, respectively, over all percentages of unrelated data for multiple draws on the default dataset. Taking the other experiments into consideration, we can conclude that either the

embedding quality is not sufficient to allow for its effective use or the encoder does not learn to sufficiently transform between the semantics of latent and embedding space. Interesting in that regard are the results of the hyperparameter tuning (see Table 14.1), which weight the knowledge loss to a surprising degree, 0.63, for EBR. This could indicate that this architecture struggles specifically on the synthetic dataset. This impression is reinforced when comparing EBR on PDPK_ρ to its performance and configuration on FDM, where the target loss is weighted stronger leading to EBR exhibiting a lower amount of outliers and variance which positively impact its overall performance.

Part V.

Conclusion & Outlook

This thesis has highlighted challenges confronting manufacturing. To meet these, we have proposed a neuro-symbolic approach that utilises expert knowledge that can be extracted in a data-based manner. This part will provide a summary and discussion of the findings and evaluation results of this thesis in Chapter 15 and an outlook future research directions in Chapter 16.

15 Conclusion

This thesis set out to provide the foundation for an intelligent assistance system that is able to recommend parametrisations to operators to mitigate occurring quality defects in discrete and continuous manufacturing scenarios. To this end, we investigated whether neuro-symbolic approaches can overcome manufacturing specific challenges, e.g. low amounts of data of erroneous parts or products and generalisability. To aid in the extraction of formalised expert knowledge that can be used in the symbolic component, we presented a data-based knowledge extraction approach. To benchmark our approaches on an additional datasets and to be able to evaluate edge cases, we developed a framework for synthesising process data and accompanying procedural knowledge. In the following, we summarise these approaches and the respective evaluation results.

Findings Through analysing the parametrisation processes prevalent in manufacturing, we identified periods in which the cognitive load of the operator is lower and which are consequently more suited for eliciting their procedural knowledge. These periods are used to prompt operators to highlight quality parameters that influenced their parametrisation in an example-based manner. We have shown that this additional information decreases the dimensionality and complexity of the resulting knowledge segments that can then be more easily aggregated to a common rule base. We proposed weighting and filtering-based approaches to detect and remove outliers in the information provided by operators. These provide a benefit to the aggregation of knowledge segments compared to a naive aggregation method for monotonic functions, which are prevalent in manufacturing [Kur+21]. Since there is no single best aggregation method for knowledge segments, we recommend that the methods should be individually evaluated on each application scenario.

Since knowledge graphs and their embeddings are frequently used in neuro-symbolic systems and to integrate knowledge into a company wide knowledge repository, we developed representations for procedural knowledge in knowledge graphs. Also, we developed an embedding method for procedural knowledge that aggregates node embeddings for entities present in subgraphs, which correspond to rules. Additionally, we presented a metric to evaluate our embedding approach for downstream scenarios. We found that our proposed chained-binary representations of ternary relations outperform reification-based representations, which are frequently encountered in literature, in a majority of the cases. Furthermore we provided an indication that representing quantifications as literals does not provide a significant benefit to their representation as entities. Also, we empirically confirmed our theoretical considerations that RDF2Vec is the best embedding method for dealing with indirections which resulted from modelling ternary relations. Lastly, we discovered that options governing the subgraph aggregation are highly dependent on the representation and embedding method used.

We proposed two regularisation-based neuro-symbolic architectures of which one uses the embeddings described above (EBR), while the other directly uses the formalised expert knowledge (RBR). We found that EBR performed worse than directly using the formalised knowledge indicating that knowledge graph representations of procedural knowledge are better used for integrating into company wide knowledge graphs than predictive systems. However, both approaches provide benefits over an architecturally similar multilayer perceptron (MLP) baseline. We discovered, that depending on the dataset used, neuro-symbolic methods can indeed provide a significant benefit on low amounts of data. On the synthetic datasets, RBR provides a benefit of 5.7% to 8.49%, while on the FDM it performs best until a certain dataset size, with the maximum benefit (38.73%) achieved at the largest dataset. In contrast to that and promises of the literature [SRG23], we found that at least our regularisation-based approaches do not improve the generalisation to unrelated data compared to the MLP baseline. A possible reason is that they only use the expert knowledge during training and not during prediction.

Research Methodology & Limitations To identify and evaluate the respective approaches we used a mixed-method research methodology. Qualitative methods, e.g. unstructured interviews, were used to analyse the parametrisation process and confirm our assumptions regarding cognitive load of operators therein. The respective approaches are evaluated via quantitative and experimental research methods on case studies, culminating in Bayesian performance analysis via Calvo models [Cal+19] to arrive statistically grounded conclusions that can be aggregated over multiple experiments and datasets. The low amount of case studies signify that our results can only serve as an indication and need to be validated for other application scenarios.

Limitations arise from the breadth of research and the low amount of case studies and therefore datasets covered. The possible combinations of parametrisations of the respective approaches are beyond what is reasonable to compute. Therefore, we evaluated each approach individually and continued with the best results. Also, as in the case of data-based knowledge extraction, it is sometimes unfeasible to evaluate the different components, e.g. elicitation, aggregation and extraction, individually, leading to aggregated metrics and uncertainty where effects are introduced. Furthermore, biases due to random sampling and probabilistic methods exist. However, these biases are controlled by repeatedly executing the sampling and methods for different seeds.

Research Objectives & Implications While we were not able to validate the combination of our approaches in an industrial setting in production, we believe that our work represents an important step towards the applicability of intelligent assistance systems for parametrisation processes of complex production processes, since we proposed a methodology that improves significantly on the state of the art. Furthermore, while it does not improve generalisability, it has the potential to address data quantity issues that aversely affect the applicability of learning systems in manufacturing.

Through the introduction of data-based knowledge extraction, we address a second objective, i.e. externalising implicit operator knowledge in a quantified manner. While the proposed methodology does not automatically lead to production ready results, a validation stage can be easily appended. Apart from providing a foundation on which to increase the quality and speed of onboarding processes, this externalised knowledge allows the comparison of operator and process engineering knowledge, leading to the potential to increase the respective understanding of the process. Also, the resulting knowledge graphs can be easily integrated with other formalised

knowledge forming a company wide knowledge base.

Combined, these two objectives serve the overarching goal of facing labour shortage, currently one of the main challenges to manufacturing. While we conducted a thorough investigation into both data-based knowledge extraction as well as neuro-symbolic learning systems for the parametrisation process multiple areas of research remain beyond the scope of this thesis. Therefore, the following chapter gives an outlook into future research questions and directions.

16 Outlook

While this thesis developed and analysed data-based knowledge extraction as well as two main hindrances to the applicability of intelligent assistance systems in manufacturing, i.e. generalisability and performance on low-data, it cannot cover all relevant aspects of such systems. In the following, we explore open research challenges in the topics addressed in this thesis as well as the industrial use of intelligent assistance systems.

Data-Based Knowledge Extraction

Depending on the complexity underlying the parameter-quality relations in an application scenario, our aggregation-based quantification approaches might not perform as intended as shown in Section 7.1.4.3. While monotonic functions that are better suited to our methodology are prevalent in manufacturing [Kur+21], learning functions as quantifications could nonetheless prove beneficial for more complex functions. As a prerequisite, however, constraints regarding data quantity have to be addressed. This could be done by using the predictive performance of the initially extracted rules in the context of an assistance system, to refine their underlying quantifications.

Another possible improvement to the weighting algorithm that ascertains the reliability of knowledge segments provided is the unsupervised classification of the operators' behaviour, e.g. whether they are exploring new parametrisations or exploiting previous knowledge. The underlying assumption being that knowledge segments gathered during exploitative behaviour are more trustworthy than others. At a more fine-grained level, common defect mitigation stages of the operators and their temporal relation to each other could be discovered.

Through the evaluation in the extrusion case study at an industrial scale (see Section 3.2), we hope to gather insights into how data-based compare to interview-based knowledge extraction approaches, e.g. in the type or detail of knowledge extracted, and how they perform on additional datasets. Also, we plan to embed the data-based knowledge extraction in a more holistic knowledge extraction process with a manual validation phase to ensure the quality of the extracted knowledge. Additionally, the trade off of sequential and combined elicitation (see Chapter 5) and their interrelationship with the operators' cognitive load at different stages of the parametrisation process can be studied in this environment. Lastly, we hope to quantify the operators' cognitive load levels to arrive at conclusions regarding whether additional process steps are feasible, i.e. sequential vs. combined knowledge elicitation or gathering their feedback of the generated rules via active learning. The latter could help in weighting the relations of knowledge segments, thereby increasing the quality of the knowledge aggregation.

Representation and Embedding of Procedural Knowledge

Directly modelling the confidence of rules in the procedural knowledge graphs and embedding them via probabilistic embedding methods [Vil+18] could be used to better represent trust in operators or knowledge segments. One strength of representing procedural knowledge in knowledge graphs is the ability to seamlessly integrate it into a company wide knowledge graph that integrates different data sources, e.g. documentation of the production process, manufacturing execution systems and temporal process data. For the last, temporal knowledge graphs embedding methods, e.g. Messner, Abboud and Ceylan [MAC22] could be employed. However, how knowledge graphs containing procedural knowledge of possibly mixed representations as well as temporal process data can be embedded in a joint vector representation is an open research question.

Neuro-Symbolic Learning in Manufacturing

One challenge limiting the direct industrial application of the proposed neuro-symbolic approaches lies in their relatively high variances (see e.g. Figure 14.1) that could be explained by the fact that their predictions are better suited for some parameters than others. Automatically dividing the monolithic model we proposed into ensembles consisting of parameters behaving similarly could offer a trade-off between training time and predictive performance. Another improvement would be the adaptive computation of λ_t and λ_k, that govern the influence of the symbolic component, depending on the loss of the currently predicted sample. Also, a stronger integration between neural and symbolic components would enable the refinement of extracted rules and provide an additional feedback loop for the data-based knowledge extraction that can continuously update the weights of knowledge segments based on their applicability, i.e. the quality of predictions that were achieved containing rules based on the knowledge segment in question. This represents a co-evolution of learning system and knowledge acquisition that could become necessary because the knowledge segments given by operators are likely influenced by the increasing automation [LA16]. Furthermore, this would represent an approach of learnt knowledge selection, which we assume could benefit the generalisability. However, further research into how task specific domain knowledge can be infused into pretrained systems is required.

Additionally, it could be investigated whether infusing the knowledge rather than using it as a regularisation term could provide a benefit. If the knowledge graph contains not only procedural knowledge but a more complete representation of the domain knowledge present in a company, i.e. including factual knowledge, an interesting research question would be whether the knowledge should be infused at one or multiple different positions in the neuro-symbolic architecture. For deeper artificial neural networks (ANNs), factual knowledge could be infused in earlier layers ensuring semantic alignment while conceptual and procedural knowledge could provide more complex concepts at higher levels of abstractions in the later layers. An alternative to this approach would be to infuse the complete knowledge graph to a deeper ANN and rely on the ANNs' ability to decide which information to take into consideration, e.g. through transformer or attention layers. Moreover, the parametrisation process could be modelled in greater detail, e.g. through recurrent neural networks with one output per process iteration. This would allow for a more detailed temporal representation of coherent parametrisation steps an operator undertakes to arrive at the optimal quality or to mitigate a quality defect. However, since these approaches are more complex they probably need a higher amount of training data which is unlikely to be

present in manufacturing scenarios.

Lastly, using neuro-symbolic systems to increase the explainability of neural predictions could be investigated. As detailed in [Hei+23], different personas have different requirements regarding the overall explainability of the system or individual predictions. For a data scientist, it could be beneficial to know whether qualities or parameters fall into clearly separable clusters that lead to unique predictions. Garcia, Renoust and Nakashima [GRN20] have indicated that regularisation by symbolic systems leads to a better separation of the latent space of neural systems for classification tasks. Evaluating whether the characteristics of these clusters can lead to an increased understanding of the regression performance and the production process seems a worthwhile undertaking. On the operator level, explanations of the predictions, e.g. by highlighting similar situations that can be reconstructed via the underlying knowledge segments of the activated rules could aid in gathering process knowledge.

Industrial Assistance Systems

To be able to realise the promises of industrial assistance systems, e.g. reduced onboarding times and better problem-solving skills via new learning paradigms, higher job satisfaction of operators, simplified human-machine interaction, and higher productivity [Apt+18; Han+20; TFP20], the approaches presented in this thesis have to be integrated into such a system. The resulting system can be evaluated by its predictive accuracy, the operators' learning progress and reduction in time required for (re-)parametrisations. Another approach towards industrial assistance systems could lie in the incorporation of large language models (LLMs), that can be constrained and refined by domain specific models that provide the LLM with context via instruction training [Wan+23; Roy+22a]. These domain specific models could consist of small scale language models trained for example on process instructions, procedural knowledge or company wide knowledge graphs. The resulting domain-specific LLMs could be used as foundational models [Ala+22] to allow zero or few shot predictions.

Bibliography

[AB18] Wenke Apt and Marc Bovenschulte. 'Die Zukunft der Arbeit im demografischen Wandel'. In: *Zukunft der Arbeit–Eine praxisnahe Betrachtung* (2018), pp. 159–173 (cit. on p. 3).

[Abb+20] Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz and Tommaso Salvatori. 'Boxe: A box embedding model for knowledge base completion'. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9649–9661 (cit. on pp. 76, 77, 79, 94, 97).

[ADV22] Bram Aerts, Marjolein Deryck and Joost Vennekens. 'Knowledge-based decision support for machine component design: A case study'. In: *Expert Systems with Applications* 187 (2022), p. 115869 (cit. on p. 33).

[Ala+22] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds et al. 'Flamingo: a visual language model for few-shot learning'. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 23716–23736 (cit. on p. 137).

[Ali+21] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp and Jens Lehmann. 'PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings'. In: *J. Mach. Learn. Res.* 22.82 (2021), pp. 1–6 (cit. on p. 97).

[Ama67] Shunichi Amari. 'A theory of adaptive pattern classifiers'. In: *IEEE Transactions on Electronic Computers* 3 (1967), pp. 299–307 (cit. on p. 115).

[Ank+99] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel and Jörg Sander. 'OPTICS: Ordering points to identify the clustering structure'. In: *ACM SIGMOD Record* 28.2 (1999), pp. 49–60 (cit. on p. 49).

[Apt+18] Wenke Apt, Marc Bovenschulte, Kai Priesack, Christine Weiß and Ernst Hartmann. *Einsatz von digitalen Assistenzsystemen im Betrieb*. Bundesministerium für Arbeit und Soziales, 2018 (cit. on pp. 3, 137).

[ASA18] Forough Arabshahi, Sameer Singh and Animashree Anandkumar. 'Combining symbolic expressions and black-box function evaluations in neural programs'. In: *arXiv preprint arXiv:1801.04342* (2018) (cit. on p. 110).

[Asi+18] Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood and Hafiza Mahnoor Abbasi. 'A survey of ontology learning techniques and applications'. In: *Database* 1 (2018), p. 24 (cit. on p. 33).

[Bäc+18] Andreas Bächler, Liane Bächler, Sven Autenrieth, Hauke Behrendt, Markus Funk, Georg Krüll, Thomas Hörz, Thomas Heidenreich, Catrin Misselhorn, Albrecht Schmidt et al. 'Systeme zur Assistenz und Effizienzsteigerung in manuellen Produktionsprozessen der Industrie auf Basis von Projektion und Tiefendatenerkennung'. In: *Zukunft der Arbeit–Eine praxisnahe Betrachtung* (2018), pp. 33–49 (cit. on p. 3).

[BAH19] Ivana Balažević, Carl Allen and Timothy M. Hospedales. 'Tucker: Tensor factorization for knowledge graph completion'. In: *arXiv preprint arXiv:1901.09590* (2019) (cit. on p. 76).

[Bai79] Lisanne Bainbridge. 'Verbal reports as evidence of the process operator's knowledge'. In: *International Journal of Man-Machine Studies* 11.4 (1979), pp. 411–436 (cit. on p. 30).

[Ban+18] Tharindu Rukshan Bandaragoda, Daswin De Silva, Damminda Alahakoon, Weranja Ranasinghe and Damien Bolton. 'Text mining for personalized knowledge extraction from online support groups'. In: *Journal of the Association for Information Science and Technology* 69.12 (2018), pp. 1446–1459 (cit. on p. 33).

[BAY21] Shaked Brody, Uri Alon and Eran Yahav. 'How attentive are graph attention networks?' In: *arXiv preprint arXiv:2105.14491* (2021) (cit. on p. 77).

[BCB21] Sadeer Beden, Qiushi Cao and Arnold Beckmann. 'Semantic Asset Administration Shells in Industry 4.0: A Survey'. In: *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*. 2021, pp. 31–38 (cit. on p. 81).

[Ben+17] Alessio Benavoli, Giorgio Corani, Janez Demšar and Marco Zaffalon. 'Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis'. In: *The journal of machine learning research* 18.1 (2017), pp. 2653–2688 (cit. on p. 97).

[Ben+21] Wiem Ben Rim, Carolin Lawrence, Kiril Gashteovski, Mathias Niepert and Naoaki Okazaki. 'Behavioral Testing of Knowledge Graph Embedding Models for Link Prediction'. In: *3rd Conference on Automated Knowledge Base Construction*. 2021 (cit. on p. 77).

[Ber21] Jawad Berri. 'Knowledge Graph-based Framework for Domain Expertise Elicitation and Reuse in e-Learning'. In: *International Journal of Advanced Computer Science and Applications* 12.12 (2021) (cit. on p. 34).

[Bes+17] Tarek R. Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima et al. 'Neural-symbolic learning and reasoning: A survey and interpretation'. In: *arXiv preprint arXiv:1711.03902* (2017) (cit. on p. 109).

[BH05] Sebastian Bader and Pascal Hitzler. 'Dimensions of neural-symbolic integration-a structured survey'. In: *arXiv preprint cs/0511042* (2005) (cit. on p. 109).

[BK56] Benjamin S. Bloom and David R. Krathwohl. *Taxonomy of educational objectives: The classification of educational goals. Book 1, Cognitive domain*. longman, 1956 (cit. on p. 29).

[BL07] Stefano Borgo and Paulo Leitão. 'Foundations for a core ontology of manufacturing'. In: *Ontologies*. Springer, 2007, pp. 751–775 (cit. on p. 35).

[Blo+14] Jürgen Bloech, Ronald Bogaschewsky, Uwe Götze and Folker Roland. *Einführung in die Produktion*. 7th ed. Springer, 2014. ISBN: 978-3-642-31892-4 (cit. on p. 3).

[Boj+17] Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov. 'Enriching word vectors with subword information'. In: *Transactions of the association for computational linguistics* 5 (2017), pp. 135–146 (cit. on p. 112).

[Bol+08] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge and Jamie Taylor. 'Freebase: a collaboratively created graph database for structuring human knowledge'. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 2008, pp. 1247–1250 (cit. on p. 77).

[Boo+21] Grady Booch, Francesco Fabiano, Lior Horesh, Kiran Kate, Jonathan Lenchner, Nick Linck, Andreas Loreggia, Keerthiram Murgesan, Nicholas Mattei, Francesca Rossi et al. 'Thinking fast and slow in AI'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 15042–15046 (cit. on p. 109).

[Boo89] John H. Boose. 'A survey of knowledge acquisition techniques and tools'. In: *Knowledge acquisition* 1.1 (1989), pp. 3–37 (cit. on p. 30).

[Bor+13] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston and Oksana Yakhnenko. 'Translating embeddings for modeling multi-relational data'. In: *Advances in Neural Information Processing Systems* 26 (2013) (cit. on pp. 76, 77, 97).

[Bor+19] Agustin Borrego, Daniel Ayala, Inma Hernández, Carlos R Rivero and David Ruiz. 'Generating rules to filter candidate triples for their correctness checking by knowledge graph completion techniques'. In: *Proceedings of the 10th International Conference on Knowledge Capture*. 2019, pp. 115–122 (cit. on p. 36).

[Bou+19] Zied Bouraoui, Antoine Cornuéjols, Thierry Denœux, Sébastien Destercke, Didier Dubois, Romain Guillaume, João Marques-Silva, Jérôme Mengin, Henri Prade, Steven Schockaert, Mathieu Serrurier and Christel Vrain. *From Shallow to Deep Interactions Between Knowledge Representation, Reasoning and Machine Learning (Kay R. Amel group)*. 2019. URL: https://arxiv.org/pdf/1912.06612.pdf (cit. on p. 4).

[BP19] Derek Beach and Rasmus Brun Pedersen. *Process-tracing methods: Foundations and guidelines*. University of Michigan Press, 2019 (cit. on p. 31).

[Bra+07] Simone Braun, Andreas Schmidt, Andreas Walter, Gabor Nagypal and Valentin Zacharias. 'Ontology Maturing: a Collaborative Web 2.0 Approach to Ontology Engineering'. In: *Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at the 16th International World Wide Web Conference (WWW2007) Banff, Canada, May 8, 2007*. (2007). URL: https://www.ra.ethz.ch/cdstore/www2007/km.aifb.uni-karlsruhe.de/ws/ckc2007/papers/BraunSchmidtWalterNagypalZacharias.pdf (cit. on p. 35).

[Bre+23] Anna Breit, Laura Waltersdorfer, Fajar J. Ekaputra, Marta Sabou, Andreas Ekelhart, Andreea Iana, Heiko Paulheim, Jan Portisch, Artem Revenko, Annette ten Teije et al. 'Combining machine learning and semantic web: A systematic mapping study'. In: *ACM Computing Surveys* (2023) (cit. on pp. 109, 111, 113).

[BTW18] Xiwei Bai, Jie Tan and Xuelei Wang. 'Fault diagnosis and knowledge extraction using fast logical analysis of data with multiple rules discovery ability'. In: *International Conference on Intelligence Science*. 2018, pp. 412–421 (cit. on p. 33).

[Buc+21] Georg Buchgeher, David Gabauer, Jorge Martinez-Gil and Lisa Ehrlinger. 'Knowledge graphs in manufacturing and production: A systematic literature review'. In: *IEEE Access* 9 (2021), pp. 55537–55554. ISSN: 2169-3536 (cit. on pp. 75, 89).

[Cal+19] Borja Calvo, Ofer M. Shir, Josu Ceberio, Carola Doerr, Hao Wang, Thomas Bäck and Jose A. Lozano. 'Bayesian performance analysis for black-box optimization benchmarking'. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 2019, pp. 1789–1797 (cit. on pp. 97, 103, 121, 132).

[Cas+18] Mercedes Arguello Casteleiro, George Demetriou, Warren Read, Maria Jesus Fernandez Prieto, Nava Maroto, Diego Maseda Fernandez, Goran Nenadic, Julie Klein, John Keane and Robert Stevens. 'Deep learning meets ontologies: experiments to anchor the cardiovascular disease ontology in the biomedical literature'. In: *Journal of biomedical semantics* 9.1 (2018), pp. 1–24 (cit. on p. 112).

[CCL18] Borja Calvo, Josu Ceberio and Jose A. Lozano. 'Bayesian inference for algorithm ranking analysis'. In: *Proceedings of the genetic and evolutionary computation conference companion*. 2018, pp. 324–325 (cit. on pp. 97, 103).

[CD22] Agnese Chiatti and Enrico Daga. 'Neuro-symbolic learning for dealing with sparsity in cultural heritage image archives: an empirical journey'. In: (2022) (cit. on p. 112).

[Cha+20] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi and Christopher Ré. 'Low-Dimensional Hyperbolic Knowledge Graph Embeddings'. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* 58 (2020), pp. 6901–6914. URL: https://aclanthology.org/2020.acl-main.617.pdf (cit. on p. 77).

[CKH06] Beth Crandall, Gary A. Klein and Robert R. Hoffman. *Working minds: A practitioner's guide to cognitive task analysis*. MIT Press, 2006 (cit. on p. 30).

[CM09] Michel Chein and Marie-Laure Mugnier. *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer eBook Collection Computer Science. London: Springer London, 2009. ISBN: 9781848002869. DOI: 10.1007/978-1-84800-286-9 (cit. on p. 78).

[Coc+17] Michael Cochez, Petar Ristoski, Simone Paolo Ponzetto and Heiko Paulheim. 'Global RDF vector space embeddings'. In: *International Semantic Web Conference*. 2017, pp. 190–207 (cit. on pp. 76, 77).

[Coo94] Nancy J. Cooke. 'Varieties of knowledge elicitation techniques'. In: *International journal of human-computer studies* 41.6 (1994), pp. 801–849 (cit. on pp. 30, 31).

[Cyg+14] Richard Cyganiak, David Wood, Markus Lanthaler, Graham Klyne, Jeremy J. Carroll and Brian McBride. 'RDF 1.1 concepts and abstract syntax'. In: *W3C recommendation* 25.02 (2014), pp. 1–22 (cit. on p. 35).

[Dae+18] Pedram Daee, Tomi Peltola, Aki Vehtari and Samuel Kaski. 'User modelling for avoiding overfitting in interactive knowledge elicitation for prediction'. In: *23rd International Conference on Intelligent User Interfaces*. 2018, pp. 305–310 (cit. on p. 34).

[Dai+20] Yuanfei Dai, Shiping Wang, Neal N. Xiong and Wenzhong Guo. 'A survey on knowledge graph embedding: Approaches, applications and benchmarks'. In: *Electronics* 9.5 (2020), p. 750 (cit. on p. 76).

[dAm20] Claudia d'Amato. 'Machine learning for the semantic web: Lessons learnt and next research directions'. In: *Semantic Web* 11.1 (2020), pp. 195–203. ISSN: 1570-0844 (cit. on p. 111).

[dAv+15] Artur d'Avila Garcez, Tarek R. Besold, Luc de Raedt, Peter Földiak, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Luis C. Lamb, Risto Miikkulainen and Daniel L. Silver. 'Neural-symbolic learning and reasoning: contributions and challenges'. In: *2015 AAAI Spring Symposium Series*. 2015 (cit. on p. 109).

[DBV16] Michaël Defferrard, Xavier Bresson and Pierre Vandergheynst. 'Convolutional neural networks on graphs with fast localized spectral filtering'. In: *Advances in Neural Information Processing Systems* 29 (2016) (cit. on p. 77).

[Del+22] Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Marcus Hutter, Shane Legg and Pedro A. Ortega. 'Neural networks and the chomsky hierarchy'. In: *arXiv preprint arXiv:2207.02098* (2022) (cit. on p. 109).

[Det+18] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp and Sebastian Riedel. 'Convolutional 2d knowledge graph embeddings'. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018 (cit. on pp. 76, 77).

[Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. 'Bert: Pre-training of deep bidirectional transformers for language understanding'. In: *arXiv preprint arXiv:1810.04805* (2018) (cit. on p. 112).

[dL23] Artur d'Avila Garcez and Luis C. Lamb. 'Neurosymbolic AI: The 3rd wave'. In: *Artificial Intelligence Review* (2023), pp. 1–20 (cit. on pp. 109–111).

[Doa76] David P. Doane. 'Aesthetic frequency classifications'. In: *The American Statistician* 30.4 (1976), pp. 181–183 (cit. on p. 51).

[DP97] V Deslandres and Henri Pierreval. 'Knowledge acquisition issues in the design of decision support systems in quality control'. In: *European journal of operational research* 103.2 (1997), pp. 296–311 (cit. on p. 32).

[DR17] Mahmoud Dinar and David W. Rosen. 'A design for additive manufacturing ontology'. In: *Journal of Computing and Information Science in Engineering* 17.2 (2017) (cit. on p. 35).

[DV22] Marjolein Deryck and Joost Vennekens. 'Knowledge Base Systems in practice: Approaches, application areas and limitations'. In: *KU Leuven* (2022) (cit. on p. 75).

[EFG22] T. Eickhoff, S. Forte and J. C. Göbel. 'Approach for Developing Digital Twins of Smart Products Based on Linked Lifecycle Information'. In: *Proceedings of the Design Society* 2 (2022), pp. 1559–1568 (cit. on p. 12).

[Far+14] Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy and Noah A. Smith. 'Retrofitting word vectors to semantic lexicons'. In: *arXiv preprint arXiv:1411.4166* (2014) (cit. on p. 112).

[Fat+20] Bahare Fatemi, Perouz Taslakian, David Vázquez and David Poole. 'Knowledge Hypergraphs: Prediction Beyond Binary Relations'. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. Ed. by Christian Bessiere. ijcai.org, 2020, pp. 2191–2197. DOI: 10.24963/ijcai.2020/303 (cit. on pp. 75, 76).

[Fat+21] Bahare Fatemi, Perouz Taslakian, David Vazquez and David Poole. 'Knowledge Hypergraph Embedding Meets Relational Algebra'. In: *arXiv preprint arXiv:2102.09557* (2021) (cit. on pp. 76, 79).

[FD81] David Freedman and Persi Diaconis. 'On the histogram as a density estimator: L 2 theory'. In: *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 57.4 (1981), pp. 453–476 (cit. on pp. 25, 51).

[FKH18] Stefan Falkner, Aaron Klein and Frank Hutter. 'BOHB: Robust and Efficient Hyperparameter Optimization at Scale'. In: *Proceedings of the 35th International Conference on Machine Learning*. 2018, pp. 1436–1445 (cit. on p. 121).

[FO17] Carl Benedikt Frey and Michael A. Osborne. 'The future of employment: How susceptible are jobs to computerisation?' In: *Technological forecasting and social change* 114 (2017), pp. 254–280 (cit. on p. 4).

[Fra16] Gernot Frank. *Durchgängiges mechatronisches Engineering für Sondermaschinen*. Stuttgart: Fraunhofer Verlag, 2016 (cit. on p. 3).

[Fu+20] Xinyu Fu, Jiani Zhang, Ziqiao Meng and Irwin King. 'Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding'. In: *Proceedings of The Web Conference 2020*. 2020, pp. 2331–2341 (cit. on p. 77).

[GA12] Tatiana Gavrilova and Tatiana Andreeva. 'Knowledge elicitation techniques in a knowledge management context'. In: *Journal of Knowledge Management* 16.4 (2012), pp. 523–537 (cit. on p. 31).

[Gad+20] Mohamed H. Gad-Elrab, Daria Stepanova, Trung-Kien Tran, Heike Adel and Gerhard Weikum. 'Excut: Explainable embedding-based clustering over knowledge graphs'. In: *International Semantic Web Conference*. 2020, pp. 218–237 (cit. on p. 34).

[Gal+15] Luis Galárraga, Christina Teflioudi, Katja Hose and Fabian M. Suchanek. 'Fast rule mining in ontological knowledge bases with AMIE $$+$$ +'. In: *The VLDB Journal* 24.6 (2015), pp. 707–730. ISSN: 1066-8888. DOI: 10.1007/s00778-015-0394-1 (cit. on p. 34).

[Gar+19] Vicente García, J. Salvador Sánchez, Luis Alberto Rodríguez-Picón, Luis Carlos Méndez-González and Humberto de Jesús Ochoa-Domínguez. 'Using regression models for predicting the product quality in a tubing extrusion process'. In: *Journal of Intelligent Manufacturing* 30 (2019), pp. 2535–2544. ISSN: 1572-8145 (cit. on p. 20).

[Gau+19] Manas Gaur, Amanuel Alambo, Joy Prakash Sain, Ugur Kursuncu, Krishnaprasad Thirunarayan, Ramakanth Kavuluru, Amit Sheth, Randy Welton and Jyotishman Pathak. 'Knowledge-aware assessment of severity of suicide risk for early intervention'. In: *The world wide web conference*. 2019, pp. 514–525 (cit. on p. 112).

[Ges+21] Genet Asefa Gesese, Russa Biswas, Mehwish Alam and Harald Sack. 'A survey on knowledge graph embeddings with literals: Which model links better literal-ly?' In: *Semantic Web* 12.4 (2021), pp. 617–647. ISSN: 1570-0844 (cit. on pp. 76, 84, 94, 100).

[Ges21] Genet Asefa Gesese. 'Leveraging Literals for Knowledge Graph Embeddings'. In: *Proceedings of the Doctoral Consortium at ISWC 2021, co-located with 20th International Semantic Web Conference (ISWC 2021). Ed.: V. Tamma*. 2021, p. 9 (cit. on p. 86).

[GFS21] Manas Gaur, Keyur Faldu and Amit Sheth. 'Semantics of the black-box: Can knowledge graphs help make deep learning systems more interpretable and explainable?' In: *IEEE Internet Computing* 25.1 (2021), pp. 51–59 (cit. on p. 113).

[GL07] Sébastien Gebus and Kauko Leiviskä. 'Defect-related knowledge acquisition for decision support systems in electronics assembly.' In: *ICINCO-ICSO*. 2007, pp. 270–275 (cit. on pp. 32, 33, 37).

[GL16] Aditya Grover and Jure Leskovec. 'node2vec: Scalable feature learning for networks'. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 855–864 (cit. on p. 77).

[GLu20] Irlán Grangel-González, Felix Lösch and Anees ul Mehdi. 'Knowledge graphs for efficient integration and access of manufacturing data'. In: *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vol. 1. 2020, pp. 93–100 (cit. on p. 75).

[Gra+17] Christelle Grandvallet, Franck Pourroy, Guy Prudhomme, Frédéric Vignat et al. 'From elicitation to structuring of additive manufacturing knowledge'. In: *DS 87-6 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 6: Design Information and Knowledge, Vancouver, Canada, 21-25.08. 2017*. 2017, pp. 141–150 (cit. on p. 32).

[Gra19] Michael S. A. Graziano. *Rethinking consciousness: a scientific theory of subjective experience*. WW Norton & Company, 2019 (cit. on p. 110).

[GRN20] Noa Garcia, Benjamin Renoust and Yuta Nakashima. 'ContextNet: representation and exploration for painting classification and retrieval in context'. In: *International Journal of Multimedia Information Retrieval* 9.1 (2020), pp. 17–30 (cit. on pp. 112, 113, 116, 118, 122, 126, 137).

[GRS15] Ian Gibson, David Rosen and Brent Stucker. *Additive Manufacturing Technologies - 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. Springer-Verlag, 2015 (cit. on p. 15).

[Gru89] Thomas R. Gruber. 'Automated Knowledge Acquisition for Strategic Knowledge'. In: *Machine Learning* 4.3-4 (1989), pp. 293–336 (cit. on p. 30).

[GS18] Vıctor Gutiérrez-Basulto and Steven Schockaert. 'From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules'. In: *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning*. 2018 (cit. on p. 78).

[GS21] Claudio Gutiérrez and Juan F. Sequeda. 'Knowledge graphs'. In: *Communications of the ACM* 64.3 (2021), pp. 96–104. ISSN: 0001-0782 (cit. on p. 34).

[Gue17] David A. Guerra-Zubiaga. 'Tacit knowledge elicitation techniques applied to Complex Manufacturing Processes'. In: *ASME International Mechanical Engineering Congress and Exposition*. Vol. 58356. 2017, V002T02A051 (cit. on p. 32).

[Guo+16] Shu Guo, Quan Wang, Lihong Wang, Bin Wang and Li Guo. 'Jointly embedding knowledge graphs and logical rules'. In: *Proceedings of the 2016 conference on empirical methods in natural language processing*. 2016, pp. 192–202 (cit. on p. 78).

[Guo+18] Shu Guo, Quan Wang, Lihong Wang, Bin Wang and Li Guo. 'Knowledge Graph Embedding With Iterative Guidance From Soft Rules'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (2018). ISSN: 2374-3468. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11918 (cit. on p. 78).

[Ham20] Hamilton. 'Graph Representation Learning'. In: *Synthesis Lectures on Artifical Intelligence and Machine Learning* 14.3 (2020), pp. 1–159 (cit. on pp. 76, 77).

[Han+20] Lea Hannola, Francisco Lacueva-Pérez, Paolo Pretto, Alexander Richter, Marlene Schafler and Melanie Steinhüser. 'Assessing the impact of socio-technical interventions on shop floor work practices'. In: *International Journal of Computer Integrated Manufacturing* 33.6 (2020), pp. 550–571 (cit. on pp. 3, 4, 137).

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on p. 116).

[Hei+23] Michael Heider, Helena Stegherr, Richard Nordsieck and Jörg Hähner. 'Assessing Model Requirements for Explainable AI: A Template and Exemplary Case Study'. In: *Artificial Life* (in press) (2023) (cit. on p. 137).

[HGC95] David Heckerman, Dan Geiger and David M. Chickering. 'Learning Bayesian networks: The combination of knowledge and statistical data'. In: *Machine Learning* 20 (1995), pp. 197–243 (cit. on p. 112).

[Hit+22] Pascal Hitzler, Aaron Eberhart, Monireh Ebrahimi, Md Kamruzzaman Sarker and Lu Zhou. 'Neuro-symbolic approaches in artificial intelligence'. In: *National Science Review* 9.6 (2022), nwac035 (cit. on pp. 4, 109).

[Hit27] Frank L. Hitchcock. 'The expression of a tensor or a polyadic as a sum of products'. In: *Journal of Mathematics and Physics* 6.1-4 (1927), pp. 164–189 (cit. on p. 76).

[Ho+18] Vinh Thinh Ho, Daria Stepanova, Mohamed H. Gad-Elrab, Evgeny Kharlamov and Gerhard Weikum. 'Rule learning from knowledge graphs guided by embedding models'. In: *International Semantic Web Conference*. 2018, pp. 72–90 (cit. on p. 88).

[Hog+21] Aidan Hogan et al. 'Knowledge graphs'. In: *ACM Computing Surveys (CSUR)* 54.4 (2021), pp. 1–37 (cit. on pp. 34, 35, 78).

[HS19] Ji Han and Dirk Schaefer. 'An Ontology for Supporting Digital Manufacturability Analysis'. In: *Procedia CIRP* 81 (2019), pp. 850–855 (cit. on p. 35).

[HSB20] Lorenz Hörner, Markus Schamberger and Freimut Bodendorf. 'Externalisierung von prozess-spezifischem mitarbeiterwissen im produktionsumfeld'. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 115.6 (2020), pp. 413–417 (cit. on pp. 32, 82).

[HSB22] Lorenz Hörner, Markus Schamberger and Freimut Bodendorf. 'Using Tacit Expert Knowledge to Support Shop-floor Operators Through a Knowledge-based Assistance System'. In: *Computer Supported Cooperative Work-The Journal of Collaborative Computing* (2022). DOI: 10.1007/s10606-022-09445-4 (cit. on pp. 3, 4, 20, 21, 30, 32, 37).

[HT07] Joseph E. Hoag and Craig W. Thompson. 'A parallel general-purpose synthetic data generator'. In: *ACM SIGMOD Record* 36.1 (2007), pp. 19–24 (cit. on p. 22).

[Hub+18] Thomas Hubauer, Steffen Lamparter, Peter Haase and Daniel Markus Herzig. 'Use Cases of the Industrial Knowledge Graph at Siemens'. In: *International Semantic Web Conference (P&D/Industry/BlueSky)*. 2018 (cit. on p. 75).

[HYL17a] Will Hamilton, Zhitao Ying and Jure Leskovec. 'Inductive representation learning on large graphs'. In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on p. 77).

[HYL17b] William L. Hamilton, Rex Ying and Jure Leskovec. 'Representation learning on graphs: Methods and applications'. In: *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 40.3 (2017), pp. 52–74 (cit. on p. 96).

[Jai+21] Nitisha Jain, Jan-Christoph Kalo, Wolf-Tilo Balke and Ralf Krestel. 'Do Embeddings Actually Capture Knowledge Graph Semantics?' In: *Extended Semantic Web Conference*. 2021, pp. 143–159 (cit. on p. 77).

[Jes+05] Daniel R. Jeske, Behrokh Samadi, Pengyue J. Lin, Lan Ye, Sean Cox, Rui Xiao, Ted Younglove, Minh Ly, Douglas Holt and Ryan Rich. 'Generation of synthetic data sets for evaluating the accuracy of knowledge discovery systems'. In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 2005, pp. 756–762 (cit. on p. 22).

[Ji+21] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen and S. Yu Philip. 'A survey on knowledge graphs: Representation, acquisition, and applications'. In: *IEEE Transactions on Neural Networks and Learning Systems* (2021) (cit. on p. 76).

[Joh+19] Teegan L. Johnson, Sarah R. Fletcher, W. Baker and Rebecca L. Charles. 'How and why we need to capture tacit knowledge in manufacturing: Case studies of visual inspection'. In: *Applied ergonomics* 74 (2019), pp. 1–9 (cit. on p. 32).

[Kah11] Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011 (cit. on p. 109).

[Kal+20] Elem Güzel Kalaycı, Irlan Grangel González, Felix Lösch, Guohui Xiao, Evgeny Kharlamov, Diego Calvanese et al. 'Semantic integration of Bosch manufacturing data using virtual knowledge graphs'. In: *International Semantic Web Conference*. 2020, pp. 464–481 (cit. on p. 75).

[Kat20] Badarinath Katti. 'Ontology-Based Approach to Decentralized Production Control in the Context of Cloud Manufacturing Execution Systems'. PhD thesis. TU Kaiserslautern, 2020 (cit. on p. 33).

[KD99] Justin Kruger and David Dunning. 'Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments.' In: *Journal of personality and social psychology* 77.6 (1999), p. 1121 (cit. on p. 45).

[KGS19] Ugur Kursuncu, Manas Gaur and Amit Sheth. 'Knowledge infused learning (K-IL): Towards deep incorporation of knowledge in deep learning'. In: *arXiv preprint arXiv:1912.00512* (2019) (cit. on p. 112).

[KGS20] Ugur Kursuncu, Manas Gaur and Amit Sheth. 'Knowledge infused learning (K-IL): Towards deep incorporation of knowledge in deep learning'. In: *Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice (AAAI-MAKE)* (2020) (cit. on pp. 96, 113, 127).

[Kha+16] Evgeny Kharlamov, Bernardo Cuenca Grau, Ernesto Jiménez-Ruiz, Steffen Lamparter, Gulnar Mehdi, Martin Ringsquandl, Yavor Nenov, Stephan Grimm, Mikhail Roshchin and Ian Horrocks. 'Capturing industrial information models with ontologies and constraints'. In: *International Semantic Web Conference*. Springer. 2016, pp. 325–343 (cit. on p. 33).

[KK17] Tomasz Kozior and Czesław Kundera. 'Evaluation of the Influence of Parameters of FDM Technology on the Selected Mechanical Properties of Models'. In: *Procedia Engineering* 192 (2017), pp. 463–468. ISSN: 18777058. DOI: 10.1016/j.proeng.2017.06.080 (cit. on p. 17).

[KM06] Lukasz A. Kurgan and Petr Musilek. 'A survey of knowledge discovery and data mining process models'. In: *The Knowledge Engineering Review* 21.1 (2006), pp. 1–24 (cit. on p. 31).

[KN11] Kevin B. Korb and Ann E. Nicholson. *Bayesian artificial intelligence*. 2nd ed. Boca Raton, FL: Chapman & Hall/CRC Computer Science & Data Analysis, 2011 (cit. on p. 31).

[Ko+19] Hyunwoong Ko, Paul Witherell, Ndeye Y. Ndiaye and Yan Lu. 'Machine learning based continuous knowledge engineering for additive manufacturing'. In: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. 2019, pp. 648–654 (cit. on pp. 34, 35).

[Ko+21] Hyunwoong Ko, Paul Witherell, Yan Lu, Samyeon Kim and David W. Rosen. 'Machine learning and knowledge graph based design rule construction for additive manufacturing'. In: *Additive Manufacturing* 37 (2021), p. 101620 (cit. on p. 34).

[KP18a] Seyed Mehran Kazemi and David Poole. 'Simple embedding for link prediction in knowledge graphs'. In: *Advances in Neural Information Processing Systems* 31 (2018) (cit. on p. 76).

[KP18b] Saad Khan and Simon Parkinson. 'Eliciting and utilising knowledge for security event log analysis: An association rule mining and automated planning approach'. In: *Expert Systems with Applications* 113 (2018), pp. 116–127 (cit. on p. 34).

[Kra02] David R. Krathwohl. 'A revision of Bloom's taxonomy: An overview'. In: *Theory into practice* 41.4 (2002), pp. 212–218 (cit. on pp. 4, 29, 30, 75, 81).

[Kri+19] Agustinus Kristiadi, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann and Asja Fischer. 'Incorporating literals into knowledge graph embeddings'. In: *International Semantic Web Conference*. 2019, pp. 347–363 (cit. on pp. 94, 97).

[KSH12] Oliver Korn, Albrecht Schmidt and Thomas Hörz. 'Assistive systems in production environments: exploring motion recognition and gamification'. In: *Proceedings of the 5th international conference on pervasive technologies related to assistive environments*. 2012, pp. 1–5 (cit. on p. 3).

[Kur+21] Martin von Kurnatowski, Jochen Schmid, Patrick Link, Rebekka Zache, Lukas Morand, Torsten Kraft, Ingo Schmidt, Jan Schwientek and Anke Stoll. 'Compensating data shortages in manufacturing with monotonicity knowledge'. In: *Algorithms* 14.12 (2021), p. 345 (cit. on pp. 51, 112, 114, 131, 135).

[KW16] Thomas N. Kipf and Max Welling. 'Semi-supervised classification with graph convolutional networks'. In: *arXiv preprint arXiv:1609.02907* (2016) (cit. on p. 77).

[LA16] George Leu and Hussein Abbass. 'A multi-disciplinary review of knowledge acquisition methods: From human to autonomous eliciting agents'. In: *Knowledge-Based Systems* 105 (2016), pp. 1–22 (cit. on pp. 30–32, 34, 37, 136).

[Lam+20] Luis C. Lamb, Artur d'Avila Garcez, Marco Gori, Marcelo Prates, Pedro Avelar and Moshe Vardi. 'Graph neural networks meet neural-symbolic computing: A survey and perspective'. In: *arXiv preprint arXiv:2003.00330* (2020) (cit. on p. 110).

[LB20] Trond Linjordet and Krisztian Balog. 'Sanitizing synthetic training data generation for question answering over knowledge graphs'. In: *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 2020, pp. 121–128 (cit. on p. 22).

[LC19] Guillaume Lample and François Charton. 'Deep learning for symbolic mathematics'. In: *arXiv preprint arXiv:1912.01412* (2019) (cit. on p. 110).

[Lin+15] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu and Xuan Zhu. 'Learning entity and relation embeddings for knowledge graph completion'. In: *Twenty-ninth AAAI conference on artificial intelligence*. 2015 (cit. on p. 77).

[LLJ17] Don Libes, David Lechevalier and Sanjay Jain. 'Issues in synthetic data generation for advanced manufacturing'. In: *2017 IEEE International Conference on Big Data (Big Data)*. 2017, pp. 1746–1754 (cit. on pp. 22, 23).

[Löh+18] Mario Löhrer, Jacqueline Lemm, Daniel Kerpen, Marco Saggiomo and Yves-Simon Gloy. 'Soziotechnische Assistenzsysteme für die Produktionsarbeit in der Textilbranche: Auswirkungen von Industrie 4.0 auf die Arbeit in einer Weberei'. In: *Zukunft der Arbeit–Eine praxisnahe Betrachtung* (2018), pp. 73–85 (cit. on p. 3).

[LT+75] Harold A. Linstone, Murray Turoff et al. *The Delphi Method*. Addison-Wesley Reading, MA, 1975 (cit. on p. 33).

[LYL20] Yu Liu, Quanming Yao and Yong Li. 'Generalizing tensor decomposition for n-ary relational knowledge bases'. In: *Proceedings of The Web Conference 2020*. 2020, pp. 1104–1114 (cit. on p. 76).

[Ma+19] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma and Xiang Ren. 'Jointly learning explainable rules for recommendation with knowledge graph'. In: *The world wide web conference*. 2019, pp. 1210–1221 (cit. on p. 88).

[MAC22] Johannes Messner, Ralph Abboud and Ismail Ilkan Ceylan. 'Temporal knowledge graph completion using box embeddings'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 2022, pp. 7779–7787 (cit. on p. 136).

[Man+18] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester and Luc de Raedt. 'Deepproblog: Neural probabilistic logic programming'. In: *Advances in Neural Information Processing Systems* 31 (2018) (cit. on p. 110).

[Mao+19] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum and Jiajun Wu. 'The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision'. In: *arXiv preprint arXiv:1904.12584* (2019) (cit. on p. 110).

[Maz+16] Luca Mazzola, Patrick Kapahnke, Marko Vujic and Matthias Klusch. 'CDM-Core: A Manufacturing Domain Ontology in OWL2 for Production and Maintenance'. In: *KEOD*. 2016, pp. 136–143 (cit. on p. 35).

[Mei+19] Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli and Heiner Stuckenschmidt. 'Anytime Bottom-Up Rule Learning for Knowledge Graph Completion'. In: *IJCAI*. 2019, pp. 3137–3143 (cit. on p. 34).

[Mik+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado and Jeff Dean. 'Distributed representations of words and phrases and their compositionality'. In: *Advances in Neural Information Processing Systems* 26 (2013) (cit. on pp. 76, 77, 112).

[Mil+90] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross and Katherine J. Miller. 'Introduction to WordNet: An on-line lexical database'. In: *International journal of lexicography* 3.4 (1990), pp. 235–244 (cit. on p. 77).

[MMB15] Omar A. Mohamed, Syed H. Masood and Jahar L. Bhowmik. 'Optimization of fused deposition modeling process parameters: a review of current research and future prospects'. In: *Advances in Manufacturing* 3.1 (2015), pp. 42–53. ISSN: 2095-3127. DOI: 10.1007/s40436-014-0097-7 (cit. on p. 15).

[MO19] Romy Müller and Lukas Oehm. 'Process industries versus discrete processing: how system characteristics affect operator tasks'. In: *Cognition, Technology & Work* 21.2 (2019), pp. 337–356 (cit. on pp. 9, 10, 12, 20).

[Mog+18] Aditya Mogadala, Umanga Bista, Lexing Xie and Achim Rettinger. 'Knowledge guided attention and inference for describing images containing unseen objects'. In: *European Semantic Web Conference*. 2018, pp. 415–429 (cit. on p. 113).

[Mon+21] Sebastian Monka, Lavdim Halilaj, Stefan Schmid and Achim Rettinger. 'Learning visual models using a knowledge graph as a trainer'. In: *International Semantic Web Conference*. 2021, pp. 357–373 (cit. on p. 113).

[Mor+16] Friedrich Morlock, Thom Wienbruch, Stefan Leineweber, Dieter Kreimeier and Bernd Kuhlenkötter. 'Industrie 4.0-Transformation für produzierende Unternehmen'. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 111.5 (2016), pp. 306–309 (cit. on p. 4).

[MPT20] Mona Mohamed, Sharma Pillutla and Stella Tomasi. 'Extraction of knowledge from open government data'. In: *VINE Journal of Information and Knowledge Management Systems* (2020) (cit. on p. 77).

[Mrk+16] Nikola Mrkšić, Diarmuid O. Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen and Steve Young. 'Counter-fitting word vectors to linguistic constraints'. In: *arXiv preprint arXiv:1603.00892* (2016) (cit. on p. 112).

[Mül+22] Dennis Müller, Michael März, Stephan Scheele and Ute Schmid. 'An interactive explanatory ai system for industrial quality control'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 2022, pp. 12580–12586 (cit. on p. 114).

[Mur+18] Nikhil Muralidhar, Mohammad Raihanul Islam, Manish Marwah, Anuj Karpatne and Naren Ramakrishnan. 'Incorporating prior domain knowledge into deep neural networks'. In: *2018 IEEE international conference on big data (big data)*. 2018, pp. 36–45 (cit. on p. 112).

[NH10] Vinod Nair and Geoffrey E. Hinton. 'Rectified linear units improve restricted boltzmann machines'. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814 (cit. on p. 121).

[NHF22] Huong Giang Nguyen, Resul Habiboglu and Jörg Franke. 'Enabling deep learning using synthetic data: A case study for the automotive wiring harness manufacturing'. In: *Procedia CIRP* 107 (2022), pp. 1263–1268 (cit. on p. 22).

[Nic+20] Ann E. Nicholson, Kevin B. Korb, Erik P. Nyberg, Michael Wybrow, Ingrid Zukerman, Steven Mascaro, Shreshth Thakur, Abraham Oshni Alvandi, Jeff Riley, Ross Pearson et al. 'BARD: A structured technique for group elicitation of Bayesian networks to support analytic reasoning'. In: *arXiv preprint arXiv:2003.01207* (2020) (cit. on pp. 33, 38).

[Nor+19] Richard Nordsieck, Michael Heider, Andreas Angerer and Jörg Hähner. 'Towards Automated Parameter Optimisation of Machinery by Persisting Expert Knowledge'. In: *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2019 - Volume 1, Prague, Czech Republic, July 29-31, 2019*. Ed. by Oleg Gusikhin, Kurosh Madani and Janan Zaytoon. SCITEPRESS, 2019, pp. 406–413. DOI: 10.5220/0007953204060413 (cit. on pp. 15, 16).

[Nor+21] Richard Nordsieck, Michael Heider, Anton Winschel and Jörg Hähner. 'Knowledge Extraction via Decentralized Knowledge Graph Aggregation'. In: *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*. 2021, pp. 92–99 (cit. on pp. 9, 15, 27).

[Nor+22a] Richard Nordsieck, Michael Heider, Alwin Hoffmann and Jörg Hähner. 'Reliability-Based Aggregation of Heterogeneous Knowledge to Assist Operators in Manufacturing'. In: *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*. 2022, pp. 131–138 (cit. on pp. 9, 27, 82).

[Nor+22b] Richard Nordsieck, Michael Heider, Anton Hummel and Jörg Hähner. 'A Closer Look at Sum-based Embeddings for Knowledge Graphs Containing Procedural Knowledge'. In: *Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022)*. Ed. by Mehwish Alam, Davide Buscaldi, Michael Cochez, Francesco Osborne and Reforgiato Recupero Diego. 2022. URL: https://ceur-ws.org/Vol-3342/ (cit. on pp. 73, 81).

[Nor+22c] Richard Nordsieck, Michael Heider, Anton Hummel, Alwin Hoffmann and Jörg Hähner. 'Towards Models of Conceptual and Procedural Operator Knowledge'. In: *Proceedings of the First International Workshop on Semantic Industrial Information Modelling (SemIIM 2022) co-located with the 19th Extended Semantic Web Conference ESWC 2022*. Ed. by Arild Waaler, Evgeny Kharlamov, Baifan Zhou and Dongzhuoran Zhou. 2022. URL: https://ceur-ws.org/Vol-3355/richardnordsieck.pdf (cit. on pp. 73, 81).

[Nor+23a] Richard Nordsieck, Anton Hummel, Michael Heider, Friedline Jessica and Jörg Hähner. 'Sequential Knowledge Elicitation in Data-Based Knowledge Extraction'. In: *Proceedings of the 12th Knowledge Capture Conference (submitted to)*. 2023 (cit. on p. 27).

[Nor+23b] Richard Nordsieck, André Schweizer, Michael Heider and Jörg Hähner. 'PDPK: A Framework to Synthesise Process Data and Corresponding Procedural Knowledge for Manufacturing'. In: *arXiv preprint arXiv:2308.08371* (2023) (cit. on pp. 21, 22, 27, 73, 81).

[Noy+06] Natasha Noy, Alan Rector, Pat Hayes and Chris Welty. 'Defining n-ary relations on the semantic web'. In: *W3C working group note* (2006). URL: https://www.w3.org/TR/swbp-n-aryRelations/#useCase1 (cit. on pp. 75, 85).

[Noy+19] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson and Jamie Taylor. 'Industry-scale knowledge graphs: lessons and challenges'. In: *Communications of the ACM* 62.8 (2019), pp. 36–43. ISSN: 00010782 (cit. on pp. 75, 87, 89).

[NTK11] Maximilian Nickel, Volker Tresp and Hans-Peter Kriegel. 'A three-way model for collective learning on multi-relational data'. In: *Icml*. 2011 (cit. on p. 76).

[OHa19] Anthony O'Hagan. 'Expert knowledge elicitation: subjective but scientific'. In: *The American Statistician* 73.sup1 (2019), pp. 69–81 (cit. on p. 33).

[Par+14] Paolo Pareti, Benoit Testu, Ryutaro Ichise, Ewan Klein and Adam Barker. 'Integrating know-how into the linked data cloud'. In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer. 2014, pp. 385–396 (cit. on p. 33).

[PAS14] Bryan Perozzi, Rami Al-Rfou and Steven Skiena. 'Deepwalk: Online learning of social representations'. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 701–710 (cit. on p. 77).

[PDT12] Hervé Panetto, Michele Dassisti and Angela Tursi. 'ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment'. In: *Advanced Engineering Informatics* 26.2 (2012), pp. 334–348 (cit. on p. 35).

[Pet+17] Niklas Petersen, Lavdim Halilaj, Irlán Grangel-González, Steffen Lohmann, Christoph Lange and Sören Auer. 'Realizing an RDF-based information model for a manufacturing company–a case study'. In: *International semantic web conference*. 2017, pp. 350–366 (cit. on p. 75).

[Pet+18] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Lee Kenton and Luke Zettlemoyer. 'Deep contextualized word representations'. In: *CoRR* abs/1802.05365 (2018) (cit. on p. 112).

[Pet+20] Ana Petrovska, Sergio Quijano, Ilias Gerostathopoulos and Alexander Pretschner. 'Knowledge aggregation with subjective logic in multi-agent self-adaptive cyber-physical systems'. In: *Proceedings of the IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. 2020, pp. 149–155 (cit. on p. 32).

[PHP22] Jan Portisch, Nicolas Heist and Heiko Paulheim. 'Knowledge graph embedding for data mining vs. knowledge graph embedding for link prediction–two sides of the same coin?' In: *Semantic Web* 13.3 (2022), pp. 399–422. ISSN: 1570-0844 (cit. on pp. 76, 77, 93, 94, 97).

[Pil+20] Wenzel Pilar von Pilchau, Varun Gowtham, Maximilian Gruber, Matthias Riedl, Nikolaos-Stefanos Koutrakis, Jawad Tayyub, Jörg Hähner, Sascha Eichstädt, Eckart Uhlmann, Julian Polte et al. 'An Architectural Design for Measurement Uncertainty Evaluation in Cyber-Physical Systems'. In: *Annals of Computer Science and Information Systems* 22 (2020), pp. 53–57 (cit. on p. 81).

[Pla+19] Marie Platenius-Mohr, Somayeh Malakuti, Sten Grüner and Thomas Goldschmidt. 'Interoperable digital twins in iiot systems by transformation of information models: A case study with asset administration shell'. In: *Proceedings of the 9th International Conference on the Internet of Things*. 2019, pp. 1–8 (cit. on p. 81).

[PP21] Jan Portisch and Heiko Paulheim. 'Putting RDF2vec in order'. In: *arXiv preprint arXiv:2108.05280* (2021) (cit. on p. 94).

[PP22a] Jan Portisch and Heiko Paulheim. 'The DLCC Node Classification Benchmark for Analyzing Knowledge Graph Embeddings'. In: *International Semantic Web Conference*. 2022, pp. 592–609 (cit. on p. 77).

[PP22b] Jan Portisch and Heiko Paulheim. 'Walk this way! entity walks and property walks for rdf2vec'. In: *arXiv preprint arXiv:2204.02777* (2022) (cit. on p. 77).

[PRF08] Ely Laureano Paiva, Aleda V. Roth and Jaime Evaldo Fensterseifer. 'Organizational knowledge and the manufacturing strategy process: a resource-based view analysis'. In: *Journal of Operations Management* 26.1 (2008), pp. 115–132 (cit. on pp. 3, 30).

[PSM14] Jeffrey Pennington, Richard Socher and Christopher D. Manning. 'Glove: Global vectors for word representation'. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543 (cit. on p. 112).

[RB22] Maximilian-Peter Radtke and Jurgen Bock. 'Expert Knowledge Induced Logic Tensor Networks: A Bearing Fault Diagnosis Case Study'. In: *PHM Society European Conference*. Vol. 7. 2022, pp. 421–431 (cit. on p. 114).

[RD03] Matthew Richardson and Pedro Domingos. 'Learning with knowledge from multiple experts'. In: *Proceedings of the 20th international conference on machine learning (ICML-03)*. 2003, pp. 624–631 (cit. on p. 112).

[REH23] REHAU Industries SE & Co. KG. *Vielfalt, Service & Know-how – REHAU Kantenbänder für Möbelprofis*. 2023. URL: https://www.youtube.com/watch?v=Q08bZBMV75s (visited on 15/03/2023) (cit. on p. 22).

[Rie+20] Ryan Riegel, Alexander Gray, Francois Luus, Naweed Khan, Ndivhuwo Makondo, Ismail Yunus Akhalwaya, Haifeng Qian, Ronald Fagin, Francisco Barahona, Udit Sharma et al. 'Logical neural networks'. In: *arXiv preprint arXiv:2006.13155* (2020) (cit. on p. 110).

[Rin+15] Martin Ringsquandl, Steffen Lamparter, Sebastian Brandt, Thomas Hubauer and Raffaello Lepratti. 'Semantic-guided feature selection for industrial automation systems'. In: *International Semantic Web Conference*. 2015, pp. 225–240 (cit. on p. 114).

[Rin+17] Martin Ringsquandl, Evgeny Kharlamov, Daria Stepanova, Steffen Lamparter, Raffaello Lepratti, Ian Horrocks and Peer Kröger. 'On event-driven knowledge graph completion in digital factories'. In: *2017 IEEE International Conference on Big Data (Big Data)*. 2017, pp. 1676–1681 (cit. on pp. 75, 78).

[Rin+18] Martin Ringsquandl, Evgeny Kharlamov, Daria Stepanova, Marcel Hildebrandt, Steffen Lamparter, Raffaello Lepratti, Ian Horrocks and Peer Kröger. 'Event-enhanced learning for KG completion'. In: *European Semantic Web Conference*. 2018, pp. 541–559 (cit. on pp. 78, 87).

[Rin19] Martin Ringsquandl. 'Semantic-guided predictive modeling and relational learning within industrial knowledge graphs'. PhD thesis. Ludwig Maximilian University, 2019 (cit. on pp. 33, 37).

[RJC12] Andres J Ramirez, Adam C Jensen and Betty HC Cheng. 'A taxonomy of uncertainty for dynamically adaptive systems'. In: *2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE. 2012, pp. 99–108 (cit. on p. 32).

[RLL16] Martin Ringsquandl, Steffen Lamparter and Raffaello Lepratti. 'Graph-based predictions and recommendations in flexible manufacturing systems'. In: *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. 2016, pp. 6937–6942 (cit. on p. 75).

[Ros+21] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata and Paolo Merialdo. 'Knowledge graph embedding for link prediction: A comparative analysis'. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15.2 (2021), pp. 1–49 (cit. on pp. 76, 77).

[Roy+22a] Kaushik Roy, Manas Gaur, Vipula Rawte, Ashwin Kalyan and Amit Sheth. 'ProKnow: Process Knowledge for Safety Constrained and Explainable Question Generation for Mental Health Diagnostic Assistance'. In: *UMBC Faculty Collection* (2022) (cit. on pp. 113, 137).

[Roy+22b] Kaushik Roy, Manas Gaur, Qi Zhang and Amit Sheth. 'Process knowledge-infused learning for suicidality assessment on social media'. In: *arXiv preprint arXiv:2204.12560* (2022) (cit. on p. 113).

[RP16] Petar Ristoski and Heiko Paulheim. 'Rdf2vec: Rdf graph embeddings for data mining'. In: *International Semantic Web Conference*. 2016, pp. 498–514 (cit. on pp. 77, 97).

[Rud16] Sebastian Ruder. 'An overview of gradient descent optimization algorithms'. In: *arXiv preprint arXiv:1609.04747* (2016) (cit. on p. 121).

[Rue+21] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy et al. 'Informed Machine Learning–A taxonomy and survey of integrating prior knowledge into learning systems'. In: *IEEE Transactions on Knowledge and Data Engineering* 35.1 (2021), pp. 614–633. ISSN: 1041-4347 (cit. on pp. 112, 122).

[Sar+21] Md Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart and Pascal Hitzler. 'Neuro-symbolic artificial intelligence'. In: *AI Communications* 34.3 (2021), pp. 197–209 (cit. on pp. 4, 109, 110).

[SC17] Renata Stasiak-Betlejewsk and Agnieszka Czajkowska. 'Quantification of the quality problems in the construction machinery production'. In: *MATEC Web of Conferences*. Vol. 94. 2017, p. 04011 (cit. on p. 3).

[Sch+12] Pol Schumacher, Mirjam Minor, Kirstin Walter and Ralph Bergmann. 'Extraction of procedural knowledge from the web: A comparison of two workflow extraction approaches'. In: *Proceedings of the 21st International Conference on World Wide Web*. 2012, pp. 739–747 (cit. on p. 33).

[Sch+18] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne den van Berg, Ivan Titov and Max Welling. 'Modeling relational data with graph convolutional networks'. In: *European semantic web conference*. 2018, pp. 593–607 (cit. on pp. 76, 77).

[Sch+22] Max Schröder, Susanne Staehlke, Paul Groth, J. Barbara Nebe, Sascha Spors and Frank Krüger. 'Structure-based knowledge acquisition from electronic lab notebooks for research data provenance documentation'. In: *Journal of biomedical semantics* 13.1 (2022), pp. 1–22 (cit. on p. 34).

[Sey+19] Tom Seymoens, Femke Ongenae, Jacobs, Stijn Verstichel and Ann Ackaert. 'A methodology to involve domain experts and machine learning techniques in the design of human-centered algorithms'. In: *Human Work Interaction Design. Designing Engaging Automation: 5th IFIP WG 13.6 Working Conference, HWID 2018, Espoo, Finland, August 20-21, 2018, Revised Selected Papers 5*. 2019, pp. 200–214 (cit. on p. 32).

[SG16] Luciano Serafini and Artur d'Avila Garcez. 'Logic tensor networks: Deep learning and logical reasoning from data and knowledge'. In: *arXiv preprint arXiv:1606.04422* (2016) (cit. on p. 111).

[SG87] Mildred LG Shaw and Brian R Gaines. 'An interactive knowledge-elicitation technique using personal construct technology'. In: *Knowledge acquisition for expert systems*. Springer, 1987, pp. 109–136 (cit. on p. 37).

[Sha+15] Nigel Shadbolt, Paul R. Smart, Wilson and S. Sharples. 'Knowledge elicitation'. In: *Evaluation of human work* (2015), pp. 163–200 (cit. on p. 30).

[She+18] Ying Shen, Yang Deng, Min Yang, Yaliang Li, Nan Du, Wei Fan and Kai Lei. 'Knowledge-aware attentive neural network for ranking question answer pairs'. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 2018, pp. 901–904 (cit. on p. 112).

[She+22] Amit Sheth, Manas Gaur, Kaushik Roy, Revathy Venkataraman and Vedant Khandelwal. 'Process Knowledge-Infused AI: Toward User-Level Explainability, Interpretability, and Safety'. In: *IEEE Internet Computing* 26.5 (2022), pp. 76–84 (cit. on p. 113).

[Shi+16] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun and S. Yu Philip. 'A survey of heterogeneous information network analysis'. In: *IEEE Transactions on Knowledge and Data Engineering* 29.1 (2016), pp. 17–37. ISSN: 1041-4347 (cit. on p. 78).

[Shi+20] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace and Sameer Singh. 'Autoprompt: Eliciting knowledge from language models with automatically generated prompts'. In: *arXiv preprint arXiv:2010.15980* (2020) (cit. on p. 34).

[Sil+16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot et al. 'Mastering the game of Go with deep neural networks and tree search'. In: *nature* 529.7587 (2016), pp. 484–489 (cit. on p. 110).

[SPG19] Amit Sheth, Swati Padhee and Amelie Gyrard. 'Knowledge graphs and knowledge networks: The story in brief'. In: *IEEE Internet Computing* 23.4 (2019), pp. 67–75 (cit. on pp. 4, 111–113).

[SRG23] Amit Sheth, Kaushik Roy and Manas Gaur. 'Neurosymbolic Artificial Intelligence (Why, What, and How)'. In: *IEEE Intelligent Systems* 38.3 (2023), pp. 56–62. DOI: 10.1109/MIS.2023.3268724 (cit. on pp. 82, 109, 111, 122, 127, 132).

[Sri+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov. 'Dropout: a simple way to prevent neural networks from overfitting'. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958 (cit. on p. 121).

[SS15] Nigel Shadbolt and Paul R. Smart. 'Knowledge Elicitation: Methods, Tools and Techniques. In, Wilson, John Rand Sharples, Sarah (eds.) Evaluation of Human Work'. In: *Boca Raton, Florida, USA, CRC Press* 163 (2015), p. 200 (cit. on pp. 29, 30).

[ST21] Amit Sheth and Krishnaprasad Thirunarayan. 'The Duality of Data and Knowledge Across the Three Waves of AI'. In: *IT professional* 23.3 (2021), pp. 35–45 (cit. on p. 109).

[Sta22] Statistisches Bundesamt. 'Inlandsproduktberechnung – Detaillierte Jahresergebnisse 2021'. In: *Fachserie 18 Reihe 1.4* (2022). URL: https://www.statistische bibliothek.de/mir/receive/DEHeft_mods_00144587 (cit. on p. 3).

[Ste+10] Jan-Philipp Steghöfer, Rolf Kiefhaber, Karin Leichtenstern, Yvonne Bernard, Lukas Klejnowski, Wolfgang Reif, Theo Ungerer, Elisabeth André, Jörg Hähner and Christian Müller-Schloer. 'Trustworthy Organic Computing Systems: Challenges and Perspectives'. In: *Autonomic and Trusted Computing*. Ed. by Bing Xie, Juergen Branke, S. Masoud Sadjadi, Daqing Zhang and Xingshe Zhou. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 62–76. ISBN: 978-3-642-16576-4 (cit. on p. 32).

[Sun+14] Jingyu Sun, Kazuo Hiekata, Hiroyuki Yamato, Norito Nakagaki and Akiyoshi Sugawara. 'A Knowledge-Based Approach for Facilitating Design of Curved Shell Plates' Manufacturing Plans'. In: *Moving Integrated Product Development to Service Clouds in the Global Economy*. IOS Press, 2014, pp. 143–152 (cit. on p. 33).

[Sun+17] Chen Sun, Abhinav Shrivastava, Saurabh Singh and Abhinav Gupta. 'Revisiting unreasonable effectiveness of data in deep learning era'. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 843–852 (cit. on p. 112).

[Sun+18] Iiris Sundin, Tomi Peltola, Luana Micallef, Homayun Afrabandpey, Marta Soare, Muntasir Mamun Majumder, Pedram Daee, Chen He, Baris Serim, Aki Havulinna et al. 'Improving genomics-based predictions for precision medicine through active elicitation of expert knowledge'. In: *Bioinformatics* 34.13 (2018), pp. i395–i403 (cit. on p. 34).

[Sun+19] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie and Jian Tang. 'Rotate: Knowledge graph embedding by relational rotation in complex space'. In: *arXiv preprint arXiv:1902.10197* (2019) (cit. on pp. 76, 77, 97).

[SVJ23] Aziz Siyaev, Dilmurod Valiev and Geun-Sik Jo. 'Interaction with Industrial Digital Twin Using Neuro-Symbolic Reasoning'. In: *Sensors* 23.3 (2023), p. 1729 (cit. on p. 114).

[SW18] Baoxu Shi and Tim Weninger. 'Open-world knowledge graph completion'. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 2018 (cit. on p. 77).

[TBH19] Johannes Philipp Terhechte, Anna-Lena Berscheid and Ilona Horwath. 'Development of a cyber-physical simulation assistance system for partially automated optimization of the extrusion process'. In: *36th Danubia-Adria Symposium on Advances in Experimental Mechanics* (2019) (cit. on pp. 3, 4).

[TC19] Pedro Tabacof and Luca Costabello. 'Probability calibration for knowledge graph embedding models'. In: *arXiv preprint arXiv:1912.10000* (2019) (cit. on p. 77).

[TFP20] Stefan Thalmann, Angela Fessl and Viktoria Pammer-Schindler. 'How Large Manufacturing Firms Understand the Impact of Digitization: A Learning Perspective'. In: *HICSS*. 2020, pp. 1–10 (cit. on p. 137).

[THM21] Efthymia Tsamoura, Timothy Hospedales and Loizos Michael. 'Neural-symbolic integration: A compositional perspective'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 5051–5060 (cit. on p. 111).

[TM22] Hasan Tercan and Tobias Meisen. 'Machine learning and deep learning based predictive quality in manufacturing: a systematic review'. In: *Journal of Intelligent Manufacturing* 33.7 (2022), pp. 1879–1905. ISSN: 1572-8145 (cit. on pp. 4, 19, 26, 114, 115).

[Tou+15] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury and Michael Gamon. 'Representing text for joint embedding of text and knowledge bases'. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 1499–1509 (cit. on p. 77).

[Tow93] Geoffrey G. Towell. 'Symbolic knowledge and neural networks: Insertion, refinement and extraction'. In: (1993) (cit. on p. 110).

[Tro+16] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier and Guillaume Bouchard. 'Complex embeddings for simple link prediction'. In: *International conference on machine learning*. 2016, pp. 2071–2080 (cit. on pp. 76, 77, 97).

[Usm+11] Zahid Usman, Robert Ian Marr Young, Nitishal Chungoora, Claire Palmer, Keith Case and Jenny Harding. 'A manufacturing core concepts ontology for product lifecycle interoperability'. In: *International IFIP Working Conference on Enterprise Interoperability*. 2011, pp. 5–18 (cit. on p. 35).

[Van+22] Gilles Vandewiele, Bram Steenwinckel, Terencio Agozzino and Femke Ongenae. 'pyRDF2Vec: A Python Implementation and Extension of RDF2Vec'. In: *arXiv preprint arXiv:2205.02283* (2022) (cit. on p. 97).

[VDM17] VDMA. *Maschinenbau in Zahl und Bild 2017*. 2017. URL: https://www.vdma.org/documents/105628/805395/Maschinenbau%20in%20Zahl%20und%20Bild%20%202017.pdf/ (cit. on p. 3).

[Vel+18] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò and Yoshua Bengio. 'Graph Attention Networks'. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=rJXMpikCZ (cit. on p. 77).

[Ven+03] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Kewen Yin and Surya N. Kavuri. 'A review of process fault detection and diagnosis: Part I: Quantitative model-based methods'. In: *Computers & chemical engineering* 27.3 (2003), pp. 293–311 (cit. on p. 20).

[Vil+18] Luke Vilnis, Xiang Li, Shikhar Murty and Andrew McCallum. 'Probabilistic embedding of knowledge graphs with box lattice measures'. In: *arXiv preprint arXiv:1805.06627* (2018) (cit. on p. 136).

[Vo+17] Khuong Vo, Dang Pham, Mao Nguyen, Trung Mai and Tho Quan. 'Combination of domain knowledge and deep learning for sentiment analysis'. In: *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*. 2017, pp. 162–173 (cit. on p. 112).

[vT19] Frank van Harmelen and Annette ten Teije. 'A boxology of design patterns for hybrid learning and reasoning systems'. In: *arXiv preprint arXiv:1905.12389* (2019) (cit. on p. 111).

[Wan+14] Zhen Wang, Jianwen Zhang, Jianlin Feng and Zheng Chen. 'Knowledge graph embedding by translating on hyperplanes'. In: *Twenty-Eighth AAAI conference on artificial intelligence*. 2014 (cit. on pp. 76, 77).

[Wan+17] Jin Wang, Zhongyuan Wang, Dawei Zhang and Jun Yan. 'Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification'. In: *IJCAI*. 2017, pp. 2915–2921 (cit. on pp. 76, 77).

[Wan+19] Shuai Wang, Chenchen Huang, Juanjuan Li, Yong Yuan and Fei-Yue Wang. 'Decentralized construction of knowledge graphs for deep recommender systems based on blockchain-powered smart contracts'. In: *IEEE Access* 7 (2019), pp. 136951–136961 (cit. on p. 33).

[Wan+23] Zezhong Wang, Fangkai Yang, Pu Zhao, Lu Wang, Jue Zhang, Mohit Garg, Qingwei Lin and Dongmei Zhang. 'Empower Large Language Model to Perform Better on Industrial Domain-Specific Question Answering'. In: *arXiv preprint arXiv:2305.11541* (2023) (cit. on p. 137).

[WBD17] Xander Wilcke, Peter Bloem and Victor De Boer. 'The knowledge graph as the default data model for learning on heterogeneous knowledge'. In: *Data Science* 1.1-2 (2017), pp. 39–57 (cit. on p. 39).

[Wel+20] Tobias Weller, Tobias Dillig, Maribel Acosta and York Sure-Vetter. 'Mining Latent Features of Knowledge Graphs for Predicting Missing Relations'. In: *International Conference on Knowledge Engineering and Knowledge Management*. 2020, pp. 158–170 (cit. on p. 36).

[Wen+16] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen and Richong Zhang. 'On the representation and embedding of knowledge bases beyond binary relations'. In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 2016, pp. 1300–1307 (cit. on p. 76).

[WG14] Terry Wohlers and Tim Gornet. 'History of additive manufacturing'. In: *Wohlers report* 24.2014 (2014), p. 118 (cit. on p. 15).

[Wie+21] Andreas Wiedholz, Michael Heider, Richard Nordsieck, Andreas Angerer, Simon Dietrich and Jörg Hähner. 'CAD-based Grasp and Motion Planning for Process Automation in Fused Deposition Modelling'. In: *Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2021, Online Streaming, July 6-8, 2021*. Ed. by Oleg Gusikhin, Henk Nijmeijer and Kurosh Madani. SCITEPRESS, 2021, pp. 450–458. DOI: 10.5220/0010571204500458 (cit. on p. 17).

[Wil+16] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino Da Silva Santos, Philip E. Bourne et al. 'The FAIR Guiding Principles for scientific data management and stewardship'. In: *Scientific data* 3.1 (2016), pp. 1–9 (cit. on p. 22).

[Wil+22] Peter Wilson, Peter Kinnell, Yee Goh, Chris Pretty and Andrew Walpole. 'A transdisciplinary approach to a manufacturing problem with a machine learning solution'. In: (2022) (cit. on pp. 32, 114).

[WS04] June Wei and Gavriel Salvendy. 'The cognitive task analysis methods for job and task design: review and reappraisal'. In: *Behaviour & Information Technology* 23.4 (2004), pp. 273–299 (cit. on p. 30).

[WSV22] Christian Wirth, Ute Schmid and Stefan Voget. 'Humanzentrierte Künstliche Intelligenz: Erklärendes interaktives maschinelles Lernen für Effizienzsteigerung von Parametrieraufgaben'. In: *Digitalisierung souverän gestalten II: Handlungsspielräume in digitalen Wertschöpfungsnetzwerken*. 2022, pp. 80–92 (cit. on p. 114).

[WZ89] Ronald J. Williams and David Zipser. 'A learning algorithm for continually running fully recurrent neural networks'. In: *Neural computation* 1.2 (1989), pp. 270–280 (cit. on p. 112).

[Xu+21] Xun Xu, Yuqian Lu, Birgit Vogel-Heuser and Lihui Wang. 'Industry 4.0 and Industry 5.0—Inception, conception and perception'. In: *Journal of Manufacturing Systems* 61 (2021), pp. 530–535 (cit. on p. 4).

[Yan+14] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao and Li Deng. 'Embedding entities and relations for learning and inference in knowledge bases'. In: *arXiv preprint arXiv:1412.6575* (2014) (cit. on pp. 76, 97).

[Yan+21] Han Yang, Leilei Zhang, Bingning Wang, Ting Yao and Junfei Liu. 'Cycle or Minkowski: Which is More Appropriate for Knowledge Graph Embedding?' In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021, pp. 2301–2310 (cit. on p. 77).

[Yet+14] Barbaros Yet, Zane Perkins, Norman Fenton, Nigel Tai and William Marsh. 'Not just data: A method for improving prediction with knowledge'. In: *Journal of biomedical informatics* 48 (2014), pp. 28–37 (cit. on p. 112).

[YF11] Kenneth A. Yates and David F. Feldon. 'Advancing the practice of cognitive task analysis: a call for taxonomic research'. In: *Theoretical Issues in Ergonomics Science* 12.6 (2011), pp. 472–495 (cit. on p. 30).

[Yi+18] Kai Yi, Zhiqiang Jian, Shitao Chen and Nanning Zheng. 'Feature selective small object detection via knowledge-based recurrent attentive neural network'. In: *arXiv preprint arXiv:1803.05263* (2018) (cit. on p. 112).

[YM17] Bishan Yang and Tom Mitchell. 'Leveraging Knowledge Bases in LSTMs for Improving Machine Reading'. In: (2017). URL: http://arxiv.org/pdf/1902.09091v1 (cit. on p. 112).

[Zha+19a] Shuai Zhang, Yi Tay, Lina Yao and Qi Liu. 'Quaternion knowledge graph embeddings'. In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 94).

[Zha+19b] Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein and Huajun Chen. 'Iteratively learning embeddings and rules for knowledge graph reasoning'. In: *The World Wide Web Conference*. 2019, pp. 2366–2377 (cit. on pp. 34, 77, 78).

List of Figures

List of Tables

A
Publications

Concepts, ideas and approaches presented in this thesis have been previously published in the following publications.

[Nor+19] Richard Nordsieck, Michael Heider, Andreas Angerer and Jörg Hähner. 'Towards Automated Parameter Optimisation of Machinery by Persisting Expert Knowledge'. In: *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2019 - Volume 1, Prague, Czech Republic, July 29-31, 2019*. Ed. by Oleg Gusikhin, Kurosh Madani and Janan Zaytoon. SCITEPRESS, 2019, pp. 406–413. DOI: 10.5220/0007953204060413.

[NH20] Richard Nordsieck and Jörg Hähner. 'Interactive Knowledge-Guided Learning'. In: *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*. 2020, pp. 237–239.

[Nor+20] Richard Nordsieck, Michael Heider, Andreas Angerer and Jörg Hähner. 'Evaluating the effect of user-given guiding attention on the learning process'. In: *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. 2020, pp. 215–221.

[HNH21] Michael Heider, Richard Nordsieck and Jörg Hähner. 'Learning classifier systems for self-explaining socio-technical-systems'. In: *Proceedings of the LIFELIKE 2021 - 9th Edition in the Evolution of the Workshop Series of Autonomously Learning and Optimizing Systems (SAOS)* (2021). URL: http://ceur-ws.org/Vol-3007/2021-short-3.pdf.

[Nor+21] Richard Nordsieck, Michael Heider, Anton Winschel and Jörg Hähner. 'Knowledge Extraction via Decentralized Knowledge Graph Aggregation'. In: *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*. 2021, pp. 92–99.

[Wie+21] Andreas Wiedholz, Michael Heider, Richard Nordsieck, Andreas Angerer, Simon Dietrich and Jörg Hähner. 'CAD-based Grasp and Motion Planning for Process Automation in Fused Deposition Modelling'. In: *Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2021, Online Streaming, July 6-8, 2021*. Ed. by Oleg Gusikhin, Henk Nijmeijer and Kurosh Madani. SCITEPRESS, 2021, pp. 450–458. DOI: 10.5220/0010571204500458.

[Nor+22a] Richard Nordsieck, Michael Heider, Alwin Hoffmann and Jörg Hähner. 'Reliability-Based Aggregation of Heterogeneous Knowledge to Assist Operators in Manufacturing'. In: *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*. 2022, pp. 131–138.

[Nor+22b] Richard Nordsieck, Michael Heider, Anton Hummel and Jörg Hähner. 'A Closer Look at Sum-based Embeddings for Knowledge Graphs Containing Procedural Knowledge'. In: *Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022)*. Ed. by Mehwish Alam, Davide Buscaldi, Michael Cochez, Francesco Osborne and Reforgiato Recupero Diego. 2022. URL: https://ceur-ws.org/Vol-3342/.

[Nor+22c] Richard Nordsieck, Michael Heider, Anton Hummel, Alwin Hoffmann and Jörg Hähner. 'Towards Models of Conceptual and Procedural Operator Knowledge'. In: *Proceedings of the First International Workshop on Semantic Industrial Information Modelling (SemIIM 2022) co-located with the 19th Extended Semantic Web Conference ESWC 2022*. Ed. by Arild Waaler, Evgeny Kharlamov, Baifan Zhou and Dongzhuoran Zhou. 2022. URL: https://ceur-ws.org/Vol-3355/richardnordsieck.pdf.

[Hei+23] Michael Heider, Helena Stegherr, Richard Nordsieck and Jörg Hähner. 'Assessing Model Requirements for Explainable AI: A Template and Exemplary Case Study'. In: *Artificial Life* (in press) (2023).

[Nor+23a] Richard Nordsieck, Anton Hummel, Michael Heider, Friedline Jessica and Jörg Hähner. 'Sequential Knowledge Elicitation in Data-Based Knowledge Extraction'. In: *Proceedings of the 12th Knowledge Capture Conference (submitted to)*. 2023.

[Nor+23b] Richard Nordsieck, André Schweizer, Michael Heider and Jörg Hähner. 'PDPK: A Framework to Synthesise Process Data and Corresponding Procedural Knowledge for Manufacturing'. In: *arXiv preprint arXiv:2308.08371* (2023).