

## Design representation for performance evaluation of 3D shapes in structure-aware generative design

Xingang Li<sup>1</sup>, Charles Xie<sup>2</sup> and Zhenghui Sha <sup>1</sup>

<sup>1</sup>Walker Department of Mechanical Engineering, University of Texas at Austin, Austin, TX, USA

<sup>2</sup>Institute for Future Intelligence, Natick, MA, USA

### Abstract

Data-driven generative design (DDGD) methods utilize deep neural networks to create novel designs based on existing data. The structure-aware DDGD method can handle complex geometries and automate the assembly of separate components into systems, showing promise in facilitating creative designs. However, determining the appropriate vectorized design representation (VDR) to evaluate 3D shapes generated from the structure-aware DDGD model remains largely unexplored. To that end, we conducted a comparative analysis of surrogate models' performance in predicting the engineering performance of 3D shapes using VDRs from two sources: the trained latent space of structure-aware DDGD models encoding structural and geometric information and an embedding method encoding only geometric information. We conducted two case studies: one involving 3D car models focusing on drag coefficients and the other involving 3D aircraft models considering both drag and lift coefficients. Our results demonstrate that using latent vectors as VDRs can significantly deteriorate surrogate models' predictions. Moreover, increasing the dimensionality of the VDRs in the embedding method may not necessarily improve the prediction, especially when the VDRs contain more information irrelevant to the engineering performance. Therefore, when selecting VDRs for surrogate modeling, the latent vectors obtained from training structure-aware DDGD models must be used with caution, although they are more accessible once training is complete. The underlying physics associated with the engineering performance should be paid attention. This paper provides empirical evidence for the effectiveness of different types of VDRs of structure-aware DDGD for surrogate modeling, thus facilitating the construction of better surrogate models for AI-generated designs.

**Keywords:** Data-driven generative design, Structure-aware generative design, Vectorized design representation, Surrogate modeling, Engineering performance prediction, Automated machine learning

### 1. Introduction

Design researchers have applied artificial intelligence (AI) techniques to support various design activities, including design exploration and optimization, design synthesis and the extraction of human preferences for designs to help human designers make decisions during the design process (McComb, Cagan & Kotovsky 2017; Panchal *et al.* 2019; Rahman, Xie & Sha 2019, 2021; Rahman *et al.* 2020). Among various AI techniques, generative design (GD) techniques are receiving

Received 11 November 2022

Revised 12 August 2023

Accepted 14 August 2023

Corresponding author

Zhenghui Sha

[zsha@austin.utexas.edu](mailto:zsha@austin.utexas.edu)

© The Author(s), 2023. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike licence (<http://creativecommons.org/licenses/by-nc-sa/4.0>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the same Creative Commons licence is used to distribute the re-used or adapted article and the original article is properly cited. The written permission of Cambridge University Press must be obtained prior to any commercial use.

*Des. Sci.*, vol. 9, e27

[journals.cambridge.org/dsj](https://journals.cambridge.org/dsj)

DOI: [10.1017/dsj.2023.25](https://doi.org/10.1017/dsj.2023.25)

The Design Society  
a worldwide community

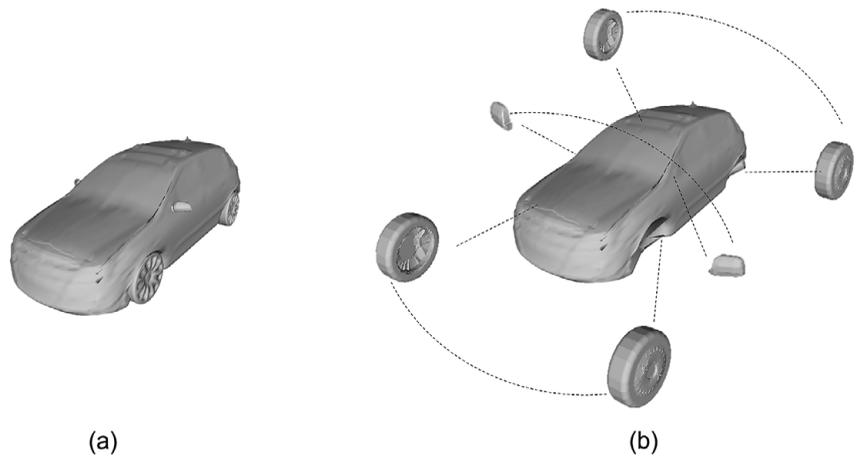
CAMBRIDGE  
UNIVERSITY PRESS

more attention in both industry and academic fields (Krish 2011; McKnight 2017; Matejka *et al.* 2018; Chen, Chiu & Fuge 2020; McComb *et al.* 2020). GD is a term for a class of tools that can generate novel yet realistic designs leveraging computational and manufacturing capabilities (Shea, Aish & Gourtovaia 2005). There are some widely used GD techniques, such as genetic algorithms and shape grammars (Singh & Gu 2012). GD has been applied in several commercial CAD software, such as Autodesk Fusion 360, PTC Creo and Siemens NX. However, current GD methods are driven primarily and solely by engineering performance, so the generated designs often do not agree with conventional aesthetics (Oh *et al.* 2019). Additionally, generated designs may be too complex to be created without using additive manufacturing (McKnight 2017). These problems can be alleviated by deep generative models (Oh *et al.* 2019), capable of learning to produce new data given a set of training examples. State-of-the-art deep generative models, such as the variational autoencoder (VAE) (Kingma & Welling 2013) and the generative adversarial network (GAN) (Goodfellow *et al.* 2014), have been applied in various fields, including computer vision, computational creativity, architecture and engineering design (Yi, Walia & Babyn 2019; de *et al.* 2020; Regenwetter, Nobari & Ahmed 2022; Li, Wang & Sha 2023).

In the design literature, these deep generative models are often referred to as data-driven generative design (DDGD) methods. DDGD methods have been increasingly used to improve design creativity and facilitate conceptual design, such as airfoil design (Chen *et al.* 2020; Chen & Ahmed 2021), car wheel design (Oh *et al.* 2019; Yoo *et al.* 2021) and car shape design (Li, Xie & Sha 2021, 2022). DDGD methods can learn to synthesize designs from data without explicit human configuration by training a deep neural network model and learning a latent vector space with a predefined (often reduced) dimensionality. Such a latent vector space is a low-dimensional representation of the design space from which the data were observed. Since the training process combines features from all existing designs, new designs that are not seen from existing data can be sampled from the latent design space (Krish 2011; Cunningham *et al.* 2020). Therefore, DDGD methods have become an important tool for the generation of conceptual design ideas due to their ability to quickly generate a large number of novel designs.

Traditionally, most DDGD methods treat each design data as a monolithic whole (i.e. one single object without considering the interconnections of components as shown in Figure 1(a)) for model training (Shu *et al.* 2020). In this paper, we refer to them as traditional DDGD methods. Recently, there have been emerging interests in developing structure-aware DDGD methods (Chen & Fuge 2019; Mo *et al.* 2019; Gao *et al.* 2019b; Li *et al.* 2021). Compared to traditional DDGD methods, structure-aware DDGD methods can handle complex geometries consisting of interconnected components and learn interdependencies between components (i.e. structure-aware shapes as shown in Figure 1(b)) to enable automatic assembly of deep-generated components in a system (Gao *et al.* 2019b). It can also provide designers with increased flexibility to make local design modifications by altering or substituting individual parts.

To evaluate the engineering performance of designs from DDGD methods, there are two different ways. One is to conduct high-fidelity simulations, for example, based on computational fluid dynamics (CFD) or finite element analysis (FEA). However, the downside of these simulations is the high computational cost. For example, assessing the aerodynamic performance of a 3D car model using CFD



**Figure 1.** Comparison between a monolithic shape (a) and a structure-aware shape (b), where dash lines indicate structural interdependencies (i.e. support and symmetry) between components.

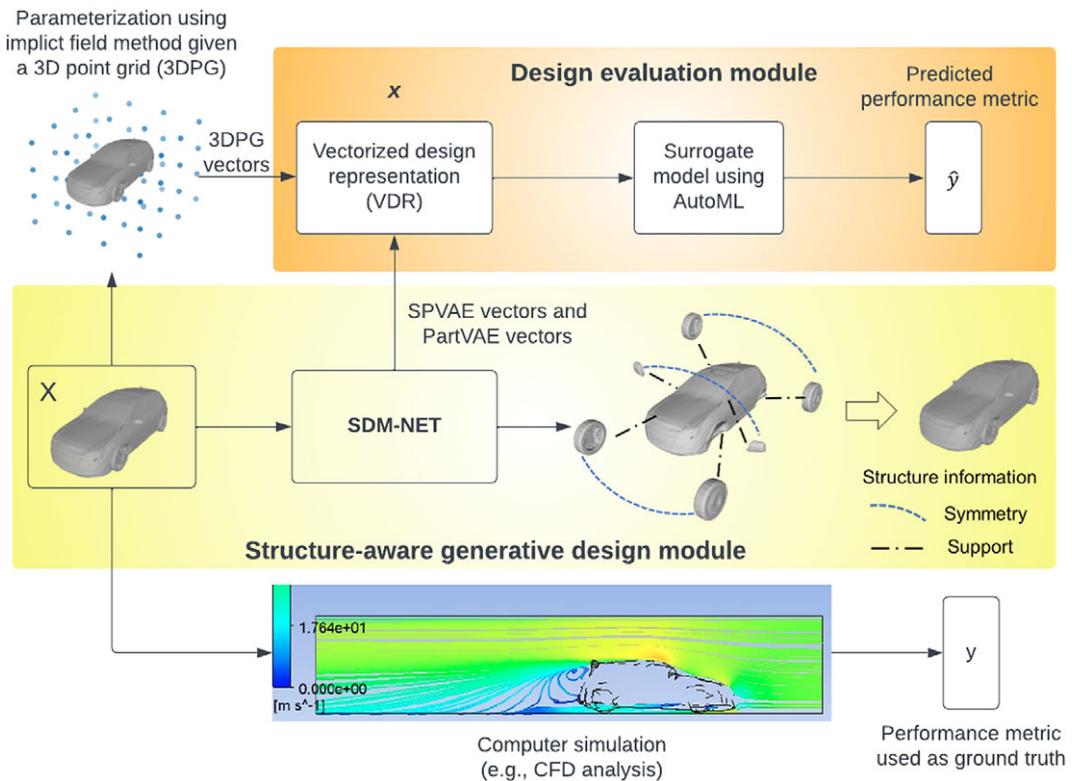
software could take hours to complete. Therefore, it is impractical to evaluate the vast number of design alternatives obtained from DDGD methods in support of fast design decision-making. The other way is to use surrogate models that have a relatively lower fidelity but can significantly accelerate the evaluation process. Surrogate modeling is a supervised machine learning technique to approximate the output based on the labeled training dataset (i.e. pairs of inputs and their corresponding outputs) (Sun *et al.* 2020; Whalen & Mueller 2022). Generally, in these surrogate modeling methods, each design is represented as a fixed-length vector of design parameters, referred to as vectorized design representation (VDR). VDR enables compact encoding of complex design configurations, making it easy to process and analyze design data mathematically and computationally. As one type of VDRs, latent vectors have recently been widely adopted in design generation, evaluation and optimization (Burnap *et al.* 2016; Umetani & Bickel 2018; Chen *et al.* 2020; Li, Xie & Sha 2022). Latent vectors are obtained from a latent space during the training process in neural network models, such as VAEs and GANs. A latent space is often continuous and low-dimensional (compared to the dimensionality of the training data) and packs complex data distributions. Vectors in such a latent space can capture the underlying structure and important features of the training data.

Previous studies have primarily concentrated on utilizing latent vectors from traditional DDGD methods as the VDR for design evaluation. Little is known about the efficacy of latent vectors acquired from the structure-aware DDGD training process, which encompasses both part-to-part structural information and geometric information. The research question, therefore, arises: What would be the appropriate VDR for a computational pipeline in support of the evaluation of structure-aware deep-generated shapes? In particular, is it reliable to directly use the latent vectors readily available from the training process of a structure-aware DDGD model?

To answer this question, we performed experiments to compare the performance of the latent vectors obtained from the training process of a structure-aware

DDGD model in predicting the engineering performance of the designs, with those obtained by embedding the generated 3D shapes (after training) using a 3D point grid (3DPG), as shown in Figure 2. We conducted the comparative study in two case studies: 1) predicting the drag coefficients of car designs and 2) predicting both the drag and lift of aircraft designs. Our results indicate that while latent vectors are frequently used in surrogate models, they may not be suitable when the encoded information includes factors that have minimal relevance to the engineering performance under investigation (e.g. the SPVAE vectors containing structural information in our study). Employing such VDRs can actually hinder the prediction accuracy of a surrogate model. This new knowledge is significant because a proper VDR is crucial to the accuracy of the engineering analysis and, therefore, the validity of a 3D shape and its associated economic impact. For example, the drag evaluation of car body shapes significantly influences their fuel economy estimate. With a 10% reduction in aerodynamic drag, the highway fuel economy will improve by approximately 5% and the city fuel economy by approximately 2% (ARC, n.d.). Therefore, it is crucial to consider the physics underlying engineering performance metrics and select VDRs that integrate relevant information, such as geometric information, to enhance the prediction accuracy of surrogate models.

The remainder of this paper is organized as follows. Section 2 provides a review of relevant research on both traditional and structure-aware DDGD methods and



**Figure 2.** Overview of the research approach consisting of two key modules: the structure-aware generative design module and the design evaluation module

surrogate models. The DDGD methods and surrogate models adopted, as well as the proposed research approach, are presented in [Section 3](#). We then present and discuss the experimental results and summarize the main findings in [Sections 4](#) and [5](#). The paper is concluded in [Section 6](#), in which we summarize the closing insights and potential future research directions.

## 2. Literature review

In this section, we present a review of the existing literature that is most relevant to this study, including data-driven generative design methods, structure-aware generative design methods and surrogate models for design evaluation.

### 2.1. Data-driven generative design methods in engineering design

Data-driven generative design (DDGD) methods can conduct an efficient design space exploration by generating a large number of various new design concepts from existing design data (e.g. images or 3D shapes) without an explicit set of design variables (Achour *et al.* 2020). In engineering design, DDGD methods are developed mainly based on two techniques, generative adversarial networks (GANs) and variational autoencoders (VAEs), in addition to a few others, such as recurrent neural networks (RNNs) and reinforcement learning (RL) (Regenwetter *et al.* 2022).

For example, focusing on 2D design applications, Oh *et al.* (2019) integrate a topology optimization (TO) technique with GANs to generate numerous aesthetic design options taking into account engineering performance. Their method was applied to the design of 2D car wheel rims. Chen *et al.* (2020) develop a Bezier-GAN model that can learn from shape variations in an existing 2D airfoil database to parameterize aerodynamic designs so that the resulting parameterization can accelerate design optimization. Dering *et al.* (2018) set up a physics-based virtual environment that combines an RNN model to enhance the quality of deep-generated designs of 2D cruise ships. Fujita *et al.* (2021) propose a framework for the generation of design concepts by applying TO and a variational deep embedding method in a 2D bridge design problem.

In 3D design applications, Shu *et al.* (2020) present a method that combines GANs and a physics-based virtual environment introduced by Dering *et al.* (2018) to generate high-performance 3D aircraft models. Zhang *et al.* (2019) propose a method using VAEs, a physics-based simulator, and a functional design optimizer to synthesize 3D aircraft with prescribed engineering performance. Building on the 2D wheel generative design work (Oh *et al.* 2019), Yoo *et al.* (2021) develop a deep learning-based CAD/CAE framework that can automatically generate 3D car wheels from 2D images by point extraction (i.e. to extract the points from contour lines of the wheels) and sketch extrusion.

All of the methods mentioned above address the form and functionality of designs in either 2D or 3D forms, which can help designers explore the design space by automatically generating a large number of design concepts with informed engineering performance. However, they all consider designs as one single monolithic piece and ignore the interrelations between components in a product or an assembly.

## 2.2. Structure-aware design generation methods

Acknowledging that real-world designs usually consist of multiple parts, Chen & Fuge (2019) develop hierarchical GANs to synthesize designs with inter-part dependencies. The method is demonstrated using a design case of 2D airfoils. However, 3D shapes are often the final form of most products, and structure-aware 3D design studies mostly come from the computer science community. Li *et al.* (2017) introduce a generative recursive autoencoder for shape structures (GRASS) based on recursive neural networks. GRASS trains independent networks for the geometry and structure of parts and generates 3D voxel shapes by producing a hierarchical series of bounding boxes filled with voxels. Nash & Williams (2017) propose a generative model of part-segmented 3D objects, namely, the shape variational autoencoder (ShapeVAE). Given a collection of dense surface points with surface normals of part-segmented objects, ShapeVAE can learn a low-dimensional shape embedding to synthesize new and realistic 3D shapes represented by point clouds, which can then be converted to 3D meshes using the surface normals. Mo *et al.* (2019) introduce StructureNet, a generative autoencoder that learns shape structure using graph neural networks. StructureNet uses graphs to encode hierarchical representations of shapes. After training, it can generate 3D shapes formed by box structures or 3D point cloud shapes. Gao *et al.* (2019b) propose Structured Deformable Meshes Net (SDM-NET) which consists of PartVAEs and a Structured-Part VAE (SPVAE) for the generation of 3D mesh shapes. PartVAE is used to learn individual part geometry, and SPVAE is used to learn part geometries and the structure of the 3D models. SDM-NET can directly output 3D mesh shapes with high surface quality. Compared to point clouds and voxels, meshes can better capture the geometric details (e.g. smoothness, curvature) of 3D objects without consuming large storage space. Therefore, mesh representation is more suitable for engineering design that requires fine-grained details of geometry so that they can be accurately measured, prototyped and tested for engineering performance. In this study, we adopt SDM-NET as our structure-aware generative design module to generate 3D mesh shapes with high surface quality.

## 2.3. Surrogate models and AutoML in engineering design

Engineering performance evaluation is a critical link in engineering design, optimization and computational manufacturing. But it is usually computationally expensive. For example, CFD evaluation of the aerodynamic performance of 3D automobile models requires solving the Navier–Stokes equation, which could take hours and days depending on the level of fidelity and computer configurations (Umetani & Bickel 2018). Therefore, the development of a cost-effective surrogate model holds practical significance. The primary purpose of a surrogate model is to act as an approximation model, replacing intricate and time-consuming computations, to facilitate a fast evaluation of designs without compromising on accuracy (see Queipo *et al.* 2005 and Wang & Shan 2006 for a review). Typically, surrogate models require that the design be represented as a fixed-length vector (i.e. vectorized design representation (VDR)) (Umetani & Bickel 2018; Chen *et al.* 2020).

The process of training surrogate models typically involves utilizing labeled design data, where the labels represent performance metrics of interest. This training can be approached as a supervised learning problem by employing machine learning (ML) techniques. Creating an ML model that achieves excellent performance often requires much investment in terms of computational time and resources in tasks such as feature engineering, model selection and hyperparameter optimization. Recently, there has been a growing interest in applying Automated Machine Learning (AutoML) to accelerate the process of training optimal surrogate models. AutoML leverages sophisticated algorithms to explore a wide range of models and hyperparameters. This automated search process often leads to better-performing models compared to manual experimentation due to its ability to navigate effectively through extensive search spaces and discover the most favorable configurations (He, Zhao & Chu 2021).

Although design researchers have been employing common surrogate models outlined in the literature and following industry practices (Cunningham, Simpson & Tucker 2019; Whalen & Mueller 2022), there is limited awareness of AutoML within the engineering design community (Regenwetter, Weaver & Ahmed 2023). Regenwetter *et al.* (2023) took a lead in this regard by comparing the performance of surrogate models constructed using traditional methods (e.g. decision trees, k-nearest neighbors, XGBoost (Chen & Guestrin 2016) and neural networks with Bayesian optimization) against those built using AutoML frameworks. They demonstrated that AutoML outperforms other surrogate models in a bicycle design application and called for the attention of the design community to explore the use of AutoML frameworks. Based on their findings, we compare the performance of different VDRs by adopting two AutoML frameworks, that is Auto-sklearn (Feurer *et al.* 2015) and AutoGluon (Erickson *et al.* 2020), due to their superior performance compared to other alternatives.

### 3. Research approach

As shown in Figure 2, the proposed approach consists of two key modules: the structure-aware generative design module, which employs the SDM-NET (Gao *et al.* 2019b); and the design evaluation module, which utilizes surrogate models implemented with AutoML frameworks. The structure-aware generative design module (Section 3.1) harnesses the capabilities of SDM-NET to enable efficient exploration of design spaces by incorporating structural information and can generate designs that not only exhibit aesthetic appeal but also possess fine geometric details. On the other hand, the design evaluation module (Section 3.2) uses AutoML techniques to construct surrogate models that approximate the engineering performance of interest. While focusing on a car body design as the primary case study to showcase the proposed approach, we ensure that the methodology maintains its generalizability. Therefore, we present a second case study on the aircraft design. See Section 4 for details.

### 3.1. Structure-aware generative design module

We implement SDM-NET (Gao *et al.* 2019b) for the structure-aware generative design module to generate 3D mesh shapes. This module consists of two types of VAEs: PartVAE and SPVAE. Given a 3D shape consisting of several parts, a PartVAE can learn the geometry of an individual part, and SPVAE can learn the geometries and the structure of parts jointly. We make no novel modifications to the network architecture of SDM-NET. Therefore, we only explain the key steps (see Figure 3) to facilitate the understanding of the latent spaces of the structure-aware generative design module. The generative design module is trained using a two-stage training strategy by training the PartVAEs first and then the SPVAE.

#### 3.1.1. Two-stage training of the PartVAEs and SPVAE

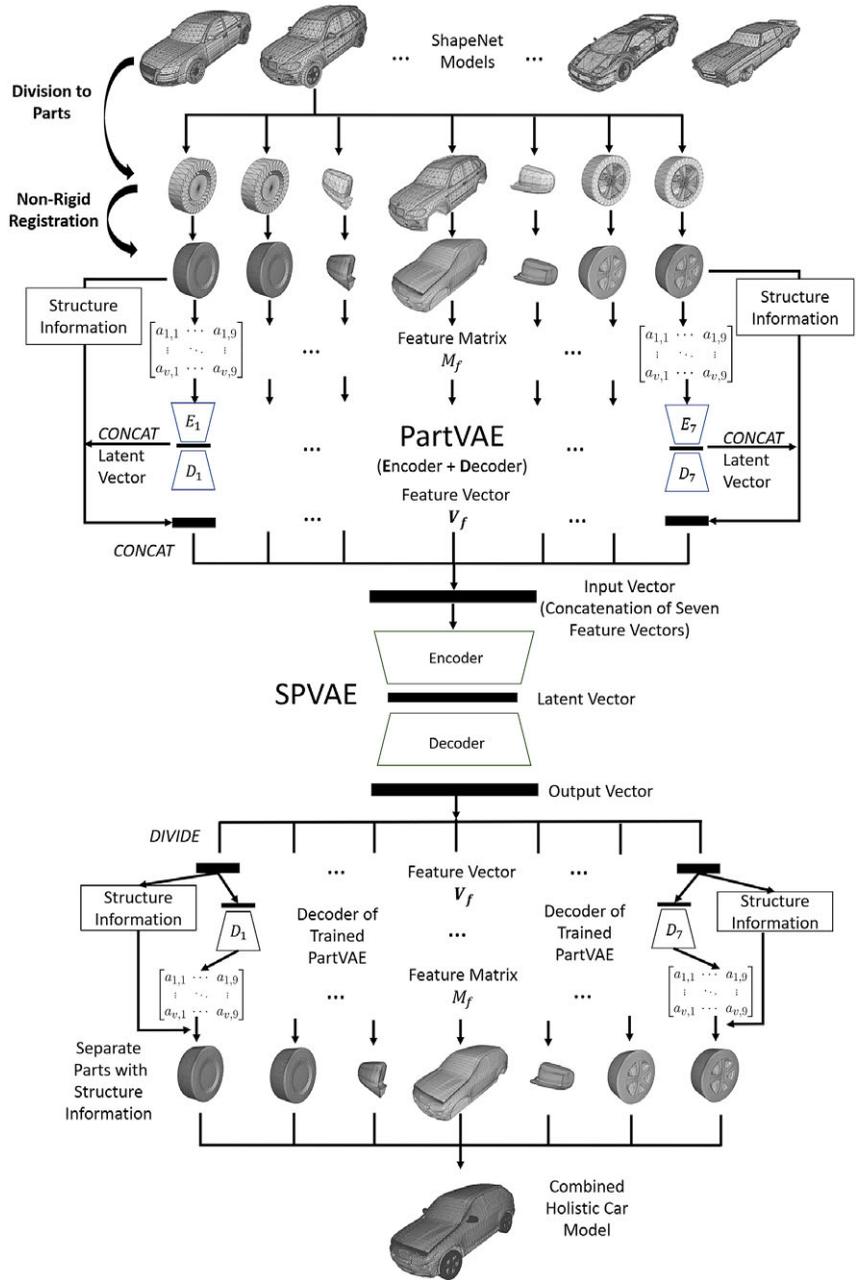
A 3D car model is first segmented into seven parts (i.e. one car body, two mirrors and four wheels). 3D models from open source databases, such as ShapeNet (Chang *et al.* 2015), are often unstructured and unoriented triangle meshes. Thus, such 3D mesh shapes cannot be directly used in DDGD methods without proper preprocessing (e.g. voxelized or re-meshed). These shapes may also contain interior parts (e.g. seats and steering wheels) that are not desired, since we focus on the external geometry only. As a widely used technology in computer graphics that maps one point set to another, non-rigid registration (Zollhöfer *et al.* 2014) is applied to re-mesh each part (e.g. car body) using a watertight template mesh shape. In our study, we use a cube as the template mesh that contains 19.2 k triangles (9602 vertices). All re-meshed parts are watertight with the same mesh connectivity of the template mesh from which design features will be extracted.

The As Consistent As Possible (ACAP) method (Gao *et al.* 2019a) is applied to extract the design features of a part for the input of its corresponding PartVAE. We deform the same cube mesh as the one used in non-rigid registration to a target part by multiplying transformation matrices, from which nine unique numbers can be extracted for each vertex of the mesh shape. Thus, a part with  $\nu$  vertices can be represented by a feature matrix  $M_f \in \mathbb{R}^{\nu \times 9}$ , where  $\nu = 9602$  in our implementation. One feature matrix can be obtained from each part of a car model, which will be the input to one PartVAE. Thus, seven PartVAEs are trained for car models. After training, the latent space of each PartVAE can be obtained and the latent vector corresponding to a part will be concatenated with the structural information (i.e. support and symmetry information) of the part to form a feature vector  $\mathbf{v}_f$ . All feature vectors from all parts of a car model will then be concatenated to form the input vector of SPVAE. The SPVAE can then be trained using the concatenated input vectors.

#### 3.1.2. Structure-aware generative design of 3D shapes in meshes

The trained generative design module can enable structure-aware generative design tasks, such as shape interpolation and random shape generation. As introduced, both the geometric information of parts and inter-part structural information are encoded into the SPVAE latent space.

As shown in Figure 3, when provided with an SPVAE vector obtained from the latent space learned by SPVAE, the SPVAE decoder can transform it into an output vector. This resulting vector can be subdivided into seven distinct vectors, each representing a specific car part. These individual vectors contain both the encoded



**Figure 3.** Overview of the data-driven structure-aware generative design module demonstrated using a car design case. The design module is implemented using SDM-NET (Gao *et al.* 2019b) that consists of PartVAEs and SPVAE.

structural information and a separate vector that encodes the geometry of the corresponding part. Afterward, the vector can be decoded using the decoder of the corresponding PartVAE model, resulting in a feature matrix. This feature matrix can then be further processed with the template cube mesh to create a car part using

the reverse process of extracting the feature matrix as introduced in [Section 3.1.1](#). Separate parts can be combined into one holistic car model with their structural information. It should be noted that independent of the SPVAE, the seven PartVAEs can also be used to generate individual car parts. However, it is not guaranteed to obtain a reasonable car model when combining those parts, as the structural information is not included.

## 3.2. Design evaluation module

As shown in [Figure 2](#), we construct surrogate models to enable a rapid and reliable evaluation of the engineering performance of interest for the design evaluation module. To achieve that, we need to determine the appropriate vectorized design representation (VDR) and the surrogate modeling frameworks. To train a surrogate model, the label data (the engineering performance of interest, such as drag and lift coefficients) can be obtained from computer simulations, for example computational fluid dynamics (CFD) analysis.

### 3.2.1. Vectorized design representation

Research has demonstrated the effectiveness of latent vectors derived from the latent space of a trained data-driven generative design (DDGD) model in design evaluation (Umetani & Bickel 2018; Chen *et al.* 2020). As shown in [Figure 3](#), there are two types of latent vectors that can be obtained from the trained structure-aware generative design module, namely SPVAE vectors and PartVAE vectors. They are readily available once the training process is completed.

In addition to the commonly used latent vectors for VDR, we propose a new method that combines a signed distance field (SDF) technique with a 3D point grid (3DPG) inspired by the work (Badías *et al.* 2019) to generate an alternative form of VDR, namely 3DPG vectors. As illustrated in [Figure 2](#), we first construct a 3DPG filled with evenly distributed points. The dimensions of the 3DPG (**Length** × **Width** × **Height**) are chosen according to the largest bounding box of the 3D models in the dataset so that the 3DPG can include all the 3D models. Once a 3D model is put into the 3DPG, each point will be assigned a value of 1 if it falls inside the 3D model or a value of 0, otherwise. The SDF method is used to determine the status of each point. Signed-distance is the distance of a given point  $\mathbf{p}$  from the boundary of a set, with its sign determined by whether the point is in the set or not. The signed distance of each point can be calculated and transferred to 0 or 1 using [Equation \(1\)](#). Then, all binary values for all points will be concatenated into a 3DPG vector.

$$\phi(\mathbf{p}) = \begin{cases} 1, & \text{if } \mathbf{p} \in \Omega, \text{ i.e., } \text{SDF}(\mathbf{p}) < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The status of points in the 3DPG varies for different 3D models, so each 3D model can be uniquely parameterized into a 3DPG vector with a dimension equal to the number of points in the 3DPG. For example, a car model will be parameterized into a 20,000-dimensional vector if there are 20,000 points in the 3DPG. The SDF method (Badías *et al.* 2019) is effective in handling meshes that are not watertight, but requires the mesh to represent the outer surface (shell) of a 3D object in order to accurately determine the status of individual points in the 3DPG.

However, in our case, we cannot directly apply this method to the final holistic 3D shapes created by combining parts from the structure-aware generative design module. This is because these shapes are essentially a combination of multiple shells of the parts. To ensure that the 3DPG vectors can better capture the geometric information of these 3D shapes, we first convert each combined shape into a single shell shape using ManifoldPlus (Huang, Zhou & Guibas 2020) before sending it to the 3DPG.

### 3.2.2. Surrogate models using AutoML frameworks

There are three main reasons for us to utilize AutoML frameworks in constructing the surrogate models: 1) AutoML routinely outperforms experienced data scientists in identifying optimal supervised learning models (Hutter, Kotthoff & Vanschoren 2019); 2) all of the best-performing AutoML frameworks today rely on some forms of model ensembling techniques that combine predictions from multiple basic models and have long been known to outperform individual models (Dietterich 2000); and 3) it has been demonstrated that AutoML outperforms the strongest gradient-boosting and neural network surrogate models identified through Bayesian optimization in a bicycle design application (Regenwetter *et al.* 2023). In summary, AutoML provides a streamlined workflow for training and deploying models, making it suitable for various machine learning applications. Therefore, we use AutoML to build optimal surrogate models to fully understand the potential of different VDRs in performance evaluation and prediction.

Specifically, we apply two popular AutoML frameworks, Auto-sklearn (Feurer *et al.* 2015) and AutoGluon (Erickson *et al.* 2020). Auto-sklearn has been the winner of numerous AutoML competitions (Guyon *et al.* 2019). It employs efficient multi-fidelity hyperparameter optimization strategies and the combination of numerous models through an ensemble selection strategy. AutoGluon was introduced recently and has been reported to outperform many other alternatives in various applications (Erickson *et al.* 2020). It applies an innovative layer-stack ensembling method. Additionally, AutoGluon incorporates k-fold bagging to minimize the risk of overfitting.

### 3.2.3. The objectives of the design evaluation module

The design evaluation module is to approximate the input–output relationship defined by the computer simulation  $f(\cdot)$  as shown by Equation (2), where  $\mathbf{x}$  represents a VDR of a 3D shape  $X$ , and  $y$  denotes the corresponding performance metric of interest which is used as the ground truth value. Similarly, the surrogate model can be defined by Equation (3), where  $\hat{y}$  is the predicted performance of interest and  $g(\cdot)$  is the approximation of  $f(\cdot)$  implied by the surrogate model. The objective of the surrogate model can be seen as an optimization problem defined by Equation (4).

$$y = f(\mathbf{x}) \quad (2)$$

$$\hat{y} = g(\mathbf{x}) \quad (3)$$

$$\hat{y}^* = \arg \min_{\hat{y}} |y - \hat{y}|, \forall \mathbf{x} \quad (4)$$

The pair data, that is the VDRs and their corresponding engineering performance values, form the training dataset for the surrogate models. We split the training dataset into a train set and a test set. The surrogate models will be trained using the train set only, and the test set serves as unseen data to test the generalizability of the trained surrogate models. In order to conduct a fair comparison of the performance of various types of VDRs in predicting engineering performance, we train an optimal surrogate model for each combination (i.e. one type of VDR and one particular AutoML framework) under identical conditions. For example, the training data and the configurations of the AutoML framework are kept the same. In addition, we use Auto-sklearn (Feurer *et al.* 2015) and AutoGluon (Erickson *et al.* 2020) to examine whether the results would be independent of a particular AutoML framework used.

For a comprehensive comparison, we adopt three evaluation metrics: the mean absolute error (MAE), the root-mean-squared error (RMSE) and the coefficient of determination ( $R^2$ ) as calculated by the following equations, where  $n$  is the total number of observations,  $y_i$  is the actual value of the observation  $i$ ,  $\hat{y}_i$  is the predicted value of the observation  $i$  and  $\bar{y}$  is the mean of the actual values. We also perform the paired t-test on the absolute errors (AE) values ( $|y_i - \hat{y}_i|$ ) of two groups to test if there is a statistically significant difference between the prediction accuracy when using different VDRs. The paired t-test is not performed on squared errors (SE) since they are essentially squared AE, and the test is not conducted on the  $R^2$  as it represents a statistics of a group of data.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (6)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (7)$$

## 4. Implementation details and results

In this section, we present the implementation details and results of the structure-aware generative design module and the design evaluation module. All experiments were run on a Linux workstation with a TITAN RTX GPU and a 20-core Intel Xeon Silver 4114 CPU.

### 4.1. Design cases and datasets

We demonstrated the proposed approach and conducted a comparative study in two design cases: the car and aircraft designs. The datasets used for the

**Table 1.** Summary of the datasets for the two design cases: the car and aircraft designs

			Car	Aircraft
Structure-aware generative design module	3D model data	Source	Gao et al. (2019) derived from ShapeNet (Chang et al. 2015) and ModelNet (Wu et al. 2015)	Gao et al. (2019) derived from ShapeNet (Chang et al. 2015) and ModelNet (Wu et al. 2015)
		Number	1824	2690
	Filter		<ul style="list-style-type: none"> <li>Automatic filter: seven parts (i.e., one body, two mirrors, and four wheels)</li> <li>Manual filter: regular passenger car models</li> </ul>	<ul style="list-style-type: none"> <li>Automatic filter: eight parts (i.e., one fuselage, two wings, three tails, and two engines)</li> </ul>
	Number of training data		1161	1597
Design evaluation module	Engineering performance data	Source	Song et al. (2023) developed from ShapeNet (Chang et al. 2015) using data augmentation and OpenFoam (Jasak et al. 2007)	Edwards et al. (2021) developed from ShapeNet (Chang et al. 2015) using OpenFoam (Jasak et al. 2007)
		Number	9070	4045
	Filter		<ul style="list-style-type: none"> <li>Overlapping with the training data of the design module</li> <li>Value range (0, 1)</li> </ul>	<ul style="list-style-type: none"> <li>Overlapping with the training data of the design module</li> <li>Value range (0, 1)</li> </ul>
	Number of training data		439	1047

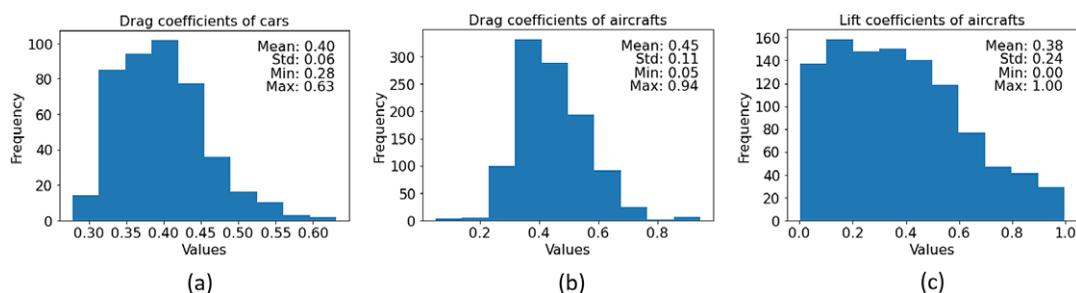
structure-aware generative design module and the design evaluation module are summarized in [Table 1](#).

**4.1.1. Training data for the structure-aware generative design module**

For the training of the structure-aware generative design module, we collected 1824 car and 2690 aircraft mesh models from Gao *et al.* (2019b), which have been divided into parts using a semantic segmentation approach (Yi *et al.* 2016). These data are derived from ShapeNet (Chang *et al.* 2015) and ModelNet (Wu *et al.* 2015). We developed an algorithm to automatically select models that have all seven parts (i.e. one body, two mirrors and four wheels) for the car models and all eight parts (i.e. one fuselage, two wings, three tails and two engines) for the aircraft models. Also, for car models, since we focused on regular passenger car models (e.g. sedans, SUVs), we manually excluded the other car types, including buses, Formula One and trucks. This gave us a total of 1161 car models and 1597 aircraft models, which were used as training data for the structure-aware generative design module.

**4.1.2. Training data for the design evaluation module**

We collected the engineering performance data of car models and aircraft models by leveraging two open-sourced datasets: 9070 car models labeled by drag coefficients (Song *et al.* 2023) and 4045 aircraft models with drag and lift coefficients (Edwards, Addala & Ahmed 2021). The 3D model data of the two datasets are both



**Figure 4.** Histograms of (a) drag coefficients of car models, (b) drag coefficients of aircraft models and (c) lift coefficients of aircraft models with the mean, std, min and max values.

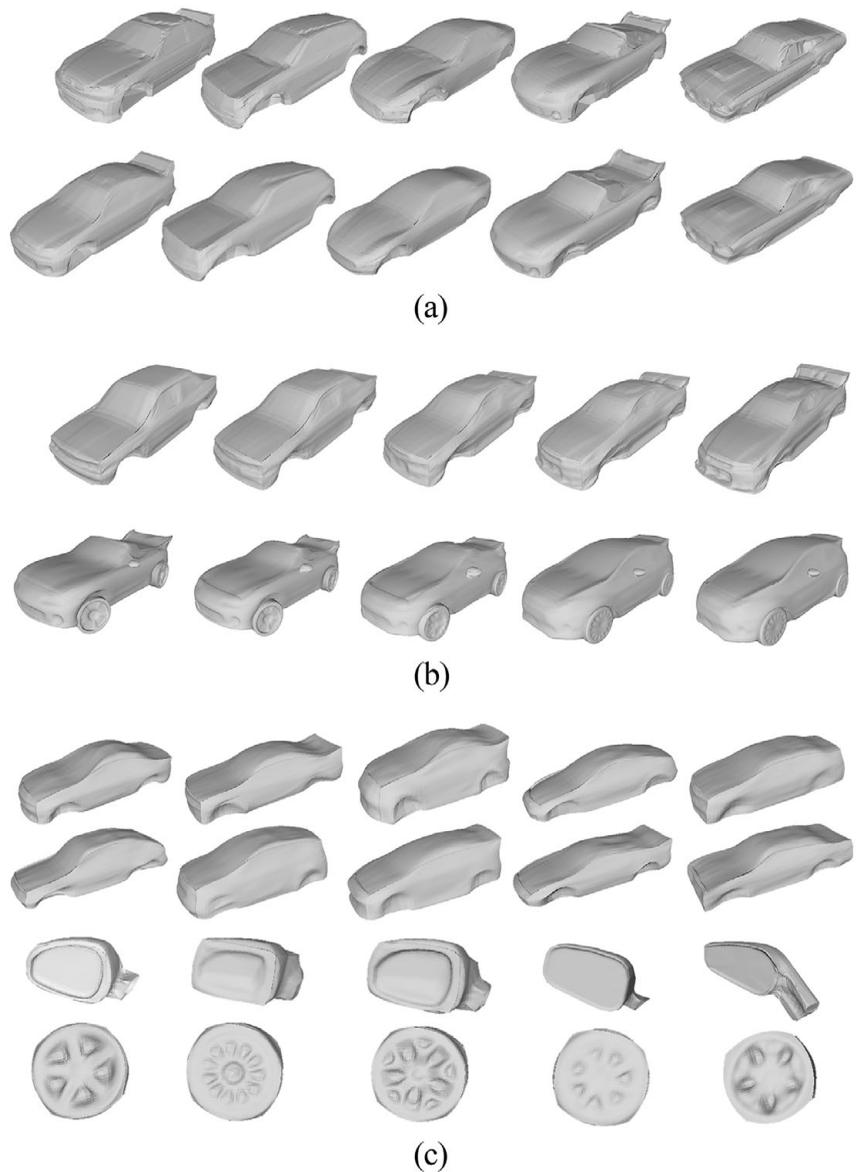
derived from ShapeNet (Chang *et al.* 2015). The corresponding performance values are obtained from the computational fluid dynamics (CFD) simulation tool OpenFOAM (Jasak 2009).

To match the labeled 3D models with the training data used in the structure-aware generative design module, we selected the overlapping models between the two datasets for each case study. Furthermore, we only selected models with drag or lift coefficients within the range of 0 to 1 to ensure data quality and reliability. This gave us a total of 439 car models with corresponding drag coefficients and 1047 aircraft models with drag and lift coefficients. These data were used as training data for the design evaluation module. Figure 4 shows the histograms of the drag coefficients for the car models, and the drag coefficients and lift coefficients for the aircraft models, along with their descriptive statistics in the legend of Figure 4.

## 4.2. Structure-aware generative design module

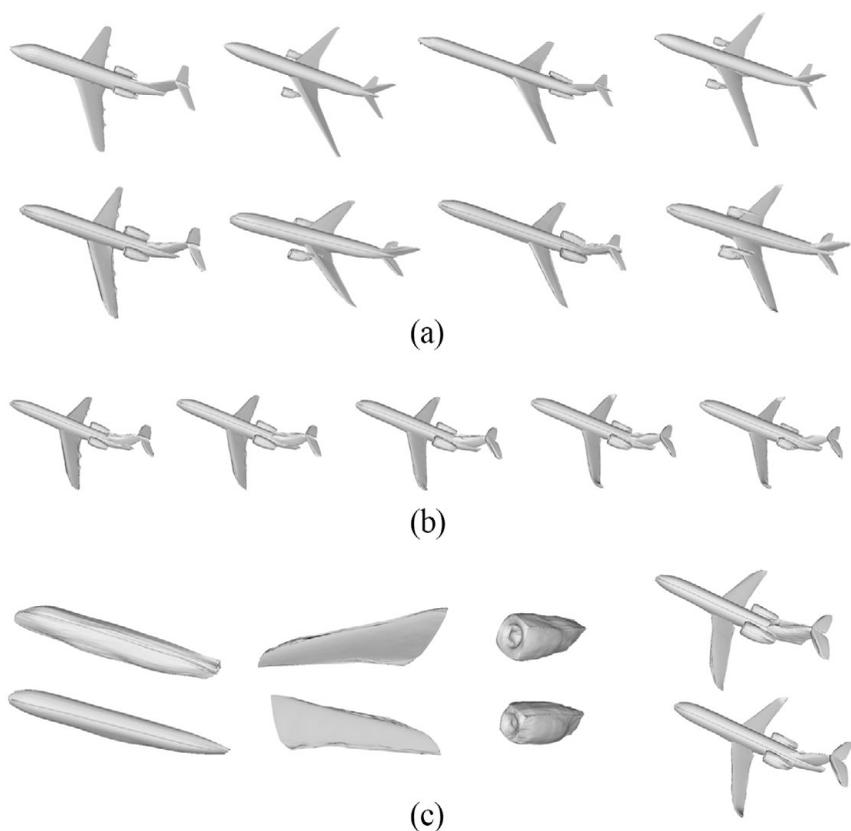
We used the same strategies for both car models (1161) and aircraft models (1597) to train the structure-aware generative design module as detailed below. We randomly split the training dataset into train data (75%) and test data (25%). 64 and 128 were chosen for the dimensionality of the latent spaces of PartVAEs and the SPVAE, respectively, because they yield the lowest reconstruction errors, as shown by Gao *et al.* (2019b). To ensure effective training, we implemented the two-stage training strategy discussed in Section 3.1.1. This involved initially training PartVAEs, followed by training the SPVAE until the networks converged. Throughout the training process, we monitored and evaluated all associated training loss terms (i.e. reconstruction loss and Kullback–Leibler (KL) divergence loss for both train and test data). The convergence of these loss terms indicated that the networks were successfully trained. In the Appendix, we document the training loss values throughout the training process in Figure A1. The training for the car models took approximately 72 hours (i.e. PartVAEs: 10000 epochs; SPVAE: 20000 epochs), while it took approximately 120 hours (i.e. PartVAEs: 5000 epochs; SPVAE: 10000 epochs) to complete the training for the aircraft models.

In Figure 5(a), several reconstructed car body models are displayed. The first row presents the original models, while the second row displays the corresponding reconstructed models. Figure 5(b) shows a few car bodies and



**Figure 5.** Examples of the generated shapes for car models. (a) Reconstruction of car bodies. (b) Shape interpolation of car bodies and merged car models. (c) Random generation of separate parts.

combined car models by linearly interpolating the shapes of the first and last columns. The in-between columns from the second to the fourth column are linearly interpolated shapes. We can observe a gradual transition of the geometry from the first column to the last column. In addition, SPVAE vectors can be randomly sampled from the learned latent space to generate random shapes (car bodies, mirrors and wheels) as shown in Figure 5(c). Similarly, the results for aircraft models are presented in Figure 6. Figure 6(a) exhibits several

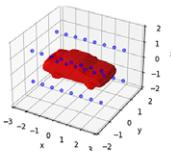
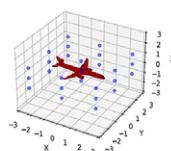


**Figure 6.** Examples of the generated shapes for aircraft models. (a) Reconstruction of aircraft models. (b) Shape interpolation of merged aircraft models. (c) Random generation of separate parts (fuselage, wings and engines) and merged aircraft models.

instances using shape reconstruction, where the top row shows the original models, and the bottom row shows their corresponding reconstructed versions. Figure 6(b) illustrates the combined aircraft models created by linearly interpolating the shapes of the first and last columns. Notably, the interpolation effectively captures the transformation of the wings, progressing from a completely flat configuration to a slightly curved shape towards the wingtips. Figure 6(c) exhibits examples of randomly generated aircraft parts: fuselage, wings and engines, from left to right, as well as combined aircraft models.

Theoretically, we can sample as many latent vectors as possible from the latent space for random shape generation. Shape interpolation can be performed between every pair of car models with any number of in-between interpolation shapes. Thus, we can generate thousands of unseen designs, and the generated designs look reasonable in terms of visual appearance and have great geometry details. The results also indicate that the latent spaces of PartVAEs and the SPVAE are trained well which can serve as VDRs for the design evaluation module.

**Table 2.** Summary of the vectorized design representations (VDRs) for the car and aircraft models

		Car	Aircraft
Latent vectors (dimension)	PartVAE	<ul style="list-style-type: none"> <li>All_parts vectors (448)</li> <li>Body vectors (64)</li> </ul>	<ul style="list-style-type: none"> <li>Wings vectors (128)</li> <li>All_parts vectors (512)</li> </ul>
	SPVAE	SPVAE vectors (128)	SPVAE vectors (128)
	Time to obtain a latent vector	<ul style="list-style-type: none"> <li>≈ 0s (inference time)</li> <li>≈ 223s (training time per car model)</li> </ul>	<ul style="list-style-type: none"> <li>≈ 0s (inference time)</li> <li>≈ 271s (training time per aircraft model)</li> </ul>
3DPG vectors	Largest bounding box in $L \times W \times H$	0.86 × 0.37 × 0.29	0.91 × 0.88 × 0.30
	Scale factor of 3D models	× 5	× 5
	Dimension of the 3DPG in $L \times W \times H$	5 × 2 × 2	5 × 5 × 2
	Number of points in the 3DPG in $L \times W \times H$	<ul style="list-style-type: none"> <li>35 × 12 × 12 = 5040</li> <li>40 × 16 × 16 = 10240</li> <li>50 × 20 × 20 = 20000</li> </ul>	<ul style="list-style-type: none"> <li>35 × 12 × 12 = 5040</li> <li>40 × 16 × 16 = 10240</li> <li>50 × 20 × 20 = 20000</li> </ul>
	Illustration of the SDF method		
	Time to obtain a 3DPG vector	≈ 35s	≈ 35s

### 4.3. Design evaluation module

#### 4.3.1. Latent vectors and 3DPG vectors for the VDR

To evaluate and determine the most effective VDR in predicting the engineering performance investigated, we prepared two representative VDRs: latent vectors (the commonly used VDR; and are obtained from the training process of the DDGD models) and 3DPG vectors (the proposed VDR; and require additional steps to vectorize the generated designs) for the training of the design evaluation module as introduced in Section 3.2.1. The configurations of these VDRs are summarized in Table 2.

Regarding the latent vectors (as depicted in Figure 3), we utilized two types of VAE vectors. First, we employed the 128-dimensional SPVAE vectors for both car and aircraft models. Second, we concatenated the PartVAE vectors from all parts, resulting in the creation of all\_parts vectors. For car models, the all\_parts vectors have a dimension of  $64 \times 7 = 448$ , while for aircraft models, the dimension is  $64 \times 8 = 512$ . The major difference between all\_parts vectors and SPVAE vectors

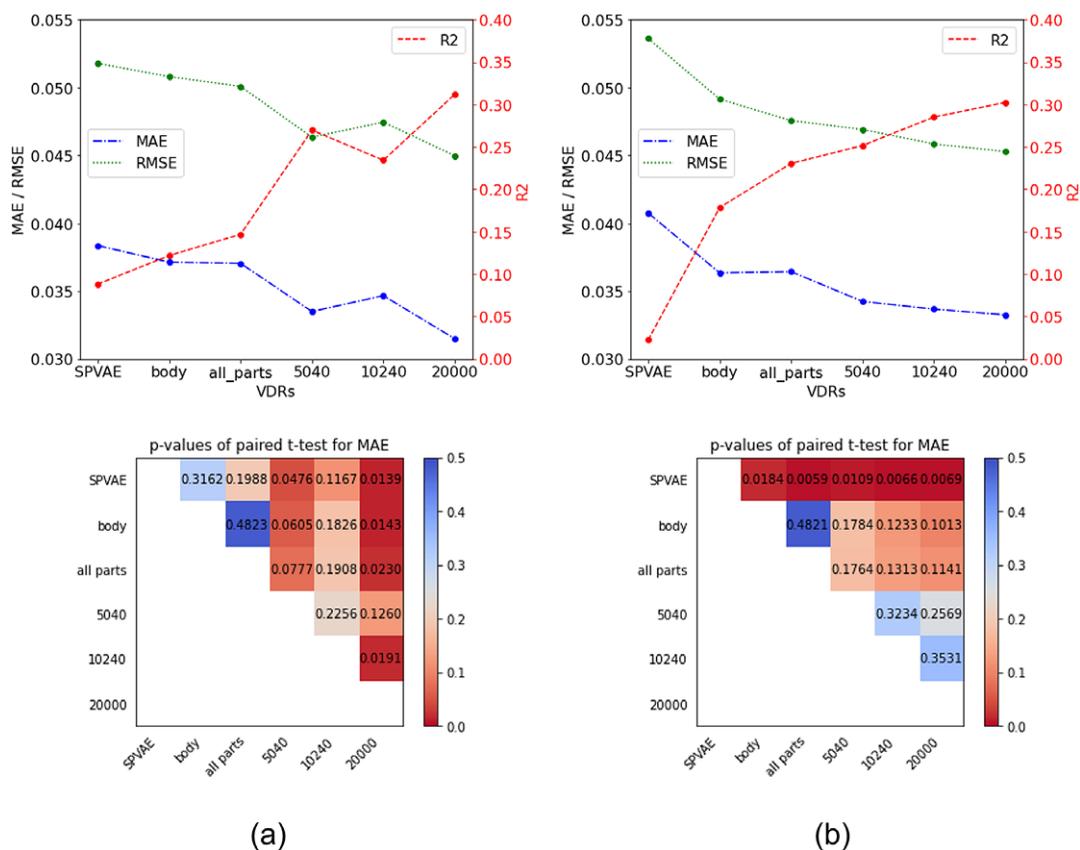
is that all\_parts vectors encode geometric information only, while the SPVAE vectors encode both geometric and structural information. Additionally, we took into account the significance of the car body in calculating the drag coefficient, as well as the significance of the wings (airfoils) in determining the lift coefficient (Fairman 1996). Thus, we specifically chose the body vectors (64-dimensional) for the car models and the wing vectors (two wings,  $64 \times 2 = 128$ -dimensional) for the lift prediction of the aircraft models.

For 3DPG vectors, we found that the largest bounding box dimensions ( $L \times W \times H$ ) for car models to be  $0.86 \times 0.37 \times 0.29$ , while for aircraft models, it was found to be  $0.91 \times 0.88 \times 0.30$ . The 3D models are all normalized (Chang *et al.* 2015) and the values of the bounding boxes in the mesh files do not have a unit because they are dimensionless, but they are proportional to the size of actual 3D models. To ensure the inclusion of all models in the training data for the design evaluation module, we set the 3DPG dimensions to  $5 \times 2 \times 2$  for car models and  $5 \times 5 \times 2$  for aircraft models for convenience. They can be set to different values as long as the ratio  $L/W/H$  is maintained. Each car or aircraft model can then be scaled up by a factor of 5 to better fit into the corresponding 3DPG.

After setting up the 3DPG, we can obtain a 3DPG vector as outlined in Section 3.2.1 by utilizing the SDF method. Although only the 20,000-dimensional vector was shown to be effective in Badiás *et al.* (2019), we tested three different configurations for 3DPG vectors: 1)  $35 \times 12 \times 12 = 5040$ , 2)  $40 \times 16 \times 16 = 10240$  and 3)  $50 \times 20 \times 20 = 20000$ . The time cost for parameterization remains constant at approximately 35 seconds, regardless of any changes in the dimension of the VDRs or the specific car or aircraft models being used. The reason for this is that the SDF method involves performing a virtual laser scan of the input 3D model, and the computational time is dominated by the resolution of the scan. Although it appears that it took no time to obtain the latent vectors from a trained structure-aware generative design model, the training itself can be time-consuming. It took approximately 223 seconds per car model and 271 seconds per aircraft model according to the training time introduced in Section 4.2 while obtaining 3DPG vectors does not involve any training process.

#### 4.3.2. Prediction results using different VDRs and surrogate models

We ended up with a total of 439 car models, each having an associated drag coefficient. These car models were represented by six types of VDRs, including 128-dimensional SPVAE vectors, 448-dimensional all\_parts vectors, 64-dimensional body vectors, as well as three types of 3DPG vectors with dimensions of 5040, 10240 and 20000, respectively. Similarly, for the 1047 aircraft models with associated drag and lift coefficients, we had 128-dimensional SPVAE vectors, 512-dimensional all\_parts vectors, 128-dimensional wing vectors and the same types of 3DPG vectors as car models. By utilizing each type of VDR along with the corresponding engineering performance label data, we randomly divided the training dataset into two parts: a train set (80%) and a test set (20%). Importantly, this division remained the same for all VDRs within a particular design case to make a fair and consistent comparison. We then trained an optimal surrogate model using Auto-sklearn (Feurer *et al.* 2015) and AutoGluon (Erickson *et al.* 2020).



**Figure 7.** The comparison of prediction accuracy for the test set data of car models using different VDRs with drag coefficients with three evaluation metrics: MAE (↓), RMSE (↓) and the R<sup>2</sup> (↓) using (a) Auto-sklearn or (b) AutoGluon. The p-values resulting from the paired t-test for AE values are used to show the statistical difference.

These AutoML models have the capability to automatically reserve a portion of the train set data as validation data. We trained all AutoML models by minimizing RMSE on the validation data for optimal surrogate models. Given our focus on understanding the generalizability of the trained surrogate models to unseen data, we primarily present the prediction results specifically for the test data. We have made the dataset, code and all results for the design evaluation module open-source for the purpose of reproducibility and for further research interests.<sup>1</sup>

*Results for predicting drag coefficients of the car models.* Several insights can be drawn based on the results of MAE, RMSE and R<sup>2</sup> presented in Figure 7.<sup>2</sup>

<sup>1</sup>[https://github.com/Xingang-Li/structure\\_aware\\_design\\_evaluation](https://github.com/Xingang-Li/structure_aware_design_evaluation)

<sup>2</sup>The root-mean-squared error (RMSE) and R<sup>2</sup> (coefficient of determination) are both statistics for group data. Consequently, the error bars are not applicable to them. Additionally, the error bars are not included for the mean absolute error (MAE) because the variance of absolute error (AE) across the entire dataset is not relevant in this analysis due to the use of the paired t-test rather than the independent two-sample t-test. This is because the different VDRs were obtained from the same data group (i.e. the test dataset), so the datasets for the t-test are not independent.

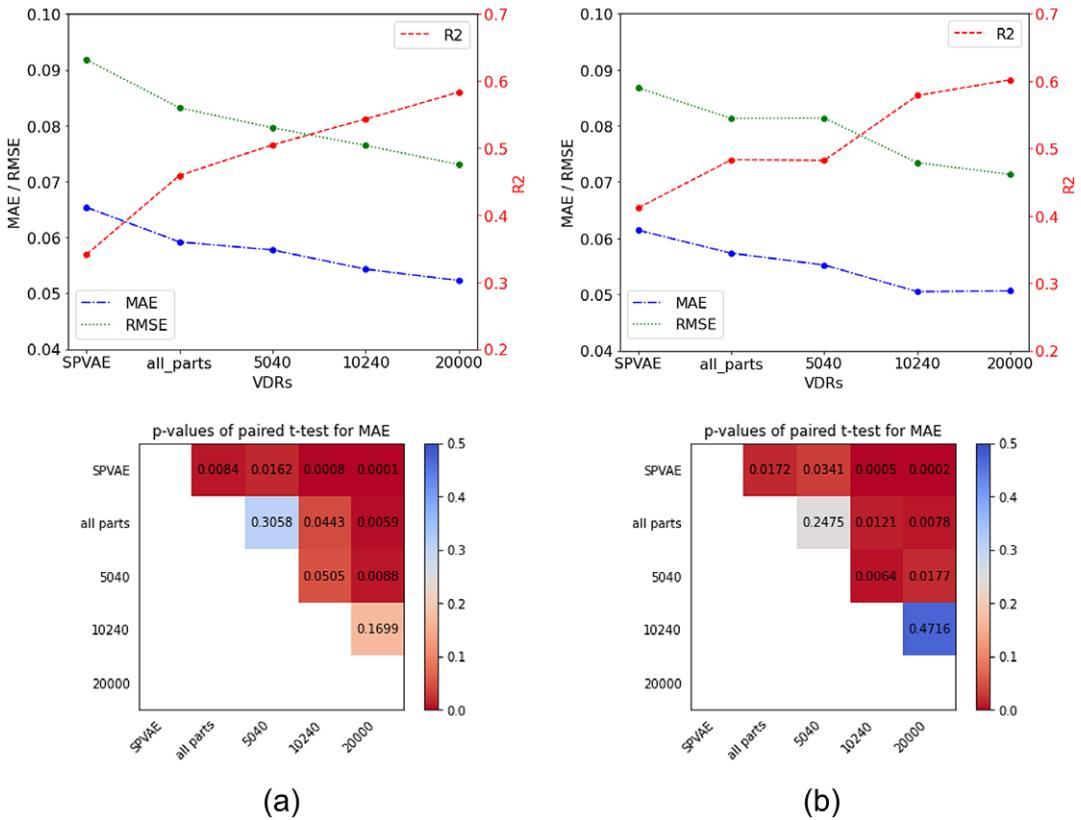
Regardless of the AutoML frameworks used, the 3DPG vectors consistently exhibit higher accuracy than the latent vectors. SPVAE vectors achieve the lowest accuracy, while the 20000-dimensional 3DPG vectors achieve the highest accuracy among all alternative VDRs. For 3DPG vectors, the mean accuracy increases in general with higher dimensions. The best combination of the VDR and AutoML framework is observed to be the 20000-dimensional 3DPG vectors and Auto-sklearn, resulting in an  $R^2$  value of 0.312. On the other hand, the worst combination is observed to be the SPVAE vectors and AutoGluon, resulting in an  $R^2$  value of 0.021.

The heatmaps in Figure 7 show the p-values associated with the paired t-test performed on the AE values. The results indicate that regardless of the specific AutoML framework utilized, the performance of the SPVAE vectors is consistently inferior to that of the 20000-dimensional 3DPG vectors, and this difference is statistically significant, that is  $p = 0.0139$  when using Auto-sklearn and  $p = 0.0069$  when using AutoGluon. Regarding the latent vectors, no significant differences are observed between the SPVAE vectors, the body vectors and the all\_parts vectors. The only exception is the difference between the SPVAE vectors and the all\_parts vectors ( $p = 0.0059$ ) in AutoGluon. Similarly, for 3DPG vectors, while the mean values of all three metrics show an increasing trend, no significant differences are observed between different VDRs.

*Results for predicting drag coefficients of the aircraft models.* Based on the results of MAE, RMSE and  $R^2$  presented in Figure 8, we can get several insights as follows. Similar to the findings in car models, the 3DPG vectors consistently demonstrate superior accuracy compared to the latent vectors across both AutoML frameworks employed. Among all the alternative VDRs, it is also observed that SPVAE vectors exhibit the lowest accuracy, while the 20000-dimensional 3DPG vectors achieve the highest accuracy. There is a notable trend of increasing accuracy as the dimension of the 3DPG vectors increases. The most favorable combination of the VDR and AutoML framework is observed with the 20000-dimensional 3DPG vectors with AutoGluon, yielding an  $R^2$  value of 0.602. Conversely, the least favorable combination is observed with the SPVAE vectors and Auto-sklearn, resulting in an  $R^2$  value of 0.341.

We also conducted a paired t-test on the AE values and the resulting p-values are visualized in the heatmaps in Figure 8. The heatmaps indicate significant differences ( $p < 0.05$ ) between most pairs of VDRs, except for three cases: 1) all\_parts vectors and 5040-dimensional 3DPG vectors with both Auto-sklearn and AutoGluon, 2) 5040 and 10240-dimensional 3DPG vectors (where the p-value slightly exceeds 0.05 with Auto-sklearn but was less than 0.05 with AutoGluon) and 3) 10240 and 20000-dimensional 3DPG vectors with both Auto-sklearn and AutoGluon.

*Results for predicting lift coefficients of the aircraft models.* As the results of MAE, RMSE and  $R^2$  shown in Figure 9, regardless of the AutoML frameworks used, the SPVAE vectors achieve the lowest accuracy while all\_parts and wing vectors achieve the top two highest accuracies among all the alternative VDRs. However, there is a discrepancy between the performance of the all\_parts and wing vectors using the two AutoML frameworks. 3DPG vectors perform poorly, and there is no noticeable trend of increasing accuracy with higher-dimensional 3DPG vectors, as previously observed in the prediction of drag coefficients. The best combination of the VDR and AutoML framework is observed with the wing vectors and



**Figure 8.** The comparison of prediction accuracy for the test set data of aircraft models using different VDRs with drag coefficients with three evaluation metrics: MAE ( $\downarrow$ ), RMSE ( $\downarrow$ ) and the  $R^2$  ( $\downarrow$ ) using (a) Auto-sklearn or (b) AutoGluon. The p-values resulting from the paired t-test for AE values are used to show the statistical difference.

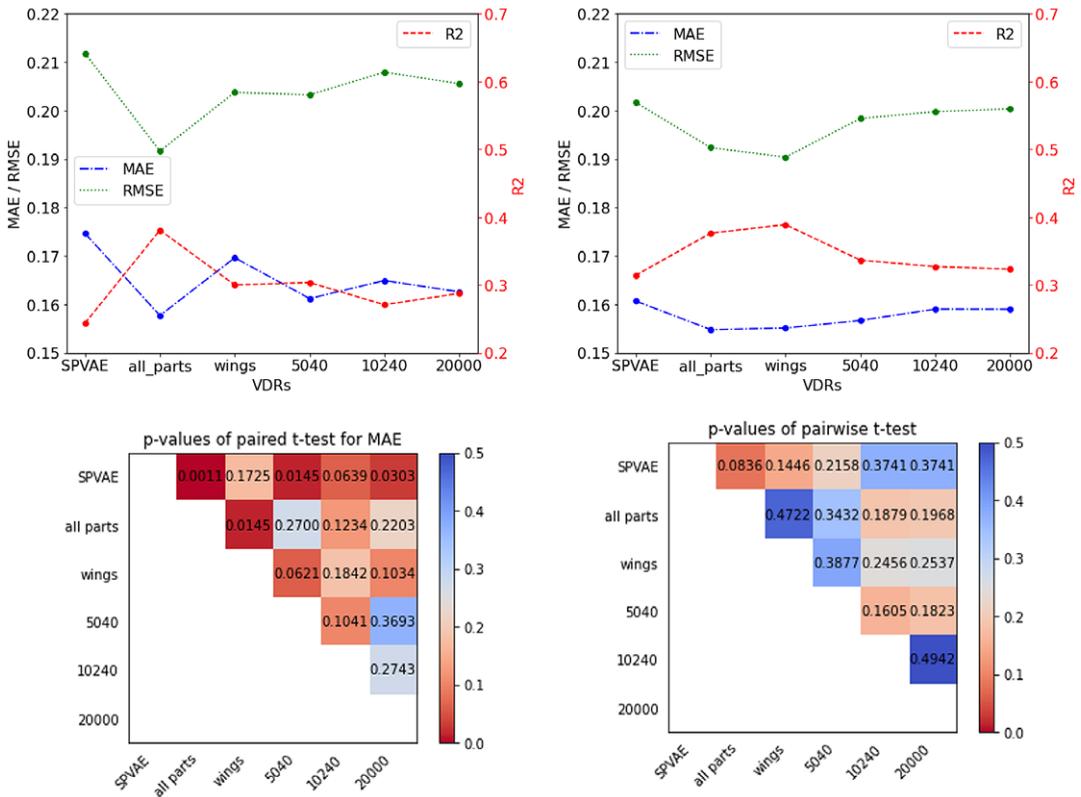
AutoGluon, resulting in an  $R^2$  value of 0.389. The worst combination is observed with the SPVAE vectors and Auto-sklearn, resulting in an  $R^2$  value of 0.244.

For the t-test conducted on the AE values, the results of the p-values are shown in Figure 9. The heatmaps show that apart from four specific pairs in Auto-sklearn: 1) SPVAE vectors and all\_parts vectors, 2) SPVAE vectors and 5040-dimensional 3DPG vectors, 3) SPVAE vectors and 20000-dimensional 3DPG vectors and all\_parts vectors and wing vectors, there is no significant difference ( $p < 0.05$ ) observed between all combinations of VDRs when using the two AutoML frameworks.

We summarize the results for the best combination of the VDR and AutoML framework for the surrogate models and the corresponding prediction performance metrics of car and aircraft models in Table 3.

### 5. Discussion

In this section, we provide a comprehensive analysis of the results obtained from the structure-aware generative design module (Section 5.1) and the design



**Figure 9.** The comparison of prediction accuracy for the test set data of aircraft models using different VDRs with lift coefficients with three evaluation metrics: MAE (↓), RMSE (↓) and the  $R^2$  (↓) using (a) Auto-sklearn or (b) AutoGluon. The p-values resulting from the paired t-test for AE values are used to show the statistical difference.

**Table 3.** The best combination of the VDR and AutoML framework for the surrogate models of car and aircraft models

		Car	Aircraft	
		Drag	Drag	Lift
VDR		20000-dim 3DPG vectors	20000-dim 3DPG vectors	Latent vectors of wings
AutoML Framework		Auto-sklearn	AutoGluon	AutoGluon
Performance Metric	$R^2$	0.312	0.602	0.389
	MAE	0.032	0.051	0.155
	RMSE	0.045	0.071	0.190

evaluation module (Section 5.2) and discuss the limitations and potential future research directions.

### 5.1. Structure-aware generative design module

Structure-aware generative design is an emerging and relatively unexplored field that holds promise in addressing the challenges of systems design problems using data-driven generative design (DDGD) methods. Unlike the generative design of monolithic shapes that prioritize optimizing overall system performance, the structure-aware generative design focuses on capturing and integrating the details of parts' geometry and structure. By considering the structural characteristics of individual parts, the structure-aware generative design enables a more comprehensive understanding of the system. It can also facilitate the exploration of various design alternatives and iterations, empowering engineers to make well-informed decisions regarding part geometries and the structural relations between parts. This opens opportunities for the discovery of novel and optimized designs that may have been overlooked using traditional generative models, especially in the early stages of design.

In our study, we implemented SDM-NET (Gao *et al.* 2019b), a data-driven structure-aware generative model, as the structure-aware generative design module. While it holds significant potential to help designers effectively explore the design space, there is a major limitation in the current methodology. The validity of the generated designs in terms of their structural integrity heavily relies on visual inspection, and there is no quantitative method available to assess. While most of the generated designs are deemed acceptable, some may have unattached parts despite the structural information learned. One possible approach to ensure structural validity is to employ optimization techniques (Gao *et al.* 2019b), but it is crucial to develop a quantitative and automatic method to assess the structural validity of the generated designs, for example rating the designs in terms of their structural integrity and surface quality.

### 5.2. Design evaluation module

#### 5.2.1. Drag prediction for car and aircraft models

The SPVAE vectors demonstrate the least accuracy, whereas the all\_parts vectors consistently enhance the predictive accuracy in both case studies. In the context of car models, the body vectors also exhibit superior performance compared to the SPVAE vectors, as indicated by lower MAE and RMSE values and higher  $R^2$ . This superiority is further supported by a paired t-test using AutoGluon ( $p = 0.0184$ ), as depicted in Figure 7. Both all\_parts vectors and SPVAE vectors encode geometric information for all components of the 3D shapes. However, the SPVAE vectors also include structural information such as support and symmetry. This structural information is crucial to generate designs that account for the underlying structures of 3D shapes, as illustrated in Figure 3. Nevertheless, when the structural information is irrelevant to the engineering performance, such as the drag here, it can have a detrimental impact on the suitability of the SPVAE vectors as VDRs for surrogate models. In such cases, using latent vectors that only capture geometric information most relevant to the engineering performance of interest, such as the all\_parts vectors or car body vectors, can be more advantageous for surrogate models. On the other hand, in situations where the structural information plays a significant role in

engineering performance, such as in determining the maximum allowable load for a bike frame design problem, using VDRs that incorporate structural information may have advantages over the VDRs that only capture geometric information.

Likewise, by capturing the geometric information of all components of 3D shapes, 3DPG vectors possess significant potential to serve as more suitable VDRs for predicting drag coefficients. There is a general trend of improved accuracy, reflected in lower MAE and RMSE values, and higher  $R^2$  values, as the dimensionality of 3DPG vectors increases. Specifically, the 20000-dimensional 3DPG vectors exhibit the highest level of accuracy. Increasing the dimensionality of the 3DPG vectors implies using more points to parameterize a design utilizing the 3D point grid, as described in Table 2. This augmentation in the number of points enables a more comprehensive capture of the geometric details of 3D shapes, thereby enhancing the prediction of the drag coefficient, which is closely influenced by the overall geometry of 3D shapes. But, it should be noted that 3DPG vectors also encode the positional information of various components due to the signed distance field, whereas all\_parts vectors (concatenation of part vectors) do not contain such information. This distinction in encoding positional information could be one of the reasons why 3DPG vectors generally exhibit better performance compared to all\_parts vectors in predicting drags in both design cases.

If considering statistical significance, however, augmenting the dimensionality of 3DPG vectors does not necessarily lead to a significant improvement in prediction accuracy as evident: 1) In the case of car models, there are generally no significant differences in the prediction accuracy between the 3DPG vectors; 2) similarly, for the aircraft models, there are no significant differences between 10240- and 20000-dimensional 3DPG vectors. The lack of significant differences could be attributed to the relatively small size of the datasets. Specifically, the dataset for car models (439) is approximately 60% smaller than the dataset for aircraft models (1047). This discrepancy in dataset size, as depicted in Figure 1, causes a less number of pairs of VDRs with significant differences in the analysis of the car models compared to the aircraft models. In addition, the curse of dimensionality could be another reason affecting the performance of 3DPG vectors with higher dimensionality. Moreover, it is important to acknowledge that in order to achieve a similar or higher level of prediction accuracy compared to latent vectors (such as SPVAE vectors or all\_parts vectors), 3DPG vectors should have a minimum of 5040 dimensions, as demonstrated in our experiments. We conducted tests using an extreme case of 128 dimensions (same as the SPVAE vectors), which resulted in a significant decrease in prediction accuracy and even yielded negative  $R^2$  values in the case study of car design.

### 5.2.2. Lift prediction for aircraft models

SPVAE vectors consistently exhibit the lowest prediction accuracy in lift prediction, similar to their performance in drag prediction regardless of the AutoML frameworks employed. Although there are variations in the outcomes produced by Auto-sklearn and AutoGluon, the use of the latent vectors of wings (i.e. wing vectors) in conjunction with AutoGluon demonstrates the highest accuracy in terms of a higher  $R^2$  value of 0.389.

In a similar manner to the drag prediction, it is evident that there are no significant differences among 3DPG vectors with varying dimensions. Additionally, there are no significant differences between all\_parts vectors and 3DPG

vectors, as supported by the p-values (all are greater than 0.1) shown in [Figure 9](#). However, when considering the MAE, RMSE and  $R^2$  metrics, the mean values indicate a slightly decreasing trend in prediction accuracy for 3DPG vectors as the dimensionality increases, and the performance of all\_parts vectors surpasses that of 3DPG vectors, which deviates from the drag prediction scenario. This discrepancy can be attributed to two major factors: the encoded geometric information and the curse of dimensionality. The drag coefficient is affected by all components of the 3D shapes, while the lift coefficient is mainly determined by the wings (NASA n.d.; Fairman 1996). Although the 3DPG vectors, which encode geometric information of all components, can be advantageous for drag prediction, they pose challenges when predicting lift because they include a considerable amount of irrelevant geometric information from non-wing parts. Consequently, the advantage of increasing the dimensionality of 3DPG vectors to capture more geometric details is diminished by the curse of dimensionality. As a result of this phenomenon, the performance of the 3DPG vectors in lift prediction even decreases to a level comparable to that of the SPVAE vectors, and even worse than that of the all\_parts vectors (512-dimensional), as shown in [Figure 9](#).

### 5.2.3. Summary of the two design cases

Important insights can be derived from the two design cases involving drag and lift prediction. While latent vectors have frequently been employed as VDRs in surrogate models, they may not be the most appropriate option when encoded information includes a mixture of relevant and irrelevant information for the engineering performance of interest. Specifically, when utilizing structure-aware generated design, caution should be exercised when employing latent vectors that encode both geometry and structural information (such as SPVAE vectors in our case) that are often readily obtainable from the training. Instead, the underlying physics shall be examined to determine what geometric information would contribute most and whether the structural information is relevant to the engineering performance to be predicted.

For 3DPG vectors, increasing the dimensionality does not necessarily improve the predictive performance of surrogate models. This is especially true when the vectors contain more noise, that is the information irrelevant to engineering performance, such as in the lift prediction.

*Limitations.* 1) Despite trying different combinations of AutoML frameworks and VDRs, the resulting surrogate models achieved modest  $R^2$  values of 0.312, 0.602 and 0.389 for drag prediction in cars and aircraft, and lift prediction in aircraft, respectively. These values fall short of high predictive accuracy if referring to the criterion of  $R^2 = 0.67$  (Henseler, Ringle & Sinkovics 2009). The primary reason is the limited availability of data. This has been evident by the difference between the  $R^2$  value of drag prediction in aircraft (0.602) and that in cars (0.312) because there are more data points for the aircraft models compared to the car models (1047 versus 439). 2) The SDF method, although effective in capturing the geometric information of 3D shapes, suffers from considerable computational cost, taking approximately 35 seconds to generate each high-quality 3DPG vector. We conducted experiments to explore the impact of reduced resolution in laser scans as outlined in [Section 4.3.1](#). In particular, utilizing a lower resolution could reduce the processing time to 3 seconds. However, this reduction in resolution also led to a significant drop in prediction accuracy, with the  $R^2$  value declining by up to 41%.

To ensure data quality, computational cost poses a practical limitation for its application in interactive generative design. Therefore, it is valuable to investigate different implementations that can offer faster solutions to encode 3D shapes.

## 6. Conclusion and future work

Data-driven generative design (DDGD) methods can effectively support design ideation and 3D shape synthesis. With the recent advances in structure-aware DDGD, this study is motivated to answer the following question: What are the appropriate vectorized design representations (VDRs) for fast performance evaluation of the 3D shapes generated by the structure-aware DDGD method? To answer this question, we first developed a structure-aware generative design module based on SDM-NET (Gao *et al.* 2019b) that can generate various new 3D shapes taking into account the interconnections between parts. Then, we realized the fast design evaluation module by constructing surrogate models using AutoML frameworks. Based on the integrated framework combining structure-aware DDGD for design generation and surrogate modeling for design evaluation, we tested different types of VDR, including latent vectors (i.e. PartVAE vectors and SPVAE vectors) obtained from the generative design module and the 3D point grid (3DPG) vectors.

We observed that SPVAE vectors directly from the structure-aware generative design module achieved the worst prediction accuracy regardless of the design cases and AutoML frameworks used. The results indicate that while latent vectors are commonly used as VDRs for surrogate models, they may not be suitable when the encoded information contains factors (e.g. structural information) that are of little relevance to the engineering performance of interest. Therefore, it is crucial to consider the physics underlying the engineering performance investigated and select VDRs that incorporate the most relevant information to improve the prediction accuracy of surrogate models. The results could have a broader impact on industry professionals because the use of appropriate VDR can lead to the improved predictive performance of design automation tools. A better prediction of engineering performance will also help designers make informed decisions in the early design stage when interacting with AI, facing a large number of design alternatives generated, thus potentially shortening the overall design cycle and reducing the development time.

The limitations presented in [Section 5](#) help us identify some future research directions in the development of more practical design applications for structure-aware generative design. First, the structural integrity of the generated designs is assessed through visual inspection, without a quantitative method. Developing an automatic and quantitative evaluation method for structural validity will greatly benefit future applications of structure-aware generative design. Second, we demonstrate our method in scenarios where the engineering performance of a product is closely related to its shape geometry. Interestingly, we observed that the inclusion of structural information could have a detrimental effect on the suitability of SPVAE vectors as VDRs for surrogate modeling. More research and investigation are necessary to explore design cases in which structural information can significantly impact engineering performance, so we can test whether VDRs that incorporate both structural and geometric information may offer advantages over those solely capturing geometric information. Furthermore, to generalize the findings of this study, it is important to test more design cases or collect additional data for the two design cases.

This would allow a deeper understanding and a wider application of the conclusions drawn from the study.

## Acknowledgments

The authors gratefully acknowledge the financial support from the NSF through the grant DUE-2207408.

## References

- ARC. n.d. The effect of aerodynamic drag on fuel economy — Auto Research Center, online document (downloadable on November 28th 2022) <http://www.arcindy.com/effect-of-aerodynamic-drag-on-fuel-economy.html>.
- Achour, G., Sung, W. J., Pinon-Fischer, O. J. & Mavris, D. N. 2020 Development of a conditional generative adversarial network for airfoil shape optimization. In *AIAA Scitech 2020 Forum*, p. 2261. The American Institute of Aeronautics and Astronautics, Inc.
- Badías, A., Curtit, S., González, D., Alfaro, I., Chinesta, F. & Cueto, E. 2019 An augmented reality platform for interactive aerodynamic design and analysis. *International Journal for Numerical Methods in Engineering* **120**, 125–138.
- Burnap, A., Liu, Y., Pan, Y., Lee, H., Gonzalez, R. & Papalambros, P. Y. 2016 Estimating and exploring the product form design space using deep generative models. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 50107)*, p. V02AT03A013. American Society of Mechanical Engineers.
- Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S. & Su, H. 2015 Shapenet: An information-rich 3d model repository. Preprint, [arXiv:1512.03012](https://arxiv.org/abs/1512.03012).
- Chen, T. & Guestrin, C. 2016 Xgboost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. Association for Computing Machinery.
- Chen, W. & Ahmed, F. 2021 Padgan: learning to generate high-quality novel designs. *Journal of Mechanical Design* **143**, 031703.
- Chen, W. & Fuge, M. 2019 Synthesizing designs with interpart dependencies using hierarchical generative adversarial networks. *Journal of Mechanical Design* **141**, 111403.
- Chen, W., Chiu, K. & Fuge, M. D. 2020 Airfoil design parameterization and optimization using bézier generative adversarial networks. *AIAA Journal* **58**, 4723–4735.
- Cunningham, J. D., Simpson, T. W. & Tucker, C. S. 2019 An investigation of surrogate models for efficient performance-based decoding of 3d point clouds. *Journal of Mechanical Design* **141**, 121401.
- Cunningham, J. D., Shu, D., Simpson, T. W. & Tucker, C. S. 2020 A sparsity preserving genetic algorithm for extracting diverse functional 3d designs from deep generative neural networks. *Design Science* **6**, E11.
- de Miguel Rodríguez, J., Villafañe, M. E., Piškorec, L. & Caparrini, F. S. 2020 Generation of geometric interpolations of building types with deep variational autoencoders. *Design Science* **6**, E34.
- Dering, M., Cunningham, J., Desai, R., Yukish, M. A., Simpson, T. W. & Tucker, C. S. 2018 A physics-based virtual environment for enhancing the quality of deep generative designs. In *ASME 2018 International Design Engineering Technical Conferences and*

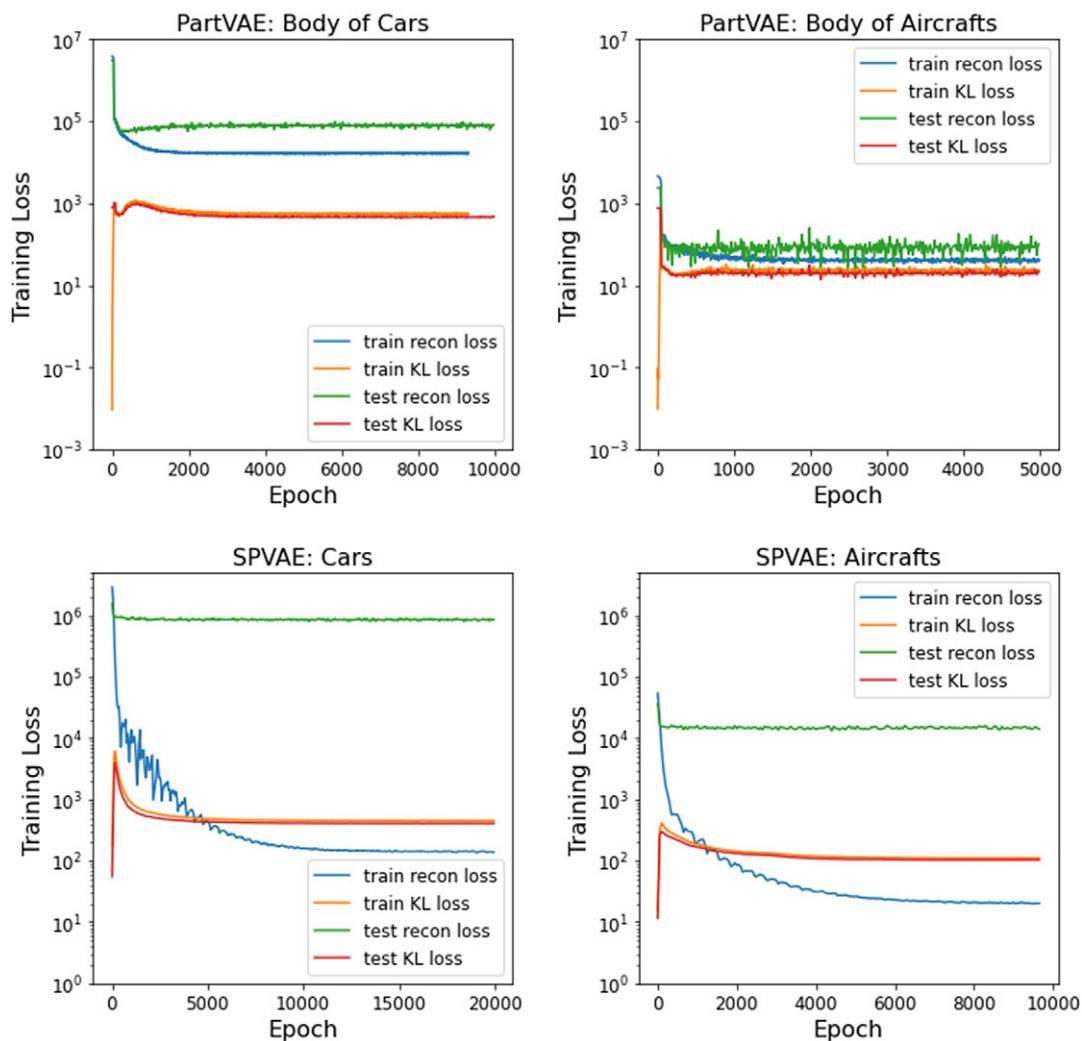
- Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection.
- Dietterich, T. G.** 2000 Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings*, pp. 1–15. Springer.
- Edwards, K. M., Addala, V. L. & Ahmed, F.** 2021 Design form and function prediction from a single image. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 85376)*, p. V002T02A032. American Society of Mechanical Engineers.
- Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M. & Smola, A.** 2020 Autogluon-tabular: robust and accurate automl for structured data. Preprint, [arXiv:2003.06505](https://arxiv.org/abs/2003.06505).
- Fairman, J. G.** 1996 Aerodynamics lift formula, online document (downloadable on June 8th 2023) [https://www.grc.nasa.gov/WWW/K-12/WindTunnel/Activities/lift\\_formula.html](https://www.grc.nasa.gov/WWW/K-12/WindTunnel/Activities/lift_formula.html)
- Feurer, M., Klein, A., Eggenberger, K., Springenberg, J., Blum, M. & Hutter, F.** 2015 Efficient and robust automated machine learning. *Advances in Neural Information Processing Systems* **28**, 2755–2763.
- Fujita, K., Minowa, K., Nomaguchi, Y., Yamasaki, S. & Yaji, K.** 2021 Design concept generation with variational deep embedding over comprehensive optimization. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (vol. 85390)*, p. V03BT03A038. American Society of Mechanical Engineers.
- Gao, L., Lai, Y.-K., Yang, J., Ling-Xiao, Z., Xia, S. & Kobbelt, L.** 2019a Sparse data driven mesh deformation. *IEEE Transactions on Visualization and Computer Graphics* **27** (3), 2085–2100.
- Gao, L., Yang, J., Wu, T., Yuan, Y. J., Fu, H., Lai, Y.-K. & Zhang, H.** 2019b SDM-NET: deep generative network for structured deformable mesh. *ACM Transactions on Graphics* **38**, 1–15; doi:[10.1145/3355089.3356488](https://doi.org/10.1145/3355089.3356488).
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y.** 2014 Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680. Curran Associates, Inc.
- Guyon, I., Sun-Hosoya, L., Boullé, M., Escalante, H. J., Escalera, S., Liu, Z., Jajetic, D., Ray, B., Saeed, M., Sebag, M., Statnikov, A., Tu, W.-W. & Viegas, E.** 2019 Analysis of the automl challenge series. *Automated Machine Learning* **177**, 177–219.
- He, X., Zhao, K. & Chu, X.** 2021 Automl: a survey of the state-of-the-art. *Knowledge-Based Systems* **212**, 106622.
- Henseler, J., Ringle, C. M. & Sinkovics, R. R.** 2009 The use of partial least squares path modeling in international marketing. In *New Challenges to International Marketing*. Emerald Group Publishing Limited.
- Huang, J., Zhou, Y. & Guibas, L.** 2020 Manifoldplus: a robust and scalable watertight manifold surface generation method for triangle soups. Preprint, [arXiv:2005.11621](https://arxiv.org/abs/2005.11621).
- Hutter, F., Kotthoff, L. & Vanschoren, J.** 2019 *Automated Machine Learning: Methods, Systems, Challenges*. Springer Nature.
- Jasak, H.** 2009 OpenFOAM: open source CFD in research and industry. *International Journal of Naval Architecture and Ocean Engineering* **1** (2), 89–94.
- Kingma, D. P. & Welling, M.** 2013 Auto-encoding variational Bayes. Preprint, [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
- Krish, S.** 2011 A practical generative design method. *Computer-Aided Design* **43**, 88–100.

- Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H. & Guibas, L. 2017 Grass: generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)* **36**, 1–14.
- Li, X., Xie, C. & Sha, Z. 2021 Part-aware product design agent using deep generative network and local linear embedding. In *Proceedings of the 54th Hawaii International Conference on System Sciences*, p. 5250. IEEE Computer Society Press.
- Li, X., Xie, C. & Sha, Z. 2022 A predictive and generative design approach for three-dimensional mesh shapes using target-embedding variational autoencoder. *Journal of Mechanical Design* **144**, 114501.
- Li, X., Wang, Y. & Sha, Z. 2023 Deep learning methods of cross-modal tasks for conceptual design of product shapes: A review. *Journal of Mechanical Design* **145**, 041401.
- Matejka, J., Glueck, M., Bradner, E., Hashemi, A., Grossman, T. & Fitzmaurice, G. 2018 Dream lens: exploration and visualization of large-scale generative design datasets. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–12. Association for Computing Machinery.
- McComb, C., Cagan, J. & Kotovsky, K. 2017 Mining process heuristics from designer action data via hidden markov models. *Journal of Mechanical Design* **139**, 111412.
- McComb, C., Song, B., Zurita, N. F. S., Zhang, G., Stump, G., Balon, C., Miller, S., Yukish, M. & Cagan, J. 2020 Toward hybrid teams: a platform to understand human-computer collaboration during the design of complex engineered systems. In *Proceedings of the Design Society DESIGN Conference*, pp. 1551–1560. Cambridge University Press.
- McKnight, M. 2017 Generative design: what it is? How is it being used? Why it's a game changer. *KnE Engineering* **2** (2), 176–181.
- Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N. J. & Guibas, L. J. 2019 StructureNet: hierarchical graph networks for 3D shape generation. *ACM Transactions on Graphics (TOG)*, **38** (6), 1–19.
- NASA n.d. Drag coefficient, online document (downloadable on June 8th 2023) <https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/drag-coefficient/>.
- Nash, C. & Williams, C. K. I. 2017 The shape variational autoencoder: a deep generative model of part-segmented 3d objects. In *Computer Graphics Forum (Vol. 36)*, pp. 1–12. Wiley Online Library.
- Oh, S., Jung, Y., Kim, S., Lee, I. & Kang, N. 2019 Deep generative design: integration of topology optimization and generative models. *Journal of Mechanical Design* **141**, 111405.
- Panchal, J. H., Fuge, M., Liu, Y., Missoum, S. & Tucker, C. 2019 Machine learning for engineering design. *Journal of Mechanical Design* **141**, 110301.
- Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R. & Tucker, P. K. 2005 Surrogate-based analysis and optimization. *Progress in Aerospace Sciences* **41**, 1–28.
- Rahman, M. H., Xie, C. & Sha, Z. 2019 A deep learning based approach to predict sequential design decisions. In *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 081706.
- Rahman, M. H., Yuan, S., Xie, C. & Sha, Z. 2020 Predicting human design decisions with deep recurrent neural network combining static and dynamic data. *Design Science* **6**, E15.
- Rahman, M. H., Xie, C. & Sha, Z. 2021 Predicting sequential design decisions using the function-behavior-structure design process model and recurrent neural networks. *Journal of Mechanical Design* **143**, 081706.

- Regenwetter, L., Nobari, A. H. & Ahmed, F.** 2022 Deep generative models in engineering design: a review. *Journal of Mechanical Design* **144**, 071704.
- Regenwetter, L., Weaver, C. & Ahmed, F.** 2023 Framed: an automl approach for structural performance prediction of bicycle frames. *Computer-Aided Design* **156**, 103446.
- Shea, K., Aish, R. & Gourtovaia, M.** 2005 Towards integrated performance-driven generative design tools. *Automation in Construction* **14**, 253–264; doi:[10.1016/j.aut-con.2004.07.002](https://doi.org/10.1016/j.aut-con.2004.07.002).
- Shu, D., Cunningham, J., Stump, G., Miller, S. W., Yukish, M. A., Simpson, T. W. & Tucker, C. S.** 2020 3d design using generative adversarial networks and physics-based validation. *Journal of Mechanical Design* **142**, 071701.
- Singh, V. & Gu, N.** 2012 Towards an integrated generative design framework. *Design Studies* **33**, 185–207.
- Song, B., Yuan, C., Permenter, F., Arechiga, N. & Ahmed, F.** 2023 Surrogate modeling of car drag coefficient with depth and normal renderings. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers. Preprint, arXiv:2306.06110.
- Sun, L., Gao, H., Pan, S. & Wang, J.-X.** 2020 Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering* **361**, 112732.
- Umetani, N. & Bickel, B.** 2018 Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics (TOG)* **37**, 89.
- Wang, G. G. & Shan, S.** 2006 Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design* **129**, 370–380; doi:[10.1115/1.2429697](https://doi.org/10.1115/1.2429697).
- Whalen, E. & Mueller, C.** 2022 Toward reusable surrogate models: graph-based transfer learning on trusses. *Journal of Mechanical Design* **144**, 021704.
- Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. & Xiao, J.** 2015 3d shapenets: a deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920. The Computer Vision Foundation.
- Yi, L., Kim, V. G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A. & Guibas, L.** 2016 A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (TOG)* **35**, 1–12.
- Yi, X., Walia, E. & Babyn, P.** 2019 Generative adversarial network in medical imaging: a review. *Medical Image Analysis* **58**, 101552.
- Yoo, S., Lee, S., Kim, S., Hwang, K. H., Park, J. H. & Kang, N.** 2021 Integrating deep learning into CAD/CAE system: generative design and evaluation of 3D conceptual wheel. *Struct. Multidiscip. Optim.* **64**, 4 (Oct 2021), 2725–2747. <https://doi.org/10.1007/s00158-021-02953-9>
- Zhang, W., Yang Z., Jiang, H., Nigam, S., Yamakawa, S., Furuhashi, T., Shimada, K. & Kara, L. B.** 2019 3d shape synthesis for conceptual design and optimization using variational autoencoders. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 59186)*, p. V02AT03A017. American Society of Mechanical Engineers.
- Zollhöfer, M., Nießner, M., Izadi, S., Rehmman, C., Zach, C., Fisher, M., Wu, C., Fitzgibbon, A., Loop, C. & Theobalt, C.** Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics (ToG)* **33**, 1–12.

## A. Appendix

The training loss values for both the car and aircraft models were recorded and documented in Figure A1. It was observed that all loss terms reached convergence, indicating successful training. Specifically, for car models, the PartVAEs were trained for 10000 epochs, followed by training the SPVAE for 20000 epochs. On the other hand, for the aircraft models, the PartVAEs were trained for 5000 epochs, followed by training the SPVAE for 10000 epochs.



**Figure A1.** The training loss values, including reconstruction loss and KL divergence loss, were recorded for both the car and aircraft models during the training process. However, for the purpose of showcasing the training of PartVAEs, we have chosen to highlight the loss values specifically for the training of the body components of the car and aircraft because similar trends were observed in the loss values for training other parts as well.