Aerodynamic Design Optimization and Shape Exploration using Generative Adversarial Networks

Wei Chen*, Kevin Chiu† and Mark D. Fuge‡
University of Maryland, College Park, Maryland, 20742

Global optimization of aerodynamic shapes requires a large number of expensive CFD simulations because of the high dimensionality of the design space. One means to combat that problem is to reduce the dimension of the design space—for example, by constructing low dimensional parametric functions (such as PARSEC and others)—and then optimizing over those parameters instead. Such approaches require first a parametric function that compactly describes useful variation in airfoil shape—a non-trivial and error-prone task. In contrast, we propose to use a deep generative model of aerodynamic designs (specifically airfoils) that reduces the dimensionality of the optimization problem by learning from shape variations in the UIUC airfoil database. We show that our data-driven model both (1) learns realistic and compact airfoil shape representations and (2) empirically accelerates optimization convergence by over an order of magnitude.

I. Introduction

Aerodynamic shape optimization is a necessary step in designing parts like aircraft wings and (propeller/rotor/turbine) blades. The bottleneck of most global optimization methods for aerodynamic design is the computational cost of the computational fluid dynamics (CFD) simulations. To combat this, surrogate-based modeling approaches are used [1–5] to reduce the number of simulations by balancing exploration and exploitation while sampling the design space. However, the computational cost of sampling the design space increases exponentially with the dimensionality of the design space due to the curse of dimensionality [6]. Previous research has looked into dimensionality reduction (DR) of the original parametric design space (*i.e.*, the space of designs represented by shape parameters such as B-spline control points). This permits faster exploration by capturing only those dimensions that either affect the final design's performance [7–11] or capture major shape variability [12–16]. But these DR models may not accurately capture the true variation that we observed in real-world airfoils, *e.g.*, those in the UIUC airfoils database. A vast amount of research on DR has been conducted in the field of machine learning, where deep neural networks such as variational autoencoders (VAEs) [17] and generative adversarial networks (GANs) [18] have successfully represented data from complex high-dimensional distributions, such as images, by using low-dimensional latent variables.

In this paper, we apply GANs to learn an interpretable low-dimensional space (*i.e.*, the *latent space*) that encodes how aerodynamic shapes vary. To avoid the limitation caused by shape parameterizations (*e.g.*, curve-fitting errors and the lack of representation flexibility), we learn directly the distribution of points along the curves instead of curve parameters (such as Bézier control points). However, naïve application of neural network techniques to airfoil designs does not work well because the output is noisy and does not conserve important continuity properties important for aerodynamic shapes. Therefore, we use Bézier-GANs [19] to generate smooth aerodynamic shapes.

II. Background

In this section, we introduce previous work on common algorithms used in aerodynamic design optimization (Sec. II.A), parameterization techniques (Sec. II.B), and methods for reducing design space dimensionality (Sec. II.C).

A. Optimization Methods

Aerodynamic design is, in large part, an optimization problem. One common objective is to find design variables that minimize the drag coefficient C_D, while maximizing or constraining the lift coefficient C_L [9, 11, 13]. There

*PhD Student, Department of Mechanical Engineering.
†PhD Student, Department of Mechanical Engineering.
‡Assistant Professor, Department of Mechanical Engineering.

are primarily three approaches for solving the optimization problem: evolutionary algorithms (EA), surrogate-based optimization (SBO), and automatic differentiation (AD) (otherwise known as adjoint methods).

1. Evolutionary Algorithms

Evolutionary algorithms (EA) are gradient-free optimization algorithms that mimic the process of biological evolution through mutation, recombination, and reproduction of different designs. Genetic algorithms (GA), a type of EA, is widely used in aerodynamic shape optimization [12, 15, 20]. Work has also been done to augment GA with the Bees algorithm [21] and adaptive mutation rates [22], resulting in more accurate optimization and/or faster convergence. Other EA methods applied in aerodynamic optimization are differential evolution [23] and particle swarm optimization [24, 25]. However, due to the large number of function calls needed in each generation, EAs can be prohibitively expensive computationally, especially if every evaluation requires a high-fidelity computational fluid dynamics simulation.

2. Surrogate-Based Optimization

Surrogate-based optimization (SBO) uses an inexpensive surrogate model to approximate the expensive function of the quantity of interest (QoI) (*i.e.*, the optimization objective). Bayesian optimization (BO) is a commonly used method for SBO. It consists of two components — a sampling method (*e.g.*, maximum expected improvement [26] or maximum upper confidence bound [27]) and a surrogate modeling method (*e.g.*, Gaussian process regression, also know as kriging [28]). In each iteration, the sampling method samples a point in the design space for evaluation of the QoI, and then that point and its QoI update the surrogate model. Compared to methods like genetic algorithms, surrogate-based optimization reduces the number of expensive CFD evaluations needed in aerodynamic shape optimization [7, 9, 12, 14, 29]. However, for a high-dimensional design space, the number of evaluations will still be inevitably high due to the curse of dimensionality [6, 30]. Note that in these cases, kriging can also be prohibitively expensive at the later stage when the model is trained on a large number of evaluated samples since its computational cost scales cubically with the sample size (though practical approximation methods do exist to reduce this cost).

3. Automatic Differentiation

Automatic differentiation (AD)—a generalization of adjoint methods used by the CFD community—provides a relatively fast and exact method of calculating numerical gradients. The computer records every elemental operation used to calculate a QoI ("forward pass") before reversing through this "tape" to determine the sensitivity of the QoI with respect to each parameter. Generally, gradient calculations are exact and have a computational cost within an order of magnitude of the forward pass. Because of this, previous work [31–37] have used AD for gradient calculations. Combined with optimization algorithms such as SQP [35, 36], steepest descent [34], and Newton- and quasi-Newton methods [31, 32, 37], AD can drastically accelerate gradient calculations in the optimization process, even in complex or turbulent models [31, 36, 37].

However, for optimization using state-of-the-art turbulence models such as Large Eddy Simulation, one cannot use adjoint methods because the chaotic butterfly exponential divergence of trajectories makes the adjoint ill-posed [38]. In addition, an AD gradient is only applicable at one point; thus, unlike *e.g.*analytical derivatives, where a single equation provides exact derivatives at any point, AD requires a forward pass before each new gradient calculation, contributing to a large portion of the optimization cost. In terms of memory, building the tape of operators can be expensive compared to, *e.g.*, a finite difference method. Additionally, as a method of gradient calculation, AD will still maintain the disadvantage inherent in gradient-based algorithms, *e.g.*, converging to local minima. As a workaround solution, Berguin *et al.* [10] use solutions to SBO as starting points for AD methods, hoping to find good local optima.

B. Shape Parameterization

Parameterization maps a set of parameters to points along a smooth curve or surface via a parametric function. Common parameterization for aerodynamic shapes includes splines (*e.g.*, B-spline and Bézier curves) [39–41], free-form deformation (FFD) [42, 43], class-shape transformations (CST) [44, 45], PARSEC [46, 47], and Bézier-PARSEC [48]. While this work does not study paramterization, we show the optimization performance of two paramterization approaches, namely nonuniform rational B-splines (NURBS) [49] and PARSEC [46], in comparison to our proposed method.

Usually during design optimization, parameters are sampled to generate design candidates [29]. There are two main

issues when optimizing these parameters from conventional parameterization: (1) one has to guess the limits of the parameters to form a bounding-box within which the optimization operates, and (2) the design space dimensionality is usually higher than the underlying dimensionality for representing sufficient shape variability [50] —*i.e.*, to capture sufficient shape variation, manually designed shape parameterizations require higher dimensions than are strictly necessary.

C. Design Space Dimensionality Reduction

It is computationally expensive to search for solutions in the design space directly due to the space's high dimensionality. Factor screening methods [51, 52] are used to select the most relevant design variables for a design problem while fixing the rest as constant during optimization. These methods fail to consider the correlation between design variables. Thus, researchers have found ways to capture the low-dimensional subspace that identifies important directions with respect to the change of *response* (*i.e.*, QoI or performance measure) [7–11]. This response-based dimensionality reduction usually has several issues: 1) it requires many simulations when collecting samples of response gradients; 2) variation in gradients can only capture non-linearity rather than variability in the response, so extra heuristics are required to select latent dimensions that capture steep linear response changes; 3) the learned latent space is not reusable for any different design space exploration or optimization task (*i.e.*, when a different response is used); and 4) the linear DR techniques applied in previous work may fail on responses with non-linear correlation between partial derivatives.

The first three issues can be avoided by directly applying DR on design variables without associating them with the response. Note that by doing this, we are assuming that if changes in a design are negligible, changes of responses are also negligible. In the area of aerodynamic design, researchers use linear models such as proper orthogonal decomposition (POD), also known as principal component analysis (PCA) [14–16], and nonlinear models like generative topographic mapping [12, 13] to reduce the dimension of design variables. More work on DR has been done in other fields such as image processing and computer graphics [53, 54], where DR is used for generating and visualizing data. Deep neural networks such as VAEs and GANs have been widely applied in these areas to learn the latent representation of data. These methods are known for their ability to learn complex high-dimensional data distributions. Our work extends this class of techniques by considering generation of smooth geometries such as those needed in spline-based representations.

Note that as DR models map latent variables to shapes, we can treat the latent variables and the mapping as parameters and the parametric function. Thus, in a broader sense, we will also refer to these methods as parameterization in Sec. VI.

III. Obtaining Disentangled Latent Representation Using Generative Adversarial Networks

We use a method based on GANs [55] to train a generative model that synthesizes aerodynamic shapes from interpretable low-dimensional latent codes. GANs are one type of deep neural network architecture which consists of two components: a generator and a discriminator. The generator takes in random noise from some known prior distribution P_z. Its objective is to generate samples from the desired distribution (*i.e.*, data distribution P_{data}). The discriminator takes in a sample (either from the training data or synthesized by the generator) and predicts the probability of the sample coming from the training data. The generator tries to make the generative distribution P_G look like P_{data} to fool the discriminator; the discriminator tries not to be fooled. GANs achieve this by minimizing the objective:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim P_{data}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim P_z}[\log(1 - D(G(\boldsymbol{z})))] \tag{1}$$

where D is the discriminator, and G is the generator. Both components improve via training until the discriminator cannot differentiate between real and fake inputs, implying that the generative distribution resembles the data distribution.

Standard GANs are not built for learning latent representations; thus, they cannot be used to reduce the dimensionality of the design space. To compensate for this weakness, InfoGANs [56] encourage interpretable latent representations by maximizing the mutual information between some noise variables (called *latent codes*) and the generated samples. Thus, InfoGAN's generator takes both latent codes $\boldsymbol{c}$ and random noise $\boldsymbol{z}$ as inputs. Unfortunately, it is hard to directly maximize the mutual information $I(\boldsymbol{c}; G(\boldsymbol{c}, \boldsymbol{z}))$, so instead an InfoGAN approximates the solution by maximizing a lower bound. In practice, this is realized by adding an extra fully connected layer to the discriminator to predict the latent codes. Please refer to Ref. [56] for more details about the InfoGAN. We build upon this line of work below, extending it to spline-based geometry.

IV. Spline-Based Shape Synthesis

Typical approaches to generative shape models (such as GANs) represent shapes as a collection of discrete samples (*e.g.*, as pixels or voxels) owing to the their original development in the computer vision community. For example, a naïve way of synthesizing shapes like airfoils would be to generate this *discrete representation* directly using the generator, such as generating a fixed number of coordinates sampled along the airfoils boundary curve (*e.g.*, Fig. 2, right). However, in practice, airfoils typically possess substantial smoothness/continuity and are typically represented using parametric curve families like splines, Bézier curves, or NURBS surfaces. The naïve GAN representation of predicting discretized curves from the generator usually (1) creates noisy curves that have low smoothness and (2) have parametric output that is harder for humans to interpret and use in standard CAD packages compared to equivalent curve representations (*e.g.*, Bézier curves). This creates problems, particularly in aerodynamic shape synthesis.

To solve this issue, we modified the InfoGAN's generator such that it only generates smooth shapes that conform to Bézier curves. We call this generative adversarial network a Bézier-GAN [19]. As shown in Fig. 1, most of its architecture is adapted from the InfoGAN. However, before outputting discrete coordinates along the curve, the generator synthesizes *control points* P, *weights* w, and *parameter variables* t of rational Bézier curves. The last layer—the Bézier layer—converts this rational Bézier curve representation into discrete representation X:

$$X_j = \frac{\sum_{i=0}^{n} \binom{n}{i} t_j^i (1-t_j)^{n-i} P_i w_i}{\sum_{i=0}^{n} \binom{n}{i} t_j^i (1-t_j)^{n-i} w_i}, \quad j = 0, \ldots, m \tag{2}$$

where n is the Bézier degree, and the number of discrete points to represent the curve is $m+1$. Since variables $\{P_i\}$, $\{w_i\}$, and $\{t_j\}$ are differentiable in Eq. 2, we can train the network using back propagation. Figure 2 compares synthesized shapes with and without using a Bézier layer.

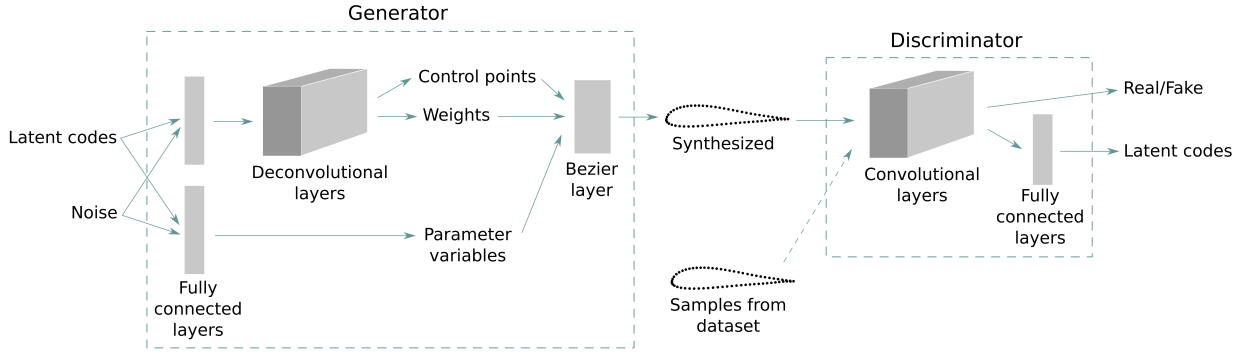

Fig. 1 Model architecture of the Bézier-GAN.

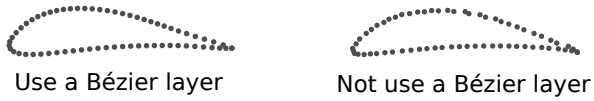

Fig. 2 Synthesized airfoils using a generator with and without a Bézier layer.

V. Optimization over the Learned Latent Space

A. The Optimization Problem in the Latent Space

The optimal aerodynamic shape can be solved by $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$, where $\boldsymbol{x}$ is an aerodynamic shape (expressed in this case by the latent codes and the Bézier curve parameters) and $f(\boldsymbol{x})$ is some performance measure defined over $\boldsymbol{x}$ (*e.g.*, lift, drag, *etc.*). Since the function f is usually non-convex, methods such as EA or SBO are often used for optimization [57–60]. These methods search for the global optimum by exploring the design space $\mathcal{X}$. However, since $\mathcal{X}$ is usually high-dimensional, it takes many performance evaluations to find the optimal solution due to the curse of

dimensionality [6]. Since we can modify $\boldsymbol{x}$ by changing the latent code $\boldsymbol{c}$, which has a lower dimension than $\boldsymbol{x}$, finding the optimal shape $\boldsymbol{x}^*$ is equivalent to finding an optimal latent code $\boldsymbol{c}^*$. Thus, we solve the following problem instead:

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} h(\boldsymbol{c}) = f(\mathbb{E}_{\boldsymbol{z}\sim P_z}[G(\boldsymbol{c}, \boldsymbol{z})]) \tag{3}$$

and then use $\boldsymbol{c}^*$ to synthesize the optimal shape $\boldsymbol{x}^* = G(\boldsymbol{c}^*, \boldsymbol{z})$.

We can then apply SBO to find $\boldsymbol{c}^*$ instead of $\boldsymbol{x}^*$. There are two major components in SBO: an inference model and an acquisition function. In this paper we use a SBO method called Efficient Global Optimization (EGO) [26], which uses Gaussian process (GP) regression [28] as the inference model, and expected improvement (EI) [26] as the acquisition function:

$$\mathbb{E}[I(\boldsymbol{c})] = \mathbb{E}[\max(h_{min} - h(\boldsymbol{c}), 0)] \tag{4}$$

where h_{min} is the current best function value. At each step t of EGO we want to find $\boldsymbol{c}^{(t)}$ that is expected to best improve upon the current optimal solution:

$$\boldsymbol{c}^{(t)} = \arg\max_{\boldsymbol{c}} \mathbb{E}[I(\boldsymbol{c})] \tag{5}$$

Now with the ingredients of GP regression and EI, the EGO process simply repeats the following steps:

1) Estimate the function h by using GP regression;
2) Compute EI using Eq. 4 and the learned GP model;
3) Search for the point $\boldsymbol{c}^{(t)}$ that has the highest EI (solving Eq. 5);
4) Evaluate the function h at $\boldsymbol{c}^{(t)}$, and add the new $(\boldsymbol{c}^{(t)}, h(\boldsymbol{c}^{(t)}))$ pair into the dataset for learning GP regression.

B. Unbounded Bayesian Optimization

To find $\boldsymbol{c}^{(t)}$ in Step 3, we can use gradient-based optimization algorithms like BFGS [61–64]. However, these methods may get stuck in local optima and are unstable if operated in an unbounded space (*i.e.*, the solution may go too far from feasible regions and thus will diverge). Random search is a simple alternative approach that searches $\boldsymbol{c}^{(t)}$ within fixed variable bounds; however, the optimal solution may be located outside those bounds. To circumvent these issues, we search for the solution near $\boldsymbol{c}^{(t-1)}$ (*i.e.*, the point evaluated at step $t-1$) without requiring a boundary. Specifically, we search for $\boldsymbol{c}^{(t)}$ among samples drawn from the distribution $\mathcal{N}(\boldsymbol{c}^{(t-1)}, \sigma^2)$, where σ controls the dispersion of the drawn samples (Fig. 3). Combined with the EI criteria, each iteration of the search area moves in the direction which is expected to improve the current optimal solution. This moving search area eliminates the limitation of variable bounds. A larger σ encourages exploration and prevents the solution from getting stuck in local optima while a smaller σ encourages exploitation and refines the current optimal solution. We use a decreasing σ over iterations (*i.e.*, in each iteration, multiply σ by a constant γ that is close to but smaller than 1), so that the algorithm first explores then focuses more on exploitation.

VI. Experiment: Airfoil Synthesis and Shape Optimization

In this section we demonstrate our method via an airfoil optimization task. Rather than targeting one specific airfoil model (*e.g.*, the NACA 0012 airfoil in Ref. [11] or the RAE 2822 airfoil in Ref. [12]) and its perturbations, we search for the optimal design within all the existing major airfoil models. We show that Bézier-GAN learns realistic shape variations from these airfoil models and that optimizing in the latent space accelerates convergence.

A. Dataset and Preprocessing

We use the UIUC airfoil database* as our training data for the Bézier-GAN. It provides the geometries of nearly 1,600 real-world airfoil designs, each of which is represented by discrete coordinates along their upper and lower surfaces. The number of coordinates for each airfoil is inconsistent across the database, so we use B-spline interpolation to obtain consistent shape representations. Specifically, we interpolate 192 points over each airfoil with the concentration of these points along the B-spline curve based on the curvature [39]. The preprocessed data are visualized at the top of Fig. 4.

*http://m-selig.ae.illinois.edu/ads/coord_database.html

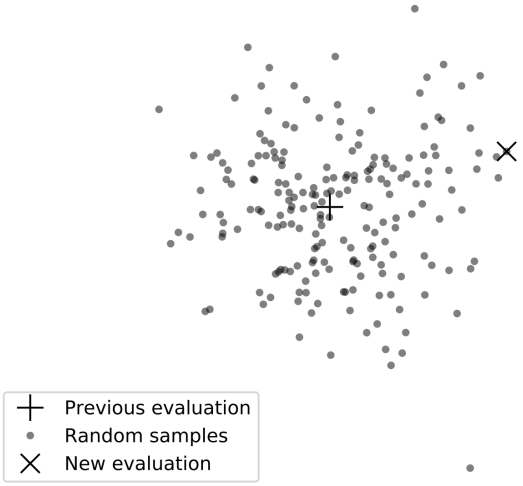

Fig. 3 Unbounded sampling in Bayesian optimization.

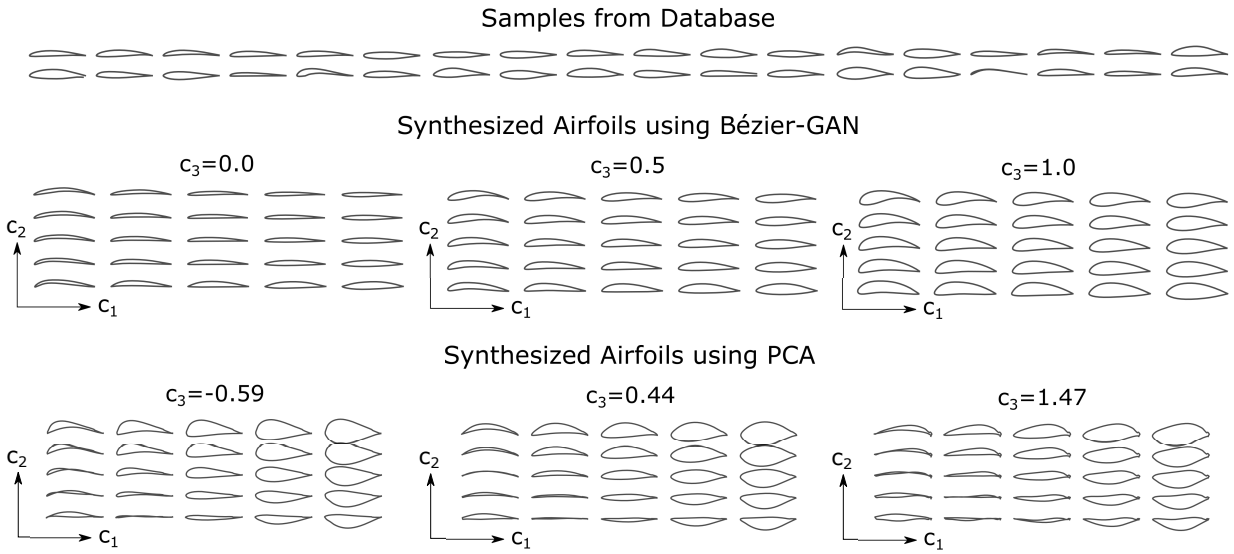

Fig. 4 Examples in the airfoil database and synthesized airfoil shapes in three-dimensional latent spaces (visualized by uniform slices of multiple two-dimensional spaces).

B. Dimensionality Reduction

We build a Bézier-GAN model based on the architecture in Fig. 1. The latent codes are from a three-dimensional uniform distribution, and the input noise is from a ten-dimensional Gaussian distribution. In the discriminator, we use six one-dimensional convolutional layers followed by fully connected layers to predict latent codes and the probability of the input data coming from the dataset. For the generator, we use three one-dimensional deconvolutional layers [65] to predict the control points $\{P_i|i = 0, ..., n\}$ and the weights $\{w_i|i = 0, ..., n\}$, and three fully connected layers followed by a softmax activation to predict discrete differences between parameter variables $\{t_{j+1} - t_j|j = 0, ..., m - 1\}$. Interested readers can refer to detailed network architectures and hyperparameters in our Tensorflow implementation available on

Github[†].

We optimize the Bézier-GAN using an Adam optimizer [66] and train it on a Nvidia Titan X GPU. The wall-clock training time is about 1 hour, and the inference takes less than 15 seconds.

Figure 4 shows synthesized airfoil shapes by linearly interpolating points in the latent space. The middle subplot shows that airfoils synthesized by Bézier-GAN are realistic and capture most variation in the airfoil dataset. We also obtained an interpretable latent space: the horizontal axis (c_1) captured the leading edge angle, the vertical axis (c_2) captured the trailing edge angle, and the third axis (c_3) captured the thickness.

We use PCA as a baseline DR method to compare the synthesis quality. The latent space is also set to three-dimensional. The results of PCA are shown at the bottom of Fig. 4[‡]. Compared to Bézier-GAN, PCA shows the limitations of a linear DR model by synthesizing unrealistic designs in some regions of the latent space.

C. Optimization

Our optimization objective is to maximize the lift to drag ratio C_L/C_D. We use XFOIL [67] to compute the lift and drag coefficients C_L and C_D.[§] The XFOIL operation conditions are set as follows: Reynolds number $Re = 1.8 \times 10^6$, Mach number $Ma = 0.01$, and angle of attack $\alpha = 0°$.

For Bézier-GAN and PCA, we apply EGO on the three-dimensional latent spaces and use the trained Bézier-GAN generator or the inverse transformation of PCA to synthesize the optimal airfoils corresponding to the optimal latent codes.

We also compare these results to optimizing directly in the parametric design space. Specifically, we use two parameterizations, NURBS and PARSEC, and two optimization algorithms, EGO and GA, as additional experiments. We use the NACA 0012 airfoil as the initial design. The NURBS parameterization is based on Ref. [12]. The design space is defined as a ± 0.1 perturbation of the initial NURBS control point coordinates or a 20% perturbation of the initial PARSEC parameters. The population size of the GA is 100, and the chance of mutation (*i.e.*, the probability of mutating an individual's parameter) is 0.1. In each generation we choose 30 best and 10 random individuals for crossover, and produce 5 children for each pair. We direct interested readers to our code for more details.

We run each experiment so that the total number of C_L/C_D evaluations is 1000. The results of each experiment setting are averaged over 10 runs. Figure 5 shows the best-so-far C_L/C_D versus the number of evaluations. It shows that the value reached in 100 XFOIL evaluations by Bézier-GAN+EGO takes other methods at least 500 XFOIL evaluations to reach. Figure 6 shows the optimal airfoils for all experiment settings. Runs from the same scenario are plotted on the same subplot. For PCA+EGO and NURBS+GA, the final optimal solutions are inconsistent compared to other methods, indicating the optimization converged to different local optima. The values of maximal C_L/C_D after 100 and 1000 evaluations are shown in Table 1.

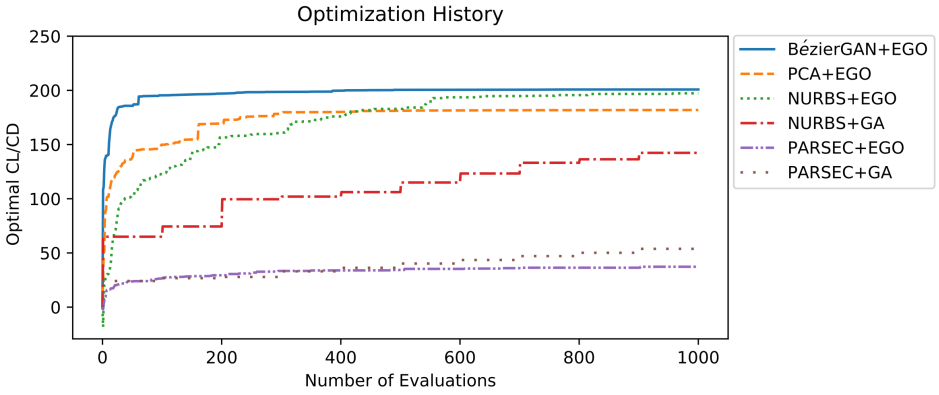

Fig. 5 Optimization history (averaged over 10 runs).

[†] https://github.com/IDEALLab/airfoil-opt-gan

[‡] The bounds of the visualized latent space are based on the latent coordinates of the data, *i.e.*, the minimum bounding box for data points projected onto the latent space.

[§] We use XFOIL here to demonstrate our scientific contributions, however, our approach is not limited to XFOIL. One can apply our techniques to any CFD or performance code including RANS or LES.

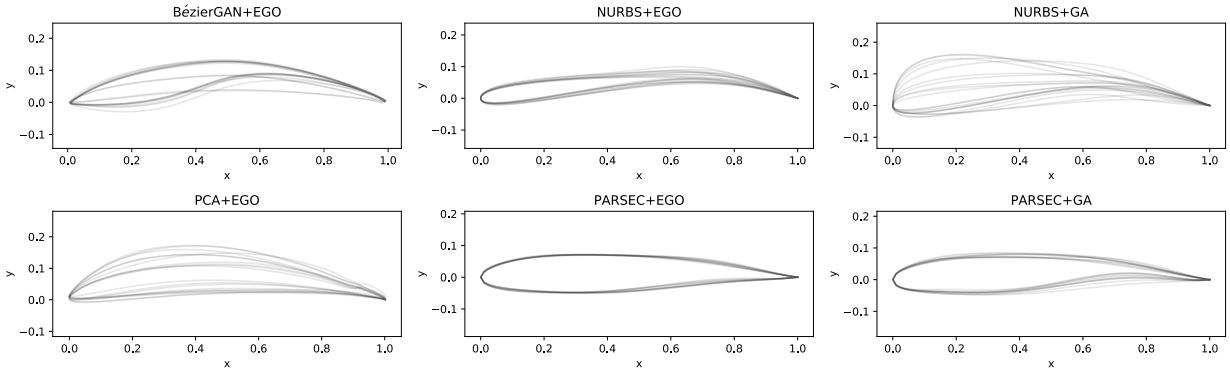

Fig. 6 Optimal airfoils (airfoils in the same subplot are under the same experimental configuration but different runs).

Table 1 Values of C_L/C_D for optimal solutions.

# Eval.	BézierGAN+EGO	PCA+EGO	NURBS+EGO	NURBS+GA	PARSEC+EGO	PARSEC+GA
100	**195.41 ± 2.94**	149.86 ± 13.46	122.75 ± 20.41	64.97 ± 4.82	26.44 ± 3.88	24.11 ± 0.90
1000	**200.80 ± 2.12**	181.86 ± 21.87	197.37 ± 3.22	142.35 ± 20.38	27.26 ± 4.07	53.77 ± 3.24

D. GA Refining

Figure 5 shows that when using Bézier-GAN, the optimal C_L/C_D stops improving after 100 evaluations, whereas the optimal C_L/C_D improves continuously, though slowly, when using the NURBS parameterization. This is because the three-dimensional latent space does not contain as much shape variation as the NURBS design space. However, while the three-dimensional latent space captures major shape variations, minor shape variations are captured by the noise space (*i.e.*, the space of the random input noise of Bézier-GAN). Therefore, we can further search for an improvement in that noise space. We achieve this by using the optimal solution of EGO after 100 evaluations as the initial design and run GAs in both the latent space and the noise space. We call this *GA refining*. Specifically, we allow larger shape variation on the noise variables while limiting the variation on the latent variables during mutation. The results are shown in Figs. 7 and 8. In this way, the optimal C_L/C_D keeps improving even after the latent space is exploited.

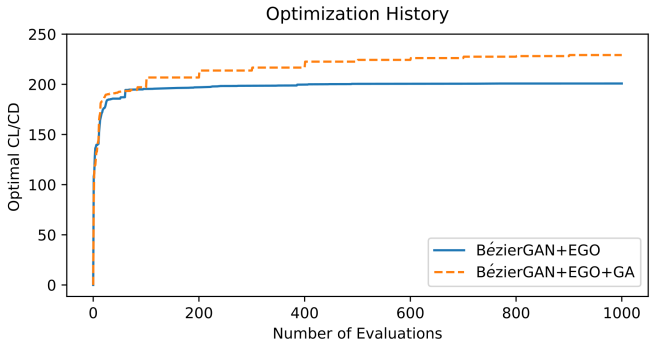

Fig. 7 Optimization history for BézierGAN+EGO with and without GA refining.

VII. Discussion and Conclusion

We use a Bézier-GAN to capture a low-dimensional latent space that encodes major shape variability of aerodynamic designs. Design optimization can then be conducted in this latent space to reduce the number of evaluations required to

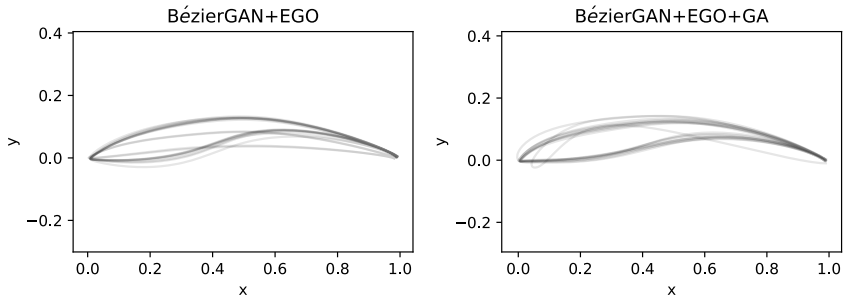

Fig. 8 Optimal airfoils for BézierGAN+EGO with and without GA refining.

find the optimal solution. Our results show that our Bézier-GAN method significantly accelerates convergence and finds optimal designs that are comparable to those found by other algorithms.

Because the latent space discards minor variability in designs that can potentially contribute to higher performance, the final optimal solution may be not as good as directly optimizing in the design space given sufficient number of evaluations. The GA refining mitigates this issue by continuing to explore the input noise space of the GAN after discovering good latent variables. There are other ways to improve the optimal solution while maintaining fast convergence. For example, the optimal solution obtained by our method can be used as a good start point for gradient-based optimization methods (*e.g.*, as in Berguin *et al.* [10]). For future research, we can concatenate a trained Bézier-GAN generator and an automatic differentiation solver to obtain the gradient of a QoI with respect to each latent variable directly. The low-dimensional gradients can then be applied to solve optimization problems.

Different from previous DR research for aerodynamic shape optimization which only targets one specific QoI (*i.e.*, response-based DR) or one airfoil model, the learned latent space in this work is reusable for optimizing any QoI for any airfoil model included in the UIUC airfoil database.

References

[1] Koziel, S., and Leifsson, L., "Multi-level surrogate-based airfoil shape optimization," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013, p. 778.

[2] Anderson, G. R., Nemec, M., and Aftosmis, M. J., "Aerodynamic shape optimization benchmarks with error control and automatic parameterization," *53rd AIAA Aerospace Sciences Meeting*, 2015, p. 1719.

[3] Fusi, F., Quaranta, G., Guardone, A., and Congedo, P. M., "Drag minimization of an isolated airfoil in transonic inviscid flow by means of genetic algorithms," *53rd AIAA Aerospace Sciences Meeting*, 2015, p. 1722.

[4] Amrit, A., Leifsson, L. T., Koziel, S., and Tesfahunegn, Y. A., "Efficient Multi-Objective Aerodynamic Optimization by Design Space Dimension Reduction and Co-Kriging," *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2016, p. 3515.

[5] Masters, D. A., Taylor, N. J., Rendall, T., Allen, C. B., and Poole, D. J., "Geometric Comparison of Aerofoil Shape Parameterization Methods," *AIAA Journal*, 2017, pp. 1575–1589.

[6] Bellman, R., *Dynamic programming*, Princeton University Press, Princeton, NY, 1957.

[7] Lukaczyk, T. W., Constantine, P., Palacios, F., and Alonso, J. J., "Active subspaces for shape optimization," *10th AIAA Multidisciplinary Design Optimization Conference*, 2014, p. 1171.

[8] Berguin, S. H., and Mavris, D. N., "Dimensional Design Space Exploration of Expensive Functions with Access to Gradient," *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2014, p. 2174.

[9] Berguin, S. H., and Mavris, D. N., "Dimensionality Reduction In Aerodynamic Design Using Principal Component Analysis With Gradient Information," *10th AIAA Multidisciplinary Design Optimization Conference*, 2014, p. 0112.

[10] Berguin, S. H., Rancourt, D., and Mavris, D. N., "Method to Facilitate High-Dimensional Design Space Exploration Using Computationally Expensive Analyses," *AIAA Journal*, Vol. 53, No. 12, 2015, pp. 3752–3765.

[11] Grey, Z. J., and Constantine, P. G., "Active subspaces of airfoil shape parameterizations," *AIAA Journal*, Vol. 56, No. 5, 2018, pp. 2003–2017.

[12] Viswanath, A., J. Forrester, A., and Keane, A., "Dimension reduction for aerodynamic design optimization," *AIAA journal*, Vol. 49, No. 6, 2011, pp. 1256–1266.

[13] Viswanath, A., Forrester, A., and Keane, A., "Constrained Design Optimization Using Generative Topographic Mapping," *AIAA journal*, Vol. 52, No. 5, 2014, pp. 1010–1023.

[14] Cinquegrana, D., and Iuliano, E., "Efficient Global Optimization of a Transonic Wing with Geometric Data Reduction," *35th AIAA Applied Aerodynamics Conference*, 2017, p. 3057.

[15] Cinquegrana, D., and Iuliano, E., "Investigation of adaptive design variables bounds in dimensionality reduction for aerodynamic shape optimization," *Computers & Fluids*, Vol. 174, 2018, pp. 89–109.

[16] Yasong, Q., Junqiang, B., Nan, L., and Chen, W., "Global aerodynamic design optimization based on data dimensionality reduction," *Chinese Journal of Aeronautics*, Vol. 31, No. 4, 2018, pp. 643–659.

[17] Kingma, D. P., and Welling, M., "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[18] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial nets," *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[19] Chen, W., and Fuge, M., "BézierGAN: Automatic Generation of Smooth Curves from Interpretable Low-Dimensional Parameters," *arXiv preprint arXiv:1808.08871*, 2018.

[20] Jeong, J.-H., and Kim, S.-H., "Optimization of thick wind turbine airfoils using a genetic algorithm," *Journal of Mechanical Science and Technology*, Vol. 32, No. 7, 2018, pp. 3191–3199. doi:10.1007/s12206-018-0622-x, URL https://doi.org/10.1007/s12206-018-0622-x.

[21] Tandis, E., and Assareh, E., "Inverse design of airfoils via an intelligent hybrid optimization technique," *Engineering with Computers*, Vol. 33, No. 3, 2017, pp. 361–374. doi:10.1007/s00366-016-0478-6, URL https://doi.org/10.1007/s00366-016-0478-6.

[22] Jahangirian, A. R., and Ebrahimi, M., "Airfoil Shape Optimization with Adaptive Mutation Genetic Algorithm," *Journal of Aerospace Science and Technology*, Vol. 11, No. 1, 2017, pp. –. URL http://jast.ias.ir/article_51638.html.

[23] LIU, Z., LIU, X., and CAI, X., "A new hybrid aerodynamic optimization framework based on differential evolution and invasive weed optimization," *Chinese Journal of Aeronautics*, Vol. 31, No. 7, 2018, pp. 1437 – 1448. doi:https://doi.org/10.1016/j.cja.2018.05.002, URL http://www.sciencedirect.com/science/article/pii/S1000936118301596.

[24] Venter, G., and Sobieszczanski-Sobieski, J., "Particle swarm optimization," *AIAA journal*, Vol. 41, No. 8, 2003, pp. 1583–1589.

[25] Ray, T., and Tsai, H., "Swarm algorithm for single-and multiobjective airfoil design optimization," *AIAA journal*, Vol. 42, No. 2, 2004, pp. 366–373.

[26] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, Vol. 13, No. 4, 1998, pp. 455–492.

[27] Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M., "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.

[28] Rasmussen, C. E., "Gaussian processes in machine learning," *Advanced lectures on machine learning*, Springer, 2004, pp. 63–71.

[29] Han, Z.-H., Chen, J., Zhang, K.-S., Xu, Z.-M., Zhu, Z., and Song, W.-P., "Aerodynamic Shape Optimization of Natural-Laminar-Flow Wing Using Surrogate-Based Approach," *AIAA Journal*, Vol. 56, No. 7, 2018, pp. 2579–2593.

[30] Regier, J. C., and Stark, P. B., "Mini-minimax uncertainty quantification for emulators," *SIAM/ASA Journal on Uncertainty Quantification*, Vol. 3, No. 1, 2015, pp. 686–708.

[31] Mani, K., Lockwood, B., and Mavriplis, D., "Adjoint-based unsteady airfoil design optimization with application to dynamic stall," *Annual Forum Proceedings - AHS International*, Vol. 3, 2012, pp. 1940–1952.

[32] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., "Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and adjoint derivative computations," *AIAA Journal*, Vol. 52, 2014, p. 935–951. doi:10.2514/1.J052255.

[33] Tesfahunegn, Y. A., Koziel, S., Leifsson, L., and Bekasiewicz, A., "Surrogate-based Airfoil Design with Space Mapping and Adjoint Sensitivity," *Procedia Computer Science*, Vol. 51, 2015, pp. 795 – 804. doi:https://doi.org/10.1016/j.procs.2015.05.201, URL http://www.sciencedirect.com/science/article/pii/S1877050915010091, international Conference On Computational Science, ICCS 2015.

[34] Schramm, M., Stoevesandt, B., and Peinke, J., "Simulation and Optimization of an Airfoil with Leading Edge Slat," *Journal of Physics: Conference Series*, Vol. 753, No. 2, 2016, p. 022052. URL http://stacks.iop.org/1742-6596/753/i=2/a=022052.

[35] Dhert, T., Ashuri, T., and Martins, J. R. R. A., "Aerodynamic Shape Optimization of Wind Turbine Blades Using a Reynolds-Averaged Navier–Stokes Model and an Adjoint Method," *Wind Energy*, Vol. 20, 2017, pp. 909–926. doi:10.1002/we.2070.

[36] Wang, K., Yu, S., and Liu, T., "Airfoil Optimization Based on Isogeometric Discontinuous Galerkin," *Proceedings of the 2018 2Nd International Conference on Algorithms, Computing and Systems*, ACM, New York, NY, USA, 2018, pp. 227–231. doi:10.1145/3242840.3242856, URL http://doi.acm.org/10.1145/3242840.3242856.

[37] Schramm, M., Stoevesandt, B., and Peinke, J., "Optimization of Airfoils Using the Adjoint Approach and the Influence of Adjoint Turbulent Viscosity," *Computation*, Vol. 6, No. 1, 2018, p. 5. doi:10.3390/computation6010005, URL http://dx.doi.org/10.3390/computation6010005.

[38] Larsson, J., and Wang, Q., "The prospect of using large eddy and detached eddy simulations in engineering design, and the research required to get there," *Phil. Trans. R. Soc. A*, Vol. 372, No. 2022, 2014, p. 20130329.

[39] Jé, r iuml, Lé, m., pine, Fran-atilde, Guibault, o., Tré, J.-Y., panier, Fran-atilde, Pé, o., et al., "Optimized nonuniform rational B-spline geometrical representation for aerodynamic design of wings," *AIAA journal*, Vol. 39, No. 11, 2001, pp. 2033–2041.

[40] Venkataraman, P., "A new procedure for airfoil definition," *13th Applied Aerodynamics Conference*, 1995, p. 1875.

[41] Rogalsky, T., Kocabiyik, S., and Derksen, R., "Differential evolution in aerodynamic optimization," *Canadian Aeronautics and Space Journal*, Vol. 46, No. 4, 2000, pp. 183–190.

[42] Sederberg, T. W., and Parry, S. R., "Free-form deformation of solid geometric models," *ACM SIGGRAPH computer graphics*, Vol. 20, No. 4, 1986, pp. 151–160.

[43] Kenway, G. K., and Martins, J. R., "Buffet-Onset Constraint Formulation for Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 55, No. 6, 2017, pp. 1930–1947.

[44] Nadarajah, S., Castonguay, P., and Mousavi, A., "Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization," *18th AIAA computational fluid dynamics conference*, 2007, p. 3837.

[45] Kulfan, B. M., "Universal parametric geometry representation method," *Journal of Aircraft*, Vol. 45, No. 1, 2008, pp. 142–158.

[46] Li, P., Seebass, R., and Sobieczky, H., "Manual aerodynamic optimization of an oblique flying wing," *36th AIAA Aerospace Sciences Meeting and Exhibit*, 1998, p. 598.

[47] Sobieczky, H., "Parametric airfoils and wings," *Recent development of aerodynamic design methodologies*, Springer, 1999, pp. 71–87.

[48] Derksen, R., and Rogalsky, T., "Bezier-PARSEC: An optimized aerofoil parameterization for design," *Advances in engineering software*, Vol. 41, No. 7-8, 2010, pp. 923–930.

[49] Lepine, J., Trepanier, J.-Y., and Pepin, F., "Wing aerodynamic design using an optimized NURBS geometrical representation," *38th Aerospace Sciences Meeting and Exhibit*, 2000, p. 669.

[50] Chen, W., Fuge, M., and Chazan, N., "Design Manifolds Capture the Intrinsic Complexity and Dimension of Design Spaces," *Journal of Mechanical Design*, Vol. 139, No. 5, 2017, pp. 051102–051102–10. doi:10.1115/1.4036134.

[51] Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D., "Screening, predicting, and computer experiments," *Technometrics*, Vol. 34, No. 1, 1992, pp. 15–25.

[52] Myers, R. H., Montgomery, D. C., et al., *Response surface methodology: process and product optimization using designed experiments*, Vol. 3, Wiley New York, 1995.

[53] Lee, J. A., and Verleysen, M., *Nonlinear dimensionality reduction*, Springer Science & Business Media, 2007.

[54] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y., *Deep learning*, Vol. 1, MIT press Cambridge, 2016.

[55] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative Adversarial Nets," *Advances in Neural Information Processing Systems 27*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Curran Associates, Inc., 2014, pp. 2672–2680. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[56] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P., "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

[57] Della Vecchia, P., Daniele, E., and D'Amato, E., "An airfoil shape optimization technique coupling PARSEC parameterization and evolutionary algorithm," *Aerospace Science and Technology*, Vol. 32, No. 1, 2014, pp. 103–110.

[58] Ebrahimi, M., and Jahangirian, A., "Aerodynamic optimization of airfoils using adaptive parameterization and genetic algorithm," *Journal of Optimization Theory and Applications*, Vol. 162, No. 1, 2014, pp. 257–271.

[59] Gaier, A., Asteroth, A., and Mouret, J.-B., "Aerodynamic Design Exploration through Surrogate-Assisted Illumination," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017, p. 3330.

[60] Zuhal, L. R., Amalinadhi, C., Dwianto, Y. B., Palar, P. S., and Shimoyama, K., "Benchmarking Multi-Objective Bayesian Global Optimization Strategies for Aerodynamic Design," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2018, p. 0914.

[61] Broyden, C. G., "The convergence of a class of double-rank minimization algorithms 1. general considerations," *IMA Journal of Applied Mathematics*, Vol. 6, No. 1, 1970, pp. 76–90.

[62] Fletcher, R., "A new approach to variable metric algorithms," *The computer journal*, Vol. 13, No. 3, 1970, pp. 317–322.

[63] Goldfarb, D., "A family of variable-metric methods derived by variational means," *Mathematics of computation*, Vol. 24, No. 109, 1970, pp. 23–26.

[64] Shanno, D. F., "Conditioning of quasi-Newton methods for function minimization," *Mathematics of computation*, Vol. 24, No. 111, 1970, pp. 647–656.

[65] Zeiler, M. D., Krishnan, D., Taylor, G. W., and Fergus, R., "Deconvolutional networks," *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, 2010, pp. 2528–2535.

[66] Kingma, D. P., and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[67] Drela, M., "XFOIL: An analysis and design system for low Reynolds number airfoils," *Low Reynolds number aerodynamics*, Springer, 1989, pp. 1–12.