Crash Simulation Exploration with Energy Absorption Features and Graph Algorithms

Anahita Pakiman[1,2*], Jochen Garcke[1,3] and Axel Schumacher[2]

[1*]Fraunhofer SCAI, Sankt Augustin, Germany.
[2]Bergische Universität Wuppertal, Wuppertal, Germany.
[3]Universität Bonn, Bonn, Germany.

*Corresponding author(s). E-mail(s): anahita.pakiman@scai.fraunhofer.de;
Contributing authors: jochen.garcke@scai.fraunhofer.de; schumacher@uni-wuppertal.de;

Abstract

We propose to represent data from finite element car crash simulations in a graph database to empower machine learning algorithms. The industrial perspective of this work is on narrowing the gap between the uptake of modern machine learning methods and the current computer aided engineering-based vehicle development workflow. The main goals are to achieve searchability and to enable trend investigations in the product development history. We introduce features for car crash simulations to enrich the graph and to provide a summary overview for the development stages. These features are based on the energy output of the finite element solver and enable filtering of the input data by identifying essential components of the vehicle. Additionally, based on these features we propose a so-called platform/vehicle fingerprint, which assists in summarizing the exploration of the design space and facilitates cross-platform and/or various load-case comparisons. Furthermore, we predict similarities between simulations using graph algorithms, which allows the identification of outliers and ranks simulations according to their similarities. We use a car sub-model for the illustration of the similarity ansatz and present results on data from real-life development stages of an automotive company.

Keywords: FE Analysis, Automotive, Searchability, Semantic Data, Outlier Detection, CAE Knowledge, Knowledge Graphs, Graph Database, SimRank

1 Introduction

In the past 30 years, the reliability of the finite element (FE) method for predicting the crash behavior of vehicles has steadily improved. FE modeling improvements resulted in more and more detailed simulations with continuously intensifying complexity of the data. Moreover, the growth of compute power has allowed an increase in the number of simulations. Due to this data and complexity growth on the one hand, and limited available engineering time on the other hand, this simulation result data is often un-explored.

We propose a data representation approach called vehicle knowledge graph, car-graph in short, to allow the identification of un-discovered trends in simulation results. The visionary goal of the car-graph ansatz is to extract the crash identity of the vehicle. A short-term outcome of this vision is an assistance tool for engineers to efficiently evaluate data, e.g. from previous simulation studies in different projects. This assistance shall encompass searchability and the capturing of trends and

aims for support when predicting outcomes or recommending solutions.

Therefore, we transfer crash simulation data to a graph database representation to generate a knowledge graph and empower ML algorithms. Determining a data representation that is based on engineering principles and with the aim to quantify crash behavior is a largely unexplored research area. This can be explained by the complexity and size of the simulation data, due to which storing the input and output of each simulation as raw metadata is not feasible. Furthermore, it prevents the direct application of machine learning (ML) algorithms to capture and detect trends in the crash simulations. This paper focuses on the representation of the data by a graph to make a step towards building a knowledge graph for crash simulations.

Besides a graph representation for simulation data, we propose the usage of features stemming from the internal energy (IE) of components for simulation data analysis. Among the various computer aided engineering (CAE) outputs, we focus on the internal energy, because energy absorption behavior is the fundamental physics for a crash accident analysis. An additional advantage is the avoidance of specific FE model preparation. We introduce feature engineering from the IE, which allows visualizations that enable the engineer to extract additional knowledge from and gain insight into simulations. For example one can study part similarity, summarize development stages, or analyse crash behavior.

We integrate the proposed energy features into our initial car-graph, which allows the prediction of similarities between simulations with the SimRank method [24] and the identification of outlier and absorption trends with a force-directed graph visualization [23].

To investigate the graph-based algorithms, we introduce an illustrative synthetic crash FE-submodel. Additionally, we will present results for energy features and graph algorithms for data exploration on data from several development stages in a project of China Euro Vehicle Technology AB (CEVT).

First, we recapture related work in section 2, followed by a description of the investigated industrial data in section 3. We present a graph database structure in section 4, after that we introduce features for energy curves, section 5. We explore these features for identifying similarities and introduce design of experience (DOE) fingerprints, section 6. Further, we investigate graph algorithms to estimate simulation similarity and use graph visualizations in section 7. A conclusion and outlook is in section 8.

2 Related work

Car-graph is inspired by the trends and development of information retrieval and mining that have transferred from the document-centric to the entity-centric [9]. Google presented the concept of a knowledge graph in 2012 as “a graph that understands real-world entities and their relationships to one another: things, not strings” [37]. [1] is a survey in domain-specific knowledge graphs and summarizes available knowledge graphs in engineering. The most relevant engineering domain for our research is manufacturing. However, ongoing research focuses more on production and manufacturing than product development. Example applications of knowledge graphs include digital twin models for industrial production [4], industry 4.0 [19], and computer-aided manufacturing (CAM) [31]. Currently, there is no knowledge graph available for the CAE/FE domain nor specifically crash, which is the focus of this paper. However, there are investigations on ontologies for FE simulation [28] or crash [16, 15]. Note that it is not well defined where ontology ends and knowledge graph starts [36]. The entanglement of ontology and knowledge graph emphasizes both.

According to [28], several studies have already applied a knowledge-based ontology system to provide simulation knowledge to FE users. These studies disregard extracting new relationships among the data or answering analytical questions of an engineer. In some, the focus has been on automating the generation of the FE simulation [28, 38] or retrieving simulation solutions from existing simulation [41, 30]. However, the case studies are simpler [28, 41, 38, 30] than a full crash simulation. [43] characterized the CAE domain and identified unsolved challenges for tailored data and metadata management as a graph. [16, 15] has looked explicitly at a crash simulation ontology and investigated the reasoning structure of engineers, in particular in view of report generation. Overall, [16, 15, 43] have a

knowledge management system orientation to understand data structure and procedures in the company, while the simulation data itself has not been studied

To summarize, previous articles have a product management perspective to represent the crash development process structure. CAM, computer-aided design (CAD), and CAE are different product data in the development phases of vehicles. There are more research studies on knowledge graphs in CAD and CAM compared to CAE. Additionally, automated CAD-CAE model integration generated ontology models and geometrical feature extractions. These studies are of interest to connect CAD knowledge to CAE. [29] uses design change vectors to enable sustained integration of FE mesh and CAD models. [17] represents an automatic approach to generate simplified and idealized geometry models for CAE simulation. Parametrized CAD models, knowledge management, and knowledge-based engineering (KBE) systems have for decades strived to capture, digitize, and automate the application of this kind of knowledge within product and production development [26]. [25] showed that the complexity of developed KBE and CAD ontologies makes it challenging to use the data, and it is essential to consider what questions will be addressed by the data.

In view of the overall application of ML for crash data in addition to knowledge graphs, it is so far not broadly used in current CAE workflows compared to typical machine learning domains. There are two main applications of ML in crash analysis. First, predicting the crash behavior to replace/support the FE simulation, e.g. [33]. Second, using dimensionality reduction on the data of the vehicle components during crash deformation for exploration of and cluster identification in the FE simulations [32, 7, 11, 22]. Here, an engineer must usually specify the critical components in advance. Besides efficiency, i.e. considering all parts is very time-consuming, analyzing all the parts together may not highlight the bifurcation behavior. This limitation emphasizes the importance of auto-detecting and filtering the essential components.

Regarding energy absorption characteristics for crash simulation, studies are showing that energy absorption characteristics enable quantifying component performance for the design of experiments (DOE) feedback in optimization studies [35, 12, 13]. However, to our knowledge, there is no research in using energy curves features to calculate the similarity of simulations or summarize development stage results.

3 OEM data from CAE development stages

We evaluate the proposed data representation and the resulting data exploration approaches on industrial data stemming from a vehicle development project undertaken at CEVT. In particular, we consider four development stages and three load-cases for front impact analysis. The development stages are so-called primary, early, middle, and late development stages, where the names reflect the sequence of the stages. The considered development window covers roughly one-third to two-thirds of the complete R&D development phase (before the first real crash test). Table 1 summarizes the three load-cases and the number of included simulations.

In particular, we aim to assess the scalability and feasibility of the introduced energy features and graph algorithms. The focus is on data visualization to summarize the behavior and trends. Note that data confidentiality hinders illustrating the developed vehicle platform or giving details about the FE model. However, we can discuss the crash behavior using the component name, so that general crashworthiness knowledge can be used to interpret and evaluate the results. Tables 3 and 4 summarize the components referred to in this paper.

Generally, what is essential in the analysis of crash behavior with regard to geometry are the positions relative to the direction of the barrier. Therefore, one can divide the components into the early, middle, and late energy absorbent components, i.e., bumper beam, crash-box, and side-member, respectively. Additionally, the vehicle's vertical axis positioning includes middle and lower load paths in the absorption, i.e., crash-box and lower load path component, respectively.

The crash-box and side-member are thin-walled tubes with well-designed cross-section shapes and crumple points, e.g., ditches and crash beads. They may collapse in a particular pattern

Test Code	Name	No. Sim	KE_i* range $[kNmm]$	Velocity $[km/h]$
ffo**	full front overload	215	328.6 - 389.4	64.0
foU	front oblique overlap, new US-NCAP	121	346.4 - 436.2	90.0
foI	front small overlap, IIHS	275	778.0 - 890.4	66.9

* Initial kinetic energy
** Internal load-case

Table 1: Properties of the investigated CEVT data, three load-cases in four development stages.

to efficiently absorb energy. The side-member is longer and stiffer in comparison to crash boxes. The deformation modes of longitudinal beams include folding, tearing, and bending. Some reinforcing components strengthen the beams and optimize the absorption of energy. However, the lower load path component is a thin-walled structure positioned vertically lower than the side member. It distributes the load in the lower part of the structure. Finally, the subframe is a structural component with a discrete structure that supports the axle, suspension, and powertrain. This component has minor crashworthiness relevant design aspects among mentioned components due to the required durable performance.

So far, we have introduced the components of the ffo load-case that are studied in-depth in section 6.3. Additional essential components for the foU and foI load-cases are the A-pillar, cowl, front fascia, wheel arch, and wheel rim. A-pillar is the most forward vertical support of the vehicle (among A, B, C, D pillars). Additionally, the cowl separates the engine compartment from the passenger cabin between two A-pillars. The rest of the components are none structural. For further background on crashworthiness, see e.g. [14].

In addition to the company data, we design and use a FE sub-model of an open YARIS FE model provided by CCSA [8], see section 7.1. This allows additional illustrative and comprehensible investigations from an engineering point of view.

4 Graph database

In this work, we build a basic graph for crash simulation to gain insight into CAE crash simulations. We consider a knowledge graph, as defined in [21], as a way to accumulate and convey knowledge of the real world. The foundation of any knowledge graph is applying a graph abstraction to data, resulting in an initial data graph. The purpose of graph relations or structures is searchability, analysis, and insight from the data.

Data processing here considers two visions, a semantic visualization for CAE data and a graph to enable ML methods. These visions imply a workflow with two stages. First, we store the data in the database, and second we export it as a graph for ML studies or web visualization. In this section, we describe the data schema.

Based on the observations of [25], it is essential to consider the questions to be tackled with the data. For our goal of introducing data representations that answer CAE questions and that work for full crash FE models, this guided us to work with an evolving schema, developed in a feedback loop.

In Figure 1 we summarize the workflow to load the data to the graph database and to explore graph mining methods. When evaluating the components of the workflow, we aim to consider the reliability of the methods, in particular for a full-scale CAE crash FE model, as well as their computational efficiency. A primary concern is that the components of the workflow are capable of detecting slight and significant crash behavior deviations in view of the different changes engineers perform during the development stages.

In the graph compositions, we look into two different visualization aspects. In section 6.3, we consider several simulations as a graph. Here, we have substantial un-labeled data, and we are just presenting some visualization for easy exploration of the data with semantic reporting. Investigating the graph mining method will be the next step. Later in section 7, we look into crash behavior identification as a graph that is detectable with human vision. Afterward, we investigate how to capture crash behavior with graph mining methods and predict simulations similarities. Before

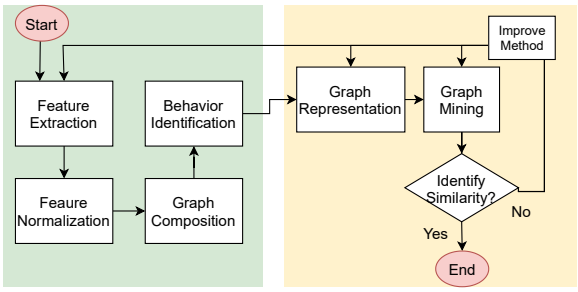


Figure 1: Feedback process during car-graph development with a focus on simulations similarity prediction. Green zone: stages with graph composition method. Orang zone: graph mining procedure and feedback process of it to graph composition.

the graph compositions, we introduce the graph schema and present the benefits of querying the database over CEVT data.

4.1 Database schema

We now introduce a database schema that aims to build a graph representation that allows the application of ML methods, Figure 2. A (Sim) node reflects a FE simulation outcome, where its properties stem from global entities of the simulation, e.g., total mass or kinetic energy. CAE simulation and crash test protocols separate into a vehicle and a barrier/impactor from the vehicle development perspective. This split allows tracing different crash scenarios (load-case) and empowers the multidisciplinary design. Due to this, we introduce nodes for the FE model (Model), barrier (Barr), and impactor (Imp) and consequently the relationships ○— SIM_MODEL —○ and ○— SIM_BARR/IMP —○, Figure 2a. Different load-cases share the same FE model from the vehicle design with this setup. Further, the focus is on the vehicle input independent of the load-case, i.e., barrier or impactor.

In addition, each part in a FE model and simulation is modeled as (Part) nodes individually and connected to the (Sim) with ○— INCL_PART —○ relationship, Figure 2b. (Part) contains information from its simulation states or the FE modeling level. Here, simulation states can consist of energy absorption features that we defined in section 5. While the FE modeling info holds properties ID (PID), box center, geometrical features material (name and ID), the center of gravity, or, when it applies, thickness (average and distribution). Geometrical features are the part's length, width, and height along with the coordinate system of the FE model (L-x, W-y, H-z).

The basic schema so far is independent of any data analysis. Whereas design (Des) and behavior (Behav) contain outcomes from specific analysis steps, e.g., feature extraction or dimensionality reduction. (Des) and (Behav) are connected to (Part) with ○— PART_DES —○ and ○— PART_BEHAV —○ respectively, Figure 2c and 2d. (Des) connects the parts that are similar at the FE modeling level while (Behav) connects the parts that have similar behavior during the simulation. In this work, the input for (Des) connection is PID similarity in one development stage. Energy features, introduced in the next section, are used for (Behav).

Moreover, we add two edges between simulations to the data FE model. First, based on the development tree that indicates the predecessor of a new FE model connects them with ○— MODEL_REF —○. Secondly, ○— SIM_SIM —○ as weighted edge that the weight refers to the similarity predictions between the simulations.

4.2 Query database

Graph mining methods allow simple data exploration, discover non-trivial patterns in the data, and reveal behaviors. One of the properties used in graph mining is the node's degree and ranking the nodes accordingly. (Des) and (Behav) nodes ranking extracts common parts and energy features in a development stage, which reflects fundamental parts and essential times during the energy absorption. This selection procedure allows automated post-processing scripts to support the CAE-ML workflow instead of user selection. High-ranked parts in a development stage for a load-case identify required parts in energy absorption. High-ranked parts are more reliable than just filtering the most energetic part in a simulation since there are outlier parts with high energy due to FE modeling error. However, low degree (Des) nodes reflect components that are outliers or in unexplored design space.

We conjecture that a ranking according to PID can summarize information relevant to the engineering process for problem-solving. For example, Figure 3c appears more stable since high ranked

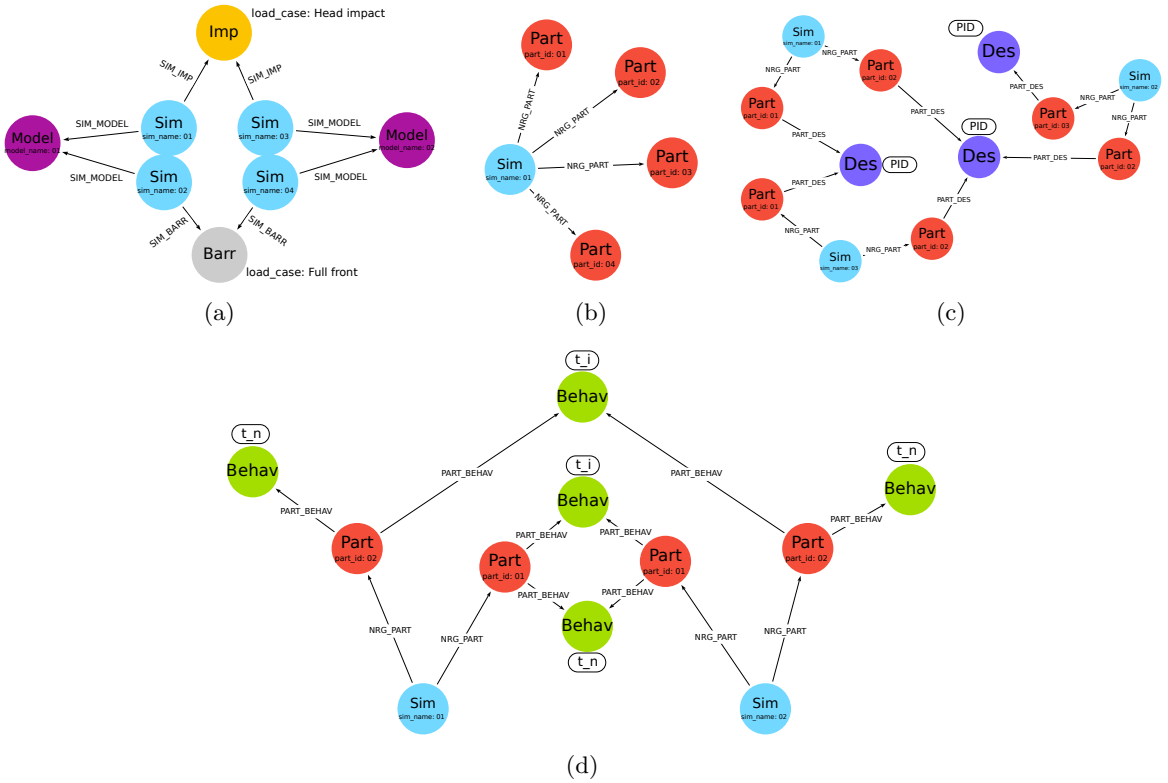
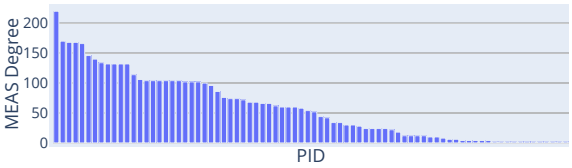


Figure 2: Database schema, (a) One simulation in high level contains one model and one impactor/barrier depending on the crash scenario, (b) More detailed components of simulation are the parts that are filtered as energetic parts (c), Design and (d) Behavior node connect parts of simulations based on similarity of design or crash behavior.



(a) ffo, foI, and ffU load-case

(b) ffo (c) foI (d) ffU

Figure 3: Degree distribution for (Des) nodes for the early stage CEVT data.

parts are more dominant than low-ranked parts. Additionally, the big step in ranking can identify the number of essential parts in this load-case.

While the node degrees in real-world, large-scale networks often follow a power-law distribution [34], this is not the case for (Des) nodes that refer to PID, Figure 3. The distribution for each load-case has some high-rank parts, some in the middle and the rest low-ranked. Whereas in power-law distribution, there is a faster decline in the ranking. High and low-rank parts refer to critical and outlier parts. However, PIDs with degrees in the middle indicate parts that change the crash behavior due to the variations among simulations. Such in-between PIDs can be valuable input for inexperienced CAE engineers to identify parts that affect the crash behavior. Moreover, the fast deviation of the degree in one load-case, Fig. 3c, compared to the rest, indicates that the number of parts affecting the load-case is limited, or that the engineer has performed a limited exploration of the design space.

5 Energy features

The deformation structures of the vehicle shall absorb the kinetic energy so that the occupants and pedestrians have the least possible injuries. In the case of shell structures, this is done by means of suitable deformation patterns. The main underlying physics of the crash analysis problem is the energy dissipation performance. CAE engineers often determine the crash behavior only by analyzing the intrusion, the acceleration, the force, the deformation, and the failure. These parameters are assessed and correlated with reality visually and quantitatively. They do not take into account the energy dissipation performance. However, the energy is a solver output available for all parts in a simulation, whereas section forces require specific FE model preparation.

The FE solver outputs energy per part over time, a so-called energy curve. Parts in the CAE model preferably refer to each vehicle component. Despite this, CAE modeling techniques require arranging vehicle components into several properties, for example, due to changes in the thickness and material. Consequently, CAE models have many parts (1500-4000). The number of parts confronts CAE engineers from a practical assessment of energy curves. Therefore, it limits the use of energy curves in the workflow, e.g., stability investigation (checks the simulation's total energy) and outlier entity identification (e.g., parts with negative energy).

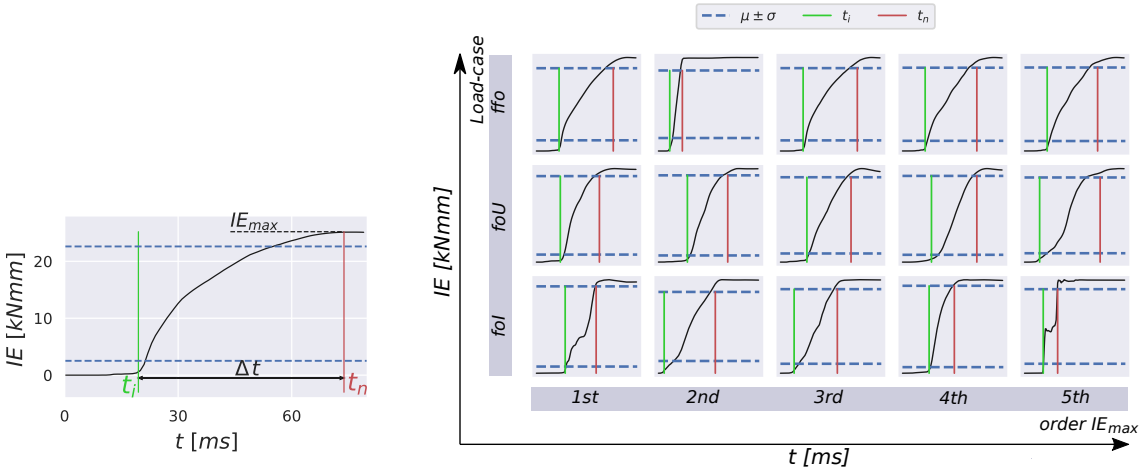
Energy Feaures			
t_i	Initial absorption time		The initial time that the energy absorption starts
IE_{max}	Absorbed energy		The max internal energy absorbed by the property
t_n	Absorption time		The time reaching to the IE_{max}
Δt			$t_n - t_i$

Table 2: Introduced features for energy curve.

We claim that energy curves hold information to characterize the simulation crash behavior. Data analysis on energy curves will simplify data processing to represent the crash behavior based on a few features. Figure 4a shows the energy curve for the most energetic part of a complete vehicle simulation in a front overload load-case, ffo, with a total initial kinetic energy of 453 $[kNmm]$ (initial velocity of 64 $[km/h]$). The shape of internal energy over time is approximately a sigmoid curve, except for parts with negative energy due to numerical error. From the crash analysis perspective, measures with the potential to analyze the crash behavior from this curve are initial absorption time, absorption end-time/period, and absorbed energy, Table 2. These features indicate three abstract characteristics of the energy curve. We define absorption time with Δt and t_n as a relative (to initial time) and absolute measuring, respectively. For now, we keep both features and study the functionality of each in different applications.

Figure 4b shows representative examples of internal energy curves and the features extraction over several simulations and parts. These curves belong to three simulations from three different front impact load-cases (Table 1) that we selected randomly. For each simulation, the five most energetic parts are plotted (part names in Table 3 and part definition in section 3). In these examples, the shape of the curve during the absorption time (Δt) is nonlinear for some parts (first and fifth part in foI load-case). This nonlinearity indicates the probable necessity of additional features or more complex methods for characterizing the absorption interval. Regardless, we estimate that the mentioned features are sufficient for the first investigation, and identifying and assessing more features is out of the scope of this paper.

We define IE_{max} as the max of the internal energy curve, and in the following, we describe the time extraction features t_i, t_n and Δt. Here, the preciseness of timings depends on the solver's time interval output. We investigate three approaches to estimate the features based on the IE behavior, namely thresholding, derivative change, and spread of standard deviation. With thresholding one considers the time when the IE crosses a pre-defined threshold value, the derivative-based method calculates the internal energy derivative ($\dot{IE}$) and determines a significant change, and using the spread $\mu \pm \sigma$ we consider upper and lower thresholds depending on the mean μ and standard deviation σ of the IE for the part over time.



(a) (b)

Figure 4: (a) Internal energy output of the solver over time for a single property with the three features $t_i, \Delta t, t_n$ that characterize the energy absorption. The dashed blue lines that intersect with the energy curve visualize the standard deviation approach. (b) Diversity of internal energy output over time for five most energetic parts, Table 3, in three load-cases, Table 1, with calculated initial and end of the absorption time and standard deviations. x and y axis are normalized.

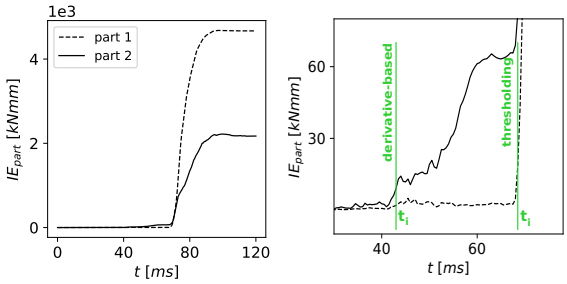
load-case	1st	2nd	3rd	4th	5th
ffo	side-member RHS-I	crash-box LHS-V	side-member LHS-I	subframe LHS-U	subframe RHS-U
foU	side-member RHS-I	crash-box LHS-V	side-member LHS-I	subframe LHS-U	subframe RHS-U
foI	A-pillar LHS-I-L	front fascia	wheel Arch -F	cowl -L	wheel rim LHS-F

RHS: right hand side, LHS: left hand side -U: upper, -V: vertical, -L: lower , -I: inner, -F: front

Table 3: Part name for Figure 4b, load-case information Table 1

Figure 4 shows the methods for t_i and t_n versus standard deviation of the time. To summarize our observations, the derivative method is more suitable for t_i due to its sensitivity in capturing trigger time. However, thresholding performs more desirable for t_n since IE growth is saturating at the end. Furthermore, the standard deviation approach fails for the parts with a long absorption time or negative IE in the initialization. In the following sections we will compare thresholding and derivative-based methods for t_i and t_n in more detail. For this, we use visual engineering judgment, where we consider both methods on random samples from three full vehicle front load-cases, 1 in four development stages, (section 3). For each simulation, we consider the 20 most energetic parts.

Note, the features are not continuous values since they depend on the solver settings for timestep output, they can vary from 1 $[ms]$ to 0.001 $[ms]$. Consequently, the binning of the features is with respect to this timestep definition. We consider a further and detailed investigation into binning and consequently resolution as out of the scope of our initial study into energy features. Further, we focus on these three features, although considering other features during the absorption may contain more component characteristics during a crash simulation. However, extracting more features is out of the scope of this work, where our focus is to investigate the potential of features from energy curves, but not to find the best approach to achieve this. The described features were



(a) (b)

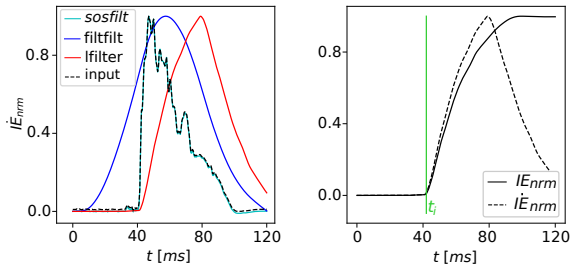
Figure 5: (a) IE curves for two different parts. (b) Zoom into (a), the derivative-based method for part two (left marker) gives an earlier, and preferred, t_i than thresholding (right marker).

selected because of their simplicity and the possible control over and interpretation of the extracted features.

5.1 Initial time

The initial time t_i for each part reflects when a part begins to absorb the impact energy. In simulations, it is imprecise to find its exact time. Here, the extracted t_i from the thresholding and derivative-based methods are close for most of the studied parts. Further, we investigated parts with large differences in the two calculated t_i for visual comparison. Figure 5 presents an example of such a part with large differences, together with a part where both approaches give similar t_i. From visual engineering judgment, part two starts to absorb energy earlier than part one, whereas the thresholding method extracts the same time ($t_i \approx 65$) for both parts. However, the derivative-based method computes for part two an earlier time ($t_i \approx 40$) than the thresholding method, which is preferred from an engineering perspective. Consequently, we perform the further investigations with the derivative-based method.

The derivative-based method requires curve filtering due to the non-smoothness of the curve. We investigate filtering methods from SciPy.signal. From lfilt[1], filtfilt[2] and sosfilt[3] we select the FIR filter (lfilt, sample number n=75, b=1/n a=1), which smoothens the curve without any time shift, Figure 6a. With this filter and a min-max normalization of the IE derivative, t_i is extracted as the time when the derivative is above 0.005. Both methods' lower limits are selected based on visual engineering judgment and visual tuning of the explored data. Figure 6b shows the result for a selected part.



(a) (b)

Figure 6: (a) Several filters applied to the derivative of the IE over time, where filtfilt[2] and sosfilt[3] miss the ramp up time. (b) Normalized IE and normalized $\dot{IE}$ that is filtered with lfilter[1], t_i extracted based on $\dot{IE}$.

5.2 Absorption time

We define the absorption time interval Δt as the time interval from when the part's internal energy increases until it stabilizes to its maximum. The start time is t_i and the end of absorption time is t_n. We treated t_i in the last section. To extract t_n, we compare the outcome of thresholding of IE_{max} and considering the derivative. For thresholding, we introduce a factor y for IE_{max} to exclude the gradual energy increase at the end of the simulation:

$$t_n = \max_t \{t \mid IE(t) \leq y \times IE_{max}\},$$
$$\Delta t = t_n - t_i. \tag{1}$$

With visual engineering judgment on the random simulations and properties, the factor y is set to 0.95. For the derivative method, we consider the second derivative of the IE equal to zero as the time for t_n. This calculation requires filtering, where we evaluate the same filters as for t_i. However, here a time shift of the filtering is inevitable,

[1]Filter data along one-dimension with an IIR or FIR filter
[2]A digital filter forward and backward to a signal.
[3]Filter data along one dimension using cascaded second-order sections

see Figure 6b. Consequently, the second derivative dissatisfies absorption time extraction. Therefore, the maximum percentage time provides the best result for t_n.

Lastly, the standard deviation of IE is calculated for the parts, Figure 4b. The crossing of IE with $\mu \pm \sigma$ refers to t_i and t_n. However, there is a deviation in the result for different parts compared to the other method. Two extreme examples, not shown in the figure, that show undesired results are curves with long absorption time and the components with spring-back FE modeling, i.e. negative IE in the initialization. In both situations, the standard deviation is relatively tiny and causes a higher value for t_i and a smaller value for t_n, respectively.

Note that parts that have common t_i and t_n in one simulation are parts that are simultaneously involved during the crash. Identifying simultaneous parts can be used to identify parts for grouping as one absorption block. Nevertheless, a grouping is out of the scope of this paper. Furthermore, considering parts that share all three features for several simulations filters out those parts that behave similarly from the energy absorption perspective.

6 Scatter visualization

Motivated by research from the information visualization community, which shows that using a visual representation of data sets enables faster analysis by the end-users since the visual perception system is the most powerful of all the human perception systems [3], we propose several data visualization for better data exploration.

We consider energy features as a data representation for each part. In a scatter plot, we visualize these to allow data exploration. Each point in the scatter plot refers to one part (PID) of the simulation, and its coordinates are the part's energy features. This visualization allows assessing the similarity of the energy curves, identifying outlier parts, finding the similarity in component-wise crash behavior, and visualizing a DOE by a fingerprint for numerous simulations.

6.1 Curve similarity

Energy features extract the main features of the energy curves. Therefore, it enables the assessment of energy curve similarities. For example,

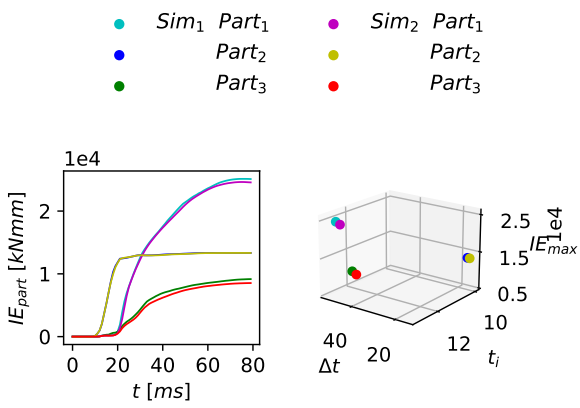


Figure 7: IE curvs similarity with scatter visualization (a) IE curves from three identical components in two simulations with same load-case and (b) the scatter plot for their energy features.

Figure 7 shows a scatter plot for three part pairs from two different simulations and the corresponding energy curves. This figure indicates that the difference between the energy features of the parts is related to the similarity of the curves. Consequently, the scatter plot of the parts energy features facilitates visualizing clusters with similar behavior from an absorption aspect.

Note that for 3D visualization, it is more illustrative to have independent variables, which facilitates the separate investigation of each feature. Therefore, in the 3D scatter plot, we depict the energy absorption time by Δt, because t_n includes the effect of t_i, whereas Δt is independent of t_i. However, it can be better to use t_n in some 2d visualizations, if one wants to include some t_i information.

Generally, the weighted sum of the energy features can be used to measure the curve similarity. Here an open question is the normalization and weighting of the energy features, which likely also depends on the analysis goal in the application. For simplicity, we concentrate on visual exploration and later individual energy features.

6.2 Part similarity

Here we investigate the detection of geometrically corresponding components with energy features. The components are correspondent if they are

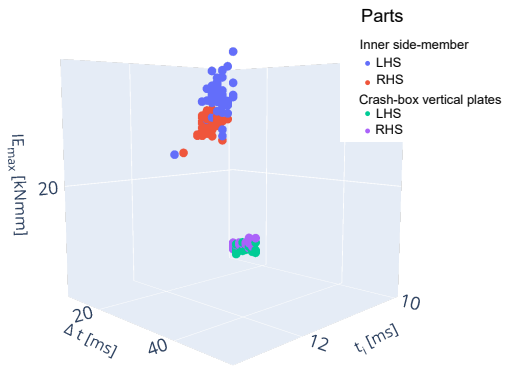


Figure 8: Energy features for symmetric components, side-member and crash-box plates.

located symmetrically in the vehicle, their undeformed geometry is mainly overlapped symmetrically, and their deformation is symmetrical. One straightforward use case is capturing similar energy absorption for symmetric parts of the vehicle structure in a full-frontal impact. The similarity is due to the almost symmetrical design of the vehicle on the left-hand side (LHS) and right-hand side (RHS). Moreover, the full-frontal load-case affects the LHS and RHS of the vehicle structure symmetrically.

Figure 8 illustrates this use case. It contains the four most energetic parts of 50 simulations of a full-front load-case in one development stage. In Figure 8, the coloring of the points is based on the PID of the parts. Similar PID between parts means the geometry of them is more relevant than the remaining parts in the vehicles[4]. This data overview shows that the four most energetic parts generate two distinct point clusters. Here, each cluster holds two parts, and each pair consists of the RHS and LHS of the corresponding geometrical part.

As a final result, we observe that energy features detect symmetrical behavior in crash simulations. While we here imposed constraints on the data set, i.e. considered only one load-case and development stage, this holds more generally. An example, which we further discuss in section 6.3.1, is for distinct point clusters, where if the PID changes for a component, one now can connect components between different development stages.

6.3 DOE fingerprint

Summarizing the behavior of a DOE with many simulations is an additional application for energy features. We introduce a DOE fingerprint as a data visualization, which is the energy features scatter distribution. The scatter plot contains energy features for energetic parts of many simulations in one or several development stages. A DOE fingerprint of a group of simulations assists in the vehicle's development process. We use four different color schemes to visually group the data points in the data exploration. The color schemes are according to PID, IE_{max} order, development stage, and load-case, respectively.

The color schemes reflect different use cases for the data exploration. The PID color scheme visualizes the design space for each part. Nonetheless, due to possible PID variations between load-case or development stages, the PID color scheme is limited to simulations in one development stage and one load-case. The second color scheme is IE_{max} order in a simulation, which visualizes the parts order in the energy absorption for each simulation. This visualization is informative if coupled with the PID color scheme to highlight parts' change in absorption permutation. Additionally, the development stage color scheme emphasizes load-cases fingerprint in one/several development stages. Finally, the load-case color scheme demonstrates the evolvement of the platform in several/single development stages independent of PID change between several load-cases.

We now show examples of data visualization by DOE fingerprints for the real life development stages from CEVT. These examples show the types of engineering information that a DOE fingerprint can visualize. To obtain a better demonstration of a 3d plot in a 2d figure, we present the DOE fingerprint as a matrix scatter plot, see Figure 9. Matrix scatter plots have two features for absorption time (Δt and t_n) since in 2d visualization t_i and Δt coupling is lost. Additionally, the range of end-time or absorption period difference remains identifiable, when comparing the spread

[4]Considering the PID remains fixed during one development stage

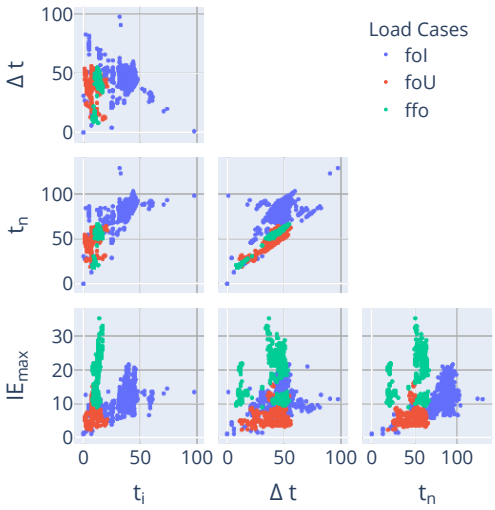


Figure 9: DOE fingerprint load-case scheme,five most energetic components in simulations from four development stages. Simulation specification in Table 1 (t_i $[ms]$, Δt $[ms]$,IE_{max} $[kNmm]$)

shape between different platform structures, by just considering just t_n or Δt.

Note that for exploring the data using DOE fingerprints, an interactive application is the most useful visualization. For example, the application can enrich the data by connecting each point in the scatter plot to additional information such as pictures, deformation videos, and metadata of the part and simulation.

6.3.1 PID scheme

The DOE fingerprint is the imprint of the scatter points distribution of the energy features in each plot independent of the PID. Consequently, the pattern of the PID color scheme relays the parts between development stages even though that PID has changed. Figure 10 uses the PID scheme for an early and a middle development stage in a IE_{max}-t_n fingerprint. This visualization shows that even though the part numbering differs in these two development stages, the shape of scatter plot and absorption order identify the pairwise components that correspond in energy absorption, see the point clouds (a) and (b) Figure in 10 and Table 4. Here, cloud (a) consists of the inner plate of the side-member. For both stages, the cloud includes only 2 PIDs referring to the LHS and RHS parts. However, there is an offset along the y-axis, which shows a decrease in the mean of IE_{max}.

Likewise, cloud (b) contains two components. The upper points belong to the subframe and the lower to the outer wall of the side-member. However, this cloud holds many different PIDs. The variation of the PID for the subframe highlights the key studied component in these analyses. Additionally, the distribution of the cloud shapes a pattern that the difference between stages addresses a change in the FE modeling technique or change in the vehicle concept. In this example, the crash-box vertical and horizontal plates have separate PID for RHS and LHS, but these are modeled as one in the mid-stage. Consequently, the absorption has doubled, see clouds (c) and (e) in Figure 10. Finally, point cloud (d) belongs to the lower load path component RHS and LHS. It keeps its dual behavior, but this visualization summrizes that the absorption interval is more stable in the later stage.

6.3.2 Order scheme

The ordering scheme visualizes IE_{max} order for each part in a simulation. The ordering scheme visualizes the point cloud for energy absorption order combined with the PID scheme. Figure 11 compares ffo load-case in two development stages with IE_{max} order scheme for the eight most energetic part for each simulation. The number of point clouds for each placement captures the number of scenarios for evaluating the permutation of the energy absorption (e.g., for third-order, it is one and two respectively in the primary and early-stage). In the primary stage, bifurcation exists for the sixth, seventh, and eighth order components; however, in the early stage, bifurcation starts right after the second part. Besides the number of scenarios, the density of the point clouds can reflect outlier simulations or unexplored design space. For example, a few simulations in the early stage has the fifth and sixth part in the left point cloud.

So far we looked at IE_{max}, t_n, and Δt features. Additionally, the t_i fingerprint has different knowledge summarization. Figure 12 shows the initial time for the same development stages as Figure 11. Here we see that the two most energetic parts, the side-members, have noticeable differences in the t_i spread. The deviation is also captured in the $t_n - \Delta t$ plot, Figure 11. The early

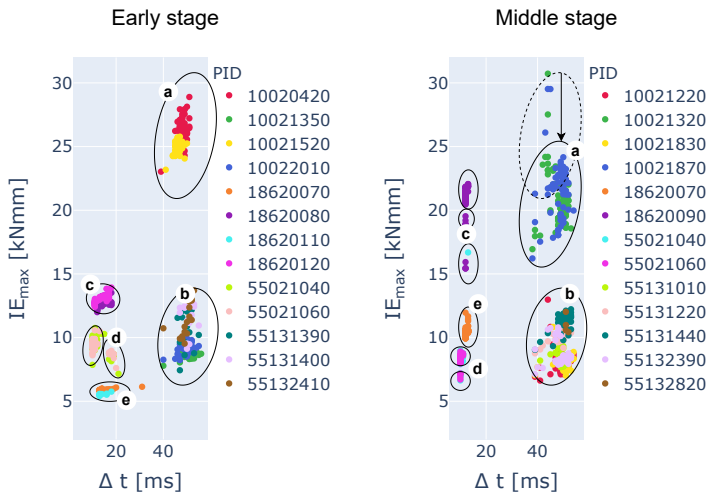


Figure 10: DOE fingerprint with PID color scheme, CEVT data ffo load-case. Part name 4

Early stage	Middle stage	Part Name (Cloud Label)	
10020420 10021520	10021870 10021320	LHS-I RHS-I	(a) side-member
10022010 10021350	10021830 10021220	LHS-O RHS-O	(b)
18620080 18620120	18620090	LHS-V RHS-V	(c) crash-box
18620070 18620110	18620070	LHS-H RHS-H	(e)
55021040 55021060		LHS RHS	lower load path (d)
55131390, 55132410 55131400	55132390, 55131220 55131440, 55132820, 55131010	LHS RHS	subframe (b)

RHS: right hand side, LHS: left hand side
-U: upper, -V: vertical, -H: horizental, I: inner, O: outer

Table 4: PID part name in two development stages for Figure 10.

development stage is more stable in trigger time than the primary development stage and limits the DOE. Consequently, the t_n and Δt relation becomes more linear. Therefore, IE_{max}-Δt and IE_{max}-t_n provide similar DOE fingerprints in the early stage. However, in the primary stage, the relation of t_n and Δt is non-linear for the side-member. Consequently, the point cloud shape of IE_{max}-Δt and IE_{max}-t_n differs in the primary development stage.

6.3.3 Development stage scheme

This coloring scheme is beneficial for summarizing the trends of the development stages. In this visualization, t_n is preferable to Δt since an absolute value is better for comparing development

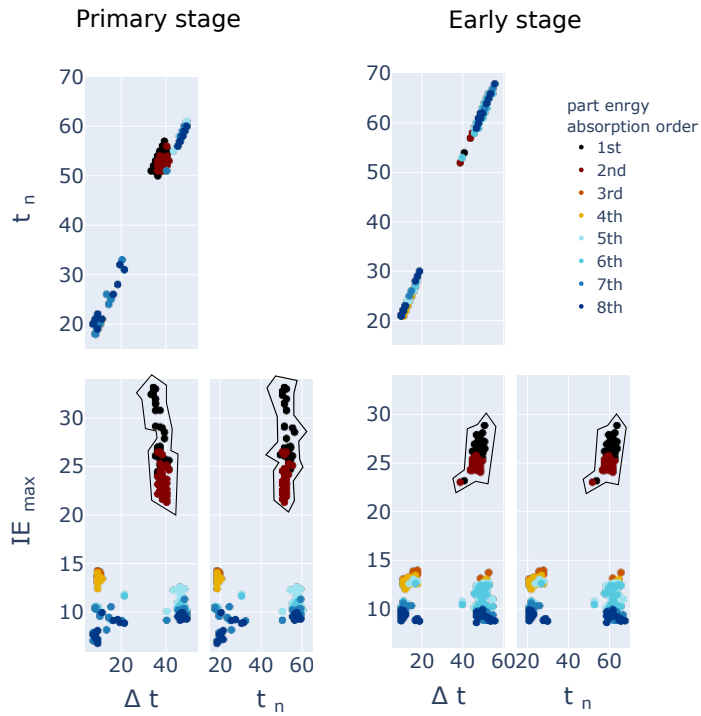


Figure 11: DOE fingerprint with order scheme, ffo load-case considering eight most energetic parts for each simulation. Primary and early development stage 50 and 53 simulations respectively. (t_n $[ms]$, Δt $[ms]$,IE_{max} $[kNmm]$)

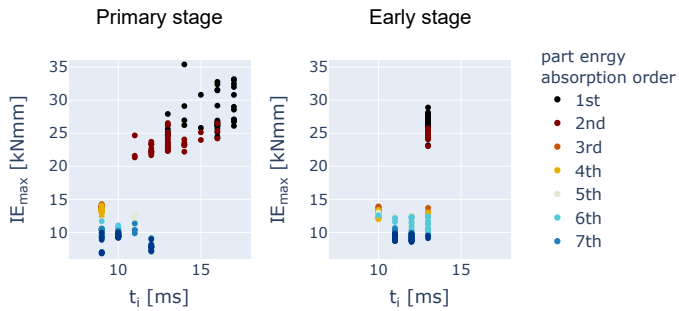


Figure 12: Initial time distribution for Figure 11, (t_i $[ms]$,IE_{max} $[kNmm]$)

stages. Figure 13 shows pair-wise comparison of four development stages with the development stage scheme coloring. In summary, remarkable detections are:

a) The initial time absorption span has been the smallest for the early development stage, and absorption initialization varies a lot for the rest.
b) The inner side-member part with the highest IE_{max} has been declining in the maximum absorbed energy during the development.
c) The 2d visualization overlays point clouds in initial absorption time.
d) The inner side-member stays almost steady in absorption time spread.
e) The spread of absorption time declines as the development stages evolve for the rest of the parts.

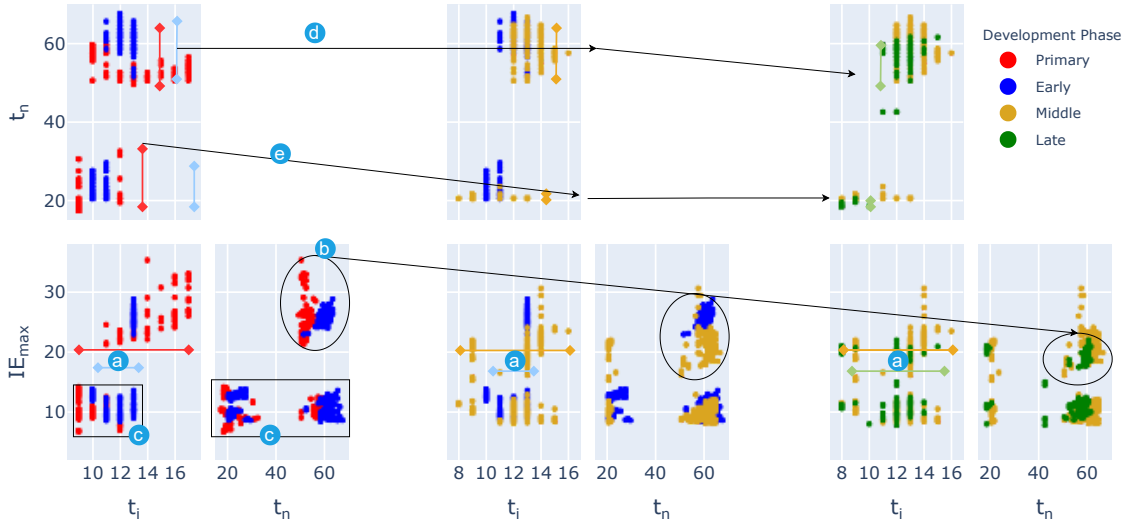


Figure 13: DOE fingerprint, development stage scheme, ffo load-case CEVT dtata. Pair-wise comparison for four different development stages. Primary, early stage respectively 50, 53 simulations, Midd-stage, 82 simulation, Late stage, 28 simulation, for 7th most energetic parts(t_i $[ms]$, Δt $[ms]$,IE_{max} $[kNmm]$)

6.3.4 Load-case scheme

This visualization enables the comparison of DOEs between load-cases, which supports detecting multi-disciplinary development challenges with different crash requirements. Figure 9 is a matrix scatter plot for three load-cases of the front crash in four development stages of CEVT data with a load-case scheme, Table 1. It includes 611 simulations with five parts with high ranked IE_{max}. The visualization indicates that the ffo load-case has discontinuous absorption compared to the other two. This gap exists for t_n values that make two clusters: early ($\approx 10\ ms$) and late absorption ($\approx 60\ ms$).

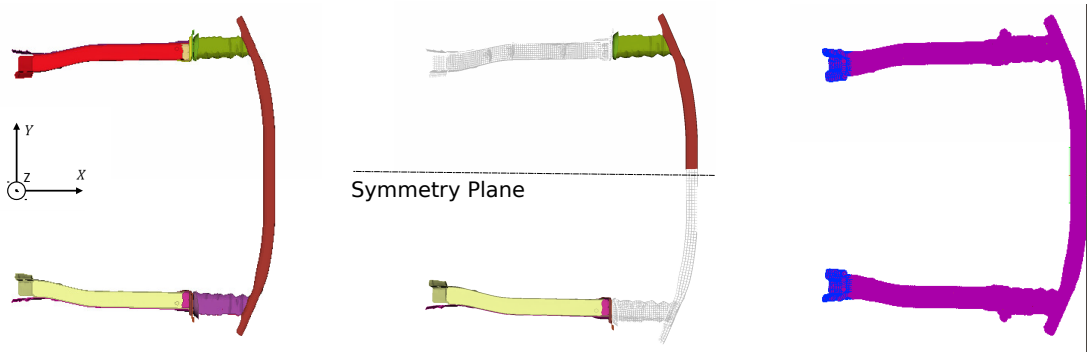
7 Simulation similarites

Predicting the similarity of simulations can assist engineers by allowing a search for the most similar simulations to a given one. This similarity connection clarifies different solutions with similar crash behavior and enables classifying the available crash behaviors or identifying outliers as those simulations dissimilar to most others. For this, we consider a weighted bipartite graph that has two types of nodes, i.e., it represents simulations as (Sim) and their energetic parts as (Des), Figure 2c. We start with an illustrative example to motivate using energy features as the weights in these graphs. In particular, we use this example to visualize crash simulations behavior as a diagram and classify their behaviors.

Afterwards, we investigate link prediction methods [24, 2] to predict similarities of simulations, i.e., (Sim) — score — (Sim). First, we examine the prediction performance on labeled crash behavior from an illustrative example. Subsequently, we explore the method on the unlabeled industrial CEVT data. Finally, we investigate force-directed graph visualization for better knowledge extraction from the (Sim) — SIM_DES — (Des) bipartite graph.

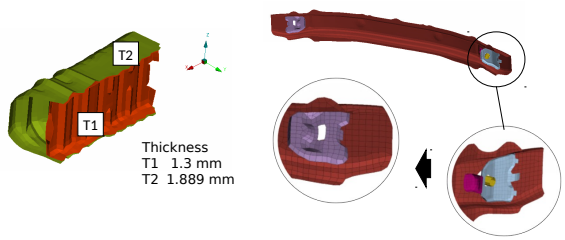
7.1 Synthetic example for illustration

Using the YARIS FE model from CCSA [8] we generate simulation data that allows easy labeling of the crash behavior. Figure 14a shows the included components, where the main components are the front bumper beam, crash-boxes, and side-members. Accordingly, each simulation includes 28 parts. Originally the right-hand side (RHS) and left-hand side (LHS) were asymmetric regarding the xz plane. Therefore the FE model is modified to make it symmetric. The resulting changes consist of removing the toe hook from the RHS,



(a) (b) (c)

Figure 14: FE-sub-model setup, (a) top view of the included components, (b) applying symmetry on crash-box, side-member and bumper beam, (c) add boundary conditions and mass.



(a) (b)

Figure 15: Detail view of FE-sub-model setup, (a) crash-box default thicknesses, (b) removing toe hook from the bumper beam.

Figure 15b, using the bumper beam and crash-box design from the LHS, Figure 14b, and using the RHS side-member reinforcement due to a simplification, Figure 14b. Moreover, the side-members end is constrained in y displacement and v_y moment to enforce the deformation (Figure 14c). To have more deformation 500 kg is added over all the components to increase the kinetic energy, Figure 14c.

7.1.1 Simulations setups

Our study consists of four full-frontal impact simulations against a rigid wall with a speed of 56.3 km/h. Simulations vary in crash-box plate thicknesses. The crash-box in this FE model is built of two sheet metal thicknesses, Figure 15a. The first base simulation has the default thickness for both RHS and LHS. For the other simulations, the thickness of the crash-box plates becomes asymmetric due to the applied changes, Table 5. These changes cause asymmetrical absorption, which results in three crash modes for the deformation, Figure 16. The crash mode indicates the yaw angle of the bumper beam. In the base simulation, the load-case and the structural stiffness are symmetric, which results in a yaw angle of zero. In the second and third simulations, the LHS is stiffer, causing the crash mode $-v_z$. For the last simulation, the RHS stiffness leads to the crash mode $+v_z$.

7.1.2 Energy diagram

We introduce an energy diagram to illustrate simulation behaviors in a crash simulation. This visualization is similar to DOE fingerprint with a new coloring scheme according to simulations. Additionally, we add the mean of energy features to the plot and connect each part. For each simulation in the illustrative example, considering the 28 parts included in each simulation will make visualization challenges. Consequently, we select the five most energetic parts for each simulation. These parts are the same for all simulations, including the four thickness part of the crash-box and the bumper beam. For simplification, we start with a 2D view using IE_{max} and t_n, where t_n contains the t_i feature and relates to Δt. Moreover, t_n

Simulation	T1*	T2*	LHS/RHS**	Crash Mode
001	**1.3**	**1.9**	-	0
002	1.5	2.1		$-v_z$
003	1.1	1.5		$-v_z$
004	1.1	1.5		$+v_z$

* modified thickness value.
** marked crash-box that its thickness has changed.

Table 5: Four different FE-model configurations. The unmarked crash-box has thicknesses as model 001.

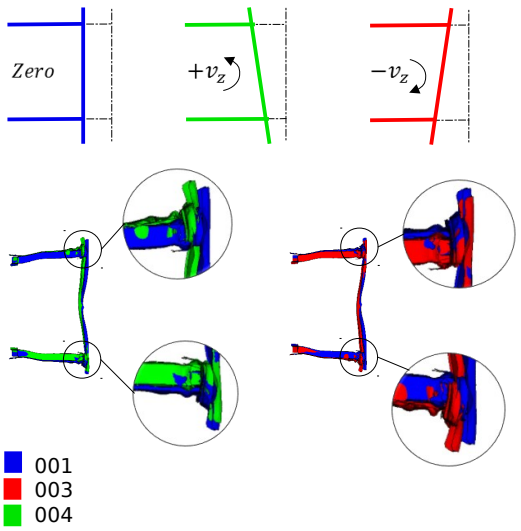


Figure 16: Defined label for crash mode based on Yaw axis rotation.

is easier to understand visually than Δt for processing the sequence of behaviors, i.e., the parts relative behavior.

Figure 17 shows the energy diagram for the base simulation. Left and right directed arrows indicate the LHS and RHS parts of the crash box, respectively, where a square represents the

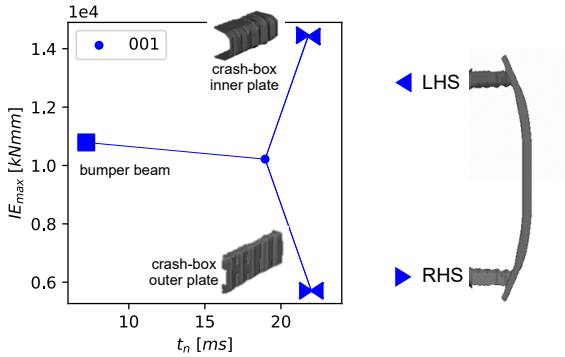
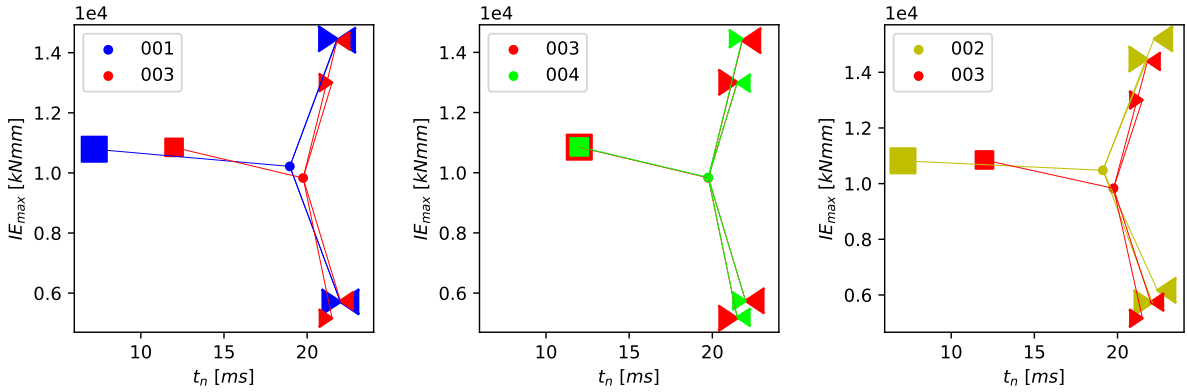


Figure 17: Base simulation energy diagram, considering five parts, simulations as in Table 5.

bumper beam. The final energy diagram is obtained by connecting each part to a point reflecting the average of the energy features of the five parts.

Figure 18b displays the energy diagrams for simulations three and four. These simulations have the same thickness value change but on opposite sides. As a result, the corresponding energy diagrams are essentially mirrored. Their structures look identical except for the switch between RHS/LHS, reflecting the change in producing negative or positive yaw. In Figure 18a we comparesone of these mirrored simulations to the base model. We observe that the IE_{max} has decreased for RHS crash-box plates, which is due to the stiffness reduction. However, in comparison t_n has not changed, which indicates a so-called stack-up state. However, the average of the energy features shows lower IE_{max}. Therefore, the side-members are absorbing the remaining energy since the total IE should remain the same over all the parts in the simulation. Another noticeable observation is that the bumper beam absorption energy is independent of crash-box, however the t_n is dependent.

Figure 18c presents energy diagrams for simulations two and three, where in both the RHS crash-box is stiffer than the LHS resulting in the negative yaw crash mode. We observe an offset in the diagrams, while the structures are similar since they reflect the similarity of the crash mode. The angle differences in each energy diagram correspond to the yaw angle. Note, the offset is due to more energy absorption in simulation two, reflecting its higher thickness values.



(a) simulations one and three (b) simulations three and four (c) simulations two and three

Figure 18: Energy graph for simulations in Table 5, Crash mode defined based on Yaw axis rotation.

7.1.3 Similarity of simulations

Representing simulations as a diagram with energy features enables the comparison of simulations. The illustrative example highlights two scenarios: a change in the diagram's structure and an offset of the whole diagram. From an engineer's perspective, these two aspects can be considered as crash mode and absorption factor. First, a crash mode reflects the parts' absorption relative to each other, represented by the diagram's structure. On the other hand, one looks at how much energy is absorbed with the absorption factor. Thus, the absorption factor operates as an offset factor in the energy diagrams in this data representation. Consequently, the same crash mode but different absorption factors exist if the relative stiffness of the components is similar.

This example shows the potential of energy features to detect slight differences between simulations. Consequently, with these visual definitions of similarity between simulations using energy diagrams, an ensuing research question is how to treat these with graph analytic methods to estimate simulation similarities. Working with unsupervised learning methods on these energy diagrams would involve treating these as separate data objects and individual weighted graphs, which is an ongoing open research question. Instead, we investigate the energy features as weights in a graph to be able to use established methods for link prediction.

7.2 Link prediction

We now investigate the SimRank method [24] to estimate similarity between simulations. Identifying similar objects based on link structure is a fundamental operation in various domains such as web mining [5], social networks [39], protein-protein interaction networks [42]. Amid the existing similarity approaches, SimRank [24] has emerged as a powerful tool for assessing structural similarities between two objects. Similar to the well-known PageRank [5], SimRank scores depend merely on the link structure, independent of the textual content of objects. The major difference between the two methods is the scoring mechanism. PageRank assigns an authority weight for each object, whereas SimRank assigns a similarity score between two objects.

SimRank is an approach that is applicable in any domain with object-to-object relationships. It measures the similarity of the structural context in which objects occur, based on their relationships with other objects. Effectively, it computes a measure that says "two objects are similar if they are related to similar objects" [24]. The similarity $s(a, b) \in [0, 1]$ between objects a and b is defined by a recursive equation. If $a = b$ then $s(a, b)$ is defined to be 1, otherwise,

$$s(a,b) = \frac{C}{|E(a)||E(b)|} \sum_{i \in E(a)} \sum_{j \in E(b)} s(i,j), \quad (2)$$

where the set $E(a)$ contains the edges of node a.

In [2] it was shown that SimRank scores are not intuitively correct for complete bipartite graphs[5]. In our application, if all the energetic parts are similar for the two simulations, we obtain a complete energy bipartite graph. Accordingly, [2] introduced SimRank++, a so-called evidence-based SimRank to work well with complete bipartite graphs, which additionally uses weights and the so-called spread to achieve similarity scores consistent with the graph's weights.

In particular, [2] introduce the notion of evidence of similarity between nodes a and b

$$evidence(a,b) := e_{a,b} := 1 - e^{-|E(a)\cap E(b)|} \quad (3)$$

as an increasing function in the number of common neighbors. Further, using normalization and scaling according to the local variance, one obtains weights W

$$W_{a,i} = \underbrace{e^{-variance(i)}}_{spread(i)} \underbrace{\frac{w(a,i)}{\sum_{j\in E(a)} w(a,j)}}_{normalized\_weight(a,i)}, \quad (4)$$

where $variance(i)$ is the variance of the edge weights w of node i. All together SimRank++ utilizes the edge weights to compute similarity scores by

$$s_w(a,b) = e_{a,b} \cdot C \sum_{i\in E(a)} \sum_{j\in E(b)} W_{a,i} W_{b,j} s_w(i,j).$$

7.2.1 Synthetic example

For the YARIS simulations, we construct the bipartite graph by querying (Sim) — SIM_DES — (Des). Based on the basic SimRank formulation, equation 2, this bipartite graph consists of two types of objects that are (Des) and (Sim). Therefore, the similarity of connections between (Sim)-(Des) is used to predict (Sim)-(Sim) connection, Figure 19. However, SimRank considers unweighted graphs. Disregarding weights means that two simulations are similar if the energetic parts are similar, which is insufficient to evaluate the similarity between simulations with similar energetic parts, but different absorption distribution.

[5]Note here, that a complete bipartite graph is a bipartite graph, where every vertex of the first node-set connects to every vertex of the second node-set.

PID		Part Name
2000000		bumper beam
LHS	RHS	
2000001	2000501	crash-box inner plate
2000002	2000502	crash-box outer plate

Table 6: Part names for PID in Figure 19.

In Figure 19, we consider just the five most energetic parts for the four simulations. This results in a fully bipartite graph, where SimRank predicts that all simulations are similar. This results differs from our goal as described in section 7.1.3.

Table 7 summarizes different approaches to predict the similary weight of ○— SIM_SIM —○ edges for the illustrative example. The columns are ordered according to the desired ranking of the result:

- Simulations (3)-(4) are the most similar due to the symmetric changes.
- Simulations (1)-(2) are the least similar since there is the most significant stiffness change among all simulations.
- The pairwise similarity of (3) or (4) to simulations (1) and (2) should be equal. The equality comes from symmetrical behavior that acts as a mirrored weight on nodes.
- Simulations (3) and (4) are more similar to (1) than (2) since the stiffness difference is less in (1)-(4)/(3) compared to (2)-(4)/(3). As a result, simulations (1)-(3) and (1)-(4) have the second-order ranking with equal values, and simulations (2)-(4) and (3)-(4) have the third-order ranking.

The used methods include SimRank (s), weighted SimRank (s_w), weighted SimRank with evidence ($s_{w,evd}$), and weighted SimRank with evidence and spread (s^{++}). We investigate two different weights of the ○— SIM_DES —○ edges: IE_{max} and the energy power absorption ($P_e = IE/\Delta t$) of the part. In the study, we use $C = 0.8$ and select the five most energetic parts, which are the bumper beam and two plates of the crash-box on LHS and RHS, Table 6.

Table 7 presents the (Sim)-(Sim) similarity predictions[6] for the illustrative example. As expected,

[6]We modify the SimRank similarity calculation in the NetworkX Python package to evidence-based SimRank with spread consideration.

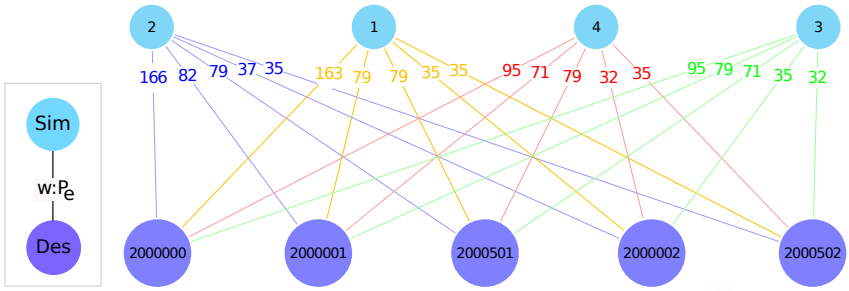


Figure 19: Bipartite graph for illustrative example weighted with P_e $[MNm/s]$, node color based on the schema in Figure 2 and edge color based on Table 5.

		3-4	1-3	1-4	2-3	2-4	1-2
s		0.665	0.665	0.665	0.665	0.665	0.665
s_w	P_e	0.687	0.690	0.690	0.690	0.690	0.698
	IE_{max}	0.681	0.681	0.681	0.681	0.681	0.681
$s_{w,evd}$	P_e	0.665	0.669	0.669	0.669	0.669	0.676
	IE_{max}	0.660	0.660	0.660	0.660	0.660	0.660
s^{++}	P_e	0.452	0.388	0.388	0.385	0.385	0.336
	IE_{max}	0.403	0.399	0.399	0.396	0.396	0.391

Table 7: Similarity prediction for considering five most energetic parts in illustrative example, Figure 19 and Table 5.

the predicted similarity for the SimRank method is the same for all simulation pairs. For s_w and $s_{w,evd}$ methods, the IE_{max} weight is failing to capture any difference between the simulations. With P_e weight, predicted similarities are still differing from our expectations for s_w and $s_{w,evd}$. Using all three modifications introduced in SimRank++ (evidence, spread, and weighted graph) gives a result that reflects our labeling. However, to get the effect of the evidence, this prediction is functioning only if the weight factors are scaled to be smaller than 2 (P_e and IE_{max} scaled relatively with $10\,e8$ and $10\,e6$ based on this model unit system, energy $[N-mm]$ and time $[s]$). If the spread is wider than two, then all similarities become zero, and if it is smaller than one, the result is similar to the weighted graph without spread.

So far, we have investigated different configurations of the SimRank++ method for similarity prediction between simulations. An additional hyperparameter in this evaluation is the number of employed parts from each simulation in the (Sim)– SIM_DES –(Des) bipartite graph. Table 8 summarizes s^{++} prediction for 2, 5, 15, and 28 (all) the parts being considered in the bipartite graph. The order of predicted similarity between simulations has the expected pattern as the labeled data from including five parts and above. However, the similarity score spread declines by including more parts. This split in similarity prediction trend identifies the smallest number of components required to predict the similarity. The split can also identify the critical parts that define the crash behavior.

7.2.2 Industrial application

After investigating the SimRank++ method for the illustrative example, we apply the approach on the CEVT data using the so far best-performing configuration, i.e. SimRank++ with evidence, spread, and scaling weights smaller than two, s^{++}. Here the bipartite graph is also considering (Sim)– SIM_MEAS –(Des) with energy power as the weight factor, $P_e = IE/\Delta t$. The similarity

No. Parts	3-4	1-3	1-4	2-3	2-4	1-2	Range
2	0.288	0.257	0.429	0.429	0.256	0.229	0.200
5	0.452	0.388	0.388	0.385	0.385	0.336	0.116
15	0.405	0.370	0.370	0.368	0.368	0.343	0.062
28	0.456	0.434	0.434	0.433	0.433	0.419	0.037

Table 8: s^{++} similarity prediction deviation regarding changing the number of energetic parts included in the illustrative example., Figure 19 and Table 5.

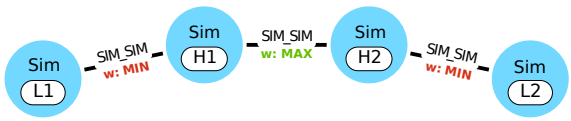


Figure 20: HH-LL simulations schema that are queried for each load-case in a development stage based on similarity prediction weight

prediction considers each load-case for each development stage separately for a specified number of parts. The load-case separation of simulations is due to the use-case that similarity between different load-cases is out of interest. Furthermore, the grouping of development stages is due to PID changes between development stages to avoid Des addressing into two irrelevant parts. The necessary parts mainly vary between load-cases and slightly among development stages.

We select HH-LL simulations pairs in each batch of simulations to investigate the similarity prediction, Figure 20. H1 and H2 are the most similar pair of simulations, according to the maximum weight on ○— SIM_SIM —○. Further, we select the least similar simulation for both of them, H1-L1 and H2-L2. Note that L1 and L2 may be the same simulation in most cases. We call these HH-LL simulations. Table 9 shows a summary of link prediction for foI load-case during the primary development stage. Each row of this table shows the predicted similarity of HH-LL simulations. We see that the HH-LL simulations remain the same for H1, L1, and L2 from eight parts upward. However, H2 is switching between two simulations from 8-15 parts. Consequently, 15 is the minimum number of parts required for this batch investigation. Similar investigation for other load-cases shows some batches are more sensitive and require even more than 20 parts to remain stable.

Table 9 shows a summary of link prediction for the foI load-case during the primary development. Each row of this table shows the predicted similarity of HH-LL simulations, Figure 20. We see that the HH-LL simulations remain the same for H1, L1, and L2 from eight parts upward. However, H2 is switching between two simulations from 8-15 parts. Consequently, 15 is the minimum number of parts required for this batch investigation. Similar investigation for other load-cases shows some batches are more sensitive and require more than 20 parts to remain stable.

Furthermore, we visualize the similarity prediction result using a histogram and a kernel density estimation (KDE)[7]. In the graph, the range on the horizontal axis refers to the predicted score of similarity, whereas on the vertical axis reflects the number of simulations for each predicted value. The density graph plots the probability of data being in a given range in the area under the density curve. Figure 21 shows the similarity predictions for foI load-case in the primary development stage when considering the 15 most energetic parts. The prediction score depends on the selection of simulations in the batch and the number of included designs. Similar evaluation for different simulations batches results in different distributions. A noticeable outcome is that the density of similarity prediction shapes clusters for simulations. In this way, density clustering detects the groups of simulations with similar scores of similarities. Figure 21 has two clusters at (a) and (b).

Overall, assessing similarity prediction between simulations is challenging since the industrial data is unlabeled. Nevertheless, we perform two cross-checks for the similarity prediction with anomaly detection and a selected examination for simulation pairs. We claim that the cluster with

[7] seaborn.distplot python package with KDE=True

nPID	H1	H2	L1	L2	w:H1-H2	w:H1-L1	w:H2-L2
2	197	227	090	090	0.530	0.195	0.196
4	086	227	386	386	0.483	0.172	0.171
6	355	387	090	090	0.419	0.290	0.291
8	354	357	090	090	0.375	0.277	0.283
10	354	387	090	090	0.337	0.267	0.272
12	354	387	090	090	0.306	0.247	0.251
14	354	357	090	090	0.283	0.233	0.238
16	354	387	090	090	0.267	0.221	0.224
18	354	387	090	090	0.253	0.211	0.214
20	354	387	090	090	0.243	0.204	0.207

Table 9: HH-LL simulations similarity prediction score with varying number of parts included in the bipartite graph from 2-20, foI load-case in primary stage, CEVT data.

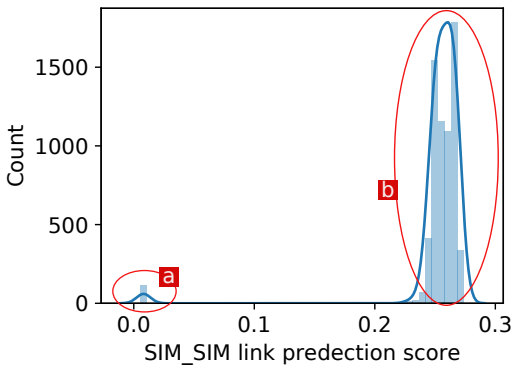


Figure 21: s^{++} link prediction histogram and estimated density for foI load-case in primary stage, CEVT data.

the lowest prediction value includes simulations that refer to an outlier for anomaly detection. For foI load-case during the primary stage, we manually identified and labeled simulation runs with early termination due to errors or unrealistic high internal energy, respectively. Afterwards, we evaluated the similarity score with and without these simulations. We observed that the low range similarity cluster is removed from the density plot when excluding the outlier simulations, i.e., zone (a) in Figure 21.

Secondly, for the simulation pair (HH-LL) investigation, we select the simulations (H1, H2) with the highest similarity prediction value of 0.306. Here, we removed the outlier simulations. Additionally, we include the least similar simulation (L1, L2) for each, where the least similar simulation is the same for both H1 and H2. To some degree, when two items are similar, where they differ the most is also similar. Figure 22 shows the energy feature scatter matrix for these simulations.

Visual comparison of the scatter matrix shows that the simulations 354-387 are more similar than 354-090 or 387-090. Here, dash lines connect parts with the same PID in the simulations. From 15 parts, eight parts of simulation 090 have different PIDs, marked with (a). Consequently, there are eight pairs of parts from simulations 354 and 387 unrelated to 090, marked as (b). These separated parts can be just due to PID change during the modeling. However, there is still a remarkable difference in the energy features of these parts that their pattern does not follow the parts from the other two simulations. Nevertheless, for simulations 354 and 387, all 15 parts have the same PID. Some parts have close energy features, and others have more distance, emphasizing the parts impacting in slight deviation of these two simulations.

Additionally, we compared the intrusion and section forces for these three simulations that confirmed the similarity relations in addition to the energy features comparison. Comparing section forces, if the curves overlay, similar buckling or bending or tearing is expected in the components[8]. In this comparison, the defined section forces vs. time curves are almost similar for 354 and 387,

[8]Look into [14] for the use of section forces in crashworthiness analysis

with a noticeable difference for simulation 090 compared to the other two.

However, while the similarity prediction shows promising results, the scores spread is small. This high density of similarity prediction is related to the number of parts included in the similarity calculation. Like in the illustrative example, a fully-connected bipartite graph has a closer prediction score, and the effect of the weights is not as strong as the structure. For example, in a group of FE simulations, a fully bipartite graph means all 20 most energetic parts are the same for all the FE simulations; however, there is a difference between them due to the difference in energy distribution. One approach to extend the deviation of the link prediction score is to include fewer parts to avoid having a fully bipartite graph.

The KDE plot and HH-LL stability pattern make it possible to find the minimum required parts. The essential outcome is to minimize the number of included parts such that the bipartite graph keeps connected and similarity prediction still has the same density distribution and a steady pattern. Change in similarity distribution identifies critical parts that affect the similarity. Additionally, sensitivity analysis of similarity prediction identifies parts that directly influence a change in the crash behavior. Therefore, it will be beneficial to have a two-step evaluation of similarity. The first step is finding the big clusters, including a more significant number of parts in similarity calculation. The second step is to evaluate similarity in each simulation cluster with reduced parts based on the sensitivity analysis.

7.3 Graph visualization

Additionally, we now investigate the visualization of the (Sim)– SIM_DES –(Des) bipartite graph that is weighted with P_e. Visualization of a network should aid the analysis and understanding of the graph by detecting, understanding, and identifying unexpected patterns [6]. Following [40], among different visualization methods, energy-based drawing algorithms constitute the most common drawing approach for undirected graphs. They are reasonably fast for medium-sized graphs, intuitive to understand, and easy to implement. The fundamental underlying idea is to model the graph as a system of interacting objects that contribute to the system's energy. A minimal energy state, according to an energy or cost function, should correspond to a nice drawing. There are various models and realizations for this approach, again see [40] for details.

From the survey [20] the three methods of Fruchterman–Reingold [18], Kamada–Kawai [27], and ForceAtlas [23]) graph are suitable for visualizing (Sim)– SIM_DES –(Des) graph concerning the size of the graph and the consideration of edge weights. However, more successful recent force-directed-based techniques have been the ones that have ignored certain principles to show off other structural properties of the graph, such as ForceAtlas [23]. These methods still support the idea of a physical system, but the principle they have tried to optimize is one of clustering rather than being concerned with edge lengths or uniform node distributions, for example [20]. In the following we summarize the use of these visualization methods for both use cases.

7.3.1 Synthetic example

For the YARIS simulations, we include all 28 parts of each simulation, Figure 23. Here, the most concrete outcome of this investigation is that ForceAtlas clusters the five most energetic parts from the rest, Table 6 and Figure 23. These parts are the same components that make the s^{++} perform the best, Table 8. These five parts are the minimum required parts to predict the similarity between the simulations and have the maximum spread.

7.3.2 Industrial application

For CEVT data, we are considering the eight most energetic parts for 115 simulations in a primary development stage and foI load-case. In Figure 24a we visualize the bipartite graph. The graph has 115 (Sim) and 33 (Des). The number of design nodes is more than eight due to differences in the most energetic part of simulations. This visualization is only capable of differentiating density of (Des) degree; nodes marked in red, orange, and green.

However, the ForceAtlas method reveals more information of this network, Figure 24b. This visualization emphasizes the outlier (Des) and (Sim) nodes by positioning them off-centered. Most of (Des) outliers are related to the beam elements and solid nuggets that are FE modeling for the

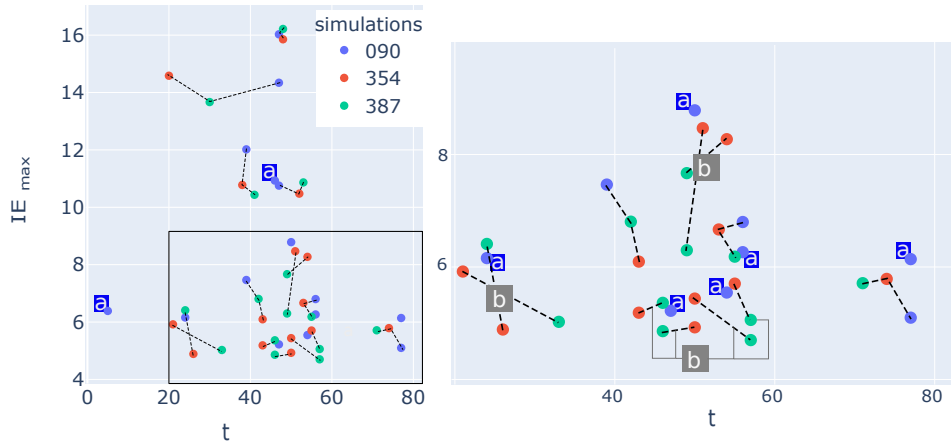


Figure 22: Analysis of SimRank++ prediction, energy features for 15 most energetic parts in HH-LL simulations for primary development stage and FoI load-case (Δt $[ms]$,IE_{max} $[kNmm]$).

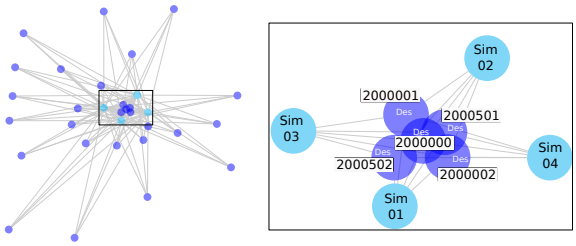


Figure 23: Illustrative example graph visualization with ForceAtlas method, weighted with P_e, nodes (Sim , Des) color based on the schema in Figure 2. Centered nodes are the minimum required parts.

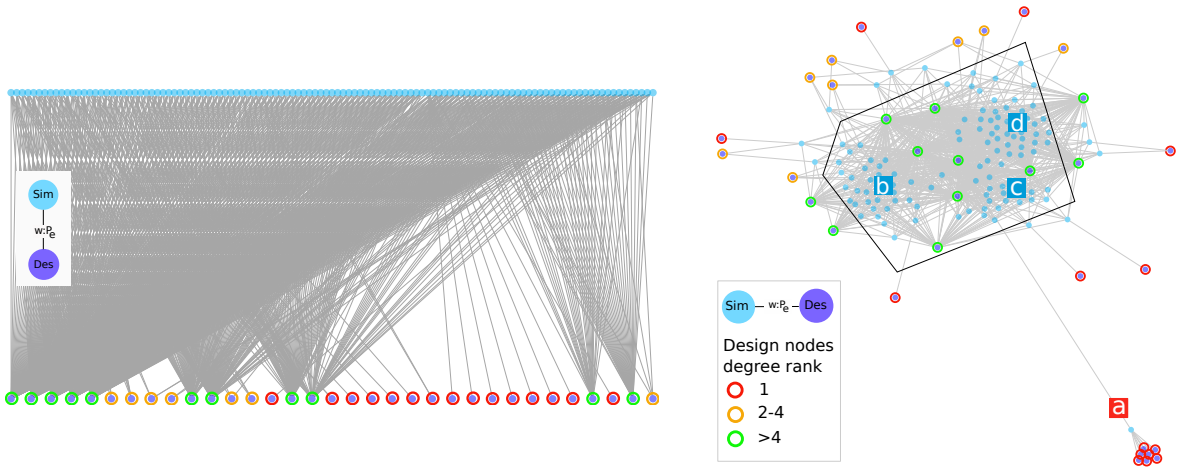
bolt and connection, respectively. Due to an error in FE modeling, parts connected to these connection parts have an intersection. Therefore, the solver tries to solve the intersection during the simulation, which causes high internal energy in the corresponding connection part. However, this is not the study's intention in the FE simulation. As a result, these are unreliable simulations and designs. In Figure 24b, the most outstanding simulation, marked with (a), is the same simulation that has the least similarity prediction score relative to the rest of the simulations.

Additionally, based on the positioning of the Sim relative to the Des clusters are shaped for simulations, zones (b), (c) and (d) in Figure 24b. From a simulations clustering perspective, ForceAtlas has an outstanding result. The other two methods, Fruchterman–Reingold and Kamada–Kawai, only separate the outlier Des nodes. The Des nodes located in the center of simulations are the Des nodes with the highest degree. These nodes are essential parts of most of the simulations. However, the difference of outer Des nodes split them into three clusters. Additionally, some simulations have more distance from the central simulation clouds. These simulations can highlight designs that are less explored.

8 Conclusion and outlook

The complexity of simulation raw data and the lack of semantics in the current vehicle development workflow causes design engineers and attribute leaders to rely on CAE engineer reporting. But this static reporting restricts the independent exploration of the data. Lack of semantics in CAE data makes the data disconnected and hinders a multi-disciplinary collaboration, which degrades efficient problem-solving. Disconnected data in an OEM, and even more between OEMs, is one of the obstacles we aim to address with the car-graph vision for an efficient exploration exploiting semantics.

We envision a web-based platform to enable semantic reporting for CAE, which targets CAE attribute leaders, CAE engineers, design engineers, and data analysts in automotive R&D. It should enable project members from different



(a) Bipartite (b) ForceAtlas

Figure 24: Information extraction with graph visualization , foI load-cas in a primary stage, 115 (Sim) simulation with eight (Des) for each simulations and weighted with P_e, node color based on the schema in Figure 2.

teams to access the CAE results, understand the design performance limitations, compare simulations, and use algorithms on the car-graph. For example, we can support a data exploration with 3d and 2d views of DOE fingerprints (section 6.3). Interpreting a DOE fingerprint involves further investigation, where a dynamic interaction and filtering facilitates the data exploration. For example, each scatter point links to the energy curve, metadata, pictures, and deformation videos of the simulations/parts.

Our research aim for this work was to introduce semantics for crash simulations, which enables searchability or filtering of FE crash simulations. Based on graph representations of the data, we proposed energy features and using these features for data visualization, while also leveraging them as weights in the data graph to allow a prediction of similarities between the simulations. We showed the capability of energy features for differentiating FE crash simulations during development stages. Moreover, it introduces a simple way of filtering the necessary parts to be studied in ML deformation-based workflows. The similarity prediction supports engineers while exploring a simulation database, it allows the finding of similar solutions and to automatically detect outliers, e.g. for data cleaning. Finally, these features with DOE fingerprint, KDE similarity prediction, and force-directed graph propose a new way of visualization for comparing many FE simulations.

However, it is still the early stage of car-graph research. For example, additional data should be loaded into the graph to enrich it. Moreover, there is a necessity for a grouping of parts and features, where for example a higher level of grouping may enable load-path detection. Additionally, a long-term target is to enable the evaluation of performance robustness. Correspondingly, we envision that the car-graph shall allow an extension of the safety evaluation from regulated tests, which are just examples of real crash scenarios, to more diverse crash scenarios.

For graph compositions, our results shows the potential to visualize graphs that identify simulations and DOE in human distinguishable graph structure. But one still requires additional data and investigations to enable graph mining methods. The SimRank++ method can predict the similarity of simulations; however, the spread of similarities is small. Therefore, further investigation on other methods, or e.g. on changing the spread function of SimRank++, is warranted in view of an improvement of the similarity prediction. Furthermore, geometrical grouping or energy feature-based grouping of the parts can improve performance and enable similarity prediction between development stages. Finally, a similarity

prediction analysis using SimRank++ may detect parts responsible for behavior changes in FE crash simulation.

ForceAtlas visualized the minimum required parts for the SimRank++ method and outliers in directed graph visualization. This visualization enables outlier detection and the clustering of the parts and simulations. However, the visualization was poor in comparing development stages. This performance can be due to the simplicity of the current graph. Therefore, including other properties such as part positions may improve the results.

One additional study to improve the visualization, can be including the SimRank++ predictions as weights of the ○— SIM_SIM —○ edges for the force-directed visualization. Additionally, edge bundling technologies can reduce the visual clutter caused by edge overlaps. It can provide a global overview of complex connection graphs while providing information on the primary connection relationships in the graph by the thickness and color of the edges [10].

References

[1] Abu-Salih B (2021) Domain-specific knowledge graphs: A survey. Journal of Network and Computer Applications 185:103,076

[2] Antonellis I, Garcia-Molina H, Chang CC (2007) Simrank++: Query rewriting through link analysis of the click graph. In: Proceedings of the 17th international conference on World Wide Web, pp 1177–1178

[3] Auber D (2004) Tulip—a huge graph visualization framework. In: Graph drawing software. Springer, p 105–126

[4] Banerjee A, Dalal R, Mittal S, et al (2017) Generating Digital Twin models using Knowledge Graphs for Industrial Production Lines. Work Ind Knowl Graphs, co-located with 9th Int ACM Web Sci Conf 2017 pp 1–5

[5] Berkhin P (2005) A survey on pagerank computing. Internet mathematics 2(1):73–120

[6] Bezerianos A, Chevalier F, Dragicevic P, et al (2010) Graphdice: A system for exploring multivariate social networks. In: Computer Graphics Forum, Wiley Online Library, pp 863–872

[7] Bohn B, Garcke J, Iza-Teran R, et al (2013) Analysis of car crash simulation data with nonlinear machine learning methods. Procedia Comput Sci 18:621–630. doi:10.1016/j.procs.2013.05.226

[8] Center for Collision Safety and Analysis (2015) Toyota Yaris Finite Element Model Validation Detail Mesh. URL https://www.ccsa.gmu.edu/models/2009-toyota-yaris/

[9] Chen C, Xing Z, Han L (2017) TechLand: Assisting technology landscape inquiries with insights from stack overflow. Proc - 2016 IEEE Int Conf Softw Maint Evol ICSME 2016 pp 356–366. doi:10.1109/ICSME.2016.17

[10] Chen Y, Guan Z, Zhang R, et al (2019) A survey on visualization approaches for exploring association relationships in graph data. Journal of Visualization 22(3):625–639

[11] Diez C, Kunze P, Toewe D, et al (2017) Big-Data based rule-finding for analysis of crash simulations. In: World Congress on Structural and Multidisciplinary Optimization, June

[12] Du X, Zhu F (2018) A new data-driven design methodology for mechanical systems with high dimensional design variables. Advances in Engineering Software 117:18–28. doi:10.1016/j.advengsoft.2017.12.006

[13] Du X, Zhu F (2019) A novel principal components analysis (pca) method for energy absorbing structural design enhanced by data mining. Advances in Engineering Software 127:17–27. doi:10.1016/j.advengsoft.2018.10.005, using pca to generate simulation

[14] Du Bois P, Chou CC, Fileta BB, et al (2004) Vehicle crashworthiness and occupant protection. Am Iron Stell Inst pp 27–280, 304–330

[15] Fatfouta N, Stal-Le Cardinal J (2020) Towards A Framework for Integrated and Collaborative Knowledge Management for Engineering Design – A Case Study. Proc Des Soc Des Conf 1:559–568. doi:10.1017/dsd.2020.136

[16] Fatfouta N, Stal-Le Cardinal J, Royer C (2019) Empirical study of car crash simulation analysis within the development phase. Proc Int Conf Eng Des ICED 2019-Augus:2843–2852. doi:10.1017/dsi.2019.291

[17] Feng Q, Zhou X, Li J (2020) A hybrid and automated approach to adapt geometry model for cad/cae integration. Engineering with Computers 36:543–563. doi:10.1007/s00366-019-00713-4

[18] Fruchterman TMJ, Reingold EM (1991) Graph Drawing by Force-directed Placement. Force-Directed Placement, in: Software-Practice and Experience vol:21no11pp1129–1164

[19] Garofalo M, Pellegrino MA, Altabba A, et al (2018) Leveraging knowledge graph embedding techniques for industry 4.0 use cases. ArXiv abs/1808.00434

[20] Gibson H, Faith J, Vickers P (2013) A survey of two-dimensional graph layout techniques for information visualisation. Information visualization 12(3-4):324–357

[21] Hogan A, Blomqvist E, Cochez M, et al (2021) Knowledge graphs. ACM Computing Surveys (CSUR) 54(4):1–37

[22] Iza-Teran R, Garcke J (2019) A Geometrical Method for Low-Dimensional Representations of Simulations. SIAM-ASA J Uncertain Quantif 7(2):472–496. doi:10.1137/17M1154205, https://arxiv.org/abs/arXiv:1903.07744

[23] Jacomy M, Bastian M (2011) Forceatlas2, a graph layout algorithm for handy network visualization

[24] Jeh G, Widom J (2002) Simrank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 538–543

[25] Johansson J (2017) Analysing engineering knowledge in cad-models and spread sheets using graph theory and filtering. Adv Transdiscipl Eng 5:629–638. doi:10.3233/978-1-61499-779-5-629

[26] Johansson J, Contero M, Company P, et al (2018) Supporting connectivism in knowledge based engineering with graph theory, filtering techniques and model quality assurance. Adv Eng Informatics 38:252–263. doi:10.1016/j.aei.2018.07.005

[27] Kamada T, Kawai S (1989) An Algorithm For Drawing General Undirected Graphs Tomihisa Kamada And Satoru Kawai. Information Processing Letters 31(1):7–15

[28] Kestel P, Kügler P, Zirngibl C, et al (2019) Ontology-based approach for the provision of simulation knowledge acquired by Data and Text Mining processes. Adv Eng Informatics 39:292–305. doi:10.1016/j.aei.2019.02.001

[29] Kirkwood R, Sherwood JA (2018) Sustained cad/cae integration: integrating with successive versions of step or iges files. Engineering with Computers 34. doi:10.1007/s00366-017-0516-z

[30] Kügler P, Kestel P, Schon C, et al (2018) Ontology-based approach for the use of intentional forgetting in product development. In: DS 92: Proceedings of the DESIGN 2018 15th International Design Conference, pp 1595–1606

[31] Li X, Zhang S, Huang R, et al (2018) Structural Modeling of Heterogeneous CAM Model Based on Process Knowledge Graph. Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/Journal Comput Des Comput Graph 30(7):1342–1355. doi:10.3724/SP.J.1089.2018.16702

[32] Mei L, Thole CA (2008) Data analysis for parallel car-crash simulation

results and model optimization. Simul Model Pract Theory 16(3):329–337. doi:10.1016/j.simpat.2007.11.018

[33] Müller M, Long X, Botsch M, et al (2018) Real-time crash severity estimation with machine learning and 2d mass-spring-damper model. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp 2036–2043

[34] Newman ME (2005) Power laws, pareto distributions and zipf's law. Contemporary physics 46(5):323–351

[35] Ortmann C, Schumacher A (2013) Graph and heuristic based topology optimization of crash loaded structures. Struct Multidiscip Optim 47(6):839–854. doi:10.1007/s00158-012-0872-7

[36] Science OOD (2018) Where Ontologies End and Knowledge Graphs Begin - Predict - Medium. URL https://medium.com/predict/where-ontologies-end-and-//knowledge-graphs-begin-6fe0cdede1ed

[37] Singhal A (2012) Introducing the knowledge graph: things, not strings. Official google blog 5:16

[38] Sun W, Ma Q, Chen S (2009) A framework for automated finite element analysis with an ontology-based approach. J Mech Sci Technol 23(12):3209–3220. doi:10.1007/s12206-009-1005-0

[39] Sá HRD, Prudêncio RB (2011) Supervised link prediction in weighted networks. Proceedings of the International Joint Conference on Neural Networks pp 2281–2288. doi:10.1109/IJCNN.2011.6033513

[40] Tamassia R (2013) Handbook of graph drawing and visualization. CRC press

[41] Wriggers P, Siplivaya M, Joukova I, et al (2008) Intelligent support of the preprocessing stage of engineering analysis using case-based reasoning. Eng Comput 24(4):383–404. doi:10.1007/s00366-007-0079-5

[42] Yu H, Braun P, Yıldırım MA, et al (2008) High-quality binary protein interaction map of the yeast interactome network. Science 322(5898):104–110

[43] Ziegler J, Reimann P, Keller F, et al (2021) A metadata model to connect isolated data silos and activities of the cae domain. In: International Conference on Advanced Information Systems Engineering, Springer, pp 213–228