Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

On the Efficiency of a Point Cloud Autoencoder as a Geometric Representation for Shape Optimization

Thiago de Jesus de Araujo Rios, Thomas Bäck, Bas van Stein, Bernhard Sendhoff, Stefan Menzel

2019

Preprint:

This is an accepted article published in 2019 IEEE Symposium Series on Computational Intelligence (SSCI). The final authenticated version is available online at: https://doi.org/10.1109/SSCI44817.2019.9003161 Copyright 2019 IEEE

On the Efficiency of a Point Cloud Autoencoder as a Geometric Representation for Shape Optimization

Thiago Rios*, Thomas Bäck†, Bas van Stein†, Bernhard Sendhoff* and Stefan Menzel*
* Honda Research Institute Europe GmbH, Carl-Legien-Str. 30, 63073 Offenbach, Germany
†Leiden Institute of Advanced Computer Science (LIACS), Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
Email: {thiago.rios, bernhard.sendhoff, stefan.menzel}@honda-ri.de, {t.h.w.baeck, b.van.stein}@liacs.leidenuniv.nl

The article has been accepted for publication in the IEEE Symposium Series on Computational Intelligence.

Copyright notice:

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

On the Efficiency of a Point Cloud Autoencoder as a Geometric Representation for Shape Optimization

Thiago Rios*, Thomas Bäck†, Bas van Stein†, Bernhard Sendhoff* and Stefan Menzel*
* Honda Research Institute Europe GmbH, Carl-Legien-Str. 30, 63073 Offenbach, Germany
†Leiden Institute of Advanced Computer Science (LIACS), Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
Email: {thiago.rios, bernhard.sendhoff, stefan.menzel}@honda-ri.de, {t.h.w.baeck, b.van.stein}@liacs.leidenuniv.nl

***Abstract*—A crucial step for optimizing a system is the formulation of the objective function, and part of it concerns the selection of the design parameters. One of the major concerns regarding the parameterization of the objective function is the trade-off between exploring feasible solutions in the design space and maintaining admissible computational effort. In order to achieve such balance in optimization problems with Computer Aided Engineering (CAE) models, the conventional constructive geometric representations are substituted by deformation methods, e.g. free form deformation, where the position of a few control points might be capable of handling large scale shape modifications. However, in light of the recent developments in the field of geometric deep learning architectures, autoencoders have risen as a promising alternative for efficiently condensing high-dimensional models into compact representations. Hence, in this paper we present a novel perspective on geometric deep learning models, by exploring the applicability of the latent space of a Point Cloud Autoencoder (PC-AE) in shape optimization problems with evolutionary algorithms. Focusing on engineering applications, a target shape matching optimization is used as a surrogate problem for computationally expensive CAE simulations. Through the evaluation of the quality of the solutions achieved in the optimization and further aspects, such as shape feasibility, PC-AE models have shown to be consistent and suitable geometric representations for such problems, adding a new perspective on the approaches for handling high-dimensional models to optimization tasks.**

***Keywords*—evolutionary design optimization, geometric representation, free form deformation, point cloud autoencoder**

I. Introduction

The formulation of a problem to optimize an engineering system often requires several simplifications and trade-offs, such as the balance between computational effort and accuracy of the model. In fact, the definition of the objective function and constraints might take as much as half of the effort to actually find the optimum design [1].

The selection of the design variables is part of the formulation process and reflects directly on the level of complexity of solving the problem. On the one hand, it is necessary to select a reasonable number of parameters to enable meaningful modifications in the modelled system, driving the optimization algorithm towards the best solution. On the other hand, high dimensional spaces increase the necessary number of fitness function evaluations to characterize the landscape, since the influence of each variable is more diluted than in the low-dimensional case. Hence, in applications with computationally expensive functions, the use of multiple parameters is undesired and might be prohibitive [2].

A problem that fits to the scenario described above is the optimization of aerodynamic shapes, such as a vehicle body or aircraft fuselage. In such problems, the design variables are mainly geometric features, since the flow and fluid characteristics are dictated by the nature of the environment where the components will operate. Therefore, the expertise of the designer plays a significant role, given that one might prefer designs related to previous successful experiences instead of creating a generic model. Additionally, the user has to define the dependency between the parameters in a way that the optimum solution is not overconstrained [3], which is usually not a trivial task.

When deformation approaches [4] came into scene, many of the discussed difficulties could be tackled. A central aspect of these methods is enabling large scale shape modifications controlled by a low number of parameters [5], favouring the balance between design space exploration and complexity of the problem. Especially for CAE models represented as meshes, the deformation methods allow a significant boost in the optimization problems, since they can be applied directly to the nodes of the discretized domain, avoiding the re-meshing phase, which is usually very time consuming [6], [7].

Neglecting the meshing costs, further methods can be considered for this purpose. Recently much effort has been applied to the research of Geometric Deep Learning (GDL) architectures, and autoencoders have risen as promising alternatives for the reduction of dimensionality in geometric problems. In the context of optimization tasks, an encode-decoder structure can be used for basis transformation, from the Euclidean to a low-dimensional latent domain, where the operations on the geometry can be performed. Further than the reduction in the dimensionality of the problem, the autoencoder might be capable of abstracting features differently than a human perspective, enabling different geometric parameterizations and the generation of novel designs.

Hence, we explore in this paper the potential of latent variables of a PC-AE as geometric representations for shape optimization tasks, as not yet reported in the literature. Furthermore, our approach focuses on engineering applications, bringing to discussion not only a numerical comparison to state-of-the-art deformation methods, but also other relevant aspects, such as shape feasibility, adaptability of the represen-

tation and interpretability of the latent variables. Our study in embedded in the framework of a target shape matching optimization problem, which substitutes the computationally expensive CAE numerical simulations, enabling the performance of several experiments.

The paper is structured according to the following order: it starts with the survey of the literature, in Section II, followed by the definition of the experimental setup and methods in Section III. The results of the experiments are discussed in Section IV, where the characteristics of the novel geometric representation is discussed in detail. The outlook and conclusions of the research close the paper and are presented in Section V.

II. Literature Survey on Geometric Representations

The term *Geometric Representation* often refers to more than the study of solid bodies, although it is historically associated to the methods for generating, modifying and visualizing shapes [8]. In order to understand and compare different representations, it is important to classify them according to the method for parameterizing the geometries. According to [3], the methods can be divided into constructive, deformative and volume based.

A. Contructive Solid Geometry

The constructive approach, also as discussed in [8], concerns the approaches based on non-parametric functions, boolean trees and PDE (Partial Differential Equations). Such techniques are widely implemented in CAD (Computer Aided Design) via combinations of CSG (Constructive Solid Geometry) and Splines, justified by the capability of the CSG to describe complex geometries as a tree of boolean operations between primitives [9], [10], added to the high compatibility of splines with visualization tools [11]–[13]. In this class, B-Splines and NURBS (Non-Uniform Rational Basis Splines) are the most used representations, and an example of application for aerodynamic shape optimization is presented by Della Vechia and Nicolosi in [14].

B. Shape Deformation Techniques

Instead of redefining the geometry based on predefined features, deformative approaches allow the optimizer to operate directly on the nodes of the mesh, treating them as design parameters [15]. Among the techniques, FFD is the most common one and is characterized by embedding the shape of interest into a tri-variate polynomial volume (e.g. Bézier), defined by a series of control points and where a $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ mapping can be applied [16].

In the literature, there are several reports on the application of FFD techniques to shape optimization. In [6] an FFD algorithm is coupled with an evolutionary strategy for the optimization of a three-dimensional high performance compressor aerofoil model. Also in [7] the optimization of a stator blade is performed using a framework that integrates FFD representations in an evolutionary optimization environment. Using a similar framework, in [17] the optimization of an element in the exhaust system of an internal combustíon engine is explored.

Sieger et al. [18] compare the performance of three deformation FFD, DFFDM (Direct FFD Manipulation) and RBF (Radial Basis Function based deformations), in shape optimization tasks. Among the criteria used for comparing the methods, the authors discussed about the adaptability and interpretability of the obtained variables, important characteristics which are also taken into account in the present paper.

C. Geometric Deep Learning

The definition of design parameters can be seen as a feature selection, where the designer abstracts the representation into a set of meaningful variables. Machine learning techniques are capable of automatically abstracting information of a high dimensional domain without the biased vision of a human user and, therefore, they have a great potential to generate efficient geometric representations.

Geometric Deep Learning (GDL) architectures have been intensively studied in the recent years. Many of the techniques are extensions of the operations used in Artificial Neural Networks (ANN) for image processing. Bronstein et al. present a detailed discussion in [19] about the topic, extending the mathematical formulation from the Euclidean domain, i.e. data defined on a uniform underlying structure, to non-Euclidean data, such as point clouds and graphs.

In many engineering applications involving geometric operations and CAE analyses, meshes are the base representation of choice, especially due to its compatibility to numerical simulation methods (e.g. Finite Element Method). These representations contain a rich amount of information, however, for GDL purposes, meshes are hard to abstract into simpler latent representations, due to the complexity of the representation, although some effort has already been reported in [20] and [21].

Nevertheless, several base representations can be derived from meshes, given their intrinsic amount of information. The simplest one is the point cloud, which consists on the list of nodes contained in the mesh and has been recently targeted by researchers due to the increasing popularizations of 3D sensoring and demand for spatial features processing. For GDL purposes, several architectures have already been published, such as PointNet [22], PointNet++ [23] and FoldingNet [24].

The most common difficulty for learning data from point clouds is imposing order-, shift- and rotation invariance to the autoencoder architecture, since the data belongs to a non-Euclidean domain. Therefore, the architectures cited before present pre-processing and modifications in the conventional multi-layer convolutional networks in order to embed the relation between the points and avoid order dependent operations.

A fourth architecture is proposed by Achlioptas et al. [25], where the operations performed by the encoder treat each point individually in five 1-D convolution layers, followed by ReLU activation functions. The sixth layer is responsible for extracting the latent representation via a maximum pooling

operation performed over the calculated features. Hence, for a given point cloud, the same latent representation is extracted, regardless of the ordering of the points, since the calculated features do not change and the maximum pooling is order independent. Finally, three fully connected layers with ReLU activation functions compose the decoder, which retrieves an approximation of the input point cloud. In [25], the authors also demonstrate geometric operations in the latent space, discussion which enabled the developments in the present study. In Figure 1 an example of car shape interpolation using latent variables is shown, which we obtained by reproducing the architecture proposed in [25] and training it on the car class of ShapeNet [26].

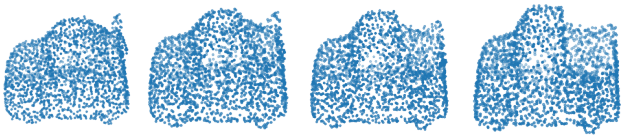

Fig. 1. Validation of our framework implementation by a reproduction of shape interpolations using the latent variables of a trained PC-AE.

III. Experimental Setup and Methods

In the light of the works reviewed in the literature survey, our objective is to evaluate the efficiency of the latent space for geometric representation in evolutionary optimization tasks. A generic single-objective optimization problem can be formulated as

$$\begin{aligned} &\min_{\mathbf{x}\in D} \ \mathrm{f}(\mathbf{x}), \ \text{subject to} \\ &\quad g_i(\mathbf{x}) \leq 0, i = 1, 2, ...p \\ &\quad h_j(\mathbf{x}) = 0, j = 1, 2, ...q \end{aligned} \tag{1}$$

where $f(\mathbf{x})$ is the objective function, and $g(\mathbf{x})$ and $h(\mathbf{x})$ are the equality and inequality constraints, respectively. In the present work, a target shape matching problem is considered, given its low computational cost and applicability of the revewed methods for geometric parameterization [27]. Therefore, the modified Hausdorff distance is used as objective function, since it is a metric of similarity between shapes and has been applied in previous work [28]. It is mathematically defined as

$$\mathcal{H}(a, \tilde{a}) = \frac{1}{2}\left(\sum_{i=1}^{N} \min_{a_i \in G_0} |a_i - \tilde{a}|^2 + \sum_{j=1}^{N} \min_{\tilde{a}_j \in G} |\tilde{a}_j - a|^2 \right) \tag{2}$$

where N is the number of samples from the geometry, a_i the points in the reference shape G_0 and $\tilde{a}_j$ the points from the deformed geometry G.

The design variables $\mathbf{x}$ in this problem are the parameters that control the deformation or shape generation method. For the experiments, two types of representations were selected: FFD and the latent variables obtained by an autoencoder. Hence, the parameters will be associated to the position of the control points in the first case and to variations in the latent variables in the second. The parameterization of the shapes will are detailed in the following sections.

For training and testing the autoencoder, as well as to provide potential target shapes for the optimization, a dataset of 500 shapes was generated using a benchmark car shape [29] and a standard FFD algorithm [**?**] which has been used throughout this paper. The deformations consist of random scalings of the vehicle along its principal axes, without conserving the aspect ratio, in order to facilitate the verification of the optimization results and increase the interpretability of variations in the latent variables.

The performance of the latent space as representation is assessed using two experiments that reflect typical cases of engineering optimization. The first consists of an unconstrained optimization using the shapes contained in the training dataset of the autoencoder, which theoretically should result in a perfect match for both representations. In the second experiment, a new set of target shapes is adopted, containing geometries that that prevent any of the optimizations to converge to a perfect match. For each experiment, 50 optimizations are carried out, in order to avoid the effects of singularities and special combinations between initial and target shapes. Additionally, the correlation between the variables in the latent and design spaces is analysed, in order to identify future directions of research on the topic.

A. Free Form Deformation Algorithm

The FFD algorithm used in the experiments has been proposed by Sederberg and Parry [30], where the deformations are calculated using the trivariate Bernstein polynomial. Therefore, by imposing a parallelepiped enclosing the geometry of interest, i.e. a control volume, any point $\mathbf{v}$ of the shape can be described with respect to a local coordinate system defined by orthogonal vectors $\vec{s}, \vec{t}, \vec{u}$, given by

$$\mathbf{v} = \mathbf{v_0} + s\vec{s} + t\vec{t} + u\vec{u} \tag{3}$$

where s, t, u are scalars, obtained using linear algebra, and $\mathbf{v_0}$ ia a point that belongs to the control volume, described in the original coordinate system. Within the parallelepiped, equally spaced grids of control points $\mathbf{p_i}$ can be defined, which globally deform the geometry when displaced, according to the following equation:

$$\begin{aligned} \mathbf{v_{FFD}} = \sum_{i=0}^{l} \binom{l}{i} (1-s)^{l-i} s^i \Bigg\{ \sum_{j=0}^{m} \binom{m}{j} (1-t)^{m-j} t^j \\ \left[\sum_{k=o}^{n} \binom{n}{k} (1-u)^{n-k} u^k \mathbf{p}_{ijk} \right] \Bigg\} \end{aligned} \tag{4}$$

where $\mathbf{v_{FFD}}$ is the deformed point, l, m, n are the number of control planes in the $\vec{s}, \vec{t}, \vec{u}$ directions, respectively.

For the points inside the control volume, the continuity C^k is ensured by the formulation of the Bezier curves, used in the algorithm. When the geometry is partially enclosed by the control volume, the kth continuity can be achieved only if

the k planes adjacent to the boundaries are not moved, which might constrain the range of shapes that can be achieved.

For the purposes of this work and for the sake of simplicity, only cases with fully enclosed geometries are considered, since the geometric properties of the FFD algorithm are not subject of investigation. Furthermore, the same configuration was used for both generating the dataset of shapes and performing the target shape matching, such that all the optimizations would have a possible and known solution.

The configuration of the control volume used in the deformations is shown in Figure 2 and contains 6 planes in X and Z directions and 4 in Y direction. For the optimization, the position of the planes are considered symmetric with respect to the geometric center of the car, resulting in 8 parameters, as shown in Figure 2.

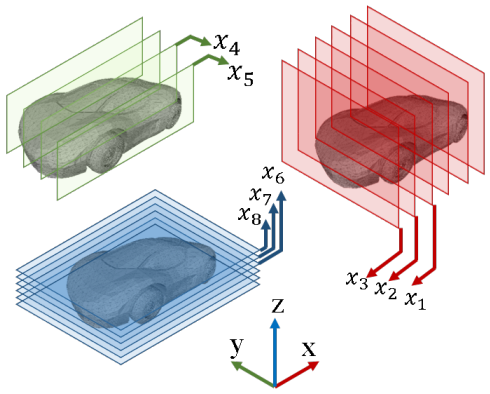

Fig. 2. Layout of the planes in the control volume and design variables adopted for the optimization.

B. Modified Vanilla Point Cloud Autoencoder

The autoencoder used in the experiments is based on the Vanilla architecture, presented in [25], and it is depicted in Figure 3. Compared to the reference, the proposed PC-AE presents minor modifications in the activation functions, in order to adapt it better to the optimization task and to the limits of normalization used for the geometries.

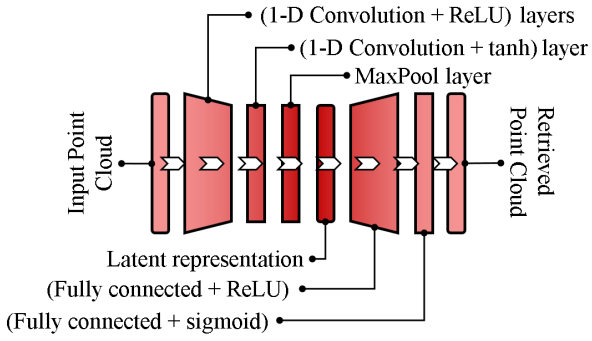

Fig. 3. Architecture of the modified Vanilla PC-AE Architecture used for encoding the geometries.

In the layer anterior to the maximum pooling, the activation function was changed from ReLU to hyperbolic tangent. Then, the values of each dimension of the latent representation are limited to the range $[-1, 1]$, which simplifies the definition of constraints. Also, in the last fully connected layer, the sigmoid function was used instead of the ReLU, since coordinates of all points were normalized to the range $[0.1, 0.9]$. In order to validate the proposed architecture, it was trained using point clouds sampled from the ShapeNet core dataset and the same training parameters and algorithm to calculate the Chamfer Distance as reported in [25], yielding the results shown in Table I.

TABLE I
EVALUATION OF THE LOSS FUNCTION AFTER TRAINING AND FOR TESTING, USING THE REFERENCE AND MODIFIED ARCHITECTURES.

Optimizer	Adam Optimizer	
Learning rate	5.00E-04	
Epochs	500	
Architectures	Reference	Modified
Chamfer Distance on the training set	3.34E-04	2.91E-04
Chamfer Distance on the test set	4.00E-04	3.03E-04
Standard deviation	9.70E-05	8.84E-05

As a last modification to the proposal in [25], the loss function considered for training the autoencoder was the sum of the mean squared distance between the points in the reference and predicted point clouds, given by the following equation:

$$Loss = \frac{1}{N}\left(\sum_{i=1}^{N} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|_2\right) \tag{5}$$

where N is the number of points, $\mathbf{x}_i$ and $\tilde{\mathbf{x}}_i$ are the i points from the ground truth and retrieved point clouds, respectively.

In the original implementation, the authors used the Chamfer and Earth Mover's Distances as loss functions, which do not impose the ordering of the points, being less restrictive than our suggested method. However, if the point clouds were generated from isometric meshes, which is the case for this work, maintaining the ordering of the points might ease the reconstruction of the mesh from the autoencoder output, since the connectivity between the nodes is already known, decreasing the post-processing costs. Also, the computational cost per iteration is considerably lower, requiring less infrastructure for training and performing experiments with the autoencoder.

The main drawbacks of forcing the order of the points during the training are the required number of iterations to achieve convergence and the restriction to datasets that contain only isomorphic meshes. Nevertheless, it is important to stress that this approach does not make the architecture itself order dependent, since the 1-dimensional convolutions in the encoder and the maximum pooling operation to retrieve the latent representation were kept according to the original implementation.

The PC-AE used in the experiments presents the same number of filters for each layer as presented in [25], except for the latent code, which is set to a 8-dimensional vector. Hence, although the landscape for the optimization with the latent

representation differs from the one obtained with FFD, both problems present at least the same dimensionality, allowing a reasonable comparison between methods. Also, the model was trained using 90% of the shapes in the dataset, selected randomly, and the Adam Optimizer [31], with learning rate $\eta = 5E{-}04$, $\beta_1 = 0.9$, $\beta_2 = 0.99$, in batches of 50 shapes and over 1700 iterations. The evaluation of the loss function on the training and test sets resulted in 0.067 and 0.082, respectively.

C. Optimization algorithm: CMA-ES

The selection of the optimizer for the present study accounted for the characteristics of the problem and objectives of the experiments. The aim is to emulate an environment as close as possible to a real case of engineering optimization, where the objective functions are computationally expensive and a high convergence ratio is required. Nevertheless, the study aims to evaluate the effects of the geometric representations rather than modifying the optimizer or tuning the hyperparameters, therefore, low number of parameters and robustness are also expected.

Considering the previous assumptions, Evolutionary Algorithms (EA) are a better fit than gradient-based optimizers. EAs do not require derivatives and their mechanism of search for the optimum allows the exploration of the design space, reducing the chances of achieving sub-optimal solutions. Nevertheless, it is important to point out that the gradient-based algorithms are still applicable to the problem, since the objective function can be derived with respect to the inputs for both representations. An assessment of their capability of handling such problems would also be an interesting scientific contribution, since they are not often explored in such domain.

The method selected for the experiments is the CMA-ES, due to its suitability for small populations [32]–[35], high convergence ratio and low number of hyperparameters. The approach implemented for the tests follows the proposal of Hansen et al. [36], considering a (3, 10)-strategy, with initial step size as 0.01 and 70 generations. The step size was defined experimentally, with the purpose to avoid aggressive shape modifications during the first stages of the optimization and, therefore, the use of constraints.

IV. Results

A. Optimization with full-match shapes

The first scenario concerns the experiment using geometries contained in training set of the PC-AE. Therefore, the true optimum in all optimizations is theoretically achievable for both representations. The behavior of the objective function was analysed statistically considering the fittest individuals of all cases, for 7 sampled generations. The results are shown in the box-plots of Figures 4 and 5.

As expected, nearly all the individuals converged to the target shape, since the dataset was known by the PC-AE and could be obtained with the parameters used in FFD. However, comparing the mean and standard deviation of the results in the last generations, the optimizations with the latent representation led to more consistent results, since the fittest individuals were often closer to the optimum, as indicated by the standard deviation values shown Table II and visually noted in Figure 5.

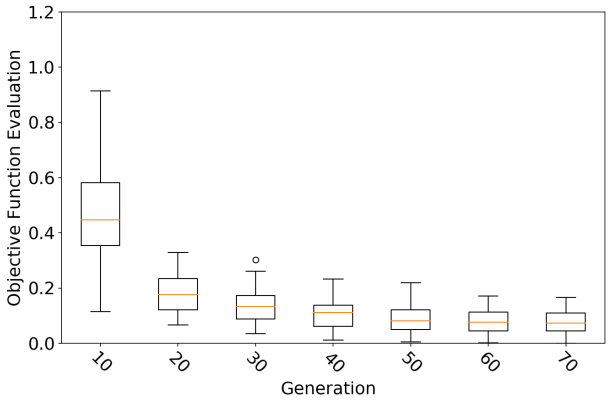

Fig. 4. Normalized objective function and statistics of the fittest individuals for all the optimizations using FFD as geometric representation.

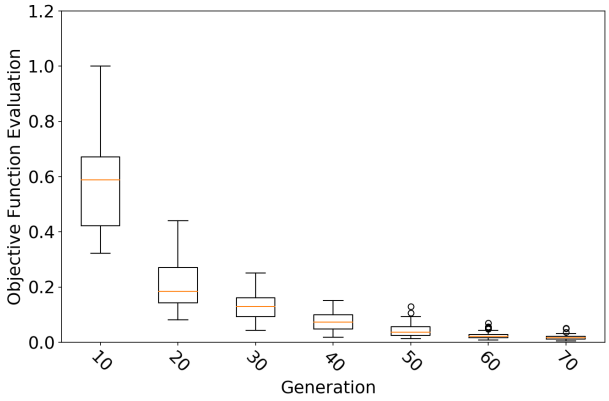

Fig. 5. Normalized objective function and statistics of the fittest individuals for all the optimizations using the latent space as geometric representation.

TABLE II
MEAN AND STANDARD DEVIATION FOR THE NORMALIZED OBJECTIVE FUNCTION EVALUATIONS OVER THE LAST GENERATIONS

	Generation					
	20	30	40	50	60	70
μ FFD	0.1397	0.1054	0.0901	0.0817	0.0784	0.076
σ FFD	0.0646	0.0496	0.0468	0.042	0.0408	0.0404
μ PC-AE	0.1364	0.0846	0.0466	0.0276	0.0188	0.0159
σ PC-AE	0.0463	0.0416	0.0265	0.0157	0.0099	0.0083

In terms of feasibility of the geometries for CAE applications, both representations led to meshes with reasonable smoothness at the latest stages. However, in some of the cases using the PC-AE, the shapes obtained at intermediate steps of the optimizations were not viable for CAE simulations, due to high irregularity of the elements, as shown in Figure 6. Hence only the FFD ensures continuity C^k of the surface, it was expected that the results obtained by the deformational method would provide higher smoothness and regularity of the surfaces.

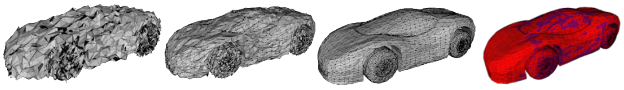

Fig. 6. Mesh reconstruction for intermediate shapes obtained in one of the cases with the PC-AE. From the left to the right: 10th, 20th, 30th and 70th generations, the last showing the difference to the target shape in blue elements.

B. Matching to unknown shapes

In the previous experiment, the optimizations were performed using target shapes that could be achieved with the selected parameterization, therefore, focusing on the exploitation capability of the representations. In order to obtain an impression of the potential for exploring new designs, a second experiment is performed using target shapes that cannot be achieved using the proposed parameters for both representations.

The target shapes were generated by positioning the control points of the lattice used in the first experiment according to harmonic functions in three axes. Performing the optimizations and analysing the best individuals over generations, the results yield the statistics shown in Figures 7 and 8.

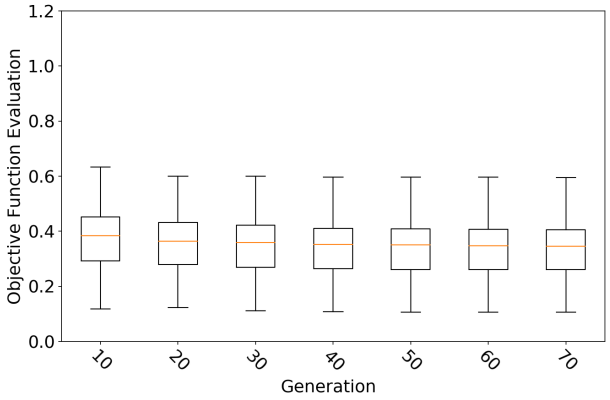

Fig. 7. Normalized objective function and statistics of the fittest individuals for all the optimizations using FFD as geometric representation.

Confirming the expectations, the optimizer was not able to converge to the target shape in any of the cases and for both representations. Statistically, the performance of the optimization using the PC-AE is still comparable to the cases with the FFD, but most of the shapes obtained with the autoencoder led to infeasible meshes for CAE applications, as shown in Figure 9.

C. Analysis of the latent variables

In order to identify potential causes for the results obtained with the PC-AE representation, it is important to understand how the latent variables are related to the parameters in the original design space. A second source of influence on the results is the configuration of the optimizer, however, for the sake of brevity, the tuning of hyperparameters will not be approached in this paper.

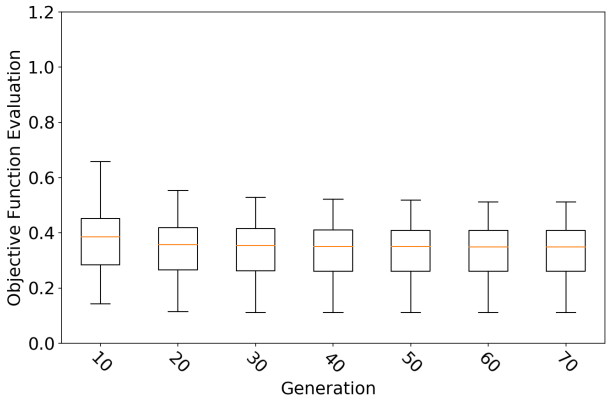

Fig. 8. Normalized objective function and statistics of the fittest individuals for all the optimizations using the latent space as geometric representation.

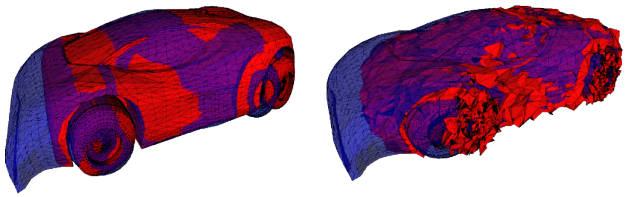

Fig. 9. In red, on the left, optimized shape obtained using FFD representation and on the right the results obtained with the PC-AE for the same target geometry. The blue meshes are the corresponding target shapes.

Returning to the deformations of the first experiment, an analysis of the correlation factor between latent variables and the factor of scale along each axis was performed. Considering 60 randomly selected shapes from the dataset, the obtained values are shown in Table III, where SF and LV stand for Scaling Factor and Latent Variable, respectively.

TABLE III
CORRELATION BETWEEN THE LATENT VARIABLES AND DESIGN PARAMETERS USED TO GENERATE THE DATASET

	SF 1	SF 2	SF 3
LV1	0.2409	-0.1723	**0.9995**
LV2	0.5623	**0.8708**	0.0084
LV3	**0.9812**	0.0682	0.3905
LV4	**0.9989**	0.0987	0.1898
LV5	0.6087	**0.7455**	0.0135
LV6	**0.9985**	0.0969	0.2061
LV7	0.1366	**0.9820**	0.0118
LV8	**0.9981**	0.0926	0.2036

As shown in the previous Table, each latent variable has a strong correlation with one component of deformation and a weaker, yet significant, with a secondary component of deformation. Hence, due to the coupling between the latent variables and multiple components of deformation, the modifications made in the latent variables are prone to generate noise and undesired displacements of the points, resulting in infeasible meshes.

In order to explore the given justification, the PC-AE was retrained considering the same conditions, but the dimensionality of the latent representation was reduced to 3 variables.

Performing the same analysis of correlation, the new factors yield the values shown in Table IV. Although the effect of the variables on the scaling along the axes is not fully decoupled, it is less aggressive than in the previous case, increasing the potential to achieve a smoother transition between shapes.

TABLE IV
CORRELATION BETWEEN THE PARAMETERS IN THE DESIGN SPACE AND LATENT VARIABLES FOR THE PC-AE RETRAINED WITH 3 LATENT VARIABLES.

	SF1	SF2	SF3
LR1	**0.9915**	0.2186	0.1968
LR2	-0.1741	**0.9641**	-0.2383
LR3	-0.1386	-0.2301	**0.9374**

For verification purposes, additional 30 target shape matching optimizations were performed using the simplified representation, leading to the results of Figure 10. Similar to the initial experiment, all the cases converged to shapes close to the true optimum, with small deviations. Also, the number of steps performed until reacing convergence was smaller, which can be explained by the reduction in the dimensionality of the design space. Finally, analysing the feasibility of the meshes at intermediate steps, most of the shapes presented high smoothness and no self-intersections, therefore, the coupling between the latent variables and multiple shape modification parameters has an influence on the feasibility of the shapes. An example retrieved from one of the cases is shown in Figure 11.

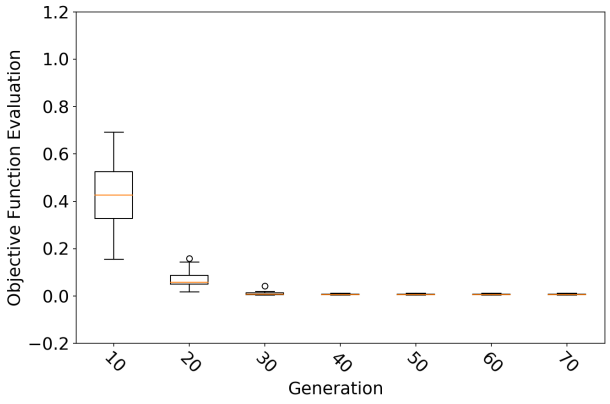

Fig. 10. Statistic of the optimizations using a simplified latent representation. The extreme values used for the normalization are 4.052 and 205.8.

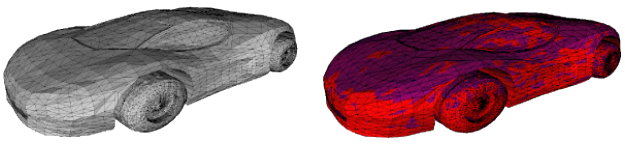

Fig. 11. Mesh reconstructed for a sampled geometry at the 5th generation of the optimization (left) and final stage (right), with the target shape shown in blue.

V. CONCLUSION

In the search for a balance between design exploration and computational costs, different geometric parameterizations for reduction of dimensionality have been developed. Among them, deformational approaches have been proven successful, especially for aerodynamic shape optimization, where the objective functions are often dependent on computationally expensive numerical simulations. In the past few years, GDL techniques started being investigated and new potential methods for compressing CAE models into efficient representations came into scene.

In this paper we explored the efficiency of the bottleneck layer of a PC-AE as a geometric representation for evolutionary shape optimization problems, implementing a breaking through application for GDL models. The representation was tested on a target shape matching optimization framework, using the CMA-ES algorithm as optimizer and an FFD technique as a reference for comparison. The experiments were designed to analyse cases where the true optimum and only local optima could be achieved. On both tests, the PC-AE representation led to results at least comparable to the FFD approach, with mean objective function values of 7.259 and 19.384 in the first case for the PC-AE and FFD representations, respectively.

Nevertheless, the proposed method requires a dataset for training the autoencoder, which is not always available prior to the optimization. Additionally, defining the dataset is already a challenging task, since it should contain the complete set of geometric features that the network should be able to reach during the optimization, which are frequently also unknown. Hence, the development of a method for exploring geometric features and designing a dataset is still required for implementing the proposed approach in real-world tasks.

Additionally, the quality of the meshes at intermediate steps of the optimization was investigated, since it is a usual requirement for CAE applications. From the two main factors that influence the shape generation process, only the footprints on how the autoencoder abstracted the geometric features was analysed. In order to do so, the correlation between the latent variables and parameters used to generate the dataset were correlated for two latent spaces of different dimensionality. The analysis pointed out that the coupling between the a latent variable and multiple geometric features is a possible source of shape infeasibility, anaspect which was tested in a third round of optimizations using a latent representation with the same dimensionality of the design space.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 766186.

REFERENCES

[1] J. S. Arora, "Chapter 2 - optimum design problem formulation," in *Introduction to Optimum Design (Fourth Edition)*, fourth edition ed., J. S. Arora, Ed. Boston: Academic Press, 2017, pp. 19 – 70. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128008065000020

[2] Y. Zhang, Z.-H. Han, L. Shi, and W.-P. Song, *Multi-round Surrogate-based Optimization for Benchmark Aerodynamic Design Problems*, 2016. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2016-1545

[3] S. Skinner and H. Zare-Behtash, "State-of-the-art in aerodynamic shape optimisation methods," *Applied Soft Computing*, vol. 62, pp. 933 – 962, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568946173056690

[4] J. A. Samaresh, "A survey of shape parameterization techniques," National Aeronautics and Space Administration, Tech. Rep. NAS-2015-02, 2015.

[5] A. Viswanath, A. I. J. Forrester, and A. J. Keane, "Dimension reduction for aerodynamic design optimization," *AIAA Journal*, vol. 49, no. 6, pp. 1256–1266, 2011. [Online]. Available: https://doi.org/10.2514/1.J050717

[6] S. Menzel, M. Olhofer, and B. Sendhoff, "Application of free form deformation techniques in evolutionary design optimisation," in *6th World Congress on Structural and Multidisciplinary Optimization (WCSMO6)*, J. Herskovits, S. Mazorche, and A. Canelas, Eds. Rio de Janeiro: COPPE Publication, 2005.

[7] S. Menzel and B. Sendhoff, *Representing the Change - Free Form Deformation for Evolutionary Design Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 63–86. [Online]. Available: https://doi.org/10.1007/978-3-540-75771-9_4

[8] V. Shapiro, "Chapter 20 - solid modeling," in *Handbook of Computer Aided Geometric Design*, G. Farin, J. Hoschek, and M.-S. Kim, Eds. Amsterdam: North-Holland, 2002, pp. 473 – 518. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780444511041500216

[9] C. D. Segura, T. Stine, and J. Yang, "Constructive solid geometry using bsp tree," 2013.

[10] *Constructive Solid Geometry*. London: Springer London, 2008, pp. 277–283. [Online]. Available: https://doi.org/10.1007/978-1-84800-115-2_30

[11] G. Farin, "Chapter 5 - bézier curve topics," in *Curves and Surfaces for Computer-Aided Geometric Design (Third Edition)*, third edition ed., G. Farin, Ed. Boston: Academic Press, 1993, pp. 65 – 81. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780122490521500106

[12] ——, "Chapter 10 - b-splines," in *Curves and Surfaces for Computer-Aided Geometric Design (Third Edition)*, third edition ed., G. Farin, Ed. Boston: Academic Press, 1993, pp. 157 – 187. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780122490521500155

[13] M. Nahas, H. Huitric, and M. Saintourens, "Animation of a b-spline figure," *The Visual Computer*, vol. 3, no. 5, pp. 272–276, Mar 1988. [Online]. Available: https://doi.org/10.1007/BF01914862

[14] P. D. Vecchia and F. Nicolosi, "Aerodynamic guidelines in the design and optimization of new regional turboprop aircraft," *Aerospace Science and Technology*, vol. 38, pp. 88 – 104, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963814001515

[15] J. Fincham and M. Friswell, "Aerodynamic optimisation of a camber morphing aerofoil," *Aerospace Science and Technology*, vol. 43, pp. 245 – 255, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963815000802

[16] G. Kenway, G. Kennedy, and J. Martins, *A CAD-Free Approach to High-Fidelity Aerostructural Optimization*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2010-9231

[17] M. Olhofer, T. Bihrer, S. Menzel, M. Fischer, and B. Sendhoff, "Evolutionary optimisation of an exhaust flow element with free form deformation," in *Simulation for Innovative Design, Proceedings of the 4th EASC - 2009 European Automotive Simulation Conference*, K. Seibert and M. Jirka, Eds. ANSYS Inc., 2009, pp. 163–174.

[18] D. Sieger, S. Menzel, and M. Botsch, *On Shape Deformation Techniques for Simulation-Based Design Optimization*, 01 2015, vol. 5, pp. 281–303.

[19] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, July 2017.

[20] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3d faces using convolutional mesh autoencoders," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 725–741.

[21] O. Litany, A. M. Bronstein, M. M. Bronstein, and A. Makadia, "Deformable shape completion with graph convolutional autoencoders," *CoRR*, vol. abs/1712.00268, 2017. [Online]. Available: http://arxiv.org/abs/1712.00268

[22] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *CoRR*, vol. abs/1612.00593, 2016. [Online]. Available: http://arxiv.org/abs/1612.00593

[23] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *CoRR*, vol. abs/1706.02413, 2017. [Online]. Available: http://arxiv.org/abs/1706.02413

[24] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Interpretable unsupervised learning on 3d point clouds," *CoRR*, vol. abs/1712.07262, 2017. [Online]. Available: http://arxiv.org/abs/1712.07262

[25] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. J. Guibas, "Representation learning and adversarial generation of 3d point clouds," *CoRR*, vol. abs/1707.02392, 2017. [Online]. Available: http://arxiv.org/abs/1707.02392

[26] A. X. Chang, T. A. Funkhouser, L. J. 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," *CoRR*, vol. abs/1512.03012, 2015. [Online]. Available: http://arxiv.org/abs/1512.03012

[27] M. de Buhan, C. Dapogny, P. Frey, and C. Nardoni, "An optimization method for elastic shape matching," *Comptes Rendus Mathematique*, vol. 354, no. 8, pp. 783 – 787, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1631073X16300802

[28] P. Zhang, X. Yao, L. Jia, B. Sendhoff, and T. Schnier, "Target shape design optimization by evolving splines," 10 2007, pp. 2009 – 2016.

[29] "Reprap action car," https://www.thingiverse.com/thing:249/files, accessed: 2019-05-17.

[30] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '86. New York, NY, USA: ACM, 1986, pp. 151–160. [Online]. Available: http://doi.acm.org/10.1145/15922.15903

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[32] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[33] N. Hansen, *The CMA Evolution Strategy: A Comparing Review*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 75–102. [Online]. Available: https://doi.org/10.1007/3-540-32494-1_4

[34] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 1–23, 1993. [Online]. Available: https://doi.org/10.1162/evco.1993.1.1.1

[35] T. Bäck, F. Hoffmeister, and H.-P. Schwefel, "A survey of evolution strategies," in *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1991, pp. 2–9.

[36] N. Hansen, Y. Akimoto, D. Brockhoff, and M. Chan, "Cma-es/pycma: r2.7.0," Apr. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2651072