A METHOD FOR EFFICIENT GENERATION AND OPTIMIZATION OF SIMULATION-BASED TRAINING DATA FOR DATA-DRIVEN INJURY PREDICTION IN VRU-VEHICLE ACCIDENT SCENARIOS

Niranjan, Ballal
Thomas, Soot
Michael, Dlugosch
Niclas, Trube
Fraunhofer-Institute for High-Speed Dynamics, Ernst-Mach-Institut, EMI
Germany

Dirk, Fressmann
DYNAmore Gesellschaft für FEM Ingenieurdienstleistungen mbH
Germany

Paper Number 23-0215

ABSTRACT

Urban traffic is characterized by limited space, varying traffic flows and multiple types of road users. Despite increasing automation and design efforts, the joint use of traffic areas poses a particular risk for vulnerable road users (VRUs). In order to make traffic as safe as possible, the severity of injuries to VRUs in unavoidable collisions must be reduced. In future applications, predicting situation-specific injury risks for VRUs in real-time using machine learning (ML) could support decision making in determining risk minimization strategies. The predictive capability of any ML model is determined by the quality of the used training data. While there are no real-world training data available for injury prediction, simulation data, which is frequently employed in passive safety engineering, can be used as synthetic data. Since deliberate training data generation consumes substantial resources, particular attention is focused on the iterative generation of optimized simulation data sets. This study presents and discusses an adaptive simulation data generation pipeline to generate simulation data sets that reflect the overall system's behavior with the overall goal of efficiency and sustainability.
The novel pipeline involving nine steps is divided into two phases, "Data Generation" and "Data Exploitation". The "Data Generation" phase predominately focusses on the adaptive strategies to generate a generalist training data set. Along with the fundamental techniques for adaptively adding new points, metrics for assessing the information content of the present data set and for tracking the iterative sampling progress are also discussed in this study. Additionally, experiments to understand the effects of batch size is conducted and the potential use of information content metrics for process termination and dynamic, adaptive batch size adjustment is discussed. The pipeline is initially tested using a generic example and is then applied to a simulation setup modeling a human head crashing onto a vehicle windshield. The observations from applying the pipeline to the simulation setup are compared with the observations from applying it to the generic function to evaluate the novel pipeline.
It is shown, that the pipeline is generally applicable to such real-world problems and that the anticipated dynamic behavior of the data generation process is confirmed in the generic and real application example. This lays fundamental groundwork which needs to be extended along multiple routes in future work.

MOTIVATION

Not only in recent years, urban traffic systems have shown a significant trend towards multimodality [1]. Next to motivating an extensive amount of research activities on the design of multimodal urban mobility systems, this poses significant challenges towards traffic safety to all stakeholders involved [2] [3]. Particularly vulnerable road users (VRUs), e.g. pedestrians or cyclists, exhibit an overproportionate share, an increased injury severity and relatively high death-rates in the accident data [4] [5] [6]. In order to respond to these VRU-specific needs in traffic safety, a consortium of industrial and academic partners has teamed up in the research project ATTENTION ("artificial intelligence for real-time injury prediction"*) to develop a framework, as well as constituting methods and tools to dynamically predict injury risks for VRUs in accident scenarios [7]. Given the requirement of (near) real-time prediction, conventional engineering methods to predict the behavior or performance of structures under dynamic loading conditions – namely the finite element method (FEM) – are not feasible. More precisely, while the comprehensive simulation of a crash scenario can take up to 30 h on an advanced compute cluster, the collision of a vehicle is avoidable until ca. 1.5 seconds before impact [8]. Hence, the potentials and applicability of artificial intelligence (AI) or – more accurately – machine learning (ML) to predict the system's responses in such scenarios are studied in this project.
ML has shown to produce promising results in predicting the behavior of vehicle structures under crash in several studies [9] [10] [11]. As outlined by Kohar et al. [10], one of the main challenges in applying ML in engineering

design for crashworthiness lies in the limited availability and heterogeneity of suitable training data. Compared to other domains, the mechanical engineering domain is generally dealing with highly complex systems and challenges, which are tackled using advanced modeling and simulation methods (such as FEM) requiring substantial (computational) resources. This constitutes two distinct characteristics of this domain hindering the widespread adoption of ML-based methods: data scarcity and data complexity. As advocated by Ng [12], shifting the ML paradigm from model-centric to data-centric approaches is an effective way to efficiently increase the overall performance in a wide range of ML applications. As opposed to rather conventional model-centric approaches, data-centric ML doesn't focus on engineering the ML-model itself (e.g. model type and architecture) to increase the overall performance, but on engineering the data used to train the model [13]. In view of aforementioned domain characteristics, this fundamental notion, that data quality has a stronger impact on the performance, efficiency and scalability of ML solutions than model sophistication or fidelity, particularly applies to the engineering domain. In order to cope with the resource-capacity induced limited availability and complexity of simulation data, following a data-centric approach in these types of applications can be considered a logical consequence. While leveraging legacy simulation data as training data poses additional – despite interesting approaches (see e.g. Vasu et al. [14] or Greve and Van de Weg [15]) widely unsolved – challenges, deliberately generating sets of training data through simulation for a specific prediction task is a common practice in respective current R&D activities [16] [17]. In the majority of applications, the training data set is initially generated using well established design-of-experiments (DoE) methods to sample the design space and FEM-simulation software. With that, an ML model is trained to predict the system response parameters of interest based on the input parameter settings as features. Since these unidirectional "one-shot" data generation approaches do not allow for any feedback from the training and model performance to the sampling phase, designing for overall process efficiency is inhibited. As seen in the works of Chec [18] and Kayvantash [19], modifying this pipeline to resemble an active learning (AL) scheme is one approach to introduce this feedback loop and thus maximize the model performance (e.g. accuracy) while minimizing the number of (often expensive) samples in the training data set. Here, by iteratively generating batches of data and evaluating the current performance of the ML model, the sampling locations for the next iteration are determined by maximizing their contribution to the learning process [20]. While these approaches have shown to yield an increased efficiency, one key shortcoming does persist in all of the existing methodologies.

As stated above, efficiently using resources (e.g. simulation) and managing all the digital assets related to the ML pipeline sustainably is always beneficial, but mandatory in the engineering domain. Hence, generating a data set for the purpose to train solely one specific ML model making one specific set of predictions – as it is also the case in regular AL schemes – does work for methods development in R&D but doesn't fulfil the domain application requirements. Rather than focusing on the ML model and its predictive capabilities while adaptively generating the training data, focusing on the data itself and optimizing the representation of the system's complex behavior in the data set holds significant potential. The goal is to generate a data set, which – within reasonable limits – represents all the relevant characteristic features of the system's (e.g. crash structure) behavior without tailoring it to fit a very specific ML application – and thus make it reusable in multiple applications. Combined with already partially employed transfer learning approaches [21] [22], this aims to increase the overall efficiency and sustainability by enabling the reuse of not only the generated data sets, but also the model(s) trained. For that, an adaptive data generation pipeline is proposed to efficiently generate information-dense and reusable data sets for training transferable ML models for traffic and vehicle safety applications. The pipeline is applied to the critical use-case of VRU safety.

INTRODUCTION AND STATE OF THE ART

As outlined above, applying ML in the domain of vehicle passive safety is an active field of current R&D efforts which have already produced quite promising results. In this chapter, two of the main ML pipeline architectures and selected fundamental methods are introduced and discussed with a particular focus on the respective data generation and sampling schemes.

Figure 1 shows the schematic flowchart of a typical ML pipeline used for data-driven predictions of system response parameters which could be constituting the crashworthiness of vehicle structures or the behavior of a dummy or human body model. While rounded rectangular shapes depict (sub-) processes in the pipeline, hexagonal and diamond shapes depict resources and process bifurcations, respectively. In general, the pipeline is divided into two major sections: the data generation and the data exploitation phase. After setting its dimensions and ranges, the DoE strategy to sample the design space is implemented in step 1. Given its comparably favorable characteristics (e.g. space filling) one widely adopted DoE-method is Latin Hypercube Sampling (LHS) [23] [24]. After defining the sample points – each of which could represent a certain crash scenario within the design space limits – the output responses are computed employing the respective FE model(s) and the compute resources (e.g. cluster) in step 2.

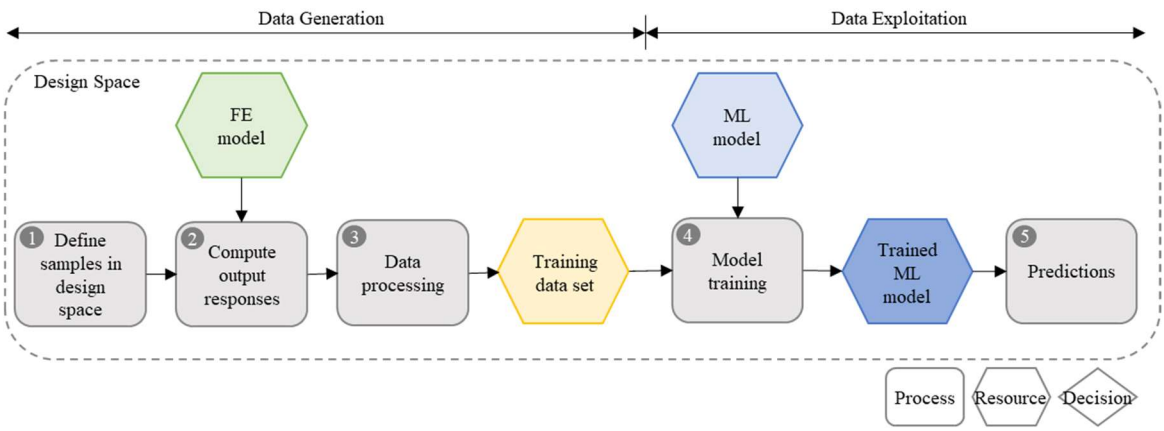

Figure 1. Generic flowchart of a conventional "one-shot sampling" ML pipeline.

Data processing in step 3 resembles conventional post-processing procedures such as the extraction and processing (e.g. re-sampling, filtering, structuring etc.) of relevant output responses from the simulation output files to generate suitable data for training purposes. The input parameters defining the sampled data points then are considered the features and the processed output responses (e.g. maximum acceleration at a defined location) are considered the labels of a resulting training data set for a supervised learning scheme [25]. In the vehicle safety domain, multiple feasible ML model types need to be assessed regarding their respective advantages and drawbacks in the specific application. For instance, while regular feed-forward neural networks (NNs) or random forests (RFs) can be used to predict discrete response values, gated recurrent units (GRUs) or long short-term memory networks (LSTMs) – both types of recurrent neural network (RNN) architectures – serve to predict time series, such as acceleration curves or node trajectories, accounting for the dependencies along the time series data [26] [27]. Potential benefits and drawbacks to consider might be model fidelity (and the corresponding amount of needed training data and computation time) or the interpretability and explainability of the model, which are key characteristics, particularly in safety critical applications [28] [29]. The training process in step 4 provides the generated training data set to the selected model reserving subsets for testing the model performance following a test-train-split or a cross-validation scheme and final validation with unseen data [25]. The trained model can then be used to predict output responses from input features within the limits of the design space in step 5.
Given this "one-shot-sampling" pipeline architecture, the information content of the training data set solely depends on the initial sampling in step 1. The linearity of this architecture does not allow for any internal feedback from data exploitation to the data generation phase, resulting in the fact that engineering the sampling methods in step 1 based on domain knowledge, experience and/or learnings from previous linear pipeline executions are the only – albeit inefficient – means to generate information dense, ideally suitable training data sets.

More advanced approaches towards training data generation are found in the field of active learning methods. Active learning – also called "query learning" – is a subfield of ML and follows the key hypothesis, that training efficiency is maximized, when the ML model can dynamically "query" additional data samples considering the current training progress [20]. Hence, through dynamically adding specific samples, the training data set continuously adapts to the information needs in the training process. Figure 2 depicts such an adaptive sampling scheme in the application scenario under consideration here.

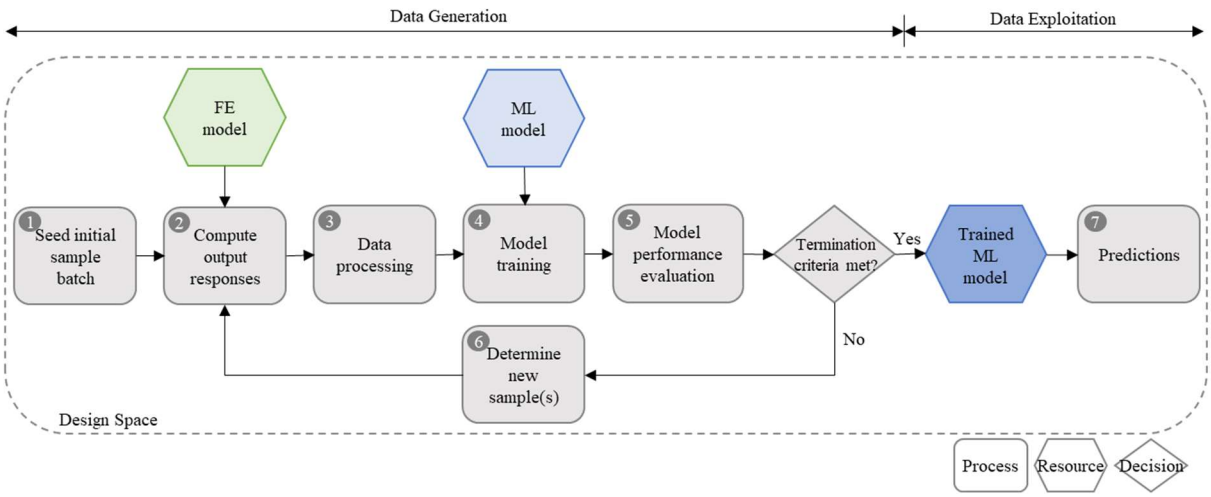

Figure 2. Generic flowchart of an adaptive sampling scheme in an AL pipeline.

Apart from potential effects on the overall process from the size of the initially sampled batch in step 1, the first four steps do not differ from the "one-shot-sampling" scheme in Figure 1. After that however, the prediction performance of the current (training) state of the model is evaluated considering the respective criteria (e.g. prediction accuracy based on error metrics such as the root-mean-squared-error (RMSE) with respect to the test samples [30]). A pre-defined threshold for the model performance can be used as a termination criterion for the data generation phase, which can be complemented by additional criteria such as a maximum number of samples in the training data set. As long as none of these termination criteria is met, new sample points to be added to the data set are determined in step 6. In AL many strategies for selecting additional samples – such as "rapid exploration", "maximized model change" or "reduction of estimated error" have been established [31] [32]. According to Settles [20] and Géron [30] one of the most popular strategies is "uncertainty sampling", where the additional samples are added in areas of the highest uncertainty of the current model, which is particularly interesting in combination with Gaussian Process (GP) models, since they indicate model uncertainty based on the variance (see also next chapter) [33] [34]. The data generation phase with adaptive sampling is terminated when the respective conditions (e.g. model prediction performance) are met. The trained model can then be used to make predictions within the design space in step 7.
Several examples in the engineering and materials science domain demonstrated the general applicability and potential benefits of such AL schemes [18] [19] and have generally shown that deliberate adaptive sampling strategies are superior to random selection in minimizing the number of experiments needed [35]. However, considering the dynamics inherent to such a pipeline architecture, the specific tailoring of the generated training data set for the model and prediction task at hand is an obvious consequence. This implies, that neither the model nor the data set are predestined to be reused in similar application scenarios in a sustainable way. While using performance metrics (e.g. accuracy) of the model to monitor and steer the data generation and training process enables optimizing for the specific prediction tasks, it also bears significant risks for blind spots regarding the representative quality of the data set.

THE ADAPTIVE DATA GENERATION PIPELINE

Pipeline overview

Building on the previously introduced and established data generation and learning schemes, a novel general pipeline architecture is proposed. With the intention to efficiently generate reusable training data sets representing the definitive characteristics of a system's behavior (here: crashworthiness of structures under crash conditions) this general architecture allows for its customization to the respective application at the most critical steps. As depicted in Figure 3, the novel pipeline consists of the two phases "Data Generation" and "Data Exploitation" as well. However, one of the main distinctive features of this architecture is that the result of the adaptive data generation phase is the generalist training data set instead of a trained ML model. Although this phase closely resembles the one from the AL scheme, there are some key differences to be pointed out. In the first iteration, after processing the data in step 3, a subset of the data is branched off to serve as an unbiased test set for later training processes in the second phase. (Depending on the overall number of samples and the size of the seed batch this might also occur in another early iteration). This is relevant, since the adaptive sampling process can also be controlled using formalized expert knowledge, which is expected to introduce a (beneficial) bias into the data set leading to more representative characteristics but does not reflect the probability distribution of the application scenario. This could also be interpreted as a "data set overfit". Steps 4 and 5 serve to analyze the current data set

for its information content and to monitor the progress of the iterative sampling loop. It is crucial to state that the one - or multiple - ML models trained in step 4 merely serve as tools to probe the data set in generation and are not used in later prediction tasks. In order to reach the goal of a generalist training data set, the response quantity predicted by the ML models during the data generation phase could be different from the response quantities considered in the later application (data exploitation phase). By combining several output quantities of the system, one could in addition introduce expert knowledge in order to evaluate the information density of the data set in a generalized way. The data quality evaluation is based on the predictive qualities developed by the trained models but can (and should) be extended by additional metrics such as importance-driven sampling density in individual dimensions or "regions" of the design space. Steps 4 and 5 is where the representative capabilities of the data set are optimized using customized metrics based on formalized expert knowledge, which, for instance, could be stated as "(relative) information density requirements".

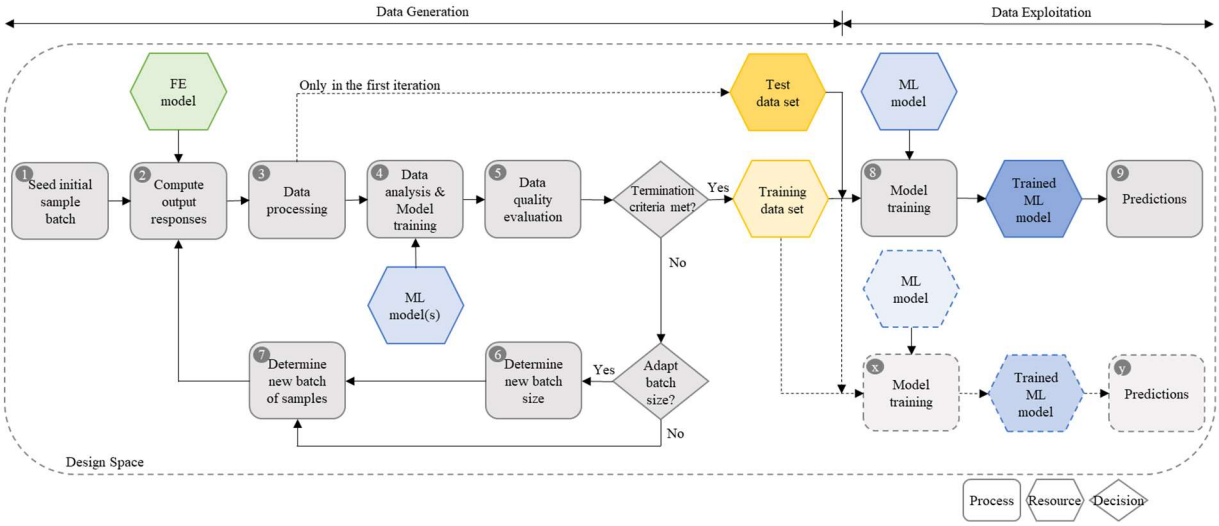

Figure 3. Generic flowchart of the proposed adaptive data generation pipeline.

After step 5 the first check taking place is for the fulfillment of any termination criteria. These could again be stated as a maximum number of samples or iterations. Additional criteria should refer to the data set quality and could be derived from model prediction quality metrics, data / information density analytics or – potentially even more powerful - gradients thereof basically representing metrics of convergence. Given that no sufficient criterion is met, these gradients can then be used in the next step checking the conditions for the adaption of the batch size (defining the number of samples generated and processed in the next iteration). These – relative and thus generalizable - conditions can be adapted individually for each application since they implement an optimum strategy to dynamically balance the resources spent for generating new samples (with expensive FE simulations) and model training or data analytics considering the current state of information content and its convergence, respectively. In step 6 the batch size is adapted to the latest gradient of the metrics used to estimate the convergence considering the predefined strategy. The new samples defined in step 7 are then selected based on the chosen sampling strategies (see previous chapter). Depending on the models and analytics employed, a promising combination of strategies might be "uncertainty sampling" with an additional "exploration" or "space filling" criterion [36]. These samples, or rather the respective system responses, are then computed back in step 2.
In Figure 3 the second phase of "Data Exploitation" schematically depicts the multiple paths of training (step 8 or x) ML models and using them to make predictions (step 9 or y) within the limits of the design space using the generated data set. This is generally enabled, since the data set is engineered to represent the systems' characteristic behavior, rather than being tailored to specifically fit a certain model and prediction task (e.g. predict structural kinematics - not critical values for specific crashworthiness performance metrics). Naturally, there is a trade-off between generalizability and respective specific prediction quality, which is expected to also be specific to the individual system and task(s). However, especially given the particular conditions in vehicle safety engineering design, there are at least three potential scenarios benefitting from such an overall approach. First, it can be expected, that rather simple – and from an expert point of view - standard prediction tasks will be trainable with the same generated data set. And that applies to a comparably larger share of the design space than with existing pipelines with a high degree of specialization. Second, by employing transfer learning schemes [37], it is generally possible to re-train a model to perform in a similar prediction task by adding few data points to the training data set, which will increase the overall process efficiency and sustainability. Third, similar to the transfer learning approach, it can reasonably be anticipated that a data set generated with the proposed pipeline and a respective set of metrics will function as a baseline or "fundamental" data set, which comprises the majority of the system's

relevant information. This baseline data set can then be extended with only a few additional sample points to efficiently customize it as a branch for a specific application. This could also happen adaptively, which could then resemble a combination of phase 1 of Figure 3 with the AL scheme depicted in Figure 2, where the initial seed batch (step 1) would be the adaptively generated baseline data set.

Detailed description of the core processes and algorithms in the current implementation

Following the conceptual introduction of the proposed pipeline architecture, this chapter describes a first base implementation and the respective algorithms used in the individual steps. As preliminary note, it is stated that this implementation is intended to study the dynamics and impact of the overall approach and can be considered the groundwork for multiple enhancements, extensions and complements in the future. In the following, all steps or sub-processes are briefly described with their algorithms and the underlying theory.

Step 1: The initial samples are generated using Latin Hypercube Sampling (LHS), which is a well-established and widely used sampling method in the engineering community. The core idea is to divide the design space into "boxes" (or hypercubes) of equal probability and sample one data point randomly within each box. This method is mainly used for relatively sparse sampling schemes and provides rather stable results (with respect to mean value and distribution) compared to other methods such as Monte Carlo Sampling. [24]. To even improve the space filling characteristics of LHS one could implement Optimal Latin Hypercube Sampling (OLHS) which enforces a certain minimum distance between samples, but might increase computational costs significantly – especially for high-dimensional design spaces [38].

Step 2: In engineering structural design applications such as the one at hand, the system's responses are computed using a regular FE simulation framework such as LS-Dyna [39]. In general, these "ground truth" observations might however also be generated by evaluating an analytic function or any other "oracle" as termed in the AL domain.

Step 3: With FE simulation frameworks the system's response is often written to a binary file, which is generally incomprehensible to humans. Python modules such as lasso-python [40] aid in reading, writing and automated processing of these binary files. Additionally, because these files are gigabyte-sized, it is efficient to read the necessary response parameters for multiple simulations and tabulate them using python modules such as pandas [41]. In this implementation, the entire process from multiple raw simulation data output to easily processable, aggregated and tabulated parameters of interest is automated with python scripts and pandas dataframes.

Step 4: In the current implementation, the ML model used is a Gaussian Process (GP) regression model. The GP represents a generalization of the Gaussian distribution and can be used to define distributions over functions [42]. A GP is defined by its mean and covariance function parametrized using hyperparameters and can be utilized as prior for Bayesian inference [42]. The posterior mean and variance are determined considering the training data and can be used to make predictions at unseen locations of the design space [42]. In this study, the python based Gaussian Process framework GPy is used [43]. The applied kernel is based on a combination of *matern*, white noise and linear kernel considering anisotropic length scale parameters. One key benefit of using a GP model is that the model indicates variances, which can be interpreted as "model uncertainties". This information can be used to significantly benefit strategic sampling of additional data points. Generally, all other model types are feasible to serve as data set probing tool in step 4. Running different models in parallel might again increase computational costs, but yield a rich assessment of the current data quality.

Step 5: Two distinct metrics are used to estimate the increase in information (density) and global convergence. The distance metric, here the root mean squared difference (RMSD), measures the change of the characteristics of the meta model response surface which is assumed to approximately quantify the relative information gain in the latest iteration – by comparing the model predictions at pre-defined evaluation points in the current iteration *(iter)* to the previous iteration *(iter-1)* using Equation (1).

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N}\left(\hat{y}_{\,iter}(i) - \hat{y}_{\,iter-1}(i)\right)^2}{N}} \qquad \textit{Equation (1)}$$

$where$

$N : number\ of\ evaluation\ points$

$\hat{y}_{\ iter}(i) : predicted\ value\ of\ i^{th}\ evaluation\ point\ in\ the\ current\ iteration$

$\hat{y}_{\ iter-1}(i) : predicted\ value\ of\ i^{th}\ evaluation\ point\ in\ the\ previous\ iteration$

The performance metric, here the root mean squared error (RMSE), computes the standard deviation of the residuals or prediction error using Equation (2) to assess the improvement with respect to the actual (ground truth) value [44].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(y(i) - \hat{y}(i)\right)^2}{N}} \qquad \textit{Equation (2)}$$

$where$

$N : number\ of\ evaluation\ points$

$y(i) : actual\ value\ at\ i^{th}\ evaluation\ point$

$\hat{y}(i) : predicted\ value\ at\ i^{th}\ evaluation\ point$

In a real application, the RMSE is typically calculated on the data points that are not utilized to train the ML model. Due to the possibility of a sparsely sampled initial seed, dividing the dataset into training and evaluation datasets may result in evaluation points that do not cover the interesting areas of the design space. Cross-validation-based methods, on the other hand, can facilitate the ability to use all the available samples for the evaluation [45].

In the engineering design application of this study, where the ground truth is not available, cross-validation for RMSE calculation is utilized. The RMSD is calculated by evaluating the GP regression model at pre-defined high-density evaluation points. These could either be defined by densely grid-sampling the entire design space or by using another LHS step adding randomness to the selection process.

The current termination criterion is merely based on the maximum number of samples to be generated in the iterative process. This suffices to analyze the essential pipeline dynamics, but should be extended to thresholds for more sophisticated data set quality metrics and especially their gradients indicating information saturation.

Step 6: In this first implementation the dynamic batch size adaption is yet left to included. The following application examples do however study the effect of different – despite static – batch sizes on the overall pipeline dynamics. The findings can indicate potential strategies in dynamic batch size adaption considering information content convergence and resource needs at the different sub-processes.

Step 7: As mentioned in Step 4, one key benefit of using a GP model is that the model indicates variances, which can be interpreted as "model uncertainties." In the regions with lower variance, it can be assumed that the model uncertainties are lower. Therefore, the new samples are selected in those regions with the highest uncertainties to minimize the variances and thus increase the anticipated model prediction quality or "confidence" [46]. In order to locate new samples in the design space this is selected as the primary criterion, mathematically represented in Equation (3).

$$x_{new}^{1\text{st crit.}} = \underset{x \in D_{cand}}{\operatorname{argmax}}\, \hat{\sigma}^2(x) \qquad \textit{Equation (3)}$$

$where$

$x_{new} : New\ sample\ point\ (location)$

$\hat{\sigma}^2(x) : Model\ prediction\ variance\ at\ point\ x$

$D_{cand} : Predefined\ sample\ candidates\ in\ the\ design\ space\ D$

As it is stated above, Equation (3) is directly applicable only to the batch size of 1. When simultaneously adding multiple samples (e.g. batch size of *n*), the *n* points of largest model variance will be selected as next sample locations. With the progression of iterations and with the introduction of larger batch sizes,

a pure variance-based addition of samples could result in a strong localization of samples in the design space. In order to prevent this local aggregation, a secondary distance-based criterion using a space-filling metric is utilized "intra-iterations" to prevent local clustering of the samples added within one batch and "inter-iterations" to prevent clustering of the samples added in the current and in previous iterations [36]. The space-filling metric *S* as mathematically represented in Equation (4) and the new sample selection in the design space using primary and secondary criterion as represented in Equation (5) are based on the works of Aute et al. [36].

$$S = 0.5 \times \max(S_N) \qquad \textit{Equation (4)}$$

where

S : *Space filling metric*
S_N : *List of Euclidian distances for all existing points in* D_{train} to respective closest neighbor

D_{train}: *Set of existing training points in the design space D*

$$x_{new}^{\text{1st \& 2nd crit.}} = \underset{x \in D_{cand}}{\operatorname{argmax}}\, \hat{\sigma}^2(x) \qquad \textit{Equation (5)}$$
$$\text{s.t. } \|x_{new} - x_k\|_2 \geq \text{S}, \qquad \forall\, x_k \in D_{train}$$

where
x_{new} : *New sample point (location)*

In this initial implementation the "Data Exploitation" phase not yet included. A full implementation can be achieved by extending the current one with a basic ML pipeline suitable to majority of applications in structural engineering. While this groundwork implementation suffices to reach the goal of this study, which is to understand the basic effects and dynamics of this adaptive data generation scheme while generating the data, it is clear, that an objective measurement of the overall performance (final prediction quality and overall data generation efficiency) is still left to be conducted in future work. A respective proposition is to be found in the last chapter of this paper.

Application of the pipeline to a generic example

As a first step, the data generation pipeline is applied to a generic mathematical example problem. This helps to demonstrate the functionality and dynamics of the pipeline while relating to rather clear expectations of the outcomes and having global access to ground truth.
Since it used in a wide range of meta-modeling and sampling methods applications, the function of choice for this study is the bimodal non-linear Hosaki function as defined in Equation (6) [47].

$$f(x) = \left(1 - 8x_1 + 7x_1^2 - \frac{7}{3}x_1^3 + \frac{1}{4}x_1^4\right) x_2^2 e^{-x_2} \qquad \textit{Equation (6)}$$

where

$$0.5 \leq x_1 \leq 4.5 \;\; and \;\; 0.5 \leq x_2 \leq 4.5$$

In order to provide the reader with a clear image of the shape, Figure 4 depicts the response surface (*f(x)*) over the given value ranges from 0.5 to 4.5 for x_1 and x_2, respectively.

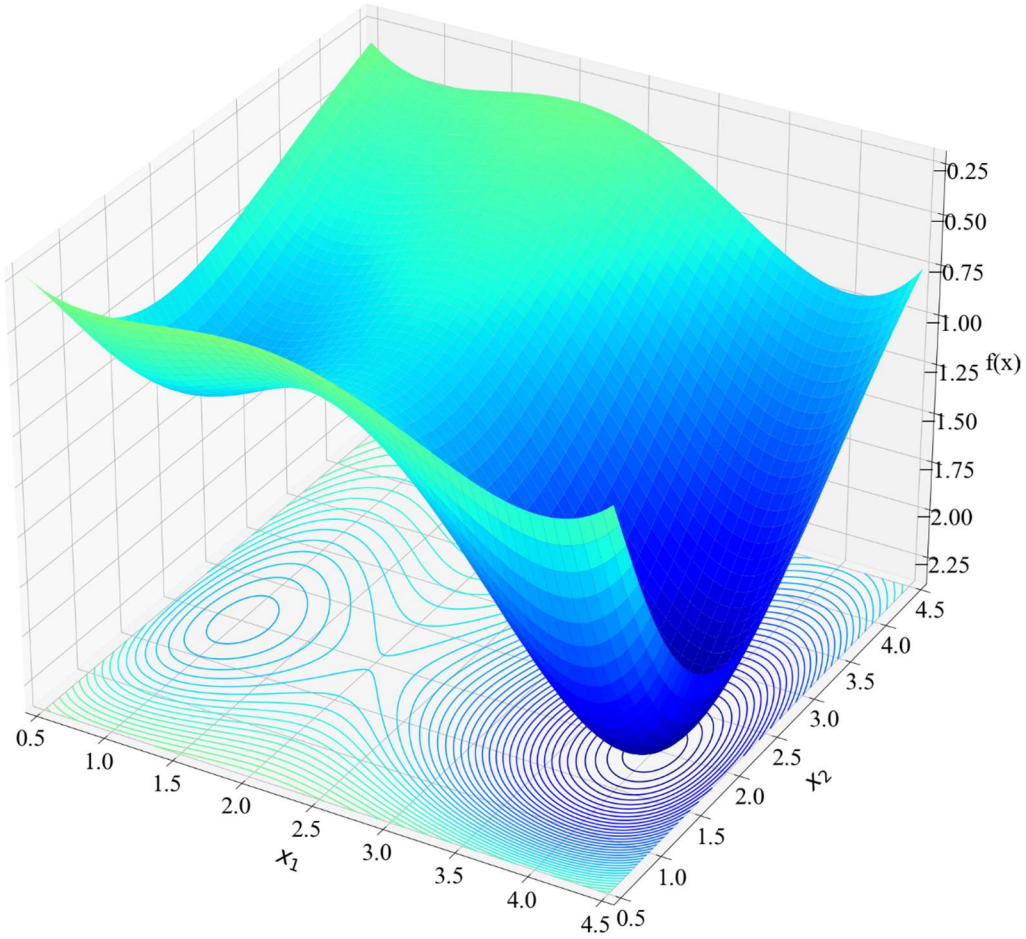

Figure 4. 3D plot of the Hosaki test function.

As for the boundary conditions of the experiment, it is stated that the evaluation points were defined by grid-sampling the design space with a total number 1000 equidistant points. The initial seed batch size was 5 and the maximum number of created samples was set to be 100. In order to study the effect of differently – despite statically – sized batches several experiments with fixed batch sizes of 1, 2, 4, 8 and ten samples per batch were conducted. The RMSE was calculated by relating the model prediction to the analytical ground truth given by Equation (6).

Figure 5 depicts the plot of RMSD and RMSE values over the generation of 100 samples (or iterations with a batch size of 1). For the RMSD and RMSE plots, a global convergence is observed as the model approximates the Hosaki ground truth with an increasing amount of information (samples) to be trained on. Even though a global convergence is anticipated overall, the early phase of sampling is also expected to see temporary increases in RMSE as a result of a sample point adding information to the very small body of existing information. This temporary increase is only observed at an initial stage and is not observed at a later stage, as would also be expected, when the body of existing information is already quite substantial. An additional effect might be that the model could, by coincidence, initially have seen critical, definitive samples. This average of high information quality is then drastically decreased by adding a sample of significantly lower learning value in an early phase.

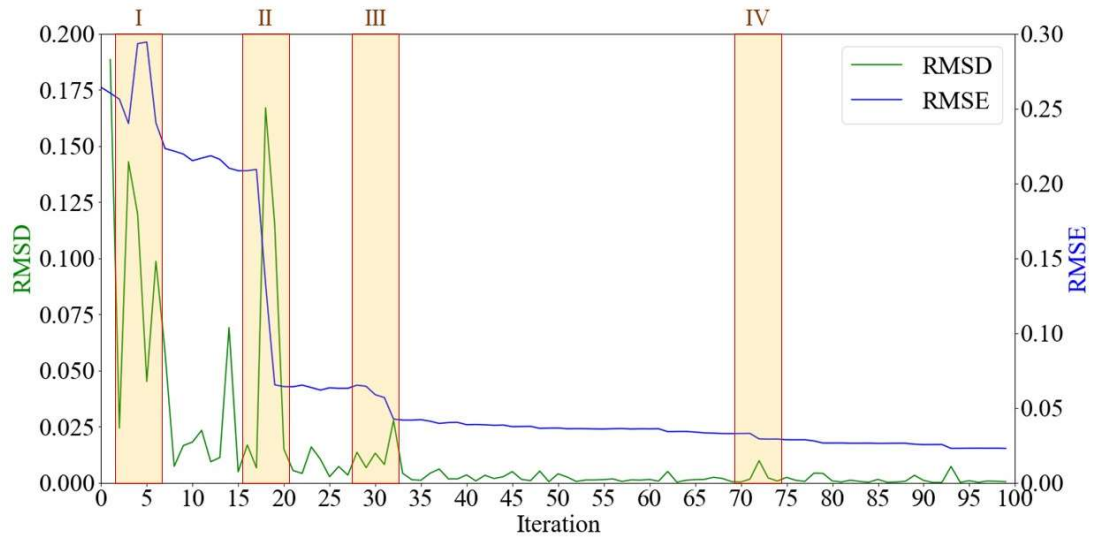

Figure 5. Plot of the RMSD and RMSE values over generated samples / iterations with batch size 1 for the Hosaki test function. Orange Boxes indicate the iteration ranges shown in detailed contour plots in Figure 6.

Along with model performance, it is seen that distinct features appear simultaneously in the RMSE and RMSD plots. This confirms the expectation that measuring a spike in RMSD, representing a significant change in the characteristic shape of the response, clearly relates to a significant change (for the better or the worse) in the RMSE. At a later stage, only decreasing RMSE values are anticipated with a "better educated" model and iterations 18 - 20 offers a remarkable illustration of the same. These peaks in RMSD are not limited to earlier stages, but are also identifiable at later stages in the sampling process. Despite the variations in RMSE being hardly noticeable when the model has already converged to a larger degree, the respective spikes in RMSD are comparably significant. This indicates their suitability to be used as termination criteria indicating information saturation.

Figure 6 depicts the contour plots for the model response surface, the residuum with respect to the Hosaki ground truth and the model variance for a selection of four times five consecutive iterations with batch size 1. These iteration streaks are also indicated in Figure 5.

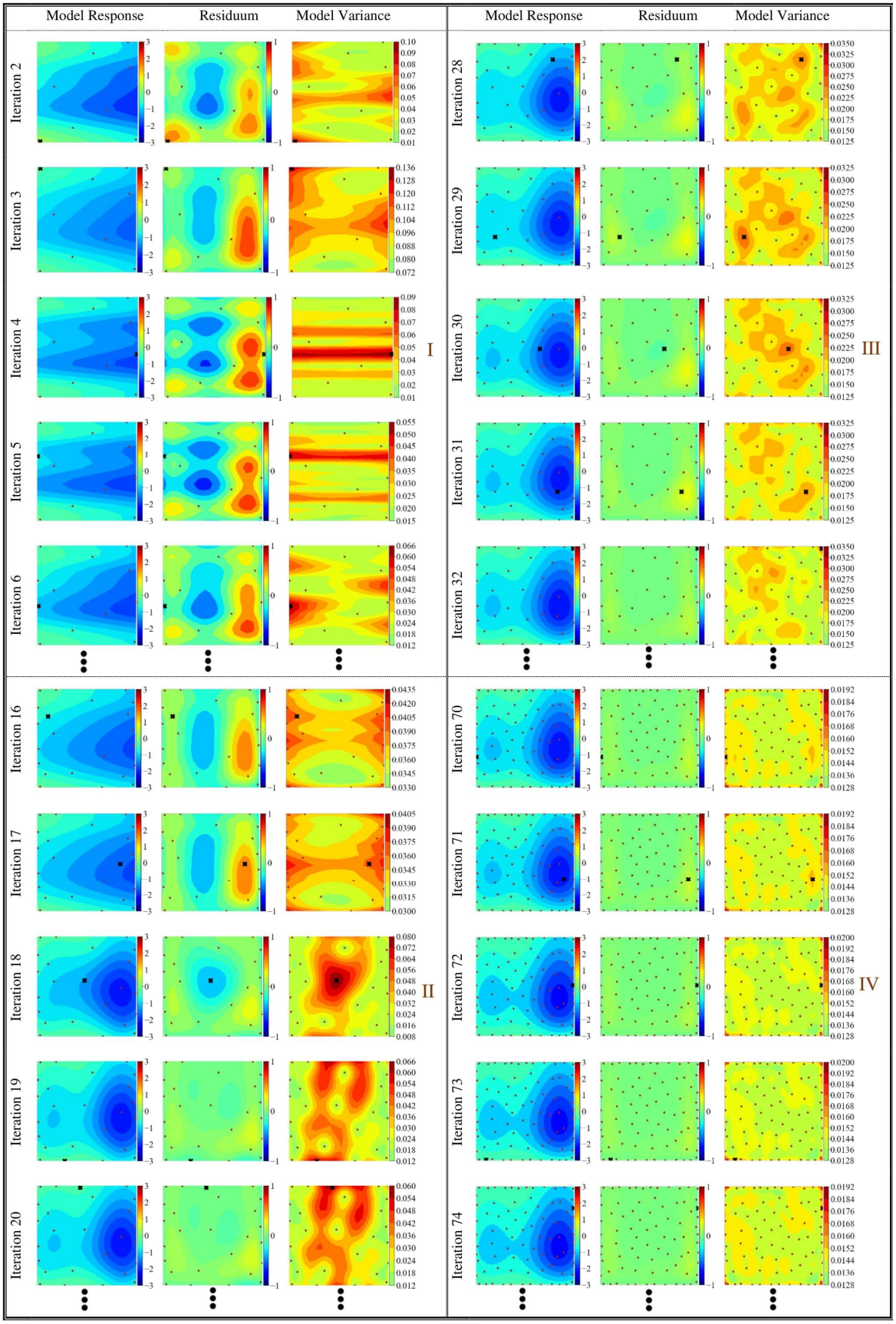

Figure 6. Contour plots of the model response surface (left), the residuum w.r.t. Hosaki ground truth (center) and model variance (right) for selected iterations (also indicated in Figure 5). The red and black dots indicate all currently available samples and the location of the new sample which is added to the training data to train the model of the next iteration (batch size 1), respectively.

As depicted in Figure 6, with increasing amounts of training information, the response surface convergence towards the Hosaki function is observed as expected. Simultaneously the residuum converges towards zero and does not show major changes in later phases. Additionally, the new point selection can be visually verified. The new sample points are always located where the highest model variance is indicated. In the following iterations this indication is shifted to a location without previously sampled information. Although the enforcement of the distance-based secondary criterion (especially the "intra-iterations") can't be observed here, it has diligently been evaluated and found to be effective.

Figure 7 depicts the plot of the RMSE over generated samples with multiple fixed batch sizes. Initially, it is seen that the resolution of the individual curves over the abscissa is defined by the batch size. This could be interpreted as a "model-wise" (as trained model) resolution of the respective performance in comparison to the models trained with a different batch size and thus different frequency.

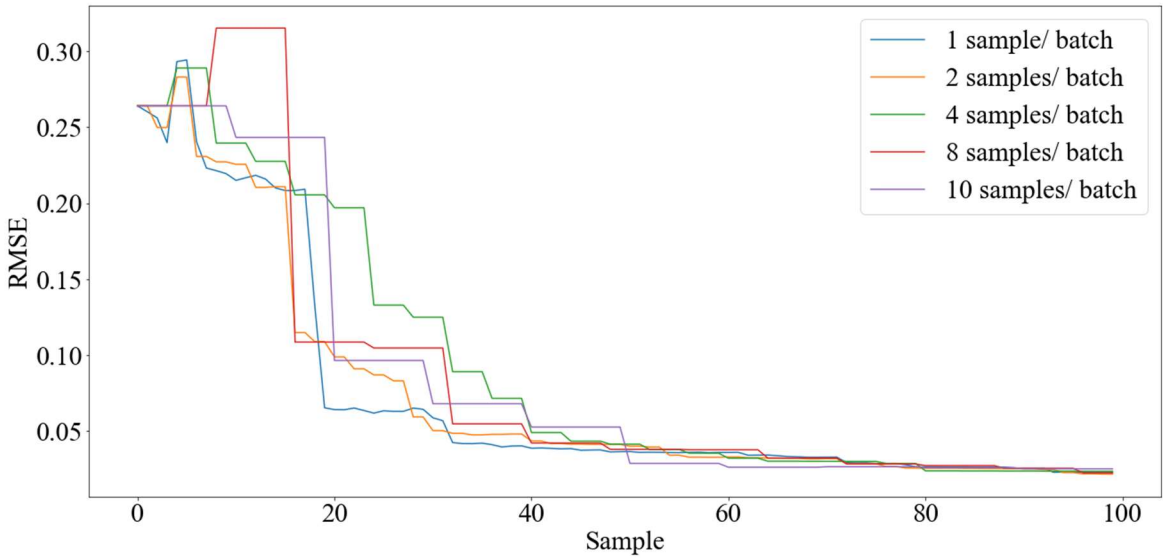

Figure 7. Plot of the RMSE over generated samples with multiple fixed (static) batch sizes for the Hosaki test function.

It can be observed that smaller batch sizes, as compared to larger ones, yield a gradual and a faster convergence towards lower error values. Particularly when looking at the higher plateau for batch size 8 over samples 8 – 16, one can conclude that adding a lot of information (larger batch) based on little (uncertain) information in an early stage does not result in a beneficial strategy regarding dynamic batch size adaption. The mutual convergence of the RMSE curves at a later stage indicates that the batch size doesn't affect the overall model convergence significantly. However, there are several other aspects (e.g. sub-process-specific resource consumption) to be considered when crafting a batch size adaption strategy.

Figure 8 depicts the plot of the RMSD value normalized by batch size over the sample generation for multiple fixed batch sizes. It can be observed that lower batch sizes yield higher values than larger batch sizes over the entire data generation process. This is expected since the information taken into consideration when defining a new sample point is maximized when minimizing the batch size. With a batch size of one, every single sample is the actual "next best sample" with respect to the (theoretically) available information. Increasing the batch size changes this ratio for the worse. This effect is additionally amplified by normalizing the "added value" per sample by batch size. Lower relative values per sample for higher batch sizes are then additionally related to a larger batch size number.

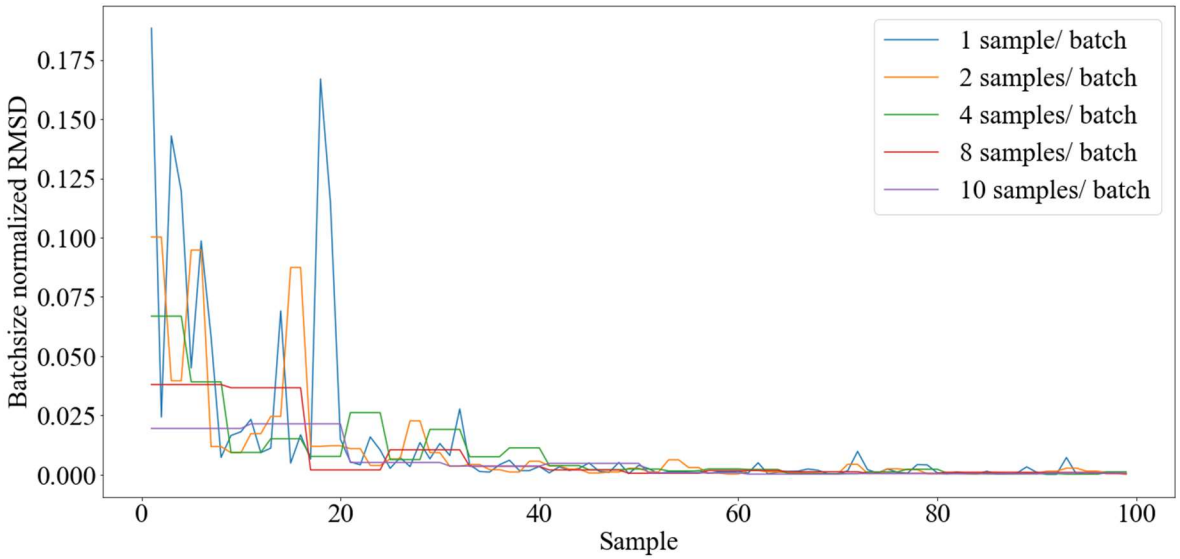

Figure 8. Plot of the RMSD values for multiple fixed (static) batch sizes normalized by batch size over the sample generation for the Hosaki test function.

Even in very late stages of the data generation process, small batch sizes still yield significantly higher relative RMSD values per sample than larger ones.

APPLYING THE PIPELINE TO GENERATE A CRASH SIMULATION TRAINING DATA SET

The FE Simulation Model – Pre-processing, Load case, Design space

The aim of project ATTENTION is to develop a framework and the constituting methods to predict the injury risk of VRUs in real-time using ML and simulation-based training data. This data is generated with a representative FE vehicle model [48] and the Total HUman Model for Safety (THUMS^{TM}) V4.02 AM50 Pedestrian [49] (see Figure 9). The overall project scope covers the extraction of vehicle-bound video data to determine parameters of accident scenarios, the generation of simulation-based training data as well as the training of advanced ML models aiming for the real-time prediction of situation-specific injury risks of VRUs. In future applications, this situation-specific injury risk prediction could support decision making in determining active risk minimization strategies.

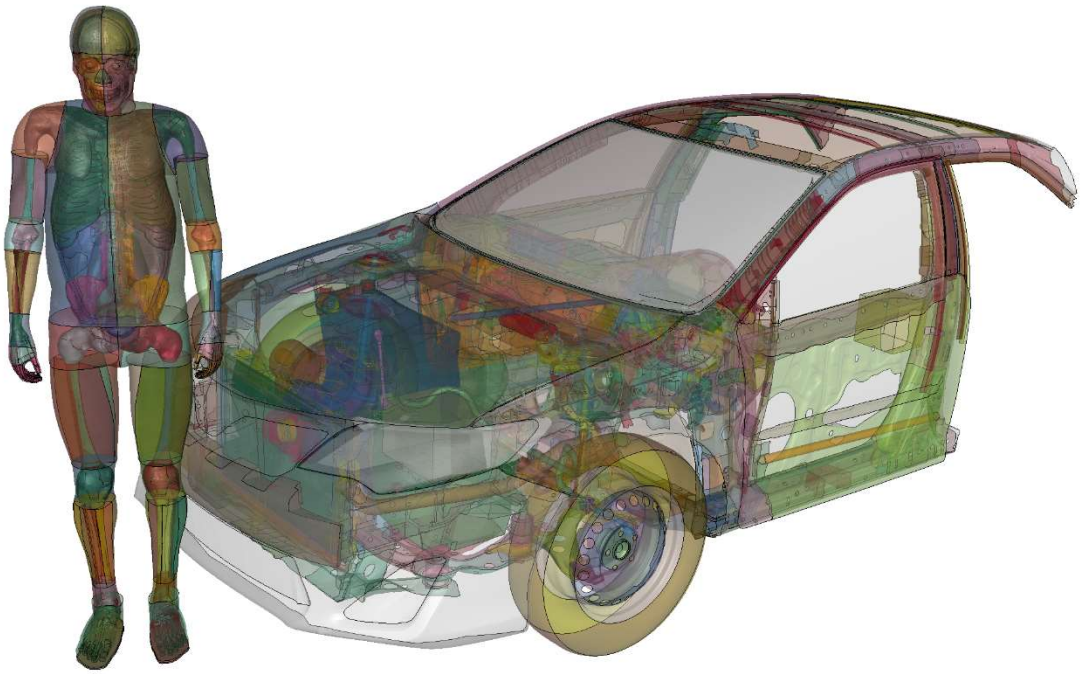

Figure 9. Full-scale simulation setup based on adapted Toyota Camry FE model and Total Human Model for Safety (THUMS™).

The first step of data generation is the preparation of the FE models. This includes a load-case-specific reduction of the vehicle model complexity in order to reduce computational costs and storage space while retaining important kinematic and dynamic properties as well as computational stability (see Figure 9). In the next step, parts with an extensive effect on the injury-relevant variables are analyzed in detail (e.g. windshield). Here, in particular, further effort is made to improve the predictive capability of the structural behavior of the windshield model, typically consisting of a PVB mid-layer and two outer glass layers (see Figure 10), making use of the latest results provided by the scientific community [50] [51] [52] [53] [54].
The PVB layer is modeled using solid elements and LS-Dyna *MAT_HYPERELASTIC_RUBBER (material data used from Osnes et al. [52], Jaware et al. [53] and Alter et al. [54]), the glass layers are modeled using shell elements and LS-Dyna * MAT_GLASS (material data used from Osnes et al. [52]). In similar fashion, the layers are connected using shared nodes for shell and solid elements. In order to bond the windshield to the body in white, an additional part, representing the bonding layer (material data used from [55]), is used at the windshield's edges in combination with LS-Dyna *CONTACT_TIED_NODES_TO_SURFACE similar to the connection used in the NHTSA Honda Accord Model [55].
Since this study focuses on the efficient generation and optimization of simulation-based training data using adaptive sampling methodology, an exemplary small-scale sub-model is extracted from the above-mentioned model setup (Toyota Camry, THUMS™) used in the ATTENTION project. This sub-model consists of the adapted windshield model impacted by a human head model isolated from the THUMS™ model. The rear side of the bonding layer (initially attached to the body in white) is fixed in space. The THUMS™ head is extracted retaining important instrumentation for injury measurement - such as accelerations, (angular) velocities and strains - which allows for the calculation of various injury criteria, such as HIC [56] [57], BrIC [58] and CSDM-calculation from the white and grey brain matter [59].

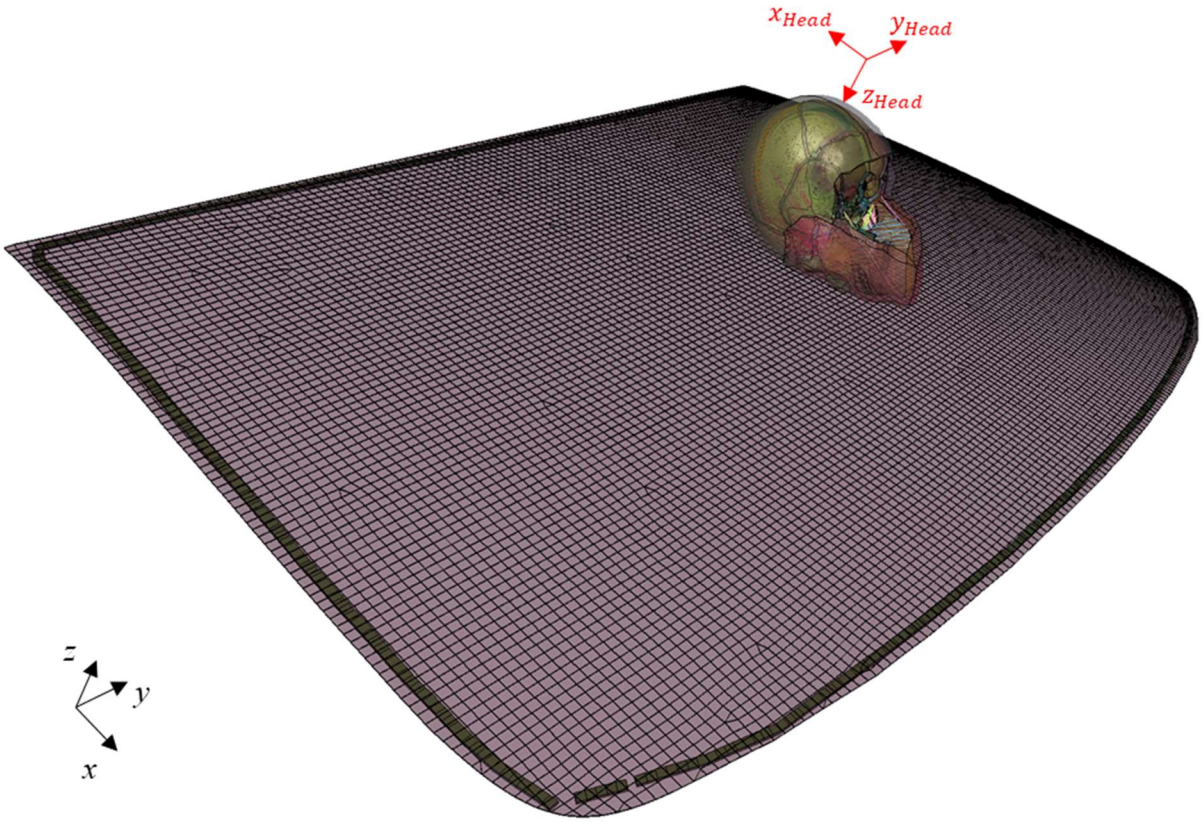

Figure 10. Sub-Model extracted from full-scale simulation model based on adapted windshield model and THUMS™ Head. *(Nominal position: x-rotation = 0°, y-rotation = 270°, x-position = y-position = 0 mm)*

The considered parameter space is spanned by the parameters listed in Table 1. These parameters and their ranges are derived from a simulation study using the above-mentioned full-scale model considering information about relevant real-world accident scenarios provided in the GIDAS (German In-Depth Accident Study) database [60]. The termination time is defined dynamically based on loss of contact, calculation time using 32 CPUs is about 10 minutes. Defined output leads to a storage consumption of 3.5 GB per sample. The training data was generated using an adapted LS-DYNA MPP 9.3 Version on AMD EPYC 7763 CPU [39].

Table 1. Parameter space used during data generation considering the sub-model

Parameter	Minimum	Maximum
Head velocity	5 km/h	45 km/h
x-rotation (Head COS)	270°	360°
y-rotation (Head COS)	180°	270°
x-position (global COS)	-300 mm	370 mm
y-position (global COS)	-550 mm	0 mm

Adaptive Data Generation Cycles

As for the boundary conditions of the experiment, the evaluation points were defined by grid-sampling the design space with a total number 1000 equidistant points. The initial seed batch size was 5 and the maximum number of created samples was set to be 120. The GP regression model is adjusted to process 5 input parameters, but the actual model is the same as the one previously explained. For simplicity, the maximum resultant acceleration of a representative head node is chosen to be the output response for the GP model. As the initial goal is to compare the results with the results from applying the pipeline to the generic Hosaki function example, experiments with fixed batch sizes of 1, 2, 4, 8 and ten samples per batch were conducted. Given the high cost of calculating the ground truth for each evaluation point for RMSE and the limited availability of training samples in the initial iterations, an RF regressor with 100 trees is utilized along with 5-fold cross-validation. As the RF regressor is only used for model evaluation and not predictions, the white-box nature of the model could potentially be used for

interpretability using input feature importance and proximity plots. The RF regressor used in this experiment can be replaced with any other ML model, including neural networks.

Figure 11 depicts the plot of RMSD and RMSE values over the generation of 120 samples with a batch size of 1. Similar to the example problem results in Figure 5, a global convergence is observed as the model approximates the maximum resultant acceleration response surface. The changes in the RMSD are seen to clearly resemble those in the RMSE, but the coherency observed in Figure 5 is more significant. In contrast to a fixed relative saturation criterion, considering a floating saturation criterion (using statistical metrics like mean and max over a particular number of iterations) may be of interest given the local spikes in RMSD.

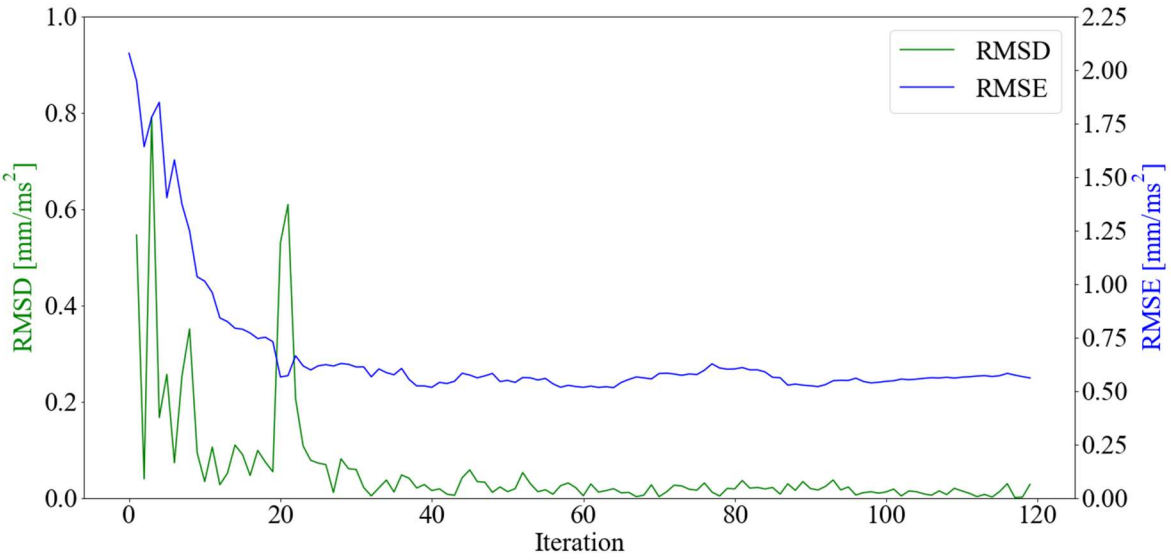

Figure 11. Plot of the RMSD and RMSE values over generated samples / iterations with batch size 1 for the sub-model (adapted windshield model and THUMS^{TM} Head).

Figure 12 depicts the plot of the RMSE over generated samples with multiple fixed batch sizes. Similar to the results in Figure 7, it can be observed that a smaller batch size results in a gradual and faster convergence towards lower error values. From the mutual convergence of the RMSE curves, it is also noted that, at a later stage, the batch size doesn't affect the overall model convergence, but a smaller batch sizes seem to have a better RMSE performance compared to bigger batch sizes.

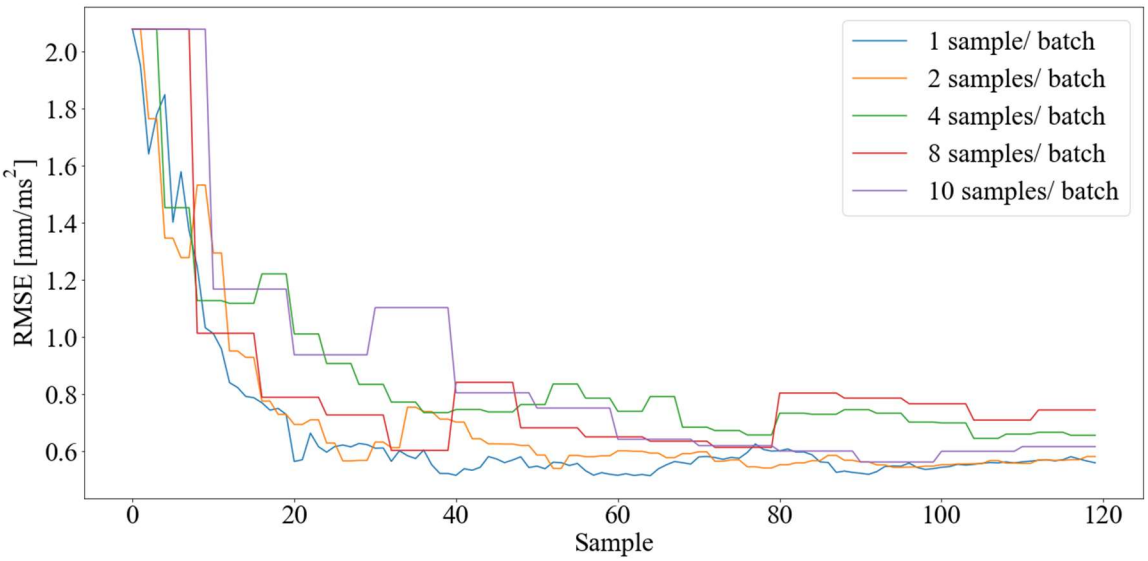

Figure 12. Plot of the RMSE over generated samples with multiple fixed batch sizes for the sub-model (adapted windshield model and THUMS^{TM} Head).

Figure 13 depicts the plot of the RMSD value normalized by batch size over the sample generation for multiple fixed batch sizes. Similar to the results shown in Figure 8, it can be observed that lower batch sizes yield higher values than larger batch sizes over the course of data generation process. Additionally, it is worth noting the similarities between batch sizes 1 and 2. This indicates the similar – and initially advantageous - behavior of smaller batch sizes with respect to the selection of "next best samples" based on the model variance and the existing body of information contained in the data set.

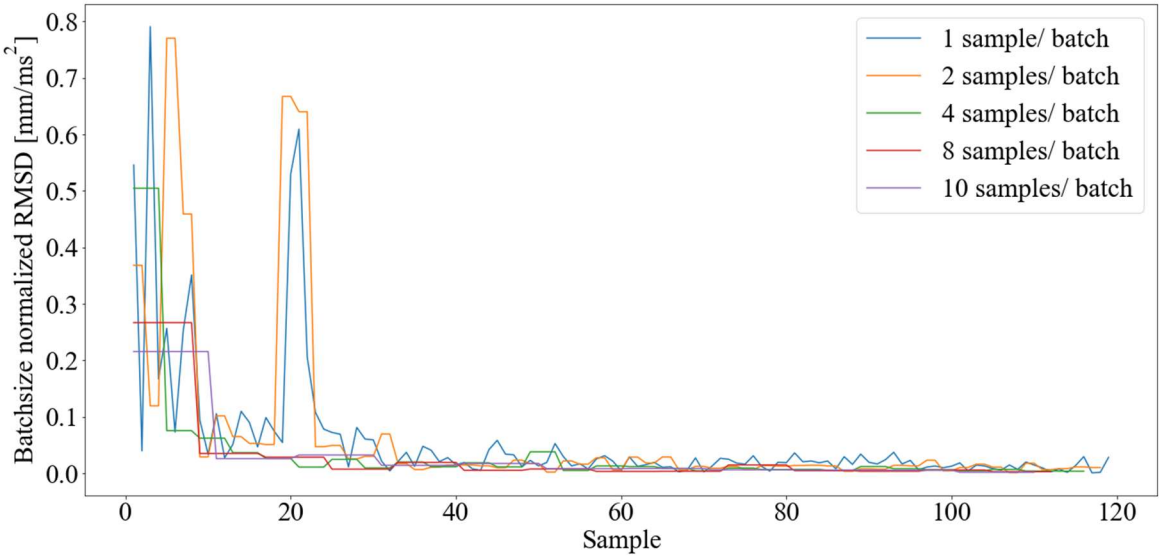

Figure 13. Plot of the RMSD values for multiple fixed batch sizes normalized by batch size over the sample generation for the sub-model (adapted windshield model and THUMS^{TM} Head).

In general, it can be subsumed, that the pipeline dynamics (relative rate of convergence, batch size effects, etc.) of the Hosaki example problem and the real engineering structural design application are very similar.

CONCLUSION

In order to successfully and sustainably apply ML methods for structural engineering tasks in vehicle or traffic safety, a robust approach towards efficiently generating training data with FE simulations is vital. A critical aspect to this success is the reusability of the generated data. Following general notions from the ML domain, one foundational hypothesis of this work is that data reusability in maximized with maximizing generalizability through the representational character of the data set from an expert perspective. As supported by the "no free lunch" theorem, no single ML algorithm universally performs the best for all the prediction tasks and the predictions quality strongly the quality of the data it is trained on [61]. This additionally motivates the key focus on the data itself and optimization of its representation of the system's complex behavior following a data-centric ML approach.

This study proposes a novel general pipeline architecture to generate data sets, which - within reasonable limits - represent all the relevant characteristic features of the system's behavior (crashworthiness characteristics) without tailoring it to a very specific ML application. After introducing the pipeline architecture with reference to existing solutions, it is applied to a generic mathematical example problem before applying it to a real vehicle safety application comprising FE simulation for data generation.

The result of this first implementation of the proposed architecture with a fixed batch size and no advanced termination criterion suggests that the pipeline can aid in generating a simulation data set that represents the relevant characteristic features of the system's behavior. The results reflect the expected behavior for the generic example and are confirmed in the real application scenario. This proves the general applicability of this novel pipeline architecture and supports the hypothesis regarding the significant potentials that lie in the multiple scaling routes and extensions of the same.
Technical conclusions include the confirmation of the potential of considering the RMSD as an indicator of convergence or the gradual saturation of information density in the data set. As indicated above, floating conditions might prove to work better than discrete threshold values as termination criteria. Furthermore, it is important to mention the strong effects of the meta model parameter settings. Considering the strong implications of changing the GP model kernel and the effects of using isotropic or anisotropic length scale parameters highlights the pitfalls

of model dependent adaptive sampling processes as seen on conventional AL schemes. Using multiple model types simultaneously might thus be a good solution to avoid overlooking such issues.

As mentioned, this implementation should be considered the groundwork for future extensions and adaptions. One major task is to examine the overall performance of the proposed architecture by comparing the final prediction quality in the exploitation phase to models trained on uniformly (LHS) sampled training data sets and by evaluating transfer learning and data set extension/adaption approaches. This requires extensive study efforts, which will be prioritized in future works. Additional future extensions include employing dimensional reduction methods, such as learned manifold mode representations [62], in step 3 (data processing) to increase the information density of the data itself. Simultaneously employing multiple model types and their effects on "data set representativeness" in step 4 is also left to be studied in the following steps. Furthermore, the learnings regarding potentially beneficial strategies for the dynamic batch size adaption need to be implemented and studied. One of the main things to be implemented is the introduction of expert-driven metrics for evaluating the data set quality and – by that – steering the data generation process. As indicated, promising directions include the definition of dimension-specific sampling densities and of design space regions of increased relevance.

*The project ATTENTION is funded by the German Federal Ministry for Economic Affairs and Climate Action.

REFERENCES

[1] M. Haag, „Verkehr und Verkehrssicherheit im urbanen Raum," in *Freiburger Dialog*, Freiburg, 2022.

[2] W. Lerner, „The Future of Urban Mobility - Towards networked, multimodal cities of 2050," Arthur D. Little, 2011.

[3] F.-J. Van Audenhove, L. Dauby, O. Korniichuk und J. Pourbaix, „The Future of Urban Mobility 2.0 - Imperatives to shape extended mobility ecosystems of tomorrow," Arthur D. Little, 2014.

[4] D. Adminaité-Fodor und G. Jost, „How safe is walking and cycling in Europe?," European Transport Safety Council, Brussels, 2020.

[5] BaWü Statistisches Landesamt, „Mehr verunglückte Fahrradfahrende im ersten Halbjahr 2020.: Baden-Württemberg: Zahl der Getöteten dennoch rückläufig – Verunglückte mit Pedelecs legen um 47 % zu," Stuttgart, 2020.

[6] J. Moennich, T. Lich, A. Gerogi und N. Reiter, „Did a higher distribution of pedelecs results in more severe accidents in Germany," 2015.

[7] Fraunhofer EMI, „ATTENTION – Artificial intelligence for real-time injury prediction," 2021. [Online]. Available: https://www.emi.fraunhofer.de/en/news/news-press/attention----artificial-intelligence-for-real-time-injury-predic.html.

[8] M. Gonter, A. Leschke und U. Seiffert, „Fahrzeugsicherheit," in *Vieweg Handbuch Kraftfahrzeugtechnik, ATZ/MTZ-Fachbuch*, Springer, 2016, pp. 1105 - 1161.

[9] L. Greve, B. von de Weg und M. Andres, „Necking Prediction using Neural Networks," Baden Baden, 2019.

[10] C. Kohar, K. Inal, D. Kracker und P. Schwanitz, „Applications of artificial intelligence in automotive engineering for crashworthiness design," in *VDI Conference - Automotive CAE*, Baden Baden, 2019.

[11] B. Van de Weg, „Surrogate modeling for finite element simulations," in *SIMVEC*, Baden Baden, 2022.

[12] Landing AI, „What is data-centric AI?," 2022. [Online]. Available: https://landing.ai/data-centric-ai/.

[13] A. Ng, „How AI is changing the future of business," Berlin, 2022.

[14] S. Vasu, N. Talabot, A. Lukoianov, P. Baque, J. Donier und P. Fua, „HybridSDF: Combining Free Form Shapes and Geometric Primitives for effectiveShape Manipulation," arxiv, 2021.

[15] L. Greve und B. Van de Weg, „Surrogate modeling of parametrized finite element simulations with varying mesh topology using recurrent neural networks," *Array,* Nr. 14, 2022.

[16] L. Greve, „KI-basierte Echtzeitmodelle für die schnelle Materialparameterkalibrierung," Freiburg, 2021.

[17] C. Kohar, L. Greve, T. Eller, D. Conolly und K. Inal, „A machine learning framework for accelerating the design process using CAE simulations: An application to finite element analysis in structural crashworthiness," *Computer Methods in Applied Mechanics and Engineering,* Nr. 385, 2021.

[18] L. Chec, „How Machine Learning and AI accelerates automotive design processes. Feedback on 3 different application: battery design, structural optimization and system design," Hanau, 2022.

[19] K. Kayvantash, „A study on the effect of sampling on the quality of AI/ML/ROM models," Hanau, 2022.

[20] B. Settles, „Active Learning Literature Survey - Computer Sciences Technical Report 1648," University of Wisconsin–Madison, Madison, 2009.

[21] Y. Yoo, C.-K. Park und J. Lee, „Deep learning-based efcient metamodeling via domain knowledge-integrated designable data augmentation with transfer learning: application to vehicle crash safety," *Structural and Multidisciplinary Optimization,* Bd. 65, 2022.

[22] J. Brownlee, „MachineLearningMastery - A Gentle Introduction to Transfer Learning for Deep Learning," 16 September 2019. [Online]. Available: https://machinelearningmastery.com/transfer-learning-for-deep-learning/.

[23] M. McKay, R. Beckman und W. Conover, „A Comparison of Three Methods for Selecting Vales of Input Variables in the Analysis of Output From a Computer Code," Bd. Technometrics , Nr. 21, 1979.

[24] F. Duddeck, „Robust Design for Car Body Development - Design of Experiments DoE," Freiburg, 2019.

[25] I. Goodfellow, Y. Bengio und A. Courville, „Machine Learning Basics," in *Deep Learning*, Cambridge, MIT Press, 2016, pp. 98 - 155.

[26] I. Goodfellow, Y. Bengio und A. Courville, „Sequence Modeling: Recurrent and Recursive Nets," in *Deep Learning*, Cambridge, MIT Press, 2016, pp. 373 - 416.

[27] D. Rumelhart, G. Hinton und R. Williams, „Learning representations by back-propagating errors," *Nature,* pp. 533 - 536, 1986.

[28] M. Turek, „Explainable Artificial Intelligence (XAI)," [Online]. Available: https://www.darpa.mil/program/explainable-artificial-intelligence.

[29] R. Roscher, B. Bohn, M. Duarte und J. Garcke, „Explainable Machine Learning for Scientific Insights and Discoveries," *IEEE Access,* pp. 42200 - 42216, 2019.

[30] A. Géron, Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow, Sebastopol. CA: O'Reilly Media Inc., 2019.

[31] B. Settles, M. Craven und S. Ray, „Multiple-Instance Active Learning," in *Advances in Neural Information Processing Systems (NIPS)*, Boston, 2008.

[32] A. Freytag, E. Rodner und J. Denzler, „Selecting Influential Examples: Active Learning with Expected Model Output Changes," in *Proceedings of the 13th European Conference on Computer Vision, ECCV 2014*, Zürich, 2014.

[33] E. Pasolli und F. Melgani, „Gaussian process regression within an active learning scheme," in *2011 IEEE International Geoscience and Remote Sensing Symposium*, Vancouver, 2011.

[34] D. Cohn, Z. Ghahramani und M. Jordan, „Active Learning with Statistical Models," *Journal of Artifical Intellgence Research,* Bd. 4, pp. 129 - 145, 1996.

[35] T. Lookman, P. Balachandran, D. Xue und R. Yuan, „Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design," *npj Computational Materials,* Bd. 5, Nr. 21, 2019.

[36] V. Aute, K. Saleh, O. Abdelaziz und S. Azarm, „Cross-validation based single response adaptive design of experiments for Kriging metamodeling of deterministic computer simulations," *Struct Multidisc Optim,* Bd. 48, pp. 581-605, 2013.

[37] B. Maschler, H. Vietz, H. Tercan, C. Bitter, T. Meisen und M. Weyrich, „Insights and Example Use Case on Industrial Transfer Learning," in *55th CIRP Conference on Manufacturing Systems*, Lugano, 2022.

[38] M. Liefvendahl und R. Stocki, „A study on algorithms for optimization of Latin hypercubes," *Journal of Statistical Planning and Inference,* Bd. 136, Nr. 9, pp. 3231 - 3247, 2006.

[39] Livermore Software Technology Corporation, „LS-DYNA R10.1," Livermore, 2022.

[40] Lasso GmbH, „Lasso Python Library GmbH," Lasso, [Online]. Available: https://lasso-gmbh.github.io/lasso-python/build/html/. [Zugriff am 22 December 2022].

[41] The pandas development team, „pandas-dev/pandas: Pandas," February 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3509134. [Zugriff am 22 December 2022].

[42] C. E. Rasmussen, „Gaussian Processes in Machine Learning," in *Advanced Lectures on Machine Learning, Lecture Notes in Computer Science*, Berlin, Heidelberg, Springer, 2004, p. 63–71.

[43] „GPy: A Gaussian Process Framework in Python," 2012. [Online]. Available: http://github.com/SheffieldML/GPy.

[44] M. Kuhn und K. Johnson, „Measuring Performance in Regression," in *Applied predictive modeling*, Springer, 2018, pp. 95-100.

[45] M. Kuhn und K. Johnson, in *Applied predictive modeling*, Springer, 2018, pp. 69-73.

[46] R. Jin, W. Chen und A. Sudjianto, „On sequential sampling for global metamodeling in engineering design," *International design engineering technical Conferences and Computers and Information in engineering conference,* pp. 539-548, 2002.

[47] G. A. Bekey und M. T. Ung., „A Comparative Evaluation of Two Global Search Algorithms," *IEEE Trans. Syst. Man Cybern.,* pp. 112-116., 1974.

[48] Center for Collision Safety and Analysis at the George Mason University (GMU), Federal Highway Administration (FHWA), „2012 Toyota Camry Detailed Finite Element Model," https://www.ccsa.gmu.edu/models/2012-toyota-camry, 2012.

[49] K. Shigeta, Y. Kitagawa und T. Yasuki, „Development of Next Generation Human FE Model capable of Organ Injury Prediction," in *Proceedings of the 21st Annual Enhanced Safety of Vehicles*, Stuttgart, 2009.

[50] N. Kulkarni, S. Deshpande und R. Mahajan, „Development of Pedestrian Headform Finite Element Model using LS-DYNA and its Validation as per AIS 100/GTR9," in *12th European LS-Dyna Conference*, Koblenz, 2019.

[51] J. Prasongngen, I. Putra, S. Koetniyom und J. Carmai, „Improvements of Windshield Laminated Glass Model for Finite Element Simulation of Head-to-Windhsield Impacts," in *IOP Conference*, Phuket, 2013.

[52] K. Osnes, S. Kreissl, J. D'Haen und T. Borvik, „Modelling of Fracture Initiation and Post-Fracture Behaviour of Head Impact on Car Windshields," in *13th European LS-Dyna Conference*, Ulm, 2021.

[53] A. Jaware, S. Chandratre, M. Perez und J. Narule, „An Advanced Methodology for Windscreen Modeling in LS Dyna," *International Journal of Mechanical Engineering and Technology (IJMET),* Bd. 10, 2019.

[54] C. Alter, S. Kolling und J. Schneider, „A new failure criterion for laminated safety glass," in *11th European LS-Dyna Conference*, Salzburg, 2018.

[55] National Highway Traffic Safety Administration (NHTSA), „Honda Accord FE-Model," NHTSA, https://www.nhtsa.gov/crash-simulation-vehicle-models.

[56] M. Kleinberger, E. Sun und R. Eppinger, „Development of improved injury criteria for the assessment of advanced automotive restraint systems," in *NHTSA Docket 4405.9*, 1998.

[57] E. Herzt, „A note on the head injury criterion (HIC) as a predictor of the risk of skull fracture," in *Proceedings: Association for the Advancement of automotive medicine annual conference*, Des Plaines, 1993.

[58] E. Takhounts, M. Craig, K. Moorhouse, J. McFadden und V. Hasija, „Development of brain injury criteria (BrIC)," in *Stapp car crash journal 57*, Orlando, 2013.

[59] E. Takhounts, R. Eppinger, Q. Campbell, R. Tannoues, E. Power und L. Shook, „On the Development of the SIMon Finite Element Head Model," in *Stapp Car Crash Journal Vol. 47*, San Diego, 2003.

[60] Verkehrsunfallforschung an der TU Dresden GmbH, „GIDAS - German In-Depth Accident Study," 2022. [Online]. Available: http://gidas.org/.

[61] D. H. Wolpert und W. G. Macready, „No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation ,* Bd. 1, Nr. 1, pp. 67 - 82, 1997.

[62] J. Garcke und R. Iza-Teran, „Machine Learning Approaches for Data from Car Crashes and Numerical Crash Simulation," in *NAFEMS World Congress*, Stockholm, 2017.