Constraint-Based Parameterization and Disentanglement of Aerodynamic Shapes using Deep Generative Models

Asmita Bhat[1] ✉, Nooshin Haji-Ghassemi[1], Deepak Nagaraj[2], and Sophie Fellenz[1][0000−0002−5385−3926]

[1] University of Kaiserslautern-Landau, Gottlieb-Daimler-Straße 47, 67663 Kaiserslautern, Germany {bhat,nooshin,fellenz}@cs.uni-kl.de
[2] Bertrandt Ingenieurbüro GmbH, Hufelandstrasse 26-28, 80939, München, Germany deepak.nagaraj@bertrandt.com

Abstract. Generating parametric shapes with respect to their structural and functional characteristics is a challenging and demanding problem. Conventional parameterization techniques are complex and require manual intervention and multiple cycles to produce plausible shapes, which makes the overall parameterization process extremely sensitive, time- consuming and error-prone. Despite these techniques' slow and iterative nature, a significant amount of data has been gathered over many years, prompting the community to turn to data-driven techniques like deep generative models for automatic parameterization. However, parameterizing shapes following necessary functional constraints is crucial but notoriously difficult and still needs to be studied. Therefore, we propose a data-driven framework that implicitly learns to generate plausible parametric aerodynamic shapes under specified constraints. We explore and compare several generative models, including generative adversarial networks and variational autoencoders, and systematically evaluate them for generation quality, diversity, and disentanglement aspects. Our framework, including a β-VAE model, enables the automatic generation of novel airfoils with watertight boundaries and interactive generation with its distributed and disentangled latent space. Through rigorous evaluation of our method, we demonstrate that the generated distribution closely matches the true distribution, resulting in the generation of highly realistic airfoils. Our method dramatically outperforms the current benchmark in terms of the quality and diversity of generated airfoils and establishes a new benchmark for constraint-based parameterization.

Keywords: Aerodynamic shapes · Deep Generative Models · VAE.

1 Introduction

Parametric shapes like airfoils, hydrofoils, fans, and turbines are used in a variety of applications, including aerodynamics, electronics, and automobiles, and are developed by combining parameterization and optimization techniques. These

techniques aim to develop shapes that yield optimal performance in terms of their functionality. The shapes are represented by one or more parametric functions in a high-dimensional space, parameterized by a set of design variables. Traditional parameterization techniques are explicit, meaning the parameters of

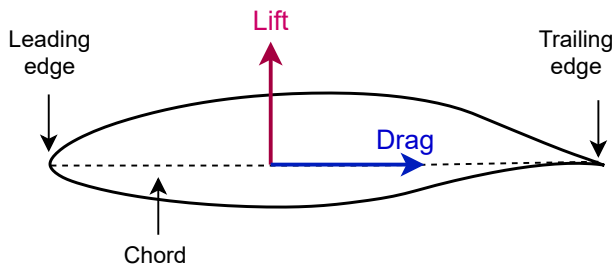


Fig. 1. An example of an airfoil from the UIUC dataset [33]. The chord of an airfoil is zero-centered. The leading edge and trailing edge points coincide with the chord. The direction of the lift component of aerodynamic force is perpendicular to the chord, and that of the drag is along the chord towards the trailing edge. Feasible airfoils have higher lift than drag.

the designs are manually set by a human expert. They are further optimized by employing numerical simulation tools such as computational fluid dynamics [31] or Computer-Aided Engineering (CAE) that consider various functional restrictions and boundary conditions. To generate feasible designs with acceptable performance using these techniques, several back-and-forth iterations through the design and optimization phases are required, making the overall process exceedingly time, memory, and computation intensive. Despite these challenges, the aerodynamics community has produced a large number of viable airfoil designs that may be utilized for a variety of tasks, including shape synthesis, optimization, and flow field prediction [1].

Modern parameterization techniques, such as BézierGAN [6], are data-driven models based on GANs [12], one of the prominent Deep Generative Models (DGMs) that can automatically parameterize aerodynamic shapes, also known as airfoils. An airfoil is a critical component of an aircraft design responsible for the aerodynamic force generated during operation. Figure 1 illustrates the two components of aerodynamic force: lift and drag. An airfoil must be streamlined to generate more lift than drag to enable an aircraft to take off. As a result, the coefficient of lift-to-drag, C_L/C_D, is an important functional characteristic of an airfoil. However, generating realistic and practically viable airfoils while ensuring such a functional characteristic is an extremely challenging problem mainly due to scarcity of data and the high dimensionality of the shapes.

This paper proposes a hybrid DGM-CAE framework for automated, constraint-based parameterization of airfoil designs while ensuring their mechanical func-

tionality. We enforce the C_L/C_D values obtained from numerical simulation software and constrain the airfoil generation using these values to generate airfoils for specific C_L/C_D ratios. This approach simplifies and accelerates the design process and can be used for effective and interactive shape parameterization. To sum up, our contributions are as follows:

1. We develop a hybrid DGM-CAE framework for automated and constraint-based airfoil parameterization while preserving their mechanical functionality.
2. We assess the results using relevant metrics for various generation aspects such as quality, diversity, and disentanglement. In addition, we also evaluate our framework's ability to parameterize original designs from the dataset and the precision of the enforced functional constraint.
3. The quantitative results demonstrate that our framework outperforms the baseline BézierGAN model in terms of quality and diversity of generated shapes.
4. To the best of our knowledge, this is a first attempt at conditionally generating airfoil designs using their C_L/C_D, which is a continuous real-valued number, unlike widely used conditional models [24] which use discrete labels.

2 Related Work

In this section, we discuss some of the traditional methods for parametric shape synthesis and the most recent popular DGMs for both high-quality synthesis and representation learning.

2.1 Parametric Shape Synthesis for Product Design

Methods for understanding design spaces and synthesizing new shapes or designs can be categorized into two broad categories: knowledge-driven methods and data-driven methods [37]. Knowledge-driven approaches use explicit rules to develop new shapes. Computational Design Synthesis is an example of a knowledge-driven method for synthesizing new shapes, which is most popularly used for gearboxes and bicycle frames [2]. Other methods include B-splines [22], Bézier curves [38], Free Form Deformation [39], Class-Shape Transformations (CST) [13], and PARSEC [9], which are parameterization methods that are used to generate curves for aerodynamic shapes. These parameterization methods adjust the control points or parameters using random perturbation or Latin hypercube sampling [39]. However, these knowledge-based methods suffer from high dimensionality of the design space [6] and unknown limits to the parameters that define the geometry. On the other hand, data-driven models [5,6,40] implicitly learn useful knowledge about the geometric representation from the existing designs in the dataset. DGMs, in particular, address the issues mentioned above by generating a compact latent representation that captures the most distinct and informative features of real-world designs and their parameter constraints.

2.2 Deep Generative Models

DGMs have proven to be successful in generating high-dimensional data in a completely unsupervised setting. Ideally, a good generative model should learn meaningful and compact representations for a qualitative and diverse generation. Two of the most popular generative models are Variational Autoencoders (VAEs) [19] and Generative Adversarial Networks (GANs) [12]. Although GANs are relatively better at synthesizing realistic data, they are notoriously difficult to train and often result in unstable models. VAEs on the other hand, are easier to train and converge faster. In addition, they are more successful than GANs in creating compact and effective representations in a continuous latent space. Such representations are useful for transferring specific characteristics and allowing control over the synthesis task, making it more productive and interactive [21]. To obtain a better trade-off in quality and training, we implement different models based on GAN and VAE frameworks for airfoil generation and systematically evaluate them to find the best suitable model for our application. In the following sections, we explain the theoretical foundations of both models in detail.

Variational Autoencoder VAEs are rooted in Bayesian inference i.e., they project the underlying training data distribution onto a distributed latent space that comprises independent factors of variation in the data. At inference, a VAE allows us to sample from the latent space to generate novel data that ideally resembles real data. Consider x to be our input data and z to be a latent vector. The VAE objective is to model the distribution of x which can be formulated as shown in Eq. 1, where $p(x|z)$ is known as conditional likelihood, and $p(z)$ is the prior distribution.

$$p(x) = \int p(x|z)p(z)\,dz \tag{1}$$

Computing the conditional likelihood requires computation of an unknown quantity known as the posterior of the true data $p(z|x)$. The VAE uses an encoder network to parameterize the variational approximation of the posterior distribution. The conditional likelihood is parameterized with a decoder network. The loss function for a VAE is given by Eq. 2 where the first term represents the reconstruction loss, and the second term is the KL divergence that minimizes the distance between the posterior and the prior. The prior distribution is usually a simple distribution, such as the standard normal distribution.

$$L = \mathbb{E}_{q(z|X)}[-\log p(X|z)] + KL\left(q(z|X)||p(z)\right) \tag{2}$$

Generative Adversarial Networks Generative adversarial networks (GANs) [12, 28] also model the true data distribution $p(x)$ but use adversarial learning instead. A typical GAN network comprises two components - a generator G and a discriminator D. A generator generates new samples using a low-dimensional noise vector and aims to fool the discriminator whose task is to distinguish real samples from fake ones (samples generated by the generator). G and D are

trained as a min-max in an adversarial fashion where each component strives to be better than the other network at their respective tasks. The GAN objective is given as

$$\min_{G} \max_{D} E(G, D) = \mathbb{E}_{x \sim P_{data}}[\log D_x] + \mathbb{E}_{z \sim P_z}[\log(1 - D(G_x))], \tag{3}$$

where x is sampled from the real data distribution P_{data}, z represents the noise vector sampled from the noise distribution P_z, and $G(z)$ is the fake distribution.

2.3 Learning Disentangled Representations

Modeling real-world data using generative models like GANs and VAEs creates a low-dimensional latent space. Ideally, this latent space represents the most crucial and distinct features of the data. In the case of VAEs, choosing an isotropic Gaussian prior has a latent space where every dimension is independent and produces what is generally known as a disentangled latent space. A latent space is called disentangled if each of its dimensions represents one and only one underlying factor of variation in the data [16]. The disentangled latent space enables interactive and controlled generation by allowing us to change specific features or to obtain data having certain features from generative distribution. Unfortunately, interactive generation is not possible with GANs as they cannot produce a disentangled representation of the data. Instead the representation is entangled, making it hard to interpret [7].

Several extensions based on the VAE framework, such as β-VAE [15], FactorVAE [17], and those based on the GAN framework, including InfoGAN [7], have been proposed to obtain a better generation quality and better disentanglement. Higgins et al. [15] and Chen et al. [4] provide simple modifications to the original VAE objective to achieve a better trade-off between generation quality and disentanglement, whereas Kim and Mnih [17] achieve this with the help of a discriminator network. InfoGAN, on the other hand, uses additional latent codes to encode some generative factors from the training data to encourage disentanglement. Other approaches, such as IDGAN [21], combine VAE and GAN frameworks to produce effective disentanglement and generate high-quality images. We explore several generative models such as β-VAE [15], DCGAN [28], and FactorVAE [16] and compare them for several generation aspects. Through evaluation of these models we find out that β-VAE is much simpler, faster and more reliable for high quality generation and disentanglement. Further, we compare our conditional β-VAE model with the popular BézierGAN model which is based on InfoGAN. In the next section, we explain our approach for reliable generation in more details.

3 Approach

Our goal is to produce high-quality airfoil designs that adhere to their performance characteristics while also producing disentangled representations of the

designs. Although GANs and VAEs can synthesize high-quality data, particularly images, their potential to synthesize parametric shapes has yet to be fully investigated. Therefore, we propose a conditional parameterization framework for synthesizing new airfoil designs constrained by specific C_L/C_D values based on GAN and VAE networks to obtain high-quality and diverse designs with disentangled representations. Conditioning the airfoil designs on their C_L/C_D values provides a mapping between the C_L/C_D and the intrinsic characteristics of the designs. Thus, we develop conditional versions of β-VAE, FactorVAE, and DCGAN networks.

Functional airfoils need to have smooth and watertight curve. To obtain such curves, our framework first constructs a binary image representation of the airfoils. The conditional DGMs are trained using these binary representations, along with their C_L/C_D values. In the following part, we discuss the data representation and conditional parameterization more deeply.

3.1 Data Representation

The parameterization of airfoils is an important stage in aircraft design. The shape, curvature, and edges of the airfoil have a significant impact on the aircraft's aerodynamic properties and the flow fields around it, influencing the optimization outcomes [41]. As a result, accurately modeling the airfoil designs, including all of the minute details regarding their geometries, is vital. In the UIUC dataset [33], airfoils are represented by a defined set of discrete design variables. These may be insufficient for complex airfoil shapes. As a result, we convert the UIUC data into binary fields in order to create smooth and watertight surfaces that preserve the geometry and its details perfectly. More importantly, the smooth curves enable us to sample as many points from the geometries as necessary. The airfoils in the UIUC dataset are first mapped onto a high-resolution Signed Distance Field (SDF) [1] and then converted to binary fields. As binary fields are equivalent to 2D images, we can learn their underlying features using convolutional operations [32] in the same way as with images. Another reason to map data onto an SDF is to have a common representation for 2D and 3D objects, which is challenging to do with alternative data formats. However, due to the lack of publicly available 3D airfoil data, we have limited our research to 2D airfoil designs.

Signed Distance Field An SDF [1] is formed using a signed distance function that calculates and assigns a distance to each point in space, with positive distance for points outside the shape, negative distance for points inside the shape, and zero distance for points on the shape. Mathematically, a signed distance function for a set of points Q is given by the distance d of all the points $q \in Q$ from the shape boundary ω as shown in Eq. 4.

$$SDF(q) = \begin{cases} d(q, \partial\omega) & q \notin \omega \\ 0 & q \in \partial\omega \\ -d(q, \partial\omega) & q \in \omega \end{cases} \tag{4}$$

Binary Fields The distances to all points within and outside the object are essential only in applications where global geometry is required. However, for airfoil parameterization, obtaining only the isosurface that represents the airfoil shape suffices. Hence, we modify the signed distance function to obtain a binary field, such that, if ω is the shape boundary, then the points on and inside the shape boundary form an isosurface, whereas points outside the shape boundary have a distance of one. Eq. 5 represents the modified signed distance function for binary fields:

$$SDF_{\text{binary}}(q) = \begin{cases} 1 & q \notin \omega \\ 0 & q \in \partial\omega \\ 0 & q \in \omega \end{cases} \tag{5}$$

We fix all distances to be equal to one because it is inconsequential to know how far the point lies outside the boundary. The obtained binary signed distance fields are analogous to binary images except that they are obtained as a result of binary SDF.

3.2 Conditional Parameterization using Generative Models

We enable the conditional synthesis of new airfoil designs by providing the C_L/C_D value as a condition to the generative component of the network. The condition is enforced by concatenating C_L/C_D values c to the latent vector $\mathbf{z}$. c is a real-valued number between 0 and 1 which is obtained as a result of the normalization of C_L/C_D values. Unlike other conditional models [35] that use a discrete label as a condition, the C_L/C_D values are continuous real-valued numbers. As a result of such conditional synthesis, the generative component of the implemented DGMs acts as an implicit parametric function that can generate airfoils for a given C_L/C_D condition. At inference, we can sample a noise vector from a Gaussian distribution $\mathcal{N}(0, 1)$ to which we can append any C_L/C_D value between zero and one and generate the design that matches the condition. We can also combine desired shape characteristics from our learned disentangled latent space to customize the airfoil designs. This enables interactive design synthesis and quick prototyping of desirable shapes. Our method can produce sharp, smooth, and desirable airfoils without any smoothing function and without having to learn any explicit parameters like control points for shape synthesis or separate latent codes for disentanglement as used in BézierGAN [6].

4 Experimental Results

We explore three DGMs – Deep Convolutional GAN (DCGAN) [28], β-VAE [15] and FactorVAE [16] – which are known to generate plausible images. We assess the generated designs quantitatively using the manifold-based metrics density and coverage. We also demonstrate the quality of disentanglement across different dimensions of the latent space.

4.1 Data Preparation

Obtaining Binary Fields We use 2D airfoil designs from the UIUC dataset [33] comprising nearly 1,600 diverse airfoil designs. It is a public dataset and is widely used for aerodynamic research. Also, obtaining C_L/C_D values using numerical simulation software is possible using this data, which aligns well with our goal. Examples of airfoil designs and their C_L/C_D values, are shown in the appendix[3]. Each airfoil design in the UIUC dataset is represented by a sequence of discrete x- and y-coordinates along the airfoil curve. The order of coordinates of every curve starts from the trailing edge point, followed by points along the upper surface of the airfoil towards the leading edge, the leading edge point and then the points along the lower surface of airfoil from leading edge towards the trailing edge. The edges and surfaces can be seen in Figure 1. The original format of the data produces rough boundaries or requires additional functions for smoothening the curves. To overcome these challenges, we convert each airfoil curve into a binary SDF of size 500×500, (refer section 3.4 for data conversion details).

Obtaining C_L/C_D The C_L/C_D value represents the lift-to-drag ratio. It can be obtained using simulation software to simulate the necessary flow fields under the required settings to obtain optimal airfoil performance. For all the airfoils in the dataset, we use XFOIL simulation [10] by setting the Reynolds number, $\mathrm{Re} = 5 \times 10^5$, the Mach number, $\mathrm{Ma} = 0.0$ and the angle of attack, $\alpha = 3°$ for around 1,200 airfoils. Please refer the appendix for more information about the distribution of the C_L/C_D values for all 1,200 airfoil designs. To condition on these values for generation, we normalize them to be between zero and one.

4.2 Evaluation Metrics

Density and coverage (DnC) [26] are used as metrics to assess our model's performance in terms of quality and diversity. Inception Score (IS) [30] and Fréchet Inception Distance (FID) [14] are some of the other metrics used to measure the overall quality of generation, but they cannot distinguish quality from diversity. For example, it is highly impractical if a generative model generates images that are very similar or generates the same image every time, even if the quality of generation is good. In that sense, IS and FID are highly uninformative. Furthermore, Kynkäänniemi et al. [20] show that IS and FID are unreliable for evaluating generative models because they do not correlate well with the image quality and produce an inconsistent evaluation.

On the other hand, DnC [26] overcame the earlier metrics' shortcomings. They are **automatic evaluation** techniques that directly compare the fake (generated) data distribution to the real, allowing us to see how well the generated distribution matches the training data. Unlike IS and FID, which rely on activations of a pre-trained Inception model based on ImageNet data [8], DnC are

[3] `https://github.com/aeroshapesynthesis/constraint_parametererization_airfoils/blob/main/Appendix.pdf`

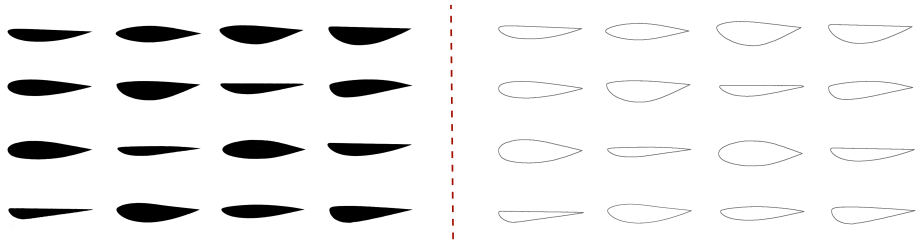


Fig. 2. Randomly sampled airfoil designs generated by β-VAE. The designs are realistic and have sharp and watertight boundaries.

independent of any dataset or model, giving a straightforward and clear evaluation method. The main idea behind DnC is to compare the manifold of real samples to the manifold of fake samples, and then quantify the quality and diversity of generated samples based on how the fake samples are placed around the real samples. Any intermediate layer, especially the fully connected layers of the generative part of the network (for example, the generator in the GAN or the decoder in the VAE), can be used to create these manifolds. More information on the metric and it's mathematical definition is given in the appendix.

4.3 Training

We train the β-VAE, FactorVAE, and DCGAN models on the UIUC airfoil dataset after preprocessing airfoils and extracting their C_L/C_D values. There are approximately 1,100 airfoils divided into a training set of 900 and a test set of 200. All models are implemented using Pytorch [27] and are trained on a single Nvidia Tesla [23] V100-SXM2 32 GB GPU. The batch size is 16 and is kept the same for all models. The learning rate for encoder and decoder in VAE, β-VAE and FactorVAE is 10^{-4}, the generator in DCGAN is $2 * 10^{-4}$ and for discriminator in DCGAN and FactorVAE is 10^{-4}. Adam optimization [18] is used for the training of all models because it can handle sparse gradients and combines the best properties of the AdaGrad [11] and RMSProp [25] algorithms. VAEs are generally stable to train and converge faster than GANs; hence we train β-VAE and FactorVAE for 300 epochs and DCGAN for 500 epochs. Hyperparameter optimization is an important part of training; therefore, using validation set we heuristically search for the best hyperparameters for each model. A brief note on hyperparameter optimization is included in the appendix.

4.4 Qualitative Results

A crucial first step is to visually inspect the results because the generated images maybe distorted or blurry, and these problems are difficult to address using quantitative analysis. From qualitative inspection, we observed that the β-VAE model with a latent dimension of 25 generates the most plausible images with

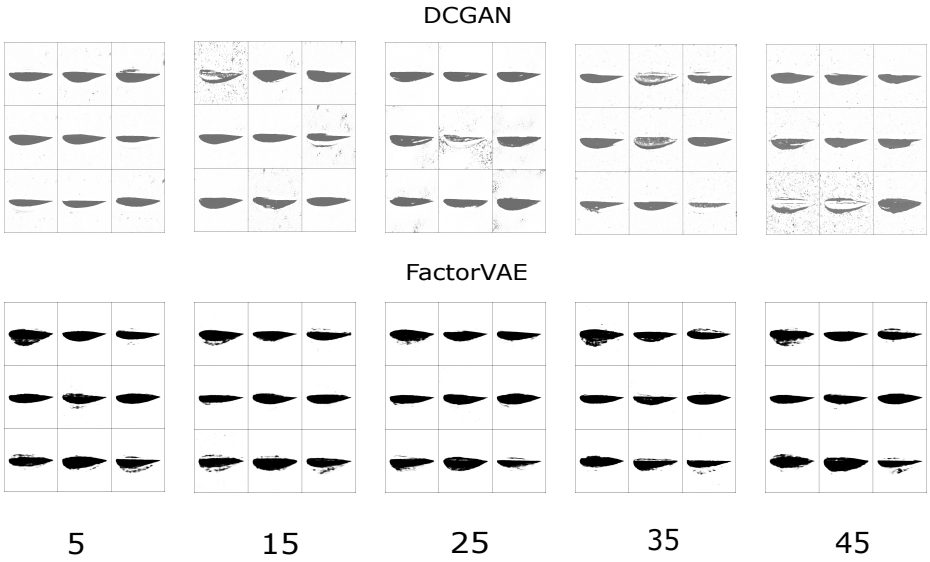


Fig. 3. Airfoil designs generated by DCGAN and FactorVAE for different latent dimension sizes. Heuristic hyperparameter search does not improve the quality of generated designs.

sharp and watertight boundaries. Figure 2 shows the airfoils and their extracted boundaries (curves). From visual inspection it is evident that the generated airfoils are very realistic as it is difficult to distinguish between real airfoils and airfoils generated using β-VAE. In the case of DCGAN and FactorVAE, the quality of generation is extremely poor as the generated designs are not sharp with closed boundaries. We tune all the models for different latent dimension sizes to find a better fit. But for DCGAN and FactorVAE, the quality of generated airfoils remain poor for all latent dimensions. Figure 3 shows airfoil designs generated using DCGAN and FactorVAE. Since, all the generated airfoils for all latent dimensions are distorted for both the models, these airfoils can't be used to extract watertight boundaries and thus, cannot be of any practical use. As β-VAE model outperforms other models, we quantitatively evaluate its results using DnC.

4.5 Quantitative Results: Analysis of Airfoil Designs

In this section, we evaluate our model using DnC and compare it to the state-of-the-art BézierGAN model. For the β-VAE model, we randomly sample from the prior and generate C_L/C_D values. The airfoils are then generated using the decoder and the latent vectors are then extracted from the encoder to calculate the DnC scores. In total, we extract latent vectors for 500 real and 500 fake images and calculate the DnC scores. Figure 4 shows the DnC scores for the β-VAE models for latent dimension size from 5 to 50 with a step size of 5. For

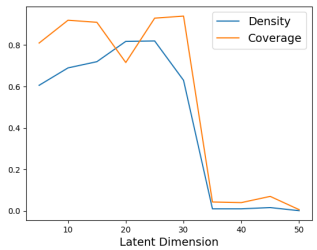


Fig. 4. Density and Coverage of designs generated using β-VAE model for different latent dimensions. Latent dimension size of 25 produces high quality designs.

Metric	β-VAE	BézierGAN
Density	**0.82**	0.63
Coverage	**0.93**	0.038

Table 1. Comparison of β-VAE vs BézierGAN for quality and diversity of generated designs. β-VAE outperforms BézierGAN on all fronts.

lower latent dimension sizes (size below 30), the quality and diversity is better than those of higher sizes. This is understandable because in high dimensional spaces, the curse of dimensionality applies and the data becomes more sparse. For latent dimension size of 25, we can see that the density is the highest which positively correlates with the observations from the qualitative results. Coverage is also high (slightly lower than the highest number) indicating more diversity in the generated samples.

We compared our conditional β-VAE model (having latent size of 25) with the popular BézierGAN model which also aims at generating novel airfoils, but without any constraints. Table 1 shows the comparison between the two models. The β-VAE model outperforms BézierGAN in terms of quality and diversity of the generated designs.

4.6 Quality of Reconstructions

For successful parameterization, representing the original airfoil design accurately is crucial. For traditional parameterization techniques like PARSEC [34], MACROS DR [36], and CST [3], the geometric error between the actual airfoil and the approximated airfoil is calculated using Root Mean Square (RMS) [41]. However, the input and output in our approach is a binary field. While obtaining curves from the generated binary fields, the points along the curve are sampled randomly and not in any particular order. Thus, RMS is not a suitable technique to measure parameterization accuracy because the points that coincide with the coordinates of an input airfoil cannot be obtained. However, the difference in geometries may be computed by directly comparing the binary fields of the orig-

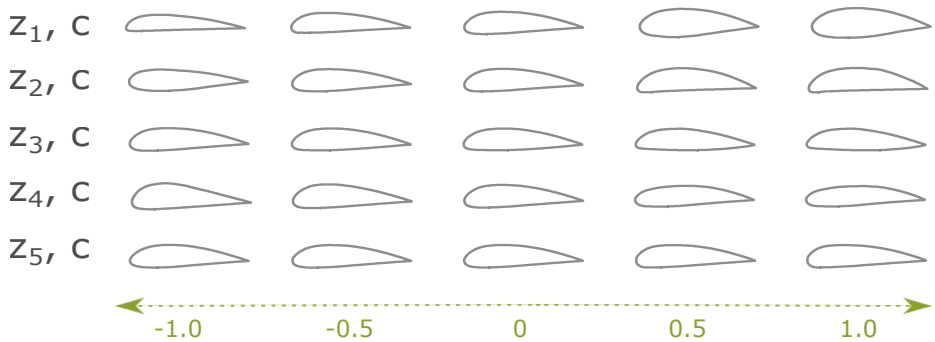


Fig. 5. Traversing across different dimensions of a latent vector using β-VAE. It shows automatic disentanglement based on geometric traits like shape, size and curvature in a distributed and a continuous latent space.

inal and reconstructed designs using Intersection over Union (IoU) [29]. IoU is a popular metric in the computer vision community to calculate the similarity of any two 2D/3D objects. Let A and B be any two 2D/3D volumes of objects, then IoU is defined as follows:

$$\frac{|A \cap B|}{|A \cup B|} \tag{6}$$

We can see that β-VAE generates designs with the sharpest and most watertight boundaries by comparing the outputs of the models shown in Figure 2. Thus, to calculate the IoU between actual and approximated airfoil designs, we use the reconstructions obtained from a β-VAE model with β=60 and a latent dimension of 25. We calculate IoU for a whole image because as long as every input and output has just one smooth and watertight airfoil without any deformity, IoU can effectively calculate similarity. Figure 3 shows that there are possibilities of distortion while generating airfoil designs and that it might result in multiple broken objects. Hence we first run all the samples through an off-the-shelf contour detection technique and select 50 samples for which only one contour is detected. The average IoU is as high as 0.975, which indicates that the β-VAE model can accurately reconstruct original airfoils, which makes the decoder of β-VAE a good parametric function. Examples of real vs reconstructed airfoils with their IoU are shown in the appendix.

4.7 Disentanglement

We illustrate latent traversal through several dimensions of the noise vector as well as interpolations between different samples to highlight our model's potential to disentangle the latent space based on the geometric properties of the airfoils and their C_L/C_D values. Latent traversal (-1 to 1) across different dimensions of a noise vector is shown in Figure 5. For each row in Figure 5, we vary one dimension in the noise vector while keeping all other dimensions, including C_L/C_D value fixed to see the contribution of the varying dimension in

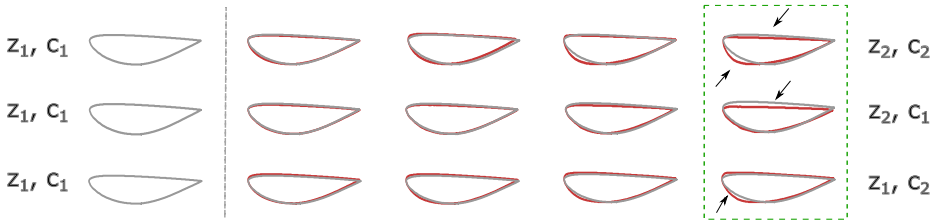


Fig. 6. Interpolations based on noise and C_L/C_D values. Noise vectors, n_1 and n_2 are randomly sampled and C_L/C_D values c_1 and c_2 are 0.1 and 0.99 respectively. Changing both noise and C_L/C_D (top) changes size and curvature, whereas, changing only noise (middle) changes the shape without affecting the curve and changing only C_L/C_D (bottom) changes the leading curvature without affecting the shape.

generating of airfoils. Different dimensions depict varying and independent factors of variation based on the size, shape, nature of leading and trailing edges of airfoils. The latent traversal is very smooth, and the model can generate feasible designs for any noise sample, which shows that the latent space is continuous and interpretable.

We further demonstrate in Figure 6 that our model can distinctly disentangle based on noise and C_L/C_D values by disentangling different properties of the airfoils. We sample two noise vectors and two C_L/C_D values and interpolate between the two samples, first by changing both noise and C_L/C_D, second by only changing the noise while keeping the C_L/C_D fixed and last by changing C_L/C_D while keeping the noise fixed, as shown in Figure 6. From the first row of Figure 6 we can observe that, changing both noise and C_L/C_D changes the size and the curve at the leading edge. In the second row, we can see that for a fixed C_L/C_D, the change in noise only changes the size of the airfoil and no change in any curves or edges. From the last row we can observe that by changing only the C_L/C_D value while keeping the noise fixed only changes the curvature at the leading edge but no change in airfoil's size. Thus, our technique achieves successful disentanglement based on geometry as well as the performance characteristics (C_L/C_D) of the airfoils.

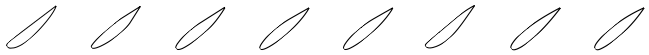
Test Airfoils								
Relative Error (%)	2.11	0.93	2.70	1.85	1.14	3.14	4.05	4.07
IoU	0.991	0.985	0.987	0.977	0.985	0.987	0.980	0.988

Table 2. The relative error between the C_L/C_D values of the original and reconstructed airfoils, along with their high IoU scores. Shown are random samples from the test set. The average relative error on the whole test set is XY.

4.8 Precision of C_L/C_D Conditions

We investigate in Table 2 if the C_L/C_D values of the real airfoils match with their reconstructed counterparts, to confirm if the reconstructed airfoils adhere to the C_L/C_D values that they were conditioned on. The reconstructed airfoils are first transformed to the data format required by XFOIL, and using the simulations in XFOIL, we obtain their C_L/C_D values. However, XFOIL is extremely sensitive to the coordinates and simulations therefore, not all airfoils converge and C_L/C_D for them cannot be obtained. Table 2 shows some examples of the converged airfoils from the test set, the IoU scores between them and their reconstructed counterparts and the relative error of the C_L/C_D value of the reconstructed airfoils. The accuracy of the reconstruction is very high, as can be observed from the high IoU values. The relative error between the C_L/C_D of the test airfoils and of the reconstructed airfoils is also low. Thus, the conditional β-VAE model can enforce the C_L/C_D condition with high precision during the reconstruction of a design.

5 Conclusion

In this paper, we proposed a hybrid DGM-CAE based framework for automated and constraint-based parameterization of airfoil designs while ensuring their mechanical functionality under the C_L/C_D value constraint. We show that the conditional β-VAE model outperforms several other popular generative models and is best at generating realistic and diverse airfoils with sharp, smooth, and watertight boundaries while also adhering to the C_L/C_D constraint. It can also disentangle several physical properties of the data enabling interactive airfoil generation much faster than traditional parameterization techniques. Our framework also outperforms the previous state of the art in terms of quality and diversity of the generated designs. To the best of our knowledge, this is the first at attempt of generating airfoil designs conditionally based on their C_L/C_D values, thereby creating a new baseline for such a constraint-based parameterization of aerodynamic shapes. In the future, this approach can be extended to parameterize 3D airfoils, and depending on the availability of data, many more additional constraints can be enforced.

Acknowledgements The second author of the paper was funded by the German Federal Ministry of Education and Research under the grant number 01IS20048. The responsibility for the content of this publication lies with the author. Furthermore, we acknowledge support by the Carl-Zeiss Foundation and the DFG awards BU 4042/2-1 and BU 4042/1-1.

References

1. Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K., Kaushik, S.: Prediction of aerodynamic flow fields using convolutional neural networks. Computational Mechanics **64**, 525–545 (2019)

2. Cagan, J., Campbell, M.I., Finger, S., Tomiyama, T.: A Framework for Computational Design Synthesis: Model and Applications. Journal of Computing and Information Science in Engineering **5**(3), 171–181 (09 2005)
3. Ceze, M., Hayashi, M., Volpe, E.: A study of the cst parameterization characteristics. In: 27th AIAA Applied Aerodynamics Conference (2009)
4. Chen, R.T.Q., Li, X., Grosse, R.B., Duvenaud, D.K.: Isolating sources of disentanglement in variational autoencoders. In: NIPS. vol. 31, pp. 2610–2620 (2018)
5. Chen, W., Ahmed, F.: Padgan: A generative adversarial network for performance augmented diverse designs. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. vol. 84003, p. V11AT11A010. American Society of Mechanical Engineers (2020)
6. Chen, W., Fuge, M.: Béziergan: Automatic generation of smooth curves from interpretable low-dimensional parameters. arXiv preprint arXiv:1808.08871 (2018)
7. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Advances in neural information processing systems **29** (2016)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
9. Derksen, R.W., Tim, R.: Bezier-parsec: An optimized aerofoil parameterization for design. Adv. Eng. Softw. **41**(7–8), 923–930 (Jul 2010)
10. Drela, M.: Xfoil: An analysis and design system for low reynolds number airfoils. In: Low Reynolds number aerodynamics, pp. 1–12. Springer (1989)
11. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(61), 2121–2159 (2011)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (oct 2020)
13. Grey, Z.J., Constantine, P.G.: Active subspaces of airfoil shape parameterizations. AIAA Journal **56**(5), 2003–2017 (2018)
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
15. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: International conference on learning representations (2017)
16. Kim, H., Mnih, A.: Disentangling by factorising. In: International Conference on Machine Learning. pp. 2649–2658. PMLR (2018)
17. Kim, H., Mnih, A.: Disentangling by factorising. In: International Conference on Machine Learning. pp. 2649–2658. PMLR (2018)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
20. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. Advances in Neural Information Processing Systems **32** (2019)

21. Lee, W., Kim, D., Hong, S., Lee, H.: High-fidelity synthesis with disentangled representation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 157–174. Springer International Publishing, Cham (2020)
22. Lepine, J., Guibault, F., Trepanier, J.Y., Pepin, F.: Optimized nonuniform rational b-spline geometrical representation for aerodynamic design of wings. AIAA journal **39**(11), 2033–2041 (2001)
23. Lindholm, E., et al.: Nvidia tesla: A unified graphics and computing architecture. IEEE micro **28**(2), 39–55 (2008)
24. Mirza, M., Osindero, S.: Conditional generative adversarial nets. ArXiv **abs/1411.1784** (2014)
25. Mukkamala, M.C., Hein, M.: Variants of rmsprop and adagrad with logarithmic regret bounds. In: International conference on machine learning. pp. 2545–2553. PMLR (2017)
26. Naeem, M.F., Oh, S.J., Uh, Y., Choi, Y., Yoo, J.: Reliable fidelity and diversity metrics for generative models. In: ICML. pp. 7176–7185. PMLR (2020)
27. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
28. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
29. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 658–666 (2019)
30. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Advances in neural information processing systems **29** (2016)
31. Samareh, J.A.: Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. AIAA Journal **39**(5), 877–884 (2001)
32. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks : the official journal of the International Neural Network Society **61**, 85–117 (2015)
33. Selig, M.S.: UIUC airfoil data site. Urbana, Ill. : Department of Aeronautical and Astronautical Engineering University of Illinois at Urbana-Champaign (1996)
34. Sobieczky, H.: Parametric airfoils and wings. Recent Development of Aerodynamic Design Methodologies. Notes on Numerical Fluid Mechanics (NNFM), vol 65. pp. 71–87 (1999)
35. Sohn, K., et al.: Learning structured output representation using deep conditional generative models. In: NIPS. vol. 28. Curran Associates, Inc. (2015)
36. Sripawadkul, V., Padulo, M., Guenov, M.: A comparison of airfoil shape parameterization techniques for early design optimization. In: 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference (2010)
37. Xu, K., Kim, V.G., Huang, Q., Mitra, N., Kalogerakis, E.: Data-driven shape analysis and processing. In: SIGGRAPH ASIA 2016 Courses. SA '16, Association for Computing Machinery, New York, NY, USA (2016)
38. Yang, C., Huang, F.: An overview of simulation-based hydrodynamic design of ship hull forms. Journal of Hydrodynamics, Ser. B **28**, 947–960 (12 2016)
39. Yasong, Q., Junqiang, B., Nan, L., Chen, W.: Global aerodynamic design optimization based on data dimensionality reduction. Chinese Journal of Aeronautics **31**(4), 643–659 (2018)

40. Yonekura, K., Wada, K., Suzuki, K.: Generating various airfoil shapes with required lift coefficient using conditional variational autoencoders (2021)
41. Zhu, F., Qin, N., Burnaev, E., Bernstein, A., Chernova, S.: Comparison of three geometric parameterization methods and their effect on aerodynamic optimization. In: Eurogen. pp. 758–772 (2011)