CarHoods10k: An Industry-grade Data Set for Representation Learning and Design Optimization in Engineering Applications

Patricia Wollstadt, Mariusz Bujny, Satchit Ramnath, Jami Shah, Duane Detwiler, Stefan Menzel

2022

Preprint:

This is an accepted article published in IEEE Transactions on Evolutionary Computation Special Issue on Benchmarking Sampling-Based Optimization Heuristics: Methodology and Software (BENCH). The final authenticated version is available online at: https://doi.org/10.1109/TEVC.2022.3147013

CarHoods10k: An Industry-grade Data Set for Representation Learning and Design Optimization in Engineering Applications

Patricia Wollstadt, Mariusz Bujny, Satchit Ramnath, Jami J. Shah, Duane Detwiler, and Stefan Menzel

***Abstract*—Large-scale, high-quality data sets are central to the development of advanced machine learning techniques that increase the effectiveness of existing optimization methods or even inspire novel ones. Especially in the engineering domain such high quality data sets are rare due to confidentiality concerns and generation costs, be it computational or manual efforts. We therefore introduce the *OSU-Honda Automobile Hood Dataset* (CarHoods10k), an industry-grade 3D vehicle hood data set of over 10000 shapes along with mechanical performance data that were validated against real-world hood designs by industry experts. CarHoods10k offers researchers and practitioners the unique opportunity to develop novel methods on realistic data with relevance to real-world vehicle design. To illustrating central use cases, we first apply methods from geometric deep learning to learn a compact latent representation for design space exploration. Second, we use machine learning models to predict mechanical hood performance from the learned latent representation. We thus demonstrate the effectiveness of machine learning for building metamodels, which are used in design optimization whenever possible to replace costly engineering simulations. Third, we integrate CarHoods10k in a topology optimization approach based on evolutionary algorithms to demonstrate its capability to search for high-performing structures, while maintaining manufacturability constraints.**

***Index Terms*—geometric deep learning, benchmark data set, topology optimization, surrogate models, engineering applications**

I. Introduction

When developing novel optimization and machine learning techniques with relevance to today's industry applications, benchmark systems and benchmark data sets with a suitable level of abstraction are key requirements for conducting meaningful experiments. In particular, machine learning and other data-driven approaches, nowadays, heavily contribute to the effectiveness of industrial evolutionary design optimization. These approaches provide, for example, improved problem parametrizations or design response prediction models, while relying heavily on the availability of realistic and large-scale benchmark data for methods development and performance assessment [1]. These benchmark data should be designed to closely resemble the data typically encountered in the targeted engineering application and concern geometric properties, such as suitable 2D or 3D shape representations and core features, as well as performance data, such as responses from aerodynamic or mechanical simulations of considered shapes. Of course, for efficient use in methods development, these data sets have to be well-curated, consistent, validated against real-world data, and sufficiently large to allow for the evaluation of computationally and numerically demanding methods. Creating such realistic benchmark data sets is central to advancing methods for vehicle design optimization.

Despite the need for industry-grade benchmarks, respective data sets are typically not readily available [2]. This is mainly due to confidentiality matters, but also to computational costs and high manual effort in their generation, which is for example required in Computer-Aided Design (CAD) processes [3], [4], or the unification and scaling of geometric representations. Especially the generation of corresponding performance data is costly in terms of computational demand and manual input, such that performance values are usually lacking.

In the present paper, we therefore introduce the *OSU-Honda Automobile Hood Dataset* (CarHoods10k), a 3D vehicle hood data set of over 10 000 shapes provided as triangulated meshes that were generated through an automated industry CAD process. Unlike existing 3D data sets, CarHoods10k provides industry-grade geometries together with performance data from structural mechanics and CAD parametrizations used to generate the data, serving as a ground truth for validation of methods. Geometries and performance values were validated by industry experts and by comparison to performance of real-world designs, such as to ensure validity, realism, manufacturability, and a sufficient variability of designs. Furthermore, geometries have genus one or higher, yielding a more realistic benchmark than existing genus zero data sets. Lastly, a major difference between the data set introduced here and existing benchmarks is that CarHoods10k provides a large set of variations of the same part. Typically, existing data sets comprise different objects to enable the development of classification or segmentation algorithms (e.g., [5]), tasks that are not central in engineering applications. CarHoods10k, on the other hand, consists of variations of the same part, where even minor geometric variations have large effects on performance. Our data set thus allows to approach challenges like building design exploration systems or metamodels for performance prediction from geometric representations (see, e.g., [6]). The data set is made public without any restrictions and is available from [7].

P. Wollstadt (*the corresponding author*), M. Bujny, and S. Menzel are with the Honda Research Institute Europe, Offenbach/Main, Germany. (e-mail: {patricia.wollstadt, mariusz.bujny, stefan.menzel}@honda-ri.de)

S. Ramnath is with the Simulation Innovation & Modeling Center, The Ohio State University Columbus, Ohio, USA. (e-mail: ramnath.17@osu.edu)

J.J. Shah is with the Digital Design & Manufacturing Lab, The Ohio State University Columbus, Ohio, USA. (e-mail: shah.493@osu.edu)

D. Detweiler is with Honda Development & Manufacturing of America, Raymond, Ohio, USA. (e-mail: ddetwiler@na.honda.com)

Manuscript received August 31, 2021; revised December 07, 2021

We believe that the listed properties of our data set offer a unique opportunity to stimulate a wide range of practically relevant research in the design optimization and machine learning domains. For illustration purposes, we present here results on the performance of methods from three central engineering design tasks: First, we apply methods from geometric deep learning (GDL) to learn a compact latent representation of all hood designs in an unsupervised fashion to perform an exploration of the car hood design space. Second, we build a surrogate model for performance prediction of new hood designs, which is of high practical relevance as an accurate metamodel significantly reduces the computational costs of performance simulations in a real-world design optimization. The model uses as input the latent representations learned in the previous task, demonstrating that the applied GDL methods successfully learned a meaningful representation in an unsupervised fashion. We further combine the metamodel with the generative capabilities of an autoencoder to automatically generate novel designs with desirable performance properties. Third, we integrate the hood data set in a topology optimization (TO) using evolutionary algorithms for searching optimal hood design structures with high similarity to designs in the data set. By this unique and novel combination, we optimize for structures which comprise a high performance, while fulfilling manufacturability constraints. Therefore, the proposed method offers to substantially improve conceptual design phases in industry, when TO methods are typically used, and currently suffer from limited practical feasibility.

The remainder of the paper is organized as follows. In Section II, we review existing geometric data sets relevant to engineering applications, the current state-of-the-art in GDL for unsupervised learning of latent representations, and structural TO methods. Section III provides details on the generation of the CarHoods10k data set and introduces the highly non-trivial process of generating industry-grade benchmark data in an automated fashion. We further describe methods from machine learning, GDL, and TO used in the experiments described in the last paragraph. In Section IV, we report experimental results, which illustrate the usefulness of CarHoods10k in engineering design tasks. Section V concludes the paper and provides an outlook on future work.

II. State of the Art

A. 3D Model Data Sets for Engineering Applications

Rich 3D geometry data sets are a valuable asset when researching and developing novel methods for improving optimization frameworks. These data sets represent historic design information which is assumed to be gathered over the course of product development in industrial scenarios. However, for practical reasons the acquisition of unified real-world data is challenging since over the course of time, shape parametrizations are likely to change substantially, e.g., by the introduction or removal of parameters required to achieve a required fidelity or reflect innovative product changes imposed by design studios, the addition and modification of processes to calculate the design responses for certain qualities or simply because the cost to generate designs is too high due to the involved manual processes and limited automation options for varying shape parameters in CAD tools.

As a consequence, a variety of 3D shape benchmark data sets has been proposed as meaningful real-world abstractions by targeting to unify representations among all design data such as boundary representation formats (B-reps), triangulated meshes, or point clouds. Data sets usually provide labelled instances, offer reasonable resolution that can be computationally processed, and are large enough to allow for the application of computationally demanding methods. Usually, each of these prominent data sets varies in scope to offer researchers options depending on their targeted application in mind. With respect to the field of machine learning, Willis et al. [3] provide a profound review on current 3D geometry data sets, which are usually relied on when developing methods, e.g., for GDL or surrogate modelling with different perspectives. Furthermore, the authors propose the *Fusion 360 Gallery*, a data set which contains geometry information of 8625 shapes and in addition related human design sequences for their generation. The authors proposed a framework of message passing networks for neurally guided search to synthesize designs and demonstrated its effectiveness in shape reconstruction tasks. *ShapeNet* [5] is a very large 3D model repository containing about 3 million shapes from which subsets have been annotated and categorized (bicycles, cars, chairs etc.). From an engineering perspective, especially both, the *ShapeNetCore* and *ShapeNetSem* [8] subsets provide value as the first subset includes about $51\,000$ clean unique 3D models which belong to 55 common categories, and the second subset includes about $12\,000$ models from 270 categories with additional features such as total volume and weight. Because of their large model numbers and quality, these data sets have been used for a variety of machine learning tasks with potential relevance to the engineering domain, e.g., hierarchical shape segmentation [9], object retrieval [10], or learning compact representations with GDL [11], [12]. The *ABC* data set [4] comprises over 1 million models and its usability has been demonstrated in a face normal estimation benchmark using a variety of state-of-the-art (deep) neural networks. Regenwetter et al. propose the *BIKED* data set which contains parametrizations of 4500 bicycles as CAD parameters which enabled them to reconstruct designs and generate images of each bicycle [13], furthermore the authors trained neural networks for bicycle type classification from parameters and variational autoencoders for shape synthesis.

From an engineering point of view, the base representation of the geometries in the data sets is important as it defines the resolution and, e.g., the input dimension for the neural networks in the context of classification or prediction tasks. *Fusion 360 Gallery* and *ABC* provide the models in B-rep format which is a standard in CAD modelling. As advantage, B-reps allow the customized sampling of the design surfaces to match the resolution desired for benchmarking a proposed machine learning algorithm. Both data sets also provide triangulated mesh data as available in *ShapeNet*. The availability of explicit parameters is an additional benefit as it allows ground truth evaluations. However, triangulated meshes are usually easier to handle, e.g., they can be smoothly transferred to point

clouds, and allow refinement but limit the resolution.

Our CarHoods10k data set includes three main characteristics. First, it provides for each hood explicit parameters which have been used in our industrial tool chain to generate the resulting hood design and can be used as ground truth data. Second, we processed the hoods to triangulated meshes to allow the calculation of further geometric features and for a smooth integration of the geometries into machine learning algorithms, since they can be easily transferred to point clouds without the need for external CAD tools. Third, we provide for each hood a quantitative performance value that results from a structural mechanics simulation which allows the development of suitable machine learning models for prediction tasks. Such models are highly desirable in practical evolutionary design optimization because they reduce the number of costly function evaluations on the real problem.

B. Geometric Deep Learning

Geometric deep learning (GDL) concerns the application of deep learning methods to geometric data in 3D space or, more generally, to non-Euclidean data [2], [14]. Existing deep learning approaches are often not directly applicable in the non-Euclidean domain because their formulation is bound to the Euclidean space. Examples are 2D convolution operations that exploit neighborhood-relationships in Euclidean space and thus require a reformulation for application in non-Euclidean domains. To still make use of the unprecedented power that deep neural networks offer, a series of network models and approaches for geometric data have been proposed, which successfully solve tasks such as segmentation [15], classification [16], and others [2]. Importantly, there is no canonical representation of geometric input data in GDL. The most popular modalities are voxels, polygon meshes, and point clouds [2], which each come with advantages and disadvantages for different applications [17]. For engineering applications, point clouds are a popular choice compared to other representations because of their lower memory and computational demand, particularly in comparison to voxels. Additionally, algorithms for polygon meshes typically make assumptions about the input data that can not be met in many engineering applications. For example, when translating convolutions from the 2D- to the 3D domain it is often required that all geometries have an identical topology (e.g., [18], [19]), due to the application of spectral mesh decomposition [20]. However, many engineering tasks deal with part geometries that differ in their topologies. A series of successful deep neural network architectures have been proposed for point clouds [11], [21], [22].

Therefore, point clouds are a promising geometric representation for processing by machine learning and GDL algorithms, in particular for engineering applications. However, the majority of existing GDL architectures are designed to handle problems that are typically not present in engineering tasks and data. Examples are partial occlusion or scans, and noise, which often occur in point clouds obtained from 3D scanning in robotics or autonomous driving. More recent work has therefore explored the applicability of GDL for point clouds specifically in engineering applications [12], [23], [24].

A prominent application example of GDL for engineering is learning low-dimensional representations of geometries in an unsupervised fashion. Finding suitable representations of 3D engineering design objects, such as to enable the application of computational design and data analytic tools, is a central task in engineering design (e.g., [6]). In particular, it is desirable to find numerical representations that preserve information about geometric properties, while having a lower dimensionality than the original geometric representation, such as to make the application of computational methods feasible (e.g., in evolutionary optimization of parts [12], [24]). GDL offers methods for the unsupervised learning of compact representations for high-dimensional (geometric) data. Here, representation learning aims at finding a low-dimensional encoding of the input data, which captures factors underlying the variability observed in the data and may be used in a downstream task where the application of methods on the original geometric data is not possible or computationally not feasible [25]. A popular deep learning approach for representation learning are *autoencoders*, i.e., models that encode high-dimensional input data into a lower-dimensional vector, from which the input can be reconstructed with minimal loss.

Autoencoders have been proposed also for point clouds by Achlioptas et al. [11], where a particular challenge was to make the architecture invariant to permutations in the ordering of the input [21]. Building on the work by Achlioptas, Rios et al. adapted the point cloud autoencoder (PC-AE) for application in engineering contexts [23]. The authors showed that the learned compact representations could be successfully used in engineering tasks, such as optimization or building metamodels, [12], [23], [24]. Here, due to the compact representation of the geometry, global optimization algorithms require less function evaluations and unsupervised representation learning potentially leads to "problem-agnostic" representations that "capture more relevant design features than a human designer would suggest" [12]. However, despite these promising results, existing work suffers from a lack of data sets that allow for an evaluation on inputs more representative of data typically encountered in engineering applications.

C. Structural Topology Optimization

Due to the rising complexity of contemporary product development processes, numerical optimization techniques start to play a vital role across different industries. In particular, topology optimization (TO) [26] is helpful in the initial design phases, by providing engineers with inspiration for novel design concepts. TO freely redistributes material within a predefined design space to optimize certain objective function, e.g., structural stiffness, based exclusively on the definition of the loads and supports in the numerical simulation model.

Typically, structural TO approaches use efficient gradient-based optimization algorithms, which utilize rigorously derived analytical sensitivities [26]. Depending on the type of the design representation, gradient-based TO approaches are frequently categorized into density-based [26] and level-set methods (LSMs) [27]. In density-based methods, mechanical properties of each element in the finite element simulation

model, e.g., Young's modulus, are explicitly controlled by design variables. Such a representation results in very high-dimensional optimization problems with thousands, up to millions of design variables, which can be efficiently solved for certain objective functions, e.g., stiffness of linearly elastic structures. The main drawback of the density-based methods, associated with the continuous parametrization of elements, is presence of regions with material properties not feasible for manufacturing. This problem is eliminated in LSMs by parametrizing the design through so-called level-set function (LSF), whose positive values indicate areas occupied by material and negative values correspond to void. As a result, only two levels of material properties are allowed and the boundary between material and void is clearly defined by the 0^{th} iso-contour of the LSF, which, in turn, can be described with use of some local basis functions [27]. Based on this concept, in the recent years, parametrized geometric shapes have been extensively used as a low-dimensional representation for TO, establishing a novel field of feature-mapping techniques [28], which allow for a smooth integration of TO in CAD systems and typically generate structures much easier to interpret and manufacture than standard LSMs or density-based techniques.

Despite the popularity and high computational efficiency of the gradient-based TO methods, in many cases they cannot be used due to unavailability of the sensitivity information of the considered objective functions and constraints. In particular, the complexity of the simulation models used in vehicle crashworthiness [29]–[31], manufacturing [32], or soft robotics [33], frequently prohibits analytical derivation or even reliable numerical estimation of sensitivities. As a result, often heuristic methods [34], [35], which redistribute the material to homogenize certain physical quantity, e.g., strain energy density or equivalent stress, are used to address different optimization problems. Nevertheless, heuristic approaches are not able to address specific objectives and constraints directly, which significantly limits their applicability. To address this problem, a field of black-box topology optimization (BBTO) emerged [36], offering a possibility to optimize any quantifiable objective functions with use of standard non-gradient optimization techniques such as evolutionary algorithms (EAs) or Bayesian optimization (BO). Here, also alternative, nature-inspired meta-heuristics proposed in the recent years (e.g., [37], [38]) could be potentially used, as demonstrated in [39]. Such methods are usually developed to efficiently solve specific types of problems. Hence, providing benchmarks as the one introduced in this paper is a necessary step for bridging the gap between the optimization and the engineering community by enabling the evaluation of performance of different optimization algorithms on industry-grade data sets and their robustness across a range of real-world applications.

In TO of hood frames, which we target in this paper, both static and crash load cases play an important role [40]. Therefore, a direct use of gradient-based techniques for such a multidisciplinary TO is not possible due to the noisiness and discontinuous character of the optimization criteria. In contrast, a BBTO technique called evolutionary level set method (EA-LSM) [30], [41] has demonstrated the ability to solve highly non-linear crash problems with use of EAs. To efficiently use non-gradient optimizers, EA-LSM employs a low-dimensional level-set representation based on parametrized geometric shapes called moving morphable components (MMCs) [42]. The generality of EA-LSM and its extensions [43] as well as other non-gradient methods relying on MMC level-set representation, e.g., kriging-guided level set method (KG-LSM) which uses BO to solve lower-dimensional TO problems more efficiently [44], [45], allows for addressing virtually any problem in structural mechanics [41], [46]. Furthermore, integration of additional criteria, e.g., manufacturing constraints [47], is straightforward in these approaches and can be easily realized by modifying the cost function minimized by the EA or by using additional surrogate models in constrained BO.

One of the criteria especially important from the perspective of industrial applications of TO is the similarity of the topology being optimized to a certain reference structure [48], [49], which can be, e.g., a prior good design or a part of the design space preferred by the engineer. As a result, manifold criteria related to manufacturability, limitations of the assembly process, economical restrictions, or even aesthetics, can be implicitly incorporated into TO [49]. Very often designs with a certain degree of similarity to the existing structures are preferred, to maximize the commonality of products within a certain group of models, as it is often the case in the automotive industry. Moreover, additional similarity criteria can help to improve the optimization process itself, by reducing the computational effort in case of addition of new load cases. Finally, similarity-based TO allows for an easier re-integration of the optimized components into a bigger system, where frequently specific structural connections are required, and might be eliminated in conventional TO.

In this article, we extend the concept of similarity-based TO by considering the similarity to an entire data set of reference structures within the EA-LSM framework. As a result, the optimization algorithm can freely choose a manufacturable reference design, interpolate between different solutions, and apply topological modifications to improve the structural performance. Hence, in a broader sense, the proposed method can be seen as a generic approach for integrating engineering design knowledge, implicitly encapsulated in a data set of parts. Please note that the manufacturability and engineering quality of the reference parts plays a critical role in such a process, and cannot be guaranteed for most of the currently available data sets.

III. Methods

A. Data Set Generation

Generating a large amount of CAD models of complex, industry-grade engineering parts, is challenging. We implemented an automated data generation pipeline for automotive hoods in CATIA v5, which results in a large set of variants of hoods that are geometrically valid, manufacturable, exhibit sufficient variability, and have functional properties comparable to real-world designs. In [1], [50]–[52], we provide a detailed description of the CAD data generation, as well as validation and post-processing of generated 3D models and corresponding performance metrics. In the following, we

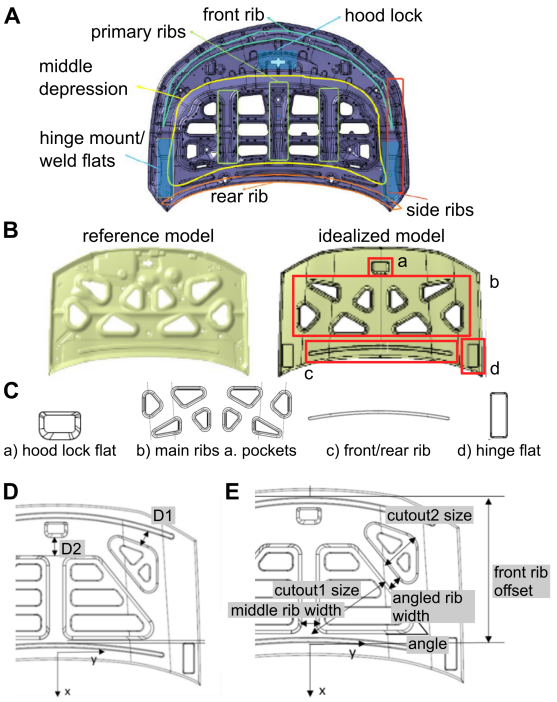


Fig. 1. A) Standard features on hood frame (from [1]). B) Example of reference hood frame models (left) versus corresponding idealized model (right). C) Examples of features on hood frame. Corresponding locations are shown on the idealized hood in panel A. D) Shortest distance measures between features. E) Sample parameters on the hood.

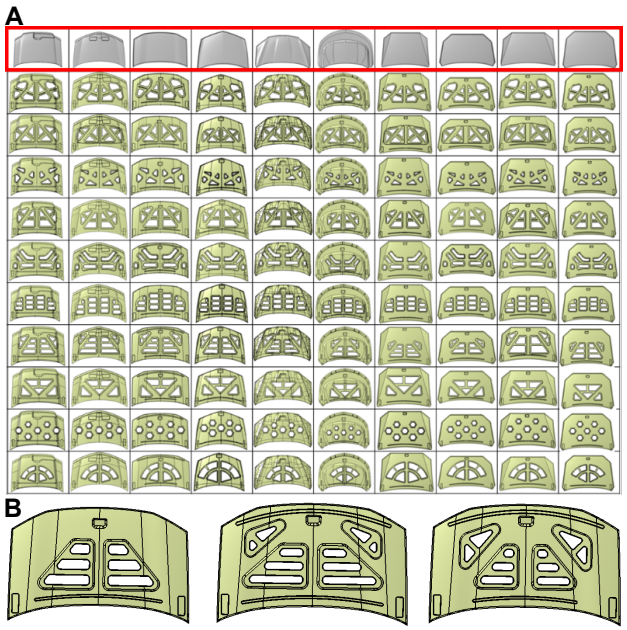


Fig. 2. A) Results from mix and match approach. B) Example of variability introduced to hood designs via placement of features.

provide a condensed summary of the primary tasks. The data set is available from [7] together with a detailed description of data types, data generation, and the organization of the repository (see Supplementary Section S1 for an example file).

1) CAD Data Generation and Validation: The first step when generating industry-grade models of engineering components is to identify relevant and irrelevant features in the corresponding CAD model. The component of interest is simplified in a process called "idealization" where irrelevant features are removed and only relevant features and their parameters are retained. Features are considered irrelevant if their removal does not compromise the overall performance of the part in critical load cases. Here, over- and under-idealization has to be balanced, where the former may lead to unrealistic geometries, whereas the latter may make it difficult to generate large number of geometry variants automatically. Fig. 1A shows a schematic of an original hood with typical features. Fig. 1B shows an example of an idealized hood and its corresponding original geometry.

To automate the process of acquiring a large amount of CAD data sets, we chose a feature-based design approach promising fast and robust design generation. Automotive hood frames must meet several structural requirements such as, maximum hood lift or twist deflections, and the ability to withstand pedestrian and frontal impact. The structure of the hood frame is made from sheet metal stamped components, while desirable properties of the component are achieved by adding features. For example, to add stiffness ribs and channels are created, while light weighting is achieved by creating cutouts or pockets. The specifications of the hood skin (like size, aspect ratio, etc.) constrains the design of the frame structure and the placement of features. Examples of features are shown in 1C. For generating large sets of geometries, features or set of features (pattern) are parametrized and parameter values are chosen such that feature do not fail during automatic generation (see below).

To generate a sufficiently large set of geometries, the process of idealizing a hood geometry, identifying relevant features and finding valid parameter ranges that led to as few invalid geometries as possible, was repeated for 100 base geometries. Base geometries were generated by combining each of 10 different skins with each of 11 existing pocket feature patters (Fig. 2A). Certain features like the hood locks and hinge flats are standard features on every design. Variations in features like front or rear ribs were limited since changing parameters on the pocket features resulted in sufficient variety.

Parameter values describing the placement and characteristics of feature patterns were generated using a sampling scheme and stored in a *design table* [1]. Examples of design variations for the same base surface and features with different parameter values are shown in Fig. 2B: In the first design, compound features at the front corners are toggled off while in the other two designs the sizes of features are varied.

This "mix-and-match" approach is designed to generate a large amount of geometries that show sufficient variability. Our workflow starts with the base component, which is an idealized hood skin, on which the various features are created and assembled. Construction entities such as offset planes, wire frame geometry, and coordinate systems, which are specific to the CAD process, are created from the base geometry. In the next step, features are instantiated using a macro, which reads parameter values from the design table and executes the modeling steps necessary to generate the features.

The geometries created using this method have to be validated to ensure proper generation. Geometrical validity of the generated model is determined based on the quality of the associated geometrical entities. There are two failure modes for geometrical entities in the process of automated CAD model generation: In the first mode, models do not generate due to errors in the CAD operations. When changing the location, shape, and size of features, based on each row of the design table, certain combinations of parameter values may lead to non-manifold geometrical entities. In the second mode, models are generated with invalid geometrical entities. The most common reasons for this type of geometrical invalidity are self-intersections, feature overlap, or inconsistent features. In this failure mode, the geometry is created in CAD and model generation does not fail as all the geometrical entities are still valid, but the result is an non-manufacturable design.

In order to avoid a high number of invalid geometries, a comprehensive analysis of the feature parameters and their limits is required. A common cause of failures of failure mode two, e.g., inconsistent or overlapping features, is that parameter values for feature positioning surpasses the allowable, i.e., manufacturable, limits. This can be avoided by setting up and solving a constraint network for feature parameters to arrive at non-conflicting parameter settings that lead to sufficient distances between outer boundaries of features. All parameter values that result in positive distances will result in CAD models that fulfill geometrical validity.

The constraints in the constraint network are solved sequentially based on the driving parameters, i.e., values in the design table, opposed to driven parameters that are automatically computed based on the driving parameters. To re-use existing constraint equations for various base geometries, parameter values were set to zero in the absence of a feature. Features and parameter values used are collected in design tables. Fig. 1D shows two distances on an idealized hood frame. These measures are the shortest distances between features on the hood. Examples of parameters describing features and feature patterns on the hood are shown in Fig. 1E. Applying the described approach of generating and solving a constraint network did not avoid invalid geometries completely, but reduced the ratio of invalid geometries from $33\,\%$ to $4\,\%$.

The total number of CAD models created with the presented library of features are $M = A \times B \times C$, where, M is the total number of designs, A the number of pocket feature patterns (11), B the number of base surfaces (10), and C the number of design variants for each surface (100). Therefore, the total number of generated geometric designs was $M = 11000$. Of these, $10\,478$ were successfully generated while 522 designs failed during the CAD process.

2) Finite Element Model Standardization and Setup: To obtain performance values for generated geometries, we used finite element analysis (FEA). The analysis was performed using Ansys Workbench R19.1. FEA was performed for hood lift and twist cases. Overall stiffness for hood lift and hood twist load cases during driving conditions is an important structural requirement when designing car hood frames. We performed load case correlation and parametric sensitivity studies prior to generating the performance data [50], [51].

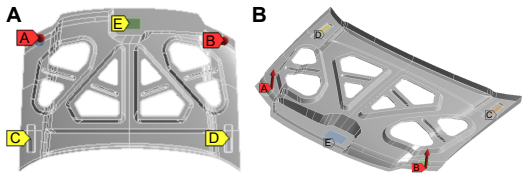


Fig. 3. Boundary conditions for hood frame finite element analysis. Red markers indicate a force of $150\,\mathrm{N}$, yellow markers indicate remote displacements.

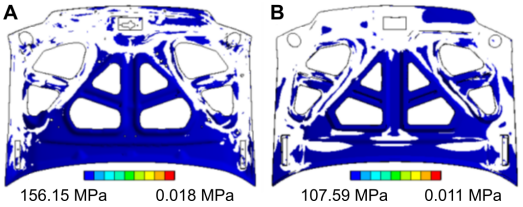


Fig. 4. Stress path comparison for reference (A) vs idealized (B) hood model (equivalent (von-mises) stress [MPa]).

We found high correlation between the performance values for both load cases and hence only report results from the hood lift load case. The obtained performance values are *maximum equivalent (van-mises) stress* [MPa], *maximum directional (z-axis) deformation* [mm], and *geometry mass* [kg].

To make the process of setting up several thousand FEA models feasible and to standardize models over geometries, we used a uniform setup for boundary conditions (loads and supports, Fig. 3A). Boundary conditions remained consistent between various FEA models with respect to fixture locations, and supports were applied on a remote point that was offset from the frame surface (Fig. 3B).

Over-idealization of the hood frames might lead to unrealistic results, while under-idealization may make it difficult to generate a large number of geometry variants automatically. Therefore, we validated designs by comparing contour patterns (along with locations of maximum and minimum values) of deflections and stresses, resulting from FEA in idealized geometries to results from real-world reference geometries. The contour patterns in the idealized hoods matched up very well with the reference hoods as shown in Fig 4. We also compared the magnitudes of the maximum von-mises stress and maximum deflection, and found that the differences were within acceptable range, given the differences in the geometries, as verified by experienced automotive designers at Honda Development and Manufacturing Americas.

As introduced above, a challenge in generating a large amount of data sets is to ensure meaningful variability in the generated designs. We measure variability in the data set not only in terms of geometric properties but also in terms of behavior in FEA. We thus extended FEA to understand the effect of the common features (pocket features in particular) on different base surfaces. We found that pocket feature patterns dominated the performance of the hood in comparison to other features. Further, varying the pocket feature patterns introduced sufficient variation in the performance of the data set. A subset of FEA simulation results in the "mix and match" models are summarized in Table I.

TABLE I
MAXIMUM DEFLECTION FOR 16 COMBINATIONS OF POCKET PATTERNS AND SKINS (m: MEAN, SD: STANDARD DEVIATION).

	base geometry					
pocket	A	B	C	D	m	SD
A	6.51	10.60	6.68	5.09	6.81	3.24
B	4.93	7.01	9.23	5.51	6.63	2.99
C	5.34	10.08	8.97	5.63	7.15	3.25
D	4.92	10.70	8.79	5.54	7.11	3.48
m	5.26	9.90	8.18	5.42		
SD	0.64	1.45	0.99	0.22		
Range: 1.53 to 10.87, grand m (SD) = 6.47 (2.91)						

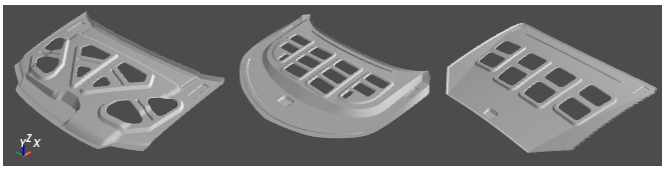


Fig. 5. Examples of STL surface meshes of hood geometries.

In terms of computational run time, the time taken to generate 100 designs—from CAD to setting up FEA models—was 30 min on a computer with Intel Xeon 2x3 GHz processor and 16 GB RAM. With the use of high performance computing the same method may be scaled up to generate over a hundred thousand models. Running FEA models and processing the results for each of 100 designs takes around 30 hrs, but the simulations may also be trivially parallelized. Note that some FEA models failed to run; hence, performance values are not available for all the geometries in the data set. We obtained results for 10 070 of the 10 478 generated CAD models. From the CAD files, we generate STL surface meshes via a custom script. All files are watertight and contain properly oriented surface normals (see Fig. 5 for examples of generated STLs).

B. Learning of Geometric Shape Representations for Metamodeling and Design Optimization

As one approach to representation learning for design optimization and other engineering tasks, we used a deep autoencoder network for point cloud representations of 3D shape data (PC-AE)[1] that was introduced by Rios et al. [23] and is based on a model originally proposed in [11]. The PC-AE accepts a point cloud as input, that is an unordered set of points in 3D space, $\mathbf{P} = \{\mathbf{p}_i\}, i \in \{1, \ldots, N\}$, where each point is a vector of coordinates, $\mathbf{p}_i = (x_i, y_i, z_i)$. To transform the STL data set into a point cloud, we randomly and uniformly sampled the STL mesh surfaces using two resolutions of 2048 and 8192 points. Note that alternative sampling schemes exist (e.g., [53]) that allow the targeted sampling of high- and low-frequency features [23]. Also, higher resolutions of point clouds (up to 200 000) have been evaluated for processing of geometric data using the PC-AE [23]. The model is implemented in Python using the TensorFlow 1.14 package [54].

We trained the autoencoder with four 1D convolutional encoder layers of sizes $[64, 128, 128, 256]$ followed by one max-pooling layer that generates the latent representation of size L. From the latent representation, three fully-connected decoder layers of sizes $[256, 256, N]$ reconstruct the input, where N denotes the size of the point cloud. All layers use rectified linear units (ReLU) as activation functions, except for the max-pooling and output layer, which uses hyperbolic and sigmoid functions, respectively. For training, we used the Adam optimizer [55] with learning rate of 5×10^{-4}, $\beta_1 = 0.9$ and $\beta_2 = 0.99$, and a batch size of 50 over 800 epochs. We used 90 % of the data for training the model and held out the remaining data as test set. The test set was randomly sampled from the full data set. Before training, we augmented the data by rotating individual geometries by random rotation around z-axis. We evaluated latent space sizes of $L = 10$ and $L = 128$.

We used the trained PC-AE to encode the whole data set (training and test data) and performed experiments to evaluate a) whether the learned representations carried relevant information about the input data set, and b) whether the learned representations were meaningful for down-stream engineering tasks. We answered both questions using machine learning. First, we trained a model to predict design parameters used in data generation from learned latent representations. Here, we evaluated whether the latent representations contained information about the variables underlying the data generation process, thus testing whether representation learning was able to capture factors underlying the variability observed in the input data. Second, we trained a model to predict FEA performance values from learned latent representations to obtain a metamodel for FEA simulations. We further used the metamodel to demonstrate the generative power of models such as deep autoencoders, by optimizing the input to the metamodel under the objective of minimizing the performance values. This optimized input was then reconstructed into a point cloud representation of a geometry via the decoder part of the PC-AE. A similar approach was proposed by [56] for generating discrete representations of molecules. In both experiments, we trained the model on the same training set as used in training of the PC-AE and applied the trained model to the unseen test set. We thus evaluate how the whole pipeline—learning of latent representation and downstream engineering task—perform on unseen test data. Machine learning was performed using Python3 and the scikit learn library [57].

C. Similarity-based Evolutionary Level Set Method

In this section, we present an extension of EA-LSM, accounting for a similarity to the data set of hood frames. Firstly, a short description of a 2D topology representation is given. Secondly, a novel formulation of the optimization problem, considering both the structural performance and similarity, is presented. Finally, the optimization algorithm is described.

1) Representation: We use a 2D MMCs level-set representation as described in [30], [41]. The distribution of the material in the design space is defined using a global LSF Φ:

$$\begin{cases} \Phi(\mathbf{x}) > 0, & \text{if } \mathbf{x} \in \Omega, \\ \Phi(\mathbf{x}) = 0, & \text{if } \mathbf{x} \in \partial\Omega, \\ \Phi(\mathbf{x}) < 0, & \text{if } \mathbf{x} \in D \backslash \Omega, \end{cases} \tag{1}$$

[1]Code available: https://github.com/HRI-EU/GDL4DesignApps.

where $\mathbf{x} = [x, y]^T$ is a position in the design space D, Ω is the area occupied by material, and $D \backslash \Omega$ corresponds to void. The boundary between material and void is denoted by $\partial\Omega$.

The global LSF is defined using m MMCs:

$$\Phi(\mathbf{x}) = \max\left(\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ..., \phi_m(\mathbf{x})\right), \tag{2}$$

where ϕ_i is given by the following formula:

$$\begin{aligned}\phi_i(\mathbf{x}) = -\Bigg(&\left(\frac{\cos\theta_i\left(x - x_{0i}\right) + \sin\theta_i\left(y - y_{0i}\right)}{l_i/2}\right)^q \\ &+ \left(\frac{-\sin\theta_i\left(x - x_{0i}\right) + \cos\theta_i\left(y - y_{0i}\right)}{t_i/2}\right)^q - 1\Bigg),\end{aligned} \tag{3}$$

with (x_{0i}, y_{0i}) being the position of the center of the i^{th} MMC rotated counterclockwise by angle θ_i w.r.t. the x-axis. The length of the MMC is denoted by l_i and thickness by t_i. As in the other works [30], [41], [42], a modelling exponent $q = 6$ is used, resulting in MMC shapes close to rectangular.

Finally, the material density at location $\mathbf{x}$ is found by applying a Heaviside function to the global LSF:

$$\rho(\mathbf{x}) = \max\left(\rho_{min}, H\left(\Phi(\mathbf{x})\right)\right), \tag{4}$$

where $\rho_{min} = 0.01$ is the minimal allowed density to guarantee numerical stability in the finite element simulations.

The density (4) is used to scale the reference Young's modulus E^0 of the material according to the following expression:

$$E(\mathbf{x}) = \rho(\mathbf{x})\, E^0, \tag{5}$$

In the numerical simulation model, the Young's modulus of each finite element is modified based on (5) computed for $\mathbf{x} = \mathbf{x}_{C_e}$, where $\mathbf{x}_{C_e}$ denotes the center of the e^{th} element.

2) Optimization Problem: We consider a TO problem with two objective functions, i.e., structural compliance (c) and a dissimilarity metric of the topology to the closest design in a data set (s). We use a weighted sum approach[2], which yields a single-objective optimization problem of a following form:

$$\begin{aligned}&\min_{\mathbf{z}}\ \left(f_{obj}(\mathbf{z}) = wc(\mathbf{z}) + (1-w)s(\mathbf{z})\right),\ \mathbf{z} \in \mathbb{R}^n; \\ &\text{s.t.}\, V(\mathbf{z}) < V_{max}; \mathbf{KU} = \mathbf{F},\end{aligned} \tag{6}$$

where $f_{obj}(\mathbf{z})$ is an aggregated objective function, while $\mathbf{z}$ is the vector of design variables collecting $n = 5m$ parameters describing the m MMCs. The weight w defines the relative importance of the two terms in $f_{obj}(\mathbf{z})$. The volume of the structure is denoted by $V(\mathbf{z})$ and V_{max} is its maximal allowed level. The global displacement vector, stiffness matrix, and the force vector in the finite element model are denoted by $\mathbf{U}$, $\mathbf{K}$, and $\mathbf{F}$, respectively. The compliance of the structure can be computed based on the stiffness matrix $\mathbf{K}$ and the displacement vector $\mathbf{U}$ obtained from the finite element solver:

$$c(\mathbf{z}) = \mathbf{U}^T\mathbf{K}\mathbf{U} \tag{7}$$

[2]Please note that here, a posteriori methods, such as multi-objective evolutionary algorithms (MOEAs), e.g., NSGA-II [58], which are able to identify multiple Pareto-optimal solutions in a single optimization run, could be alternatively used. Here, we applied a single-objective version of EA-LSM since it has been rigorously validated by comparing with the state-of-the-art TO methods in extensive experimental studies [30], [41] and successfully applied to non-standard, real-world TO problems [59].

The dissimilarity metric of the current topology to the data set, analogically to the metrics used in the other works [48], [49], is defined as follows:

$$s(\mathbf{z}) = \min_j \frac{\sum_{e=1}^{n_{el}} \left(\rho_e - \rho_e^j\right)^2}{n_{el}},\ j \in \{1, 2, ..., N\}, \tag{8}$$

where ρ_e is the density of the e^{th} finite element in the design being optimized and $\rho_e^j \in [0, 1]$ is the density of the corresponding element in the reference design, obtained by projecting the j^{th} hood frame model on a 2D plane along the z-axis. The total number of finite elements is denoted by n_{el}.

3) Optimization Algorithm: For simplicity, we use a standard evolution strategy ES(μ,λ) [60] with μ parents, λ offspring individuals, and a single step size. A repair operator is used to delete MMCs of thickness lower than t_{th} and move the ends of the MMCs going outside the design domain to its boundary. For details please refer to [41]. The volume constraint is handled using the exterior penalty method [61], leading to the following cost function:

$$f(\mathbf{z}) = wc(\mathbf{z}) + (1-w)s(\mathbf{z}) + P\max(0, V(\mathbf{z}) - V_{req}), \tag{9}$$

where P is a large constant.

IV. Experiments

A. Unsupervised Representation Learning using Geometric Deep Learning

In a first experiment, we demonstrate the usability of CarHoods10k for unsupervised representation learning. We used the recently proposed PC-AE for geometric data represented as point clouds [23] to find low-dimensional representations of sizes $L = 10$ and $L = 128$. Fig. 6 shows examples of the reconstruction quality from latent representations.

To validate that the learned latent space represents latent variables underlying the data generation process, we used the learned representations to predict design parameters. We trained machine learning models (support vector machines, decision trees, linear model, ensemble methods, k-nearest neighbor models, logistic regression) on latent representations of the training set also used for training the PC-AE and used 5-fold cross-validation to identify the best-performing model and corresponding hyper-parameters [57] (see Supplementary Material Section S2 for all hyper-parameter settings explored and Tables SI and SII for selected models and hyper-parameter settings). The identified model was applied to the test set for final performance evaluation. We used a classifier to predict the baseline skin, and regression models to predict design parameter values *RibDepth* [a.u.] and *Pocket1_Radius* [a.u.], which were used in the majority of CAD models (Table II).

For classification, support vector machines with radial basis functions as kernels performed best for all combinations of L and N. For regression, linear models such as linear support vector machines and least angle regression (LARS) performed best in the majority of cases. For some of the low-dimensional representations, $L = 10$, support vector machines with non-linear kernels performed best.

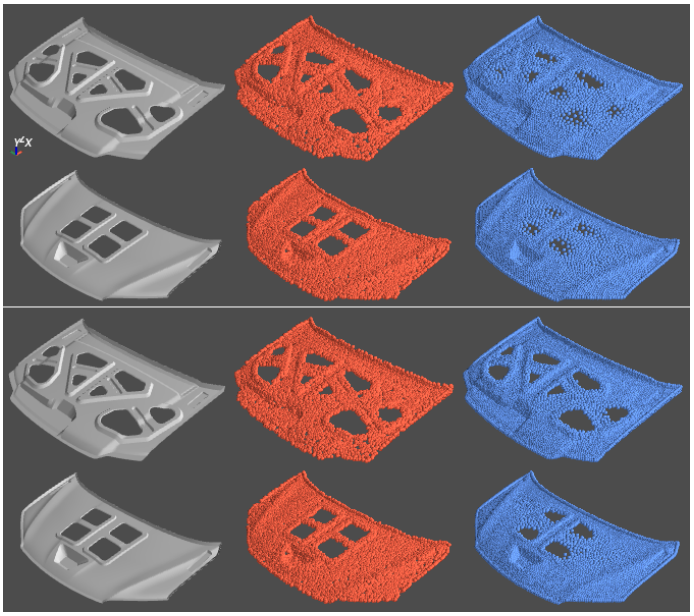


Fig. 6. Examples of reconstructed point clouds. Columns from left to right show the original mesh representation (grey), the input point clouds (red), and the reconstructed point clouds (blue). The two top rows show reconstructions for $N = 8192$, $L = 10$, the two bottom rows for $N = 8192$, $L = 128$.

TABLE II
RESULTS FOR PREDICTION OF DESIGN PARAMETERS ON TRAINING AND TEST SET.

N	L	Metric	Train	Test
Base skin				
2048	10	Accuracy	0.7697	0.6644
2048	128	Accuracy	1.0000	0.9837
8192	10	Accuracy	0.6988	0.4784
8192	128	Accuracy	1.0000	0.9962
RibDepth [a.u.]				
2048	10	MAE	4.0023	3.9322
		R^2	0.3896	0.3976
		MAPE	26.74%	26.66%
2048	128	MAE	3.4400	3.4974
		R^2	0.5677	0.5538
		MAPE	23.17%	24.15%
8192	10	MAE	4.3499	4.3313
		R^2	0.2564	0.2759
		MAPE	29.67%	29.67%
8192	128	MAE	4.7276	4.9490
		R^2	0.6428	0.6217
		MAPE	16.81%	17.53%
Pocket1_Radius [a.u.]				
2048	10	MAE	6.7184	6.6948
		R^2	0.2844	0.3252
		MAPE	23.68%	23.20%
2048	128	MAE	4.7179	4.9722
		R^2	0.6399	0.6393
		MAPE	16.72%	17.74%
8192	10	MAE	7.0569	7.2036
		R^2	0.0681	0.0479
		MAPE	22.64%	22.90%
8192	128	MAE	4.7276	4.9490
		R^2	0.6428	0.6217
		MAPE	16.81%	17.53%

All predictive regression models for design parameter values outperformed baseline models, where we predicted parameter values in the test set as the mean and median of the training set, respectively. The same was true for the classification model, where we used two baseline models that either generated predictions with uniform probability over classes or with a stratified probability, i.e., respecting the distribution of geometries across classes in the training set. In the regression task, the mean percentage error (MAPE) was relatively high (up to $30\,\%$) for all experiments. The classification performed better, particularly when predicting from latent representations of size $L = 128$, where accuracy was above $98\,\%$ for both point cloud sizes, $N = 2048$ and $N = 8192$.

In sum, we were able to learn low-dimensional latent representations of the data set, from which we were able to reconstruct design parameters underlying the data generation process to some degree. The visual reconstruction quality was sufficiently high. We conclude that the PC-AE successfully compressed the geometric information in the input data into a lower-dimensional latent representation.

B. Design Space Exploration and Generation of Novel Shapes Using Geometric Deep Learning

The space of latent representations learned by the PC-AE in the previous experiment may be used to explore the design space of the generated car hoods. Design space exploration is a central task in the engineering process, where large sets of candidate designs have to be generated and evaluated under different objectives. Autoencoders are a promising tool for exploring design spaces and even synthesizing novel shapes. To demonstrate this, we show that interpolations between latent representations lead to meaningful shapes when decoded using the PC-AE decoder. Fig. 7 shows reconstructions from such an interpolation, where the first geometry on the left (white) indicates the start of the interpolation and the last geometry on the right (dark blue) indicates the target geometry. Each geometry in between is a reconstruction from the respective step in the linear interpolation between the latent representation of the first and last geometry. Note that only start and target geometries are samples from the data set, while the interpolation generates "new" samples in the latent space that do not correspond to a geometry in the input. Hence, the autoencoder is able to generate meaningful new geometries from manipulation of latent representations.

The learned latent space may be used to easily explore the design space by manipulating values for all or individual dimensions of the latent representation, where individual dimensions may be related to certain properties of the data. Note, however, that the architecture used here places no constraints on latent representation learning and more specialized models are needed to reliably enforce a close relationship between individual learned variables and specific geometric properties of the data. Examples of such architectures are variational autoencoders or fader networks, which are specifically designed to learn interpretable latent representations where each variable codes for one feature of the input data set (e.g., [25]).

In sum, the proposed data set allowed for the evaluation of unsupervised representation learning approaches. Using a dedicated industry data set as done here allows to assess such approaches for the generation of realistic shapes, such that manufacturability or functional properties are ensured (see also the following experiments).

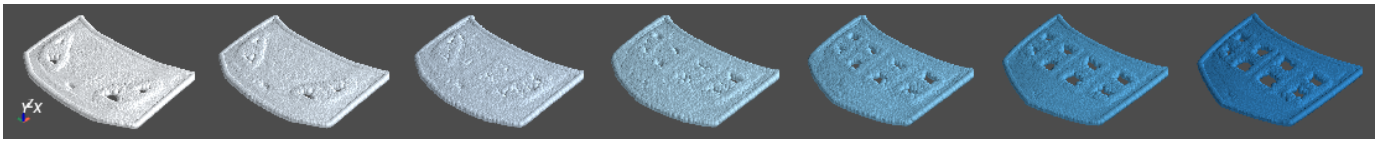


Fig. 7. Point clouds reconstructed from five interpolation steps between latent representations of two geometries in the data set ($N = 8192$, $L = 128$). Color coding indicates an increasing similarity to the start (white) and target (dark blue) of the interpolation, respectively (see also main text).

C. Metamodel for Performance Prediction from Learned Representations

In a second experiment, we used the CarHoods10k data set for leaning a meta model for performance prediction. We used machine learning to predict the *maximum equivalent stress* [MPa], the *maximum directional deformation* [mm], and the *geometry mass* [kg] from the latent representations obtained from the PC-AE trained in the previous experiment. Note that the latent space was learned without including any knowledge about the geometries' performance.

For model training, we again used the training data that was also used to train the PC-AE. The remaining $10\,\%$ of the data were used as test set. On the training set, we again trained a series of machine learning models (support vector machines, decision trees, linear model, ensemble methods, k-nearest neighbor models) and tuned their hyper-parameters using 5-fold cross validation. See Supplementary Material Section S2 for all hyper-parameter settings explored and Table SIII for selected models and hyper-parameter settings. The model performing best across cross-validation splits was chosen as the metamodel and applied to the unseen test set (see Table III). The experiment was performed for point cloud sizes of $N = 2048$ and $N = 8192$, and latent space dimension of $L = 10$ and $L = 128$. The best-performing models were support vector machines with either a polynomial or radial basis function kernel for all combinations of point cloud size and latent space dimension, except for predicting the directional deformation from $L = 10$, $N = 8192$, where the Extra-Trees ensemble method performed best.

All models significantly outperformed two evaluated baseline models that predicted the mean and the median of the training set, respectively. In general, the test error decreased for higher latent size $L = 128$, while models trained on all combinations of L and N achieved satisfying accuracy with a mean percentage error below $10\,\%$ and even below $4\,\%$ when predicting the geometry mass. The latter prediction problem is potentially easier due to a more direct relationship between geometry and mass, compared to the performance values obtained from FEA. Note that best-performing models were non-linear and thus outperformed evaluated linear variants, indicating a non-linear relationship between latent representations and performance values.

A metamodel trained on the latent space of an autoencoder offers the possibility to "guide" the generative capabilities of the autoencoder and thus generate shapes in a more informed fashion compared to simple interpolation. For example, Gómez-Bombarelli et al. [56] used a deep autoencoder to learn latent representations of chemical molecules. The authors then trained a metamodel on the latent space to predict a performance value and used gradient-based optimization in the space learned by the metamodel to find inputs that resulted in high performance values. These inputs could then be fed to the trained decoder to generate high-performing novel shapes. Previously, this approach could not be fully realized in the engineering context due to a lack in data sets comprising both engineering shapes and corresponding performance values. The CarHoods10k data set thus offers the possibility to developing approaches similar to the one used by [56] to generate functional geometries in engineering applications in a fully automated fashion.

To demonstrate the latter approach, we used the metamodel trained for performance prediction to guide the generative capabilities of the PC-AE that were demonstrated in the

TABLE III
RESULTS FOR PREDICTION OF PERFORMANCE VALUES ON TRAINING AND TEST SET.

N	L	Metric	Train	Test
Max. equivalent stress [MPa]				
2048	10	MAE	13.3570	13.0630
		R^2	0.7668	0.7899
		MAPE	7.96%	7.67%
2048	128	MAE	12.6403	11.8847
		R^2	0.7827	0.8236
		MAPE	7.51%	7.16%
8192	10	MAE	13.8635	15.3200
		R^2	0.7649	0.6866
		MAPE	8.34%	8.90%
8192	128	MAE	10.8227	12.7015
		R^2	0.8332	0.7514
		MAPE	6.48%	7.51%
Max. directional deformation [mm]				
2048	10	MAE	0.8176	0.8083
		R^2	0.8950	0.8999
		MAPE	8.25%	8.14%
2048	128	MAE	0.5365	0.5700
		R^2	0.9409	0.9417
		MAPE	5.56%	5.62%
8192	10	MAE	0.6844	0.7655
		R^2	0.9302	0.9113
		MAPE	6.53%	7.18%
8192	128	MAE	0.5342	0.6134
		R^2	0.9437	0.9270
		MAPE	5.58%	6.30%
Geometry mass [kg]				
2048	10	MAE	0.4770	0.4799
		R^2	0.8756	0.8764
		MAPE	3.51%	3.53%
2048	128	MAE	0.2241	0.2576
		R^2	0.9601	0.9486
		MAPE	1.63%	1.87%
8192	10	MAE	0.4874	0.5215
		R^2	0.8735	0.8508
		MAPE	3.60%	3.83%
8192	128	MAE	0.2710	0.2992
		R^2	0.9431	0.9202
		MAPE	1.97%	2.14%

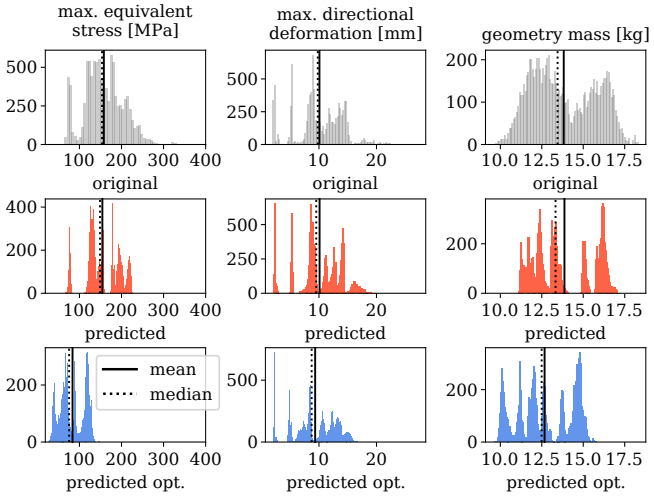


Fig. 8. Distribution of performance values: original (top row, gray), predicted from latent representations (middle row, red), and predicted from optimized latent representations (bottom row, blue). Vertical lines indicate mean (solid line) and the median (dashed line).

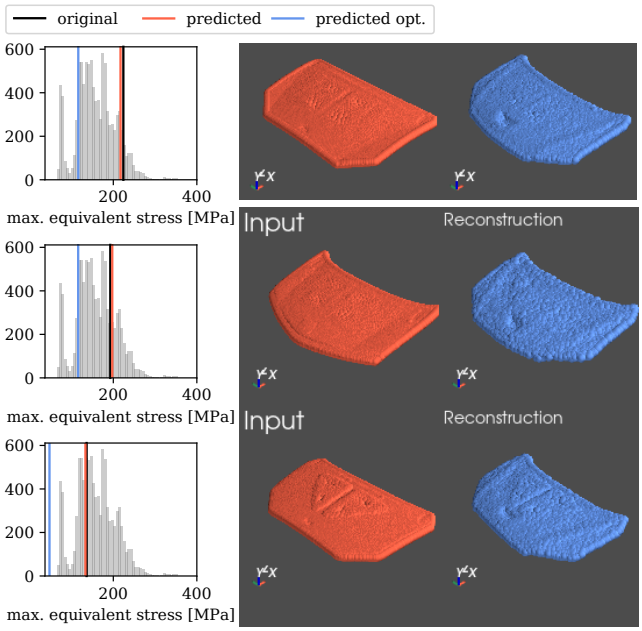


Fig. 9. Examples of optimized performance values with distribution of performance values over all geometries (left column) and corresponding geometries (middle and right column). Vertical lines indicate original and predicted values, and predicted value after optimization of the latent representation. Red point clouds show reconstructions from original and blue point clouds show reconstructions from optimized latent representations.

previous experiment. We optimized the input to the metamodel through differential evolution with the objective of minimizing the model's output. We used a population size of $NP = 15$, a crossover probability $CR = 0.7$ and a variable mutation constant or differential weight, F that was chosen uniformly from the interval $[0.1, 1)$ for each generation. The optimization terminated if a maximum number of iterations of 500 was reached, or if the standard deviation of the population fell below $10\,\%$ of the mean of the population energies.

We optimized each geometry in the data set by providing its latent representation to the optimizer with the objective of minimizing the metamodel's prediction of the *maximum equivalent stress* [MPa], the *maximum directional deformation* [mm], and the *geometry mass* [kg], respectively. Fig. 8 shows distributions of the original performance values, the performance values predicted from the geometries' latent representations, and the optimized latent representations for $N = 8192$ and $L = 10$. The optimization lead to a reduction in the predicted values for all three performance values with the highest effect on the *maximum equivalent stress*, which was on average reduced by $70.92\,\mathrm{MPa}$ ($45.90\,\%$) compared to the prediction from the original latent representations (with mean $154.50\,\mathrm{MPa}$ and SD $40.04\,\mathrm{MPa}$). The *maximum directional deformation* was on average reduced by $0.77\,\mathrm{mm}$ ($7.62\,\%$) compared to the prediction from the original latent representations (with mean $10.11\,\mathrm{mm}$ and SD $3.98\,\mathrm{mm}$). The *geometry mass* was on average reduced by $1.20\,\mathrm{kg}$ ($8.65\,\%$) compared to the prediction from the original latent representations (with mean $13.87\,\mathrm{kg}$ and SD $1.81\,\mathrm{kg}$).

The goal of the optimization was to guide the generative capabilities of the PC-AE. We evaluated through visual inspection whether the point clouds reconstructed from optimized latent representations led to meaningful hood frames. Fig. 9 shows examples of reconstructions from optimized latent representations, their corresponding performance values and original geometry represented as point cloud. Note in particular the changes in the outline and surface of the hoods. In sum, our experiments demonstrated great potential of the application of GDL and machine learning methods in engineering tasks, such as metamodeling and automated synthesis of part geometries with desirable performance values.

D. Similarity-Based Topology Optimization Using Evolutionary Level Set Method

In this section, a description of the test case and the setup of the similarity-based EA-LSM introduced in Section III-C is given. Subsequently, we present the results of the experimental evaluation and discuss the properties of the presented method.

1) Hood Lift Test Case: To allow for an easy analysis of the properties of the proposed method, a simplified, 2D finite element model of the hood frame is used (Fig. 10A).

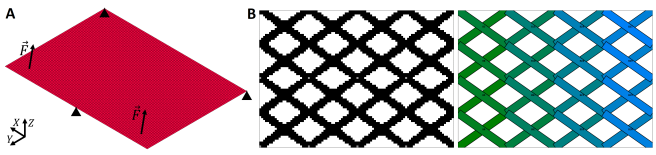


Fig. 10. A) Isometric view of the shell finite element model used in the TO with EA-LSM. Degrees of freedom of the three mesh nodes are fixed (triangles), unit static loads are applied in the corners of the design space as a simplified version of the hood lift load case (Fig. 3). B) Initial design used in TO with EA-LSM. Material distribution (left), corresponding layout of MMCs (right).

The simulation model uses a regular mesh of 100x70 square, four-node shell finite elements of unit size. A linear elastic material model with Young's modulus of $E^0 = 2.1 \cdot 10^5$ [MPa] and a Poisson's ratio of $\nu = 0.3$ is used. Due to the linearity of the model and utilization of volume as a constraint, the

exact scale of the model and magnitudes of the loads are not important for TO. The simulations are carried out with use of an open source finite element solver, CalculiX 2.9[3].

2) Setup of EA-LSM: Following the original method [30], [41], an initial (reference) design composed of 32 MMCs and shown in Fig. 10B was used in all of the optimization runs.

The initial population of individuals is created based on the reference design by varying the parameters of MMCs according to the normal distribution [41]. In order to reduce the number of design variables, a symmetry w.r.t. vertical axis going through the center of the design domain is enforced, resulting in an 80-dimensional optimization problem. The exact setup of the optimization algorithm is defined in Table IV.

TABLE IV
SETUP OF EA-LSM USED IN THE PAPER.

Property	Symbol	Value
Number of parent individuals	μ	10
Number of offspring individuals	λ	70
Initial step size	σ_{init}	0.02
Static penalty for constraint handling	P	1000.0
Number of MMCs	m	32
Initial thickness of MMCs	t_{init}	4.0
Minimal thickness of MMCs	t_{th}	0.5

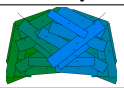
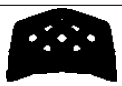
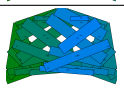
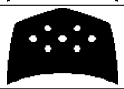
All of the optimization runs are conducted on a computational cluster using an in-house Python implementation of EA-LSM. For each of the optimization problems considered, always 30 independent optimization runs with different random seeds are carried out, to provide a sufficient basis for deriving conclusions regarding properties of the stochastic optimizers. The reported values of compliance, dissimilarity metric, and therefore, also the cost function, are normalized using the initial value computed for the reference design (Fig. 10B) to avoid strong dominance of one of the objectives. Please note, however, that the influence of w on the final values of both objectives can vary considerably depending on their local landscapes and an extensive sampling of w is usually required to generate a diverse set of solutions.

3) Results: In the following section the results of the experimental studies using similarity-based EA-LSM are presented. First of all, to evaluate the ability of EA-LSM to represent different types of designs present in the CarHoods10k data set, 30 optimization runs exclusively for similarity, i.e., for $w = 0$, and without a volume constraint, were carried out. The representatives of the three most frequent design types found by EA-LSM are shown in Table V, together with the closest designs in the data set and the corresponding values of the dissimilarity metric. The results show that 32 MMCs are sufficient to accurately represent different types of hood frames and reduce the dissimilarity metric (8) to the level of ca. $1\,\%$ to $4\,\%$ of the initial value.

Table VI shows the resulting structures for a TO for similarity ($w = 0$) with a 50% volume constraint, which was used also in case of all the remaining experiments discussed in this section. Again, EA-LSM is able to significantly reduce the dissimilarity metric compared to the initial design, reaching values of ca. 19%. To achieve that, the algorithm selects

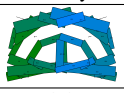
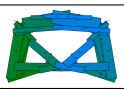
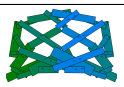
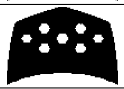
[3]Code available: http://www.calculix.de/

TABLE V
THREE MOST FREQUENT DESIGN TYPES OBTAINED IN 30 TO RUNS FOR SIMILARITY ($w = 0$) AND WITHOUT THE VOLUME CONSTRAINT.

Mat. distribution	MMC layout	Closest design	Dissimilarity
			$s = 0.0078$
			$s = 0.0233$
			$s = 0.0448$

reference designs of lower volume fractions from the data set compared to the ones presented in Table V and further modifies the topological features of the designs.

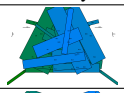
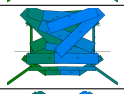
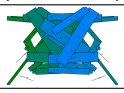
TABLE VI
THREE MOST FREQUENT DESIGN TYPES OBTAINED IN 30 TO RUNS FOR SIMILARITY ($w = 0$) AND 50% VOLUME CONSTRAINT.

Mat. distribution	MMC layout	Closest design	Dissimilarity
			$s = 0.1877$
			$s = 0.3263$
			$s = 0.4027$

The next 30 optimization runs were carried out considering exclusively the compliance (7) of the design ($w = 1$) and the 50% volume constraint, which is a typical problem formulation considered in structural TO [26]. The design prototypes presented in Table VII reduce compliance values to ca. 5–7% of the initial value and focus most of the material in the central part of the design space to maximize the stiffness of the structure against bending. Although the obtained designs significantly improve the structural performance, they are far from the hood frames used in the industry, which is a common problem in TO. To some extent, this problem might be mitigated by defining new load cases and imposing additional manufacturing constraints, however, in general case it is difficult to describe all the design criteria in a closed mathematical form [49], and cumbersome manual post-processing of TO results is necessary to obtain manufacturable structures.

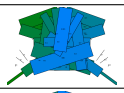
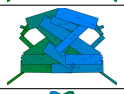
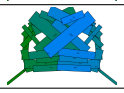
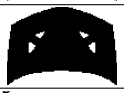
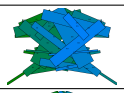
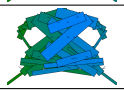
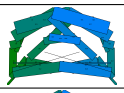
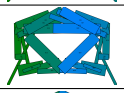
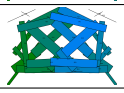
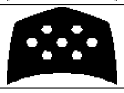
The manufacturability of the designs can be significantly improved by using the concurrent EA-LSM for compliance and similarity to the data set. Table VIII shows the most frequent design types obtained in 30 optimization runs for different weightings of the two objective functions, i.e., for $w = 0.75$, $w = 0.5$, and $w = 0.25$. The corresponding approximation of the Pareto front is shown in Figure 11. One can easily note a gradual transition from the designs similar to the ones shown in Table VII to the designs close to the ones presented in Table VI as the value of w is decreased. Interestingly, the results show that very often, significant

TABLE VII
THREE MOST FREQUENT DESIGN TYPES OBTAINED IN 30 TO RUNS FOR COMPLIANCE ($w = 1$) AND 50% VOLUME CONSTRAINT.

Material distribution	MMC layout	Compliance
		$c = 0.0533$
		$c = 0.0557$
		$c = 0.0707$

improvements in terms of similarity to the hood data set can be obtained without large reduction of structural performance compared to the standard compliance-based TO (e.g., for $w = 0.5$). This demonstrates the value of TO considering both structural performance criteria as well as prior design knowledge based on the concept of similarity to a data set proposed in this work.

TABLE VIII
MOST FREQUENT DESIGN TYPES OBTAINED IN 30 TO RUNS FOR EACH VALUE OF w IN A CONCURRENT OPTIMIZATION FOR COMPLIANCE AND SIMILARITY WITH A 50% VOLUME CONSTRAINT. DESIGNS ARE ORDERED ACCORDING TO THE COST FUNCTION (f), BEING A WEIGHTED SUM OF NORMALIZED COMPLIANCE (c) AND DISSIMILARITY (s). ASTERISK ($*$) INDICATES PARETO NON-DOMINATED SOLUTIONS.

Mat. distribution	MMC layout	Closest design	Objectives
$w = 0.75$			
			$f = 0.1361$ $c = 0.0596^*$ $s = 0.3658^*$
			$f = 0.1564$ $c = 0.0661$ $s = 0.4273$
			$f = 0.1742$ $c = 0.0645$ $s = 0.5034$
$w = 0.5$			
			$f = 0.2163$ $c = 0.0818^*$ $s = 0.3508^*$
			$f = 0.2315$ $c = 0.0847$ $s = 0.3782$
			$f = 0.2503$ $c = 0.0788$ $s = 0.4217$
$w = 0.25$			
			$f = 0.2159$ $c = 0.1849^*$ $s = 0.2256^*$
			$f = 0.2857$ $c = 0.1622^*$ $s = 0.3275^*$
			$f = 0.3382$ $c = 0.1346$ $s = 0.4093$

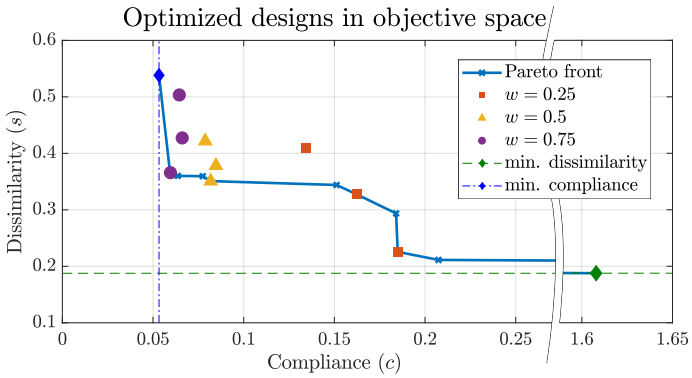


Fig. 11. Approximation of the Pareto front (blue solid line) based on 150 optimization runs with different values of w, where the Pareto-optimal solutions are marked using the cross sign, and the design types from Table VIII are indicated using square, triangular, and circular markers. The asymptotes correspond to the minimal values of compliance and dissimilarity obtained in 30 independent optimization runs for $w = 1$ and $w = 0$, respectively. The corresponding solutions, obtained in single-objective optimization runs, are marked using diamond signs.

V. CONCLUSION

We presented the CarHoods10k benchmark data set as the first set of industry-grade CAD data of over 10 000 car hood frames, including ground-truth information on design generation as well as performance values from finite element analysis (FEA) that describe structural performance properties. Data were generated using a parametrized CATIA work flow from automotive engineering and were validated by industry experts. We presented three example applications relevant to the engineering domain. First, we performed unsupervised representation learning for geometries using recently introduced geometric deep learning methods. Second, we trained a metamodel that predicted performance values as well as design parameters from learned representations with sufficient accuracy. Last, we demonstrated a topology optimization approach using evolutionary level-set methods that generated designs with high performance while fulfilling manufacturability constraints. The data set has been released to the public under a CC0 license and is freely available from [7]. Other than existing geometric data sets, CarHoods10k is specifically targeted at the engineering domain and thus provides performance data typically lacking from benchmark data sets. The data set comprises geometries of high-resolution, varying topology, and higher genus that are variations of the *same* object, which is not the case for existing data sets and enables, for example, the generation of metamodels. Last, the size of the data set enables the transfer of techniques from machine and deep learning as demonstrated in this paper. We strongly believe that CarHoods10k is a valuable tool in developing computational methods for industry design and development, allowing for the evaluation of data mining, optimization, and metamodeling methods for real-world engineering applications.

ACKNOWLEDGMENT

The authors thank Mike Berry of Honda Development & Manufacturing of America for valuable discussions and support in setting up the CAD workflow.

REFERENCES

[1] S. Ramnath, P. Haghighi, J. H. Kim, D. Detwiler, M. Berry, J. J. Shah, N. Aulig, P. Wollstadt, and S. Menzel, "Automatically generating 60,000 CAD variant for big data applications," in *ASME Int. Des. Eng. Tech. Conf. and Int. Conf. Comput. Ind. Eng., (IDETC-CIE 19)*, Anaheim, CA, USA, 2019.

[2] E. Ahmed, A. Saint, A. Shabayek, K. Cherenkova, R. Das, G. Gusev, D. Aouada, and B. Ottersten, "Deep learning advances on different 3D data representations: A survey," *arXiv:1808.01462 [cs.CV]*, 2018.

[3] K. D. Willis, Y. Pu, J. Luo, H. Chu, T. Du, J. G. Lambourne, A. Solar-Lezama, and W. Matusik, "Fusion 360 gallery: A dataset and environment for programmatic CAD construction from human design sequences," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, 2021.

[4] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo, "ABC: A big CAD model dataset for geometric deep learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR 2019)*, 2019.

[5] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," *arXiv:1512.03012 [cs.GR]*, 2015.

[6] L. Graening, S. Menzel, M. Hasenjäger, T. Bihrer, M. Olhofer, and B. Sendhoff, "Knowledge extraction from aerodynamic design data and its application to 3D turbine blade geometries," *Mathematical Modelling and Algorithms*, vol. 7, no. 329, pp. 329–350, 2008.

[7] S. Ramnath, J. J. Shah, P. Wollstadt, M. Bujny, S. Menzel, and D. Detwiler, "OSU-Honda Automobile Hood Dataset (CarHoods10k)," *Dryad, Dataset*, 2022. [Online]. Available: https://doi.org/10.5061/dryad.2fqz612pt

[8] M. Savva, A. X. Chang, and P. Hanrahan, "Semantically-enriched 3D models for common-sense knowledge," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR 2015)*, Boston, MA, USA, 2015.

[9] L. Yi, L. Guibas, A. Hertzmann, V. G. Kim, H. Su, and E. Yumer, "Learning hierarchical shape segmentation and labeling from online repositories," in *Proc. SIGGRAPH*, Los Angeles, CA, USA, 2017.

[10] Y. Li, A. Dai, L. Guibas, and M. Nießner, "Database-assisted object retrieval for real-time 3D reconstruction," *Computer Graphics Forum*, vol. 34, no. 2, pp. 435–446, 2015.

[11] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. 35th Int. Conf. on Machine Learning (ICML 2018)*, vol. 1, Stockholm, Sweden, 2018, pp. 67–85.

[12] T. Rios, B. van Stein, P. Wollstadt, T. Bäck, B. Sendhoff, and S. Menzel, "Exploiting local geometric features in vehicle design optimization with 3D point cloud autoencoders," in *Proc. IEEE Congr. Evol. Comput. (CEC 2021)*, Kraków, Poland, 2021.

[13] L. Regenwetter, B. Curry, and F. Ahmed, "BIKED: A dataset and machine learning benchmarks for data-driven bicycle design," in *ASME Int. Des. Eng. Tech. Conf. and Int. Conf. Comput. Ind. Eng., (IDETC-CIE 2021)*, 2021.

[14] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Process. Mag.*, pp. 18–42, 2017.

[15] P. Wang, Y. Gan, P. Shui, F. Yu, Y. Zhang, S. Chen, and Z. Sun, "3D shape segmentation via shape fully convolutional networks," *Computers & Graphics*, vol. 76, pp. 182–192, 2018.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv:1609.02907 [cs.LG]*, 2016.

[17] T. Friedrich, N. Aulig, and S. Menzel, "On the potential and challenges of neural style transfer for three-dimensional shape data," in *Proc. Int. Conf. on Eng. Opt. (EngOpt 2018)*, Lisboa, Portugal, 2018, pp. 581–592.

[18] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV 2015)*, Santiago, Chile, 2015, pp. 37–45.

[19] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D faces using convolutional mesh autoencoders," in *Proc. Europ. Conf. on Comp. Vis. (ECCV 2018)*, Munich, Germany, 2018.

[20] O. Sorkine, "Laplacian mesh processing," in *Proc. 26th annual conference of the European Association for Computer Graphics (EUROGRAPHICS '05)*, vol. 109, no. 2732, Dublin, Ireland, 2005, p. 318.

[21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR 2017)*, Honolulu, HI, USA, 2017, pp. 652–660.

[22] D. Rethage, J. Wald, J. Sturm, N. Navab, and F. Tombari, "Fully-convolutional point networks for large-scale point clouds," in *Proc. Europ. Conf. on Comp. Vis. (ECCV 2018)*, Munich, Germany, 2018, pp. 596–611.

[23] T. Rios, P. Wollstadt, B. V. Stein, T. Bäck, Z. Xu, B. Sendhoff, and S. Menzel, "Scalability of learning tasks on 3D CAE models using point cloud autoencoders," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI 2019)*, Xiamen, China, 2019.

[24] T. Rios, B. Sendhoff, S. Menzel, T. Bäck, and B. van Stein, "On the efficiency of a point cloud autoencoder as a geometric representation for shape optimization," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI 2019)*, Xiamen, China, 2019, pp. 791–798.

[25] M. Tschannen, O. Bachem, and M. Lucic, "Recent advances in autoencoder-based representation learning," *ArXiv Preprint: arXiv:1812.05069 [cs.LG]*, 2018.

[26] M. P. Bendsøe and O. Sigmund, *Topology Optimization*. Berlin, Heidelberg, Germany: Springer, 2004.

[27] N. v. Dijk, K. Maute, M. Langelaar, and F. v. Keulen, "Level-set methods for structural topology optimization: a review," *Struct. Multidiscip. Optim.*, vol. 48, no. 3, pp. 437–472, 2013.

[28] F. Wein, P. Dunning, and J. A. Norato, "A review on feature-mapping methods for structural optimization," *Struct. Multidiscip. Optim.*, pp. 1–42, 2019.

[29] C. Ortmann and A. Schumacher, "Graph and heuristic based topology optimization of crash loaded structures," *Struct. Multidiscip. Optim.*, vol. 47, no. 6, pp. 839–854, 2013.

[30] M. Bujny, N. Aulig, M. Olhofer, and F. Duddeck, "Identification of optimal topologies for crashworthiness with the evolutionary level set method," *Int. J. Crashworthiness*, vol. 23, no. 4, pp. 395–416, 2017.

[31] C. M. Aye, N. Pholdee, A. R. Yildiz, S. Bureerat, and S. M. Sait, "Multi-surrogate-assisted metaheuristics for crashworthiness optimisation," *Int. J. Veh. Des.*, vol. 80, no. 2-4, pp. 223–240, 2019.

[32] R. Tavakoli and P. Davami, "Optimal riser design in sand casting process with evolutionary topology optimization," *Struct. Multidiscip. Optim.*, vol. 38, no. 2, pp. 205–214, 2009.

[33] J. Hiller and H. Lipson, "Automatic Design and Manufacture of Soft Robots," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 457–466, 2012.

[34] Y. M. Xie and G. P. Steven, *Evolutionary Structural Optimization*. London, UK: Springer, 1997.

[35] A. Tovar, "Bone remodeling as a hybrid cellular automaton optimization process," PhD Thesis, University of Notre Dame, Notre Dame, IN, USA, 2004.

[36] D. Guirguis, N. Aulig, R. Picelli, B. Zhu, Y. Zhou, W. Vicente, F. Iorio, M. Olhofer, W. Matusik, C. A. Coello Coello, and K. Saitou, "Evolutionary black-box topology optimization: Challenges and promises," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 613–633, 2020.

[37] D. Molina, J. Poyatos, J. Del Ser, S. García, A. Hussain, and F. Herrera, "Comprehensive taxonomies of nature- and bio-inspired optimization: Inspiration versus algorithmic behavior, critical analysis recommendations," *Cognitive Computation*, vol. 12, no. 5, pp. 897–939, 2020.

[38] S. Gupta, H. Abderazek, B. S. Yıldız, A. R. Yildiz, S. Mirjalili, and S. M. Sait, "Comparison of metaheuristic optimization algorithms for solving constrained mechanical design optimization problems," *Expert Systems with Applications*, p. 115351, 2021.

[39] P. Rostami and J. Marzbanrad, "Identification of Optimal Topologies for Continuum Structures Using Metaheuristics: A Comparative Study," *Arch. Comput. Methods Eng.*, 2021.

[40] D.-H. Kim, K.-H. Jung, D.-J. Kim, S.-H. Park, D.-H. Kim, J. Lim, B.-G. Nam, and H.-S. Kim, "Improving pedestrian safety via the optimization of composite hood structures for automobiles based on the equivalent static load method," *Composite Structures*, vol. 176, pp. 780–789, 2017.

[41] M. Bujny, "Level set topology optimization for crashworthiness using evolutionary algorithms and machine learning," PhD Thesis, Technische Universität München, Munich, Germany, 2020.

[42] X. Guo, "Doing topology optimization explicitly and geometrically: a new moving morphable components based framework," in *Frontiers in Applied Mechanics*. Imperial College Press, 2014, pp. 31–32.

[43] M. Bujny, N. Aulig, M. Olhofer, and F. Duddeck, "Evolutionary crashworthiness topology optimization of thin-walled structures," in *Proc. 11th Int. Conf. on Numerical Optimisation Methods for Engineering Design (ASMO UK/ISSMO/NOED 2016)*, Munich, Germany, 2016.

[44] E. Raponi, M. Bujny, M. Olhofer, N. Aulig, S. Boria, and F. Duddeck, "Kriging-assisted topology optimization of crash structures," *Computer Methods in Appl. Mech. Eng.*, vol. 348, pp. 730–752, 2019.

[45] E. Raponi, M. Bujny, M. Olhofer, S. Boria, and F. Duddeck, "Hybrid strategy coupling EGO and CMA-ES for structural topology optimiza-

tion in statics and crashworthiness," in *International Joint Conference on Computational Intelligence*, 2019, pp. 55–84.

[46] E. Denimal, F. El Haddad, C. Wong, and L. Salles, "Topological optimization of under-platform dampers with moving morphable components and global optimization algorithm for nonlinear frequency response," *J Eng. Gas Turbines Power*, vol. 143, no. 2, p. 021021, 2021.

[47] M. Bujny, M. Olhofer, and F. Duddeck, "Optimal structures for crash by additive manufacturing," in *Proc. 1st ECCOMAS Thematic Conference on Simulation for Additive Manufacturing*, Munich, Germany, 2017.

[48] S. Oh, Y. Jung, S. Kim, I. Lee, and N. Kang, "Deep generative design: Integration of topology optimization and generative models," *Journal of Mechanical Design*, vol. 141, no. 11, 2019.

[49] M. S. Yousaf, M. Bujny, N. Zurbrugg, D. Detwiler, and F. Duddeck, "Similarity control in topology optimization under static and crash loading scenarios," *Eng. Optim.*, vol. 53, no. 9, pp. 1523–1538, 2021.

[50] S. Ramnath, P. Haghighi, J. Ma, J. J. Shah, and D. Detwiler, "Design science meets data science: Curating large design datasets for engineered artifacts," in *ASME Int. Des. Eng. Tech. Conf. and Int. Conf. Comput. Ind. Eng., (IDETC-CIE 2020)*, vol. 83983. American Society of Mechanical Engineers, 2020, p. V009T09A043.

[51] S. Ramnath, J. Ma, J. J. Shah, and D. Detwiler, "Intelligent design prediction aided by experiment design and machine learning in feature based product development," in *ASME Int. Des. Eng. Tech. Conf. and Int. Conf. Comput. Ind. Eng., (IDETC-CIE 2021)*, 2021, pp. 1–11.

[52] S. Ramnath, A. Li, J. J. Shah, and D. Detwiler, "Design exploration and prediction of automotive hood designs based on non-uniform feature parameters," in *NAFEMS World Congress 2021*, Salzburg, Austria, 2021, pp. 1–11 [under review].

[53] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovacevic, "Contour-enhanced resampling of 3D point clouds via graphs," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP 2017)*, pp. 2941–2945, 2017.

[54] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, and Z. Chen, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org/

[55] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. on Learning Representations (ICLR 2015)*, 12 2014.

[56] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, "Automatic chemical design using a data-driven continuous representation of molecules," *ACS Central Science*, vol. 4, no. 2, pp. 268–276, 2018.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[58] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[59] M. Bujny, M. Olhofer, N. Aulig, and F. Duddeck, "Topology Optimization of 3D-printed joints under crash loads using Evolutionary Algorithms," *Struct. Multidiscip. Optim.*, vol. 64, no. 6, pp. 4181–4206, 2021.

[60] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 1–23, 1993.

[61] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Appl. Mech. Eng.*, vol. 191, no. 11–12, pp. 1245–1287, 2002.

Patricia Wollstadt received the Dipl.-Psych. degree in psychology in 2012, her B.Sc. in computer science in 2012, the and M.Sc. in computer science in 2016 from Goethe-University, Frankfurt, Germany. She earned her Ph.D. in computer science in 2018 also from Goethe-University, Frankfurt, Germany. Since 2018, she is with the Honda Research Institute Europe, Offenbach, Germany, where she is currently a Senior Scientist in the Optimization and Creativity Group. Her current research interests include information theory and its application in data analysis and machine learning, as well as machine learning, and data mining in engineering applications and human-machines interaction.

Mariusz Bujny received his B.Sc. in Mechanical Engineering from the Warsaw University of Technology, Poland, and B.A. in Economics from the University of Warsaw in 2013. In 2015, he earned his M.Sc. in Computational Science and Engineering from the Technical University of Munich (TUM), Germany. In 2020, he completed a doctorate at the Chair of Computational Mechanics, TUM, conducted in cooperation with the Honda Research Institute Europe (HRI-EU), Offenbach/Main, Germany. Since 2018, he is a researcher at HRI-EU, where he is currently a Senior Scientist in the Optimization and Creativity Group, shaping international projects in the domains of optimization and vehicle crashworthiness. His research interests include applications of non-gradient techniques and machine learning in topology optimization.

Satchit Ramnath is a research specialist at the Simulation Innovation and Modeling Center, at The Ohio State University since 2016. He is also a Ph.D. candidate in the Integrated Systems Engineering department, at The Ohio State University. His research interests include topology optimization, size optimization, ML in product design, and software development for various CAE applications. Satchit has previously worked as a senior design engineer on the Airbus A350 program.

Jami J. Shah is Honda Professor of Engineering Design at Ohio State. He was previously Professor of Mechanical & Aerospace Engineering at Arizona State. He earned a Ph.D. in Mechanical Engineering. He is the co-author of 2 US patents, 2 books, and 300+ peer reviewed technical papers. He is the founding chief editor of ASME Transaction, the Journal of Computing & Information Science in Engineering (JCISE), Area Editor of Research in Engineering Design and Co-Chief Editor of Journal of Computational Design & Engineering. He was elected Fellow of ASME in 2001. He is the recipient of numerous awards, including the 2008 ASME Design Automation award, Siemens PLM Engineering Education Excellence Award 2009 and the ASME CIE Lifetime Achievement Award given in 2012.

Duane Detwiler is Chief Engineer and Research Lead at Honda Development & Manufacturing of America – Automotive Development Center in Raymond, Ohio, where he leads a team that engages in strategic research partnerships to innovate technologies for future Honda applications. His technical career of over 25 years at Honda has focused on research, development, and application of physics-based modeling, simulation, and optimization methods for virtual vehicle development earning him multiple patents. He obtained a B Eng. 1991 and M Eng. 1993 from The Ohio State University, and he has co-authored dozens of scientific papers and patents through his collaborations with research partners.

Stefan Menzel received the Dipl.-Ing. degree in civil engineering from RWTH Aachen, Aachen, Germany, in 1998 and the Ph.D. degree in civil engineering from Technical University Darmstadt, Darmstadt, Germany, in 2004. Since 2004, he is with the Honda Research Institute Europe, Offenbach, Germany, where he is currently a Chief Scientist with the Optimization and Creativity Group. His current research interests include evolutionary optimization with special focus on adaptive representations, machine learning for knowledge transfer, and multidisciplinary optimization for real-world applications.