



Efficient Aerodynamic Shape Optimization with Deep-Learning-Based Geometric Filtering

Jichao Li* and Mengqi Zhang†

National University of Singapore, Singapore 117575, Republic of Singapore

Joaquim R. R. A. Martins‡

University of Michigan, Ann Arbor, Michigan 48109

and

Chang Shu§

National University of Singapore, Singapore 117575, Republic of Singapore

<https://doi.org/10.2514/1.J059254>

Surrogate-based optimization has been used in aerodynamic shape optimization, but it has been limited due to the curse of dimensionality. Although a large number of variables are required for the shape parameterization, many of the shapes that the parameterization can produce are abnormal and do not add meaningful information to a surrogate model. To improve the efficiency of surrogate-based optimization, recent machine learning techniques are applied in this study to reduce the abnormality of both initial and infill samples. This paper proposes a new sampling method for airfoils and wings, which is based on a deep convolutional generative adversarial network. This network is trained to learn the underlying features among the existing airfoils and is able to generate sample airfoils that are notably more realistic than those generated by other sampling methods. In addition, a discriminative model is developed based on convolutional neural networks. This model detects the geometric abnormality of airfoils or wing sections quickly without using expensive computational fluid dynamic models. These machine learning models are embedded in a surrogate-based aerodynamic optimization framework and perform aerodynamic shape optimization for airfoils and wings. The results demonstrate that, compared with the conventional methods, our proposed models can double the optimization efficiency.

Nomenclature

A	= airfoil area
C_d	= drag coefficient
C_l	= lift coefficient
C_p	= pressure coefficient
D	= discriminative model
G	= generative model
M	= Mach number
Re	= Reynolds number
S_{validity}	= discriminative score provided by the convolution model
t	= thickness
V	= wing volume
x_{shape}	= shape design variables
α	= angle of attack
θ	= lower bound of S_{validity} in design optimization

I. Introduction

AERODYNAMIC shape optimization based on computational fluid dynamics (CFD) has been widely used in aircraft design. Various optimization approaches, such as the adjoint-based high-fidelity optimization [1,2], surrogate-based optimization [3–5], database design optimization [6–8], and hybrid methods [9,10] have been developed to satisfy different demands. As reviewed by Yondo et al. [11], surrogate-based design optimization is a popular approach

due to the easy implementation, which has been applied in airfoil design [12], wing design [13], and aerostructural design optimization [14]. This approach does not necessarily require gradient information, the computation of which is either time-consuming or requires an adjoint solver. This makes it possible to use a black-box CFD solver, such as robust commercial software or a legacy code. However, the optimization efficiency of surrogate methods leaves much to be desired, especially in high-dimensional design problems where surrogate models suffer from the curse of dimensionality. Therefore, it is important to develop new methods to improve the efficiency of surrogate-based optimization.

Previous efforts to improve the surrogate-based optimization efficiency mainly emphasize two parts: more accurate surrogate models or smaller design spaces. Kriging models [15] are the most popular surrogate models used in this field. Different variations of kriging have been developed. For example, the gradient-enhanced kriging (GEK) [16,17] uses the gradient information to improve the model accuracy, which has been shown to be more effective than conventional kriging in airfoil design optimization. Bouhlel et al. [18] proposed a kriging model with the partial least-squares method (KPLS) to reduce the computational cost of training kriging models in high-dimensional problems. Bouhlel and Martins [17] further developed a gradient-enhanced version of KPLS, GEKPLS, to improve accuracy using gradient information. According to their results, KPLS and GEKPLS have better accuracy than kriging and GEK for the same number of training points. Besides the dimensionality, the range of the design variable values is another factor influencing the accuracy of surrogate models. Liem et al. [19] coupled kriging with clustering algorithms to construct a mixture of experts (ME) strategy. This ME model has been shown to be more accurate than regular kriging in a large design space covering the whole flight envelope of aircraft.

Moreover, simplifying the design space also improves the efficiency of surrogate-based optimization, which is mainly performed by reducing the dimensionality of the design variables. For example, Li et al. [4] proposed a surrogate-based optimization approach with the active subspace method (ASM) to reduce the high-dimensional inputs. By projecting the original 220 design variables to four ASM vectors, they made the aircraft wing design optimization much more efficient. In addition, singular value decomposition (SVD) is used to

Received 20 November 2019; revision received 4 May 2020; accepted for publication 10 June 2020; published online 30 June 2020. Copyright © 2020 by the authors. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 1533-385X to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Postdoctoral Research Fellow, Department of Mechanical Engineering; cfdljc@gmail.com. Member AIAA (Corresponding Author).

†Assistant Professor, Department of Mechanical Engineering.

‡Professor, Department of Aerospace Engineering. Fellow AIAA.

§Professor, Department of Mechanical Engineering.

derive orthogonal airfoil mode shapes [20,21]. These airfoil modes have been shown to be more efficient than regular parameterization methods [22], so fewer parameters can be used in the aerodynamic shape optimization. Li et al. [6] investigated the relationships between two dominant airfoil modes and higher-order modes and managed to shrink the design space using margin surfaces of higher-order modes to exclude abnormal airfoils. Chen et al. [23] proposed a parameterization method for airfoils based on generative adversarial networks (GAN) that constructs low-dimensional airfoil shape representations using machine learning techniques. GAN is a machine-learning model proposed by Goodfellow et al. [24], which learns to generate new data with the same statistics as the training set. Compared with SVD mode shapes, GAN mode shapes produce more realistic airfoils. Chen et al. [23] achieved better optimization results by using GAN mode shapes than other parameterization methods.

Training data have a significant impact on the accuracy of surrogate models and further influence the overall efficiency of surrogate-based optimization. However, there have been few efforts toward improving the quality of training data. In surrogate-based optimization, training data are composed of initial samples and infill samples. Initial samples are generated by a sampling method in the design of experiment (DoE) process, which is typically the first step of surrogate-based optimization. Existing sampling methods, such as the Latin hypercube sampling (LHS) method [25] cannot ensure realistic samples in aerodynamic shape optimization, because they are designed to generate samples distributed evenly without considering feasibility. Because of the usage of these methods, many samples associated with abnormal aerodynamic shapes are thus included in the initial training data, which cannot be used in practice. In addition, these samples are not the most effective for training surrogate models to predict aerodynamic functions of designs with realistic shapes [6]. Using existing airfoils as initial samples could be a solution, but it is difficult to have enough samples, because the number of existing airfoils is limited.

Infill samples are added by solving suboptimization problems in surrogate-based optimization, where aerodynamic functions in the objective and constraints are evaluated by surrogate models. Because of the inaccuracy of surrogate models, the shapes of new infill samples are also not guaranteed to be realistic. In fact, many infill samples have such abnormal aerodynamic shapes that they do not add meaningful information to surrogate models. Refining surrogate models with these infill samples further reduces the accuracy in predicting aerodynamic coefficients of realistic shapes, which leads to a vicious circle. Thus, it is of great importance to avoid abnormal shapes in training data.

Abnormal airfoil shapes, such as multimodal shapes, obviously result in bad aerodynamic performance. However, these geometric abnormalities are hard to define mathematically. Thus, it is difficult to exclude abnormal shapes by explicitly defining geometrically infeasible domains in an aerodynamic design space. We seek to address this issue by using deep learning techniques. First, we propose a sampling method with a deep convolutional generative adversarial network (DCGAN). This method produces realistic airfoil and wing samples by learning the underlying distribution features among the existing airfoils in the University of Illinois Urbana-Champaign (UIUC) airfoil database. Then, we propose a discriminative model for evaluating the feasibility of airfoils or wing sections using a convolutional neural network (CNN). This CNN discriminator is trained by 20,000 realistic DCGAN airfoils and 20,000 abnormal LHS airfoils. We use this discriminator as a constraint function to avoid abnormal infill samples. Then, we couple the proposed methods with a surrogate-based optimization framework and perform aerodynamic shape optimization of airfoils and wings to verify the advantages.

The rest of this paper is organized as follows. First, we introduce the DCGAN model (Sec. II.A) and the CNN discriminator (Sec. II.B). We present the sampling method based on DCGAN in Sec. II.C. We investigate the influence of the training database on the DCGAN model in Sec. II.D. Then, we present the optimization framework in Sec. III.A. We show the airfoil design optimization and wing design optimization results in Secs. III.B and III.C, respectively. We finish with our conclusions in Sec. IV.

II. Methodology

A. Deep Convolutional Generative Adversarial Network of Airfoils

GAN consists of two “adversarial” models: a generative model G that captures the training data distribution P_{data} and a discriminative model D that estimates the probability that a sample comes from the training data rather than G . Both G and D could be a nonlinear mapping function, such as a CNN. In these circumstances, the model is referred to as DCGAN [26], where DC is short for deep convolutional. Models G and D are trained simultaneously to reach a Nash equilibrium, which is the optimal point for the value function V in a two-player minimax game defined as

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

where x and z are the training dataset and noisy inputs, respectively. By learning the underlying features of P_{data} , G manages to produce similar synthetic versions of training data just using noisy inputs.

Different variants of GAN models have been successfully used to generate realistic and even hyper-realistic image and voice synthesis [27]. In the field of aerodynamics, Chen et al. [23] used GAN to construct a compact set of airfoil shape representations. Airfoils produced by this approach are more realistic than the conventional mode shape approaches. These GAN mode shapes were then used to parameterize airfoils in aerodynamic shape optimization. In this work, we apply DCGAN, not as a parameterization method, but to directly generate realistic airfoil coordinates. Then, this DCGAN model can be further coupled with any conventional parameterization methods such as class-shape transformations, B-splines, and Hicks–Henne bump functions to obtain realistic samples [22]. The proposed approach can retain the high number of shape design variables currently used in airfoil design optimization.

The flowchart of the DCGAN model in this work is shown in Fig. 1. Instead of generating images, we use this model to produce airfoil coordinates, which are in a vector format. The training data are chosen from the UIUC airfoil database. The training of DCGAN is an adversarial process with both discriminative and generative models. The discriminative model D uses a fully connected layer to perceive the input information. Then, it uses four convolution layers to extract underlying features, which is referred to as the downsampling process. These features are linked to a fully connected layer with one scalar output, which is used to distinguish whether an input is from the training data set or synthetic data of the generative model. The generative model G is trained to provide synthetic data using noisy inputs. The noisy input is followed by a fully connected layer. This layer is reshaped and then upsampled by four transposed convolution layers. The output of the last transposed convolution layer in the generative model corresponds to the coordinates of an airfoil. The DCGAN is trained using the Adam gradient-based optimization algorithm for stochastic objective functions due to its high efficiency [28].

We use the open-source script by Li et al. [29] to preprocess raw UIUC data and obtain 1407 airfoils in a uniform x - y format with $N = 251$ points. The x coordinates are given by Eq. (A1) in the Appendix. We store each airfoil as a vector y with the corresponding y_i coordinates. The airfoils in this database have different camber and thickness characteristics and have been designed for different purposes. Thus, various scalings are needed, which is an issue when trying to capture the common features among them. To overcome this, we propose a normalization method to reduce the difference in the training data. The normalization first removes the dominant camber information from the data by subtracting the component contributed by the first camber mode. The camber mode is solved by using the camber-thickness shape mode method proposed by Li et al. [6] (see Appendix for details). Then, we scale the remaining component of the airfoil by dividing the maximum absolute value of y , which we refer to as the thickness factor. After the normalization, all values are in the $[-1, 1]$ interval. The details of this normalization method are shown in Algorithm 1 and Fig. A1 in the Appendix shows sample airfoils before and after the normalization.

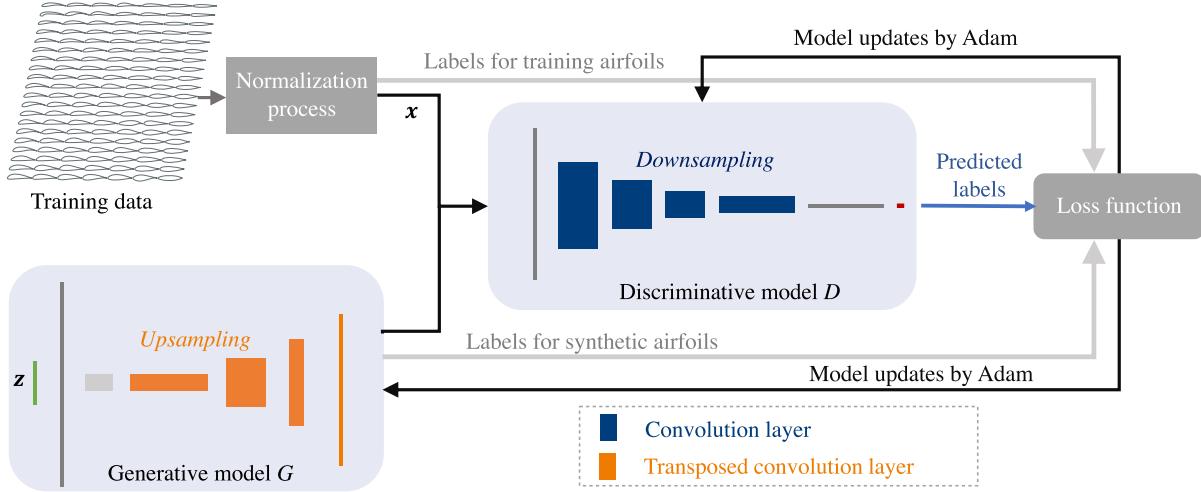


Fig. 1 Flowchart of the DCGAN trained with an airfoil database.

Algorithm 1: Normalization of airfoil data

```

1: procedure NORMALIZATION( $y, C$ )
2:    $y_{camber} = f_c(y)$                                  $\triangleright y \in \mathbb{R}^N$  is the vector of airfoil  $y$  coordinates and  $C$  is the first camber mode
3:    $c_1 = C^T y_{camber}$                              $\triangleright f_c$  computes the camber line coordinates of this airfoil
4:   for  $i = 1; i \leq n_f; i + + \text{do}$            $\triangleright$  Compute the component contributed by the first camber mode
5:      $y_i = y_i - c_1 C_i$                           $\triangleright$  Cycle to subtract the first camber component
6:      $y_{N-i-1} = y_{N-i-1} - c_1 C_i$ 
7:   end for                                          $\triangleright$  Compute the thickness factor
8:    $t = \max |y|$                                   $\triangleright$  Normalize
9:    $y = y/t$                                       $\triangleright$  Return normalized  $y$ 
10:  return  $y$ 
11: end procedure

```

The G model in DCGAN takes noise from the latent space as the input. We use a 100-dimensional latent space as in other applications of GAN [30]. This noise conforms to the standard normal distribution. In D , the downsampling process could either be realized by using a pooling layer or using stride steps larger than one in convolution layers. We use convolution layers with a stride $r = 2$ instead of adding a pooling layer because it tends to improve the overall accuracy and stability of the model [31]. Similarly, the upsampling in G is performed by using transposed convolution layers with $r = 2$, rather than by adding an unpooling layer. We adopt a four-layer structure in both G and D based on trial and error. Although this choice may not be optimal, we find that it works well with the given airfoil format. Table 1 shows the number of filters and weights in the DCGAN model.

The size n_{size} and depth n_{depth} of the filters are two important model parameters that influence the convergence and performance of DCGAN. We conduct several tests to determine good values for these parameters. Figure 2 shows the convergence histories and synthetic airfoils of generative models with different values of n_{size} and n_{depth} . For each DCGAN, we show 3×4 synthetic airfoils, which are

generated by random inputs and converted using Algorithm 2 with a fixed thickness factor and camber coefficient.

The filter size has a significant influence on the performance of DCGAN. Using $n_{size} < 3$ cannot make the generative model produce smooth airfoil shapes. Increasing n_{size} makes DCGAN models converge faster. Increasing n_{depth} will cause difficulty in training the model. However, it is necessary to make the depth larger than 5 ($n_{depth} > 5$); otherwise, the synthetic airfoils may be not reasonable (see the synthetic airfoils with $n_{size} = 3$ and $n_{depth} = 5$ in Fig. 2). Overall, if either n_{size} or n_{depth} is not too small, the synthetic airfoils are similar to real airfoils, which demonstrates the robustness of the DCGAN model. Thus, we use $n_{size} = 7$ and $n_{depth} = 10$ in this work. The chosen DCGAN has 68,394 and 30,901 trainable parameters in the generative and discriminative model, respectively. Modifying these hyperparameters, by, for example, maximizing the inception score [32], might further improve the performance. Nevertheless, the chosen model is stable with different initialization weights, as shown in the Appendix. In Sec. II.D, we show that this model works well using different airfoil datasets.

Table 1 Number of filters and weights in each convolution or transposed convolutional layer of the DCGAN model

Layers	Generative model		Discriminative model	
	Filters	Weights	Filters	Weights
First layer	$4 \times n_{depth}$	$128 \times n_{depth} \times n_{size}$	n_{depth}	$n_{depth} \times (n_{size} + 1)$
Second layer	$2 \times n_{depth}$	$8 \times n_{depth}^2 \times n_{size}$	$2 \times n_{depth}$	$2 \times n_{depth} \times (n_{size} \times n_{depth} + 1)$
Third layer	n_{depth}	$2 \times n_{depth}^2 \times n_{size}$	$4 \times n_{depth}$	$4 \times n_{depth} \times (2 \times n_{size} \times n_{depth} + 1)$
Fourth layer	1	$n_{depth} \times n_{size}$	$8 \times n_{depth}$	$8 \times n_{depth} \times (4 \times n_{size} \times n_{depth} + 1)$

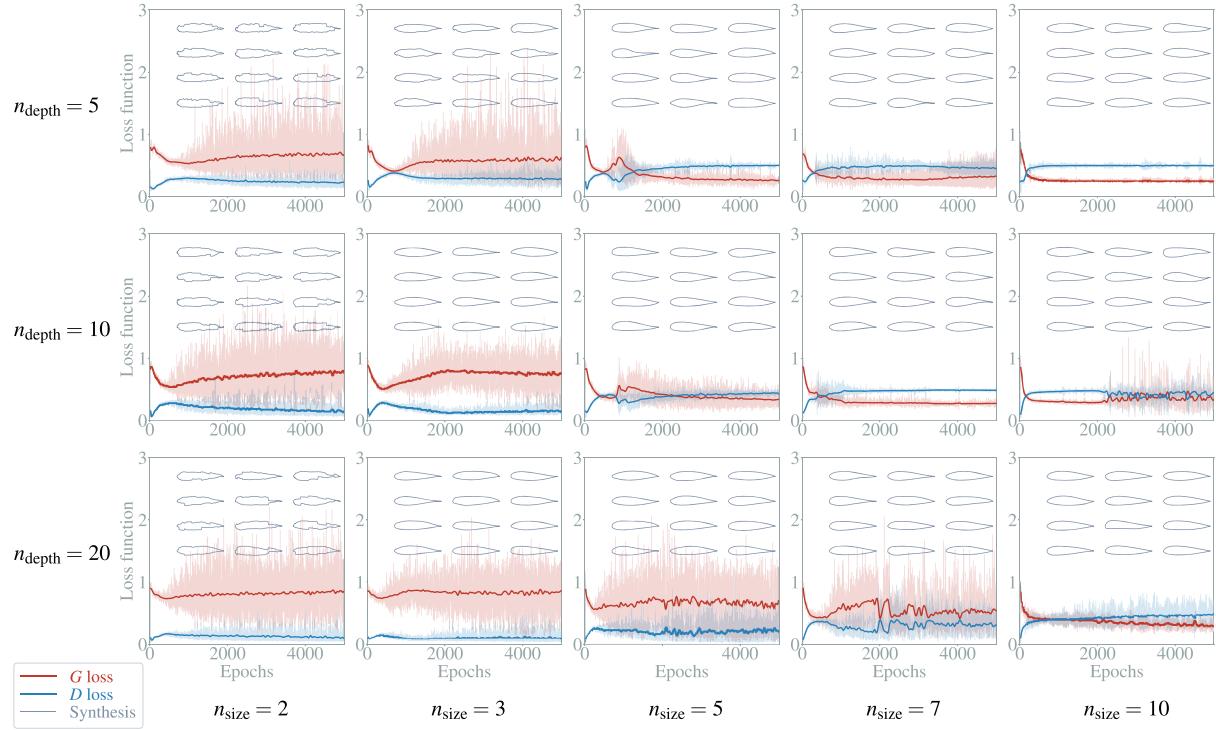


Fig. 2 Effect of n_{size} and n_{depth} on the DCGAN model.

Algorithm 2: Transformation of DCGAN synthetic airfoils to sampling airfoils

```

1: procedure TRANSFORMATION( $y, t, c$ ) $y \in \mathbb{R}^N$  is the vector of  $y$  coordinates of the DCGAN synthetic.  $t$  and  $c$  are
   the corresponding thickness factor and dominant camber coefficient, respectively.
2:    $y = y \times t$                                       $\triangleright$  Scale the synthetic airfoil coordinates
3:   for  $i = 1; i \leq n_f; i + +$  do            $\triangleright$  Cycle to add on the dominant camber component
4:      $y_i = y_i + cC_i$ 
5:      $y_{N-i-1} = y_{N-i-1} + cC_i$ 
6:   end for
7:   return  $y$                                       $\triangleright$  Return the transformed airfoil coordinates
8: end procedure

```

GAN is known to be very unstable due to the mode collapse issue, where the generative model can produce only a small portion of the trained distribution. This leads to similar or even the same synthetic airfoils, and thus DCGAN cannot contain necessary diversity for the sampling purpose. Different approaches, such as the Wasserstein GAN [33], have been proposed to address this issue. Other generative models, such as normalizing flows [34,35], have been shown to not suffer from the mode collapse issue. As shown below, we find that the airfoil normalization presented in Algorithm 1 helps avoid this issue as well.

We use the inception score (IS) proposed by Salimans et al. [32] to evaluate the diversity of DCGAN outputs. As demonstrated by Zhou et al. [36], mode-collapsed GANs usually get low inception scores. The score is given by

$$\text{IS}(G) = \exp(E[D_{\text{KL}}(p(y|x_{\text{GAN}})||p(y))]) \quad (2)$$

where x_{GAN} and y are synthetic airfoils of DCGAN and their classification labels, and D_{KL} is the Kullback–Leibler divergence between the conditional class distribution $p(y|x_{\text{GAN}})$ and the marginal class distribution $p(y)$. We compute $p(y|x_{\text{GAN}})$ using a Gaussian mixture model (GMM) [6,29], which is a mixture of finite number of Gaussian distributions trained by the UIUC database to classify airfoils. The probability density function of GMM is $p(x) = \sum_{i=1}^K \phi_i \mathcal{N}(x|\mu_i, \Sigma_i)$, where ϕ_i ($i = 1, \dots, K$) are the combination weights ($\sum_{i=1}^K \phi_i = 1$) and K is the number of Gaussian

components. Each component represents a class of airfoils. The hyperparameters in GMM are solved by maximizing the likelihood function $p(X) = \prod_{n=1}^{N_{\text{training}}} p(x_n)$ on the training data (UIUC airfoils). The chosen GMM has $K = 10$ components because we find that the likelihood function does not significantly increase afterward. The inception score has the lowest value of 1.0 and the highest value of the number of classes supported by the classification model. For the DCGAN trained without using the normalization process, an average score of $\text{IS}(G) = 3.23$ is obtained in multiple tests. This score increases to $\text{IS}(G) = 4.78$ in the proposed DCGAN. Airfoil normalization increases the diversity of DCGAN and thus helps address the mode collapse issue.

In addition to the inception score, we use another metric, the maximum mean discrepancy (MMD), to evaluate the quality of synthetic airfoils. MMD computes the dissimilarity between training data and synthetic data. A lower MMD means that the distribution of synthetic data is closer to that of training data and thus is desirable. We use an empirical estimate of MMD [37], which is formulated as

$$\begin{aligned} \widehat{\text{MMD}} &= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n k(x^i, x^j) + \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m k(x_G^i, x_G^j) \\ &\quad - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m k(x^i, x_G^j) \end{aligned} \quad (3)$$

Table 2 Influence of the latent space dimension

d_{latent}	IS				Maximum mean discrepancy	Synthetic airfoils
	Mean	SD	Mean	SD		
1	4.25	0.14	0.154	0.026		
10	4.77	0.11	0.123	0.026		
100	4.78	0.11	0.120	0.025		
1000	4.81	0.09	0.118	0.026		

where n and m are the numbers of training and synthetic samples; \mathbf{x}^i and $\mathbf{x}_{\text{GAN}}^i$ are the i th training and synthetic airfoil; k is the kernel function and we use a radial kernel $k(\mathbf{x}, \mathbf{y}) = \exp[-(\|\mathbf{x} - \mathbf{y}\|^2)/2\theta^2]$ with $\theta = 0.1$.

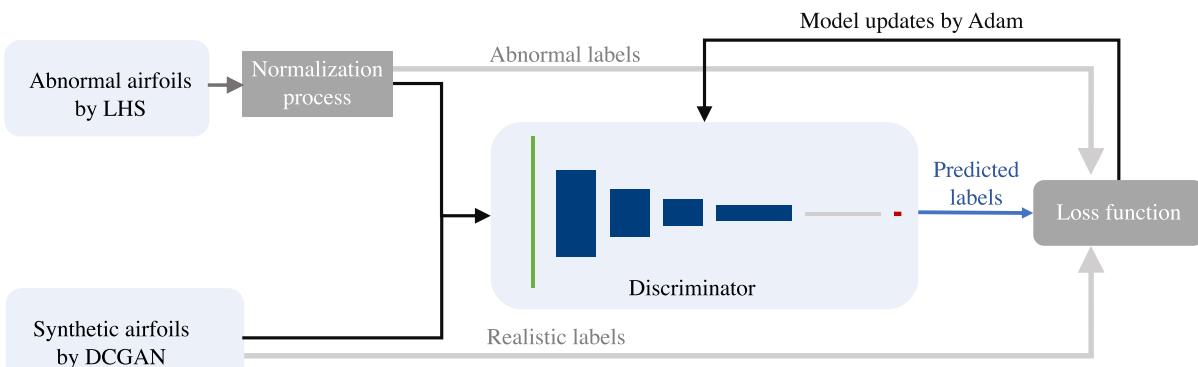
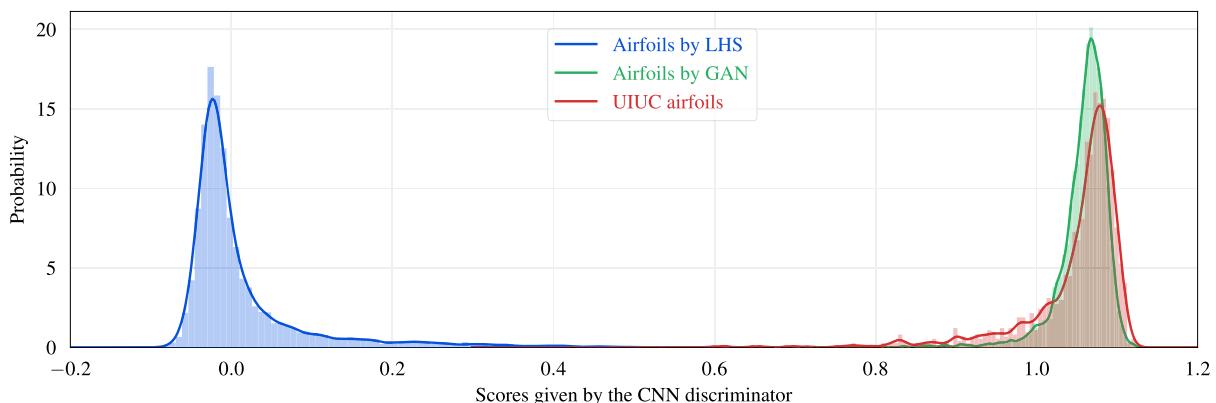
We analyze the influence of latent space dimension (from 1 to 1000) using IS and MMD scores computed in multiple tests. The mean and standard deviation (SD) of these metrics are shown in Table 2. A low latent space dimension ($d_{\text{latent}} \sim 1$) would lead to a DCGAN with insufficient control ability and thus results in geometric failures, such as crossing surfaces near trailing edges (see examples of synthetic airfoils in Table 2). Higher latent space dimensions ($d_{\text{latent}} \geq 10$) make DCGAN generate reasonable synthetic airfoils. The increase in latent space dimension improves both diversity (higher inception scores) and validity (lower MMD scores) of synthetic airfoils although no significant difference could be detected by visual inspection.

B. Airfoil Discriminator with Convolutional Neural Networks

In surrogate-based aerodynamic shape optimization, infill samples may be generated with abnormal shapes due to surrogate model inaccuracy. It is helpful to use additional simple functions to constrain the aerodynamic shape in the infill process. In the field of airfoil and wing design optimization, this can be done with geometric functions on the sectional shape. Li et al. [29] proposed a data-driven geometric constraint on the leading-edge thickness to enhance the low-speed performance in transonic wing design. Nevertheless, it is difficult to design a universal function to judge the shape abnormality with a limited number of existing airfoils (only ~ 1500).

In this work, we manage to generate numerous realistic airfoils with the DCGAN model introduced above. With sufficient training data, we construct a discriminative model shown in Fig. 3. This model uses four convolution layers to extract features in the training data. Filter sizes and depths are the same as those in the discriminative model of the DCGAN. These features are summarized by a fully connected layer with one neuron, which provides a discriminative score. In the training data, realistic airfoils and abnormal airfoils are labeled with ones and zeros, respectively. The loss function of this model is the mean square error of discriminative scores to the labels, and the binary cross-entropy function can also be used in this model. We use Adam to minimize the loss function [28].

We train the CNN discriminator using 20,000 realistic airfoils and 20,000 abnormal airfoils. The realistic airfoils are generated by the DCGAN model, which we refer to as *GAN airfoils*. The abnormal airfoils are generated by perturbing control points of UIUC airfoils using LHS. As shown in Fig. 7 and in [6], most of the airfoils generated this way are abnormal. A discriminative score near zero means that the airfoil is abnormal, whereas a score around one corresponds to a realistic airfoil. Figure 4 shows discriminative scores of another 5000 LHS airfoils, another 5000 GAN airfoils, and UIUC airfoils provided by the CNN model. The CNN discriminator manages to distinguish abnormal airfoils from realistic ones.

**Fig. 3** CNN-based discriminator of airfoil abnormality.**Fig. 4** The CNN discriminator manages to distinguish abnormal airfoils from realistic ones.

Similarly to other discriminative methods, such as logistic regression, the CNN discriminator is trained to label samples. A CNN discriminator can be advantageous because it is model-free and inherently nonlinear. It may seem to be redundant to train an independent CNN discriminator, because DCGAN already contains a discriminative model. However, the discriminative model in DCGAN is trained to distinguish synthetic airfoils from database airfoils, which are all realistic. As shown in Fig. A3 of the Appendix, we find that the DCGAN discriminative model cannot distinguish realistic airfoils from abnormal ones. Thus, it is necessary to have such an independent discriminative model. In addition, it is possible to obtain a compact model by combining the training of DCGAN and the CNN discriminator.

To avoid geometric abnormality, we use the CNN discriminative function as a constraint on airfoils and wing sections in the following optimizations, i.e., $S_{\text{validity}} \geq \theta$. The selection of θ would compromise the optimization efficiency and efficacy, and a conservative θ may even limit the discovery of novel shapes with good performance. Nevertheless, this influence can be within an acceptable range by using an appropriate θ in the constraint. This work focuses on transonic aerodynamic optimization, so we use $\theta = 0.9$ because most transonic airfoils in the database have CNN scores larger than 0.9.

The complexity of constraint functions may affect the efficiency of solving suboptimization problems in surrogate-based optimization. Li et al. [38] showed that, for aerodynamic design, solving suboptimization problems using gradient-based algorithms can be much more effective and efficient than using gradient-free algorithms. We use this discriminator as a geometric constraint of airfoil and wing sections in aerodynamic shape optimization. If this constraint is highly multimodal, gradient-based algorithms may not be able to make suboptimization problems converge, and then the efficiency would be negatively impacted. Therefore, we investigate the complexity of the CNN discriminative function. Figure 5 shows decaying lines of CNN discriminative scores from a UIUC airfoil to three LHS airfoils. The intermediate airfoils along each slice line are linearly interpolated with k and airfoils at both vertices, where k is the linear interpolation parameter describing the distance to the starting vertex. The monotonic and smooth decay of scores indicates that the CNN discriminative constraint function is suitable for gradient-based optimization.

C. Sampling Method with DCGAN

Samples of airfoils and wings can be used to train surrogate models of aerodynamic functions [16,38,39] or to construct reduced-order models [40,41] of the aerodynamic system. Conventional sampling methods often introduce aerodynamic samples with geometric abnormalities, which is inefficient. In this section, we propose to use the DCGAN model to generate realistic airfoil and wing samples whose shapes are more regular and smooth. DCGAN produces coordinates of normalized airfoils or wing sections from noisy inputs. For the purpose of DoE samples in optimization, DCGAN outputs should be further transformed into sampling airfoils and then parameterized by design variables. The dimensionality of the DCGAN inputs is

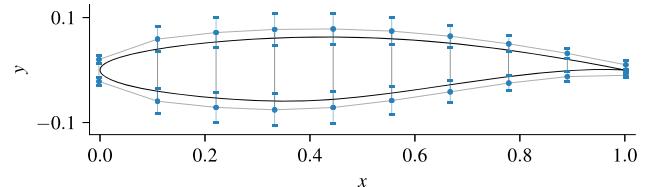


Fig. 6 FFD control points in the airfoil parameterization.

independent of that of the shape design variables and any conventional parameterization method can be used. Herein, we use the free form deformation (FFD) method to parameterize DCGAN airfoils.

1. Airfoil Samples

The generative model in the DCGAN model produces normalized synthetic airfoils with random noisy inputs. These synthetic airfoils are required to multiply a thickness factor and associate a dominant camber component to obtain the airfoils, which is the inverse process of the normalization process. The thickness factor and dominant camber coefficient are two basic airfoil parameters that determine their overall shape. For a given design problem, the bounds of these two parameters can be predetermined based on the thickness and lift requirements. We use DCGAN to produce normalized synthetic airfoils and LHS to generate the two preliminary parameters for each synthetic. Then, we scale and camber of the synthetic airfoils with the corresponding thickness factor t and dominant camber coefficient c , respectively. This transformation process is described in Algorithm 2, which is the inverse process of the normalization in Algorithm 1.

We compare the GAN airfoils with airfoils generated by a traditional sampling method (LHS). To obtain LHS airfoils, the baseline airfoil is parameterized using FFD. Then, LHS generates perturbations of the FFD control points. The bounds of FFD control points and the baseline airfoil are shown in Fig. 6. For GAN airfoils, the bounds of the thickness factor and dominant camber coefficient are chosen to have a design space of a similar size. As shown in Fig. 7, most of LHS airfoils are abnormal, and GAN airfoils are much more realistic. We compute the flow around each airfoil by solving the Reynolds-averaged Navier–Stokes (RANS) equations with $M = 0.75$, $Re = 6.5 \times 10^6$, and $\alpha = 2.5^\circ$. The Mach contours in Fig. 7 highlight the impact of geometric abnormality in LHS airfoils, which makes local flows much more complicated.

2. Wing Samples

Similarly to GAN airfoils, we use DCGAN to produce sectional airfoil shapes and then combine them to generate wing shapes, which we refer to as *GAN wings*. As shown in Fig. 8, we create five sectional airfoils using the generative model for each GAN wing. We compute the 12 intermediate section shapes along the span by the cubic interpolation. These sectional airfoils roughly represent the shape of GAN wings. The corresponding FFD control points of GAN wings are solved by the least-squares method. LHS wings

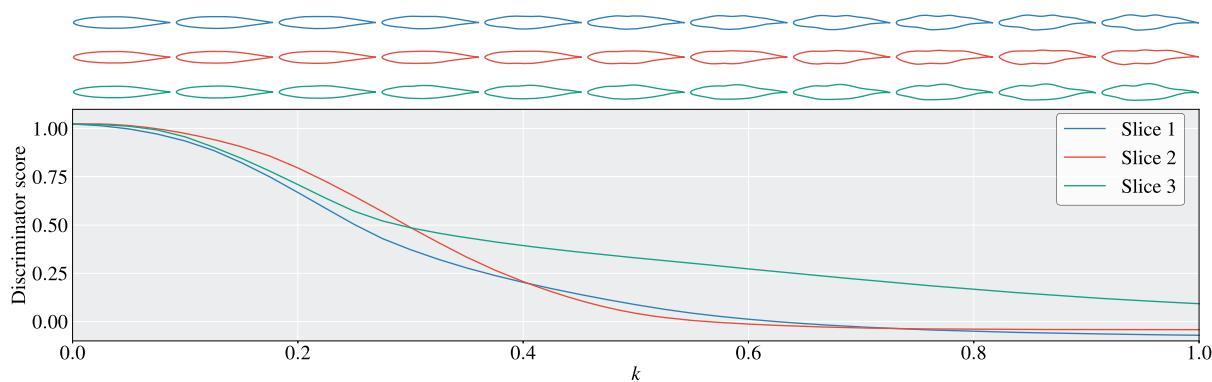
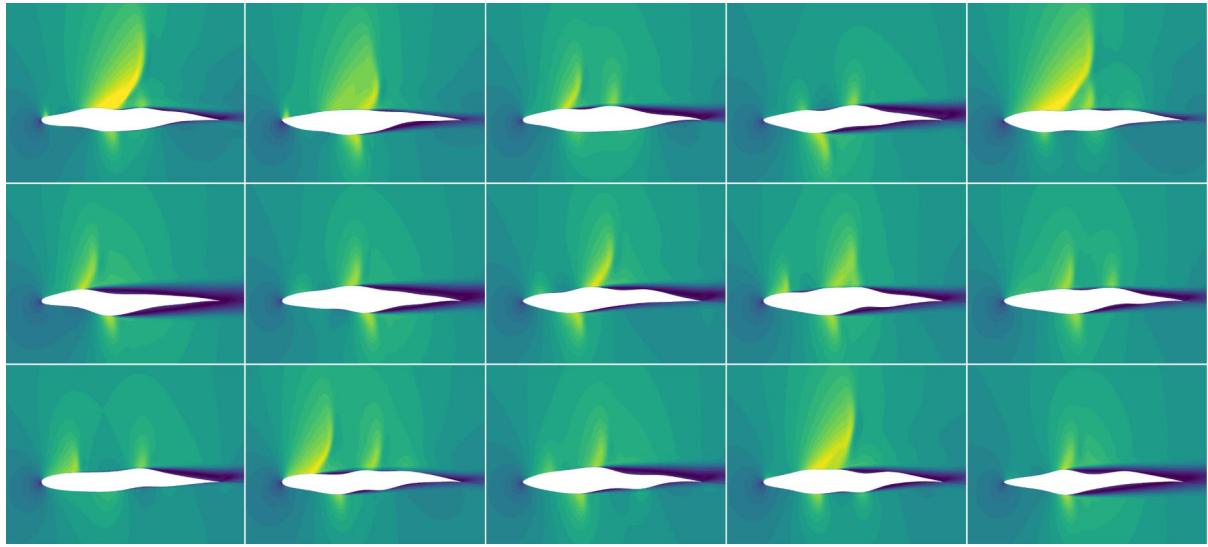
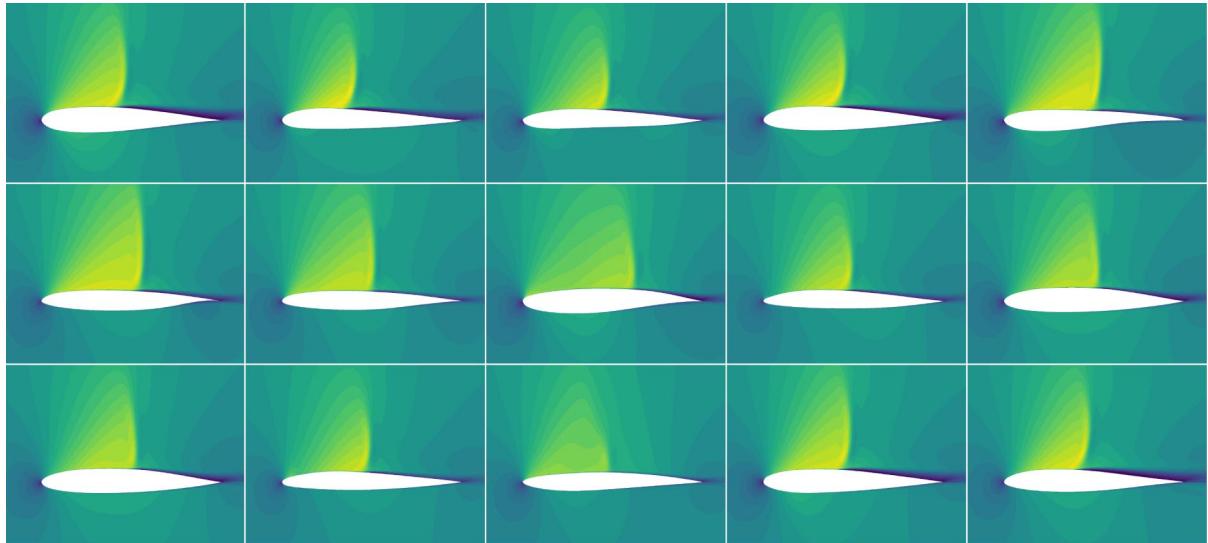


Fig. 5 Smooth monotonic decaying lines imply that the CNN discriminative function is suitable for gradient-based optimization.



a) Mach contours of LHS airfoils ($M = 0.75, \alpha = 2.5^\circ, Re = 6.5 \times 10^6$)



b) Mach contours of GAN airfoils ($M = 0.75, \alpha = 2.5^\circ, Re = 6.5 \times 10^6$)

Fig. 7 Comparison of airfoils generated by LHS and DCGAN.

and GAN wings are compared in Fig. 9, where the LHS wings are generated by perturbing FFD control points directly. Similarly to what we observed for the GAN airfoils, the GAN wings are much more realistic.

D. Impact of the Airfoil Database on DCGAN Training

Data-driven models rely on training data to learn the underlying patterns [42]. It is of great value to have the DCGAN model trained by an airfoil database that is big enough to involve all features of interest.

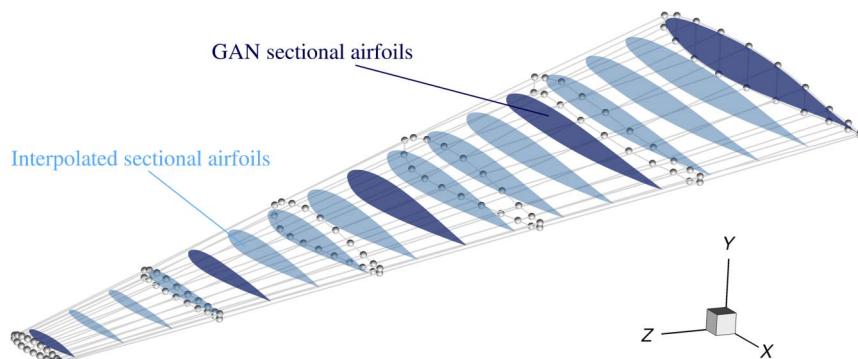


Fig. 8 From GAN sectional airfoils to wing samples with perturbed FFD control points.

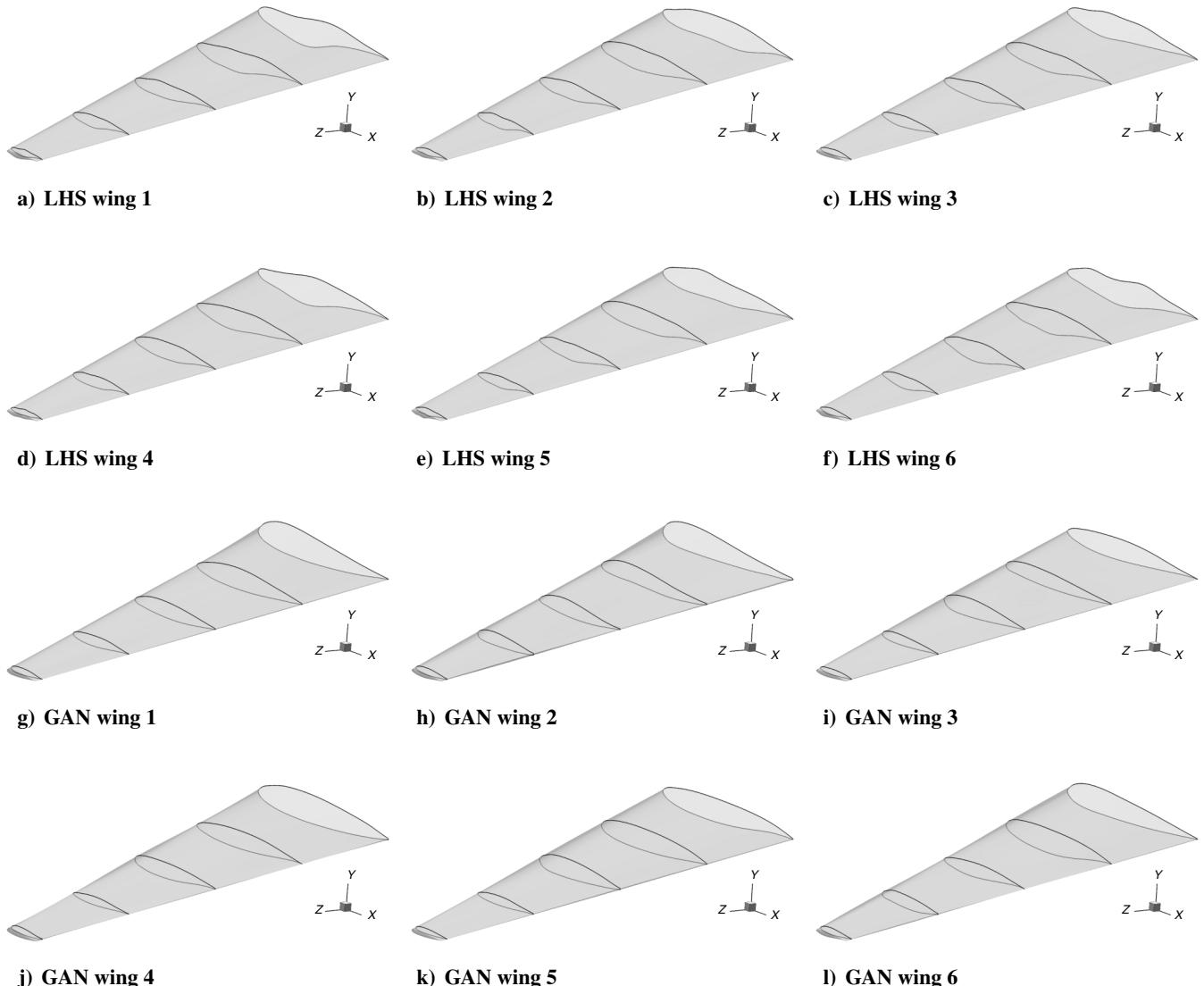


Fig. 9 GAN wings are much more realistic compared with those generated using LHS.

However, this is usually not the case because it is difficult to find airfoils beyond the UIUC database in the public domain. To investigate the influence of training data in synthetic airfoils, we train DCGAN with three different airfoil databases: NASA SC(2) database

with 21 supercritical airfoils, wind turbine database with 27 airfoils, and the NACA airfoil database with 100 airfoils. These databases are subsets of airfoils chosen from the UIUC airfoil database. As shown in Fig. 10, DCGAN models trained by these airfoil databases are all

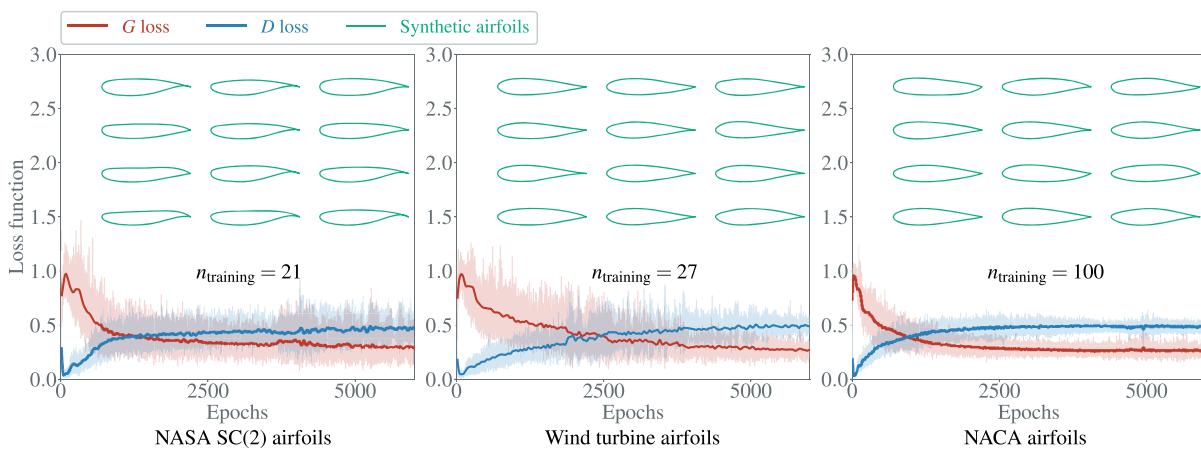


Fig. 10 DCGAN can be trained with modest airfoil databases to produce a specific series of airfoils.

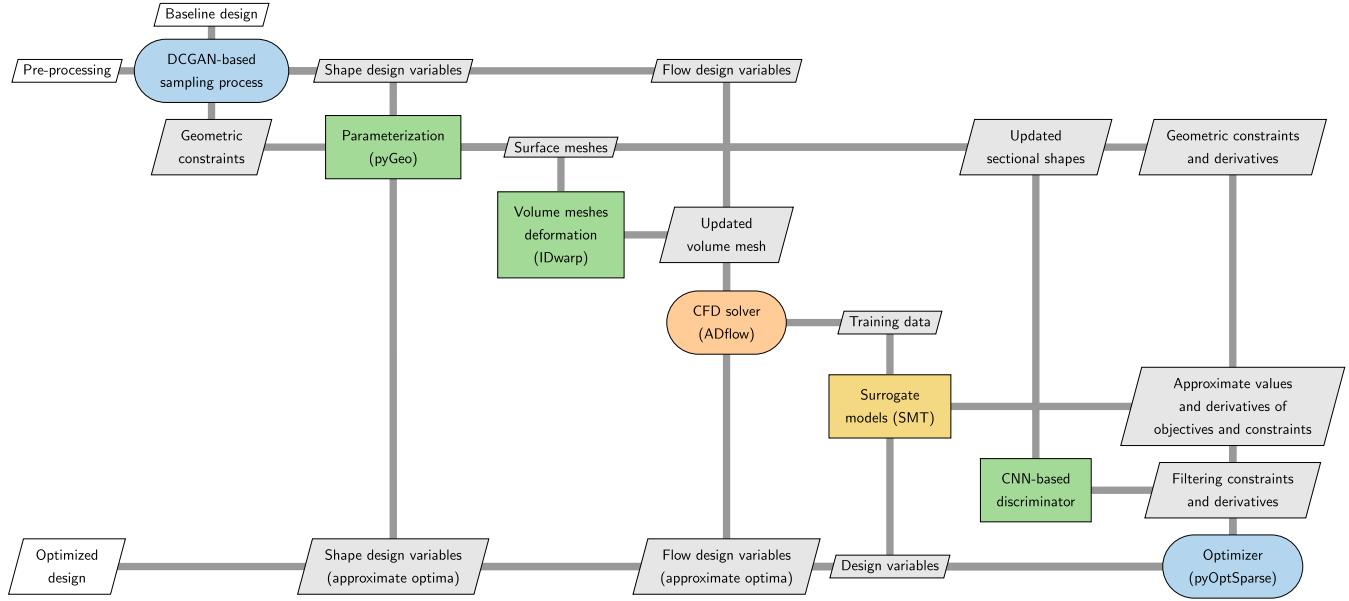


Fig. 11 XDSM diagram of pyAeroEGO.

Table 3 Optimization problem statement in the airfoil case

	Function	Quantity	Description
Minimize with respect to	C_d	1	Drag coefficient
	α	1	Angle of attack
	x_{shape}	18	y perturbation of FFD control points
subject to	$C_l \geq 0.824$	1	Lift constraint
	$A \geq A_{initial}$	1	Area constraint
	$S_{validity} \geq 0.9$	1	Discriminative score constraint

converged, which means that the DCGAN model works well with different sizes of training datasets. Similarly to Fig. 2, we show 3×4 synthetic airfoils for each DCGAN. These synthetic airfoils are generated by random inputs, i.e., latent space variables, thickness factors, and dominant camber coefficients. These inputs are the same for three DCGAN models. Nevertheless, as shown in Fig. 10, each particular DCGAN produces similar designs to the given series of airfoils. We notice that MMD decreases in these models. For example, the average MMD score of the DCGAN trained by NASA SC(2) airfoils is reduced to 0.043 (MMD = 0.120 in the DCGAN trained by all UIUC airfoils). Consequently, the synthetic airfoils of this

DCGAN model accurately retain the geometric features of the training data, which can be visually identified as supercritical airfoils.

The inception scores of the three DCGAN models are 3.96 (NASA SC(2) airfoils), 4.14 (wind turbine airfoils), and 4.40 (NACA airfoils). Compared with the DCGAN model trained by 1407 UIUC airfoils, there is a slight decrease. Nevertheless, the decrease is not large enough to be treated as a sign of mode collapse. Actually, the decrease of diversity enables synthetic airfoils to preserve specific geometric features of training airfoils. This is useful for the airfoil design of a given category, which is of potential to further improve optimization efficiency. For the aerodynamic design optimization of specific usage, like transonic wings in commercial aircraft, the training airfoil database of DCGAN can be more specific.

III. Aerodynamic Shape Optimization

A. Optimization Framework: pyAeroEGO

We use the efficient global optimization method (EGO) [3] to perform aerodynamic shape design in this paper. In EGO, ordinary kriging [15] is used to predict aerodynamic coefficients. Kriging also provides the variance of the prediction, which is used to evaluate the expected improvement (EI) function in EGO. The details of EGO in aerodynamic shape optimization can be found in previous work [4,43].

As shown in Fig. 11, we develop an aerodynamic EGO framework (pyAeroEGO) based on the open-source aerodynamic shape optimization suite MACH-Aero (<https://github.com/mdolab/MACH-Aero>). Aerodynamic shapes in pyAeroEGO are parameterized by FFD

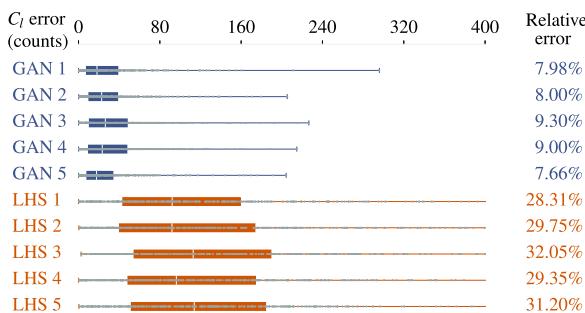
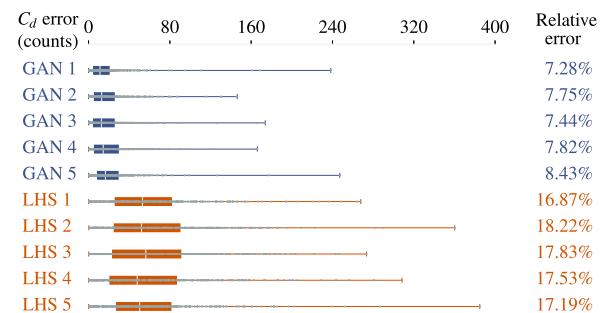
a) Errors of surrogate models for C_l b) Errors of surrogate models for C_d

Fig. 12 Surrogate models trained by GAN samples are much more accurate.

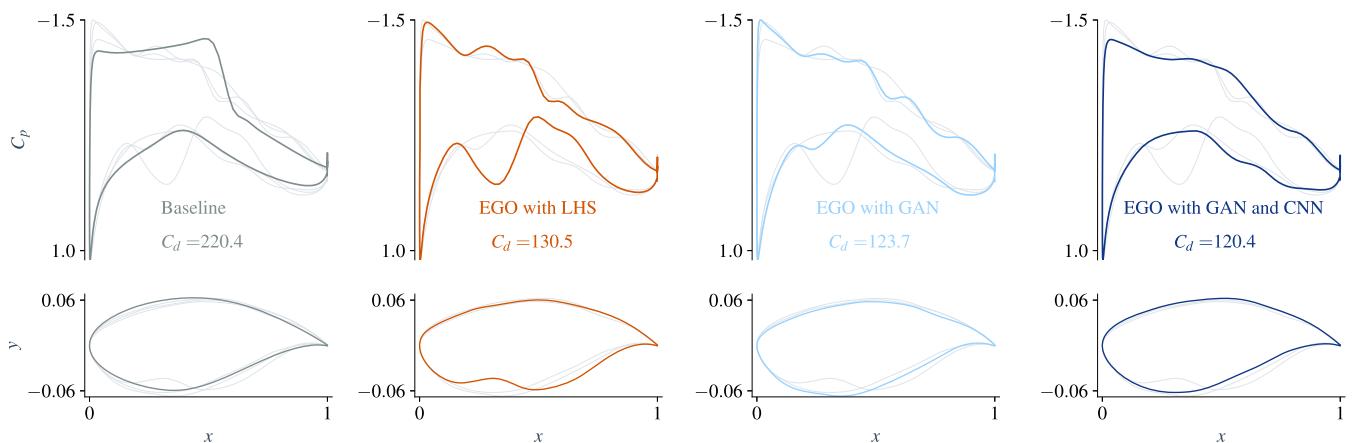


Fig. 13 EGO with GAN and CNN reduces drag the most.

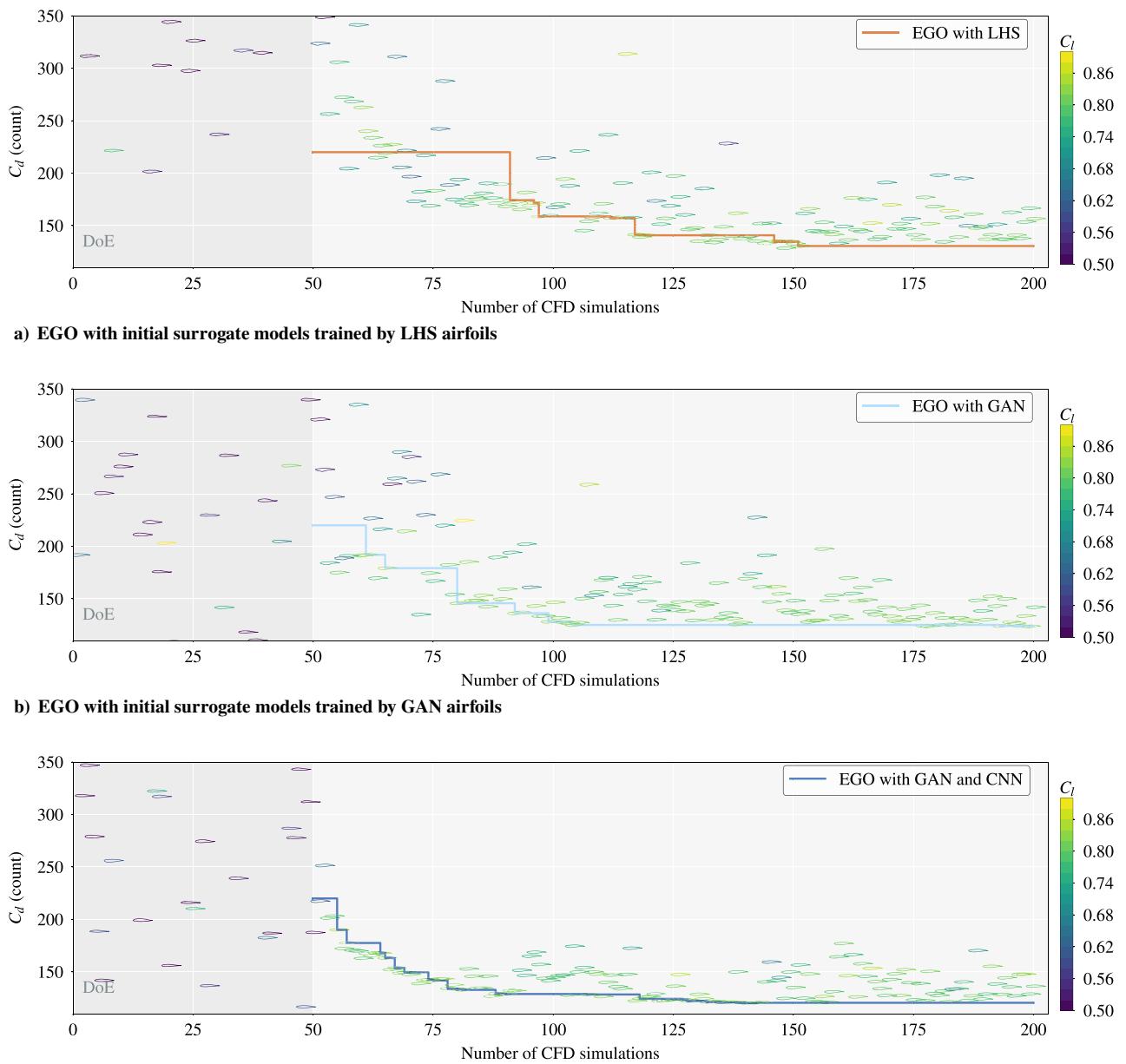


Fig. 14 GAN and CNN discriminator improve the efficiency of EGO in airfoil design.

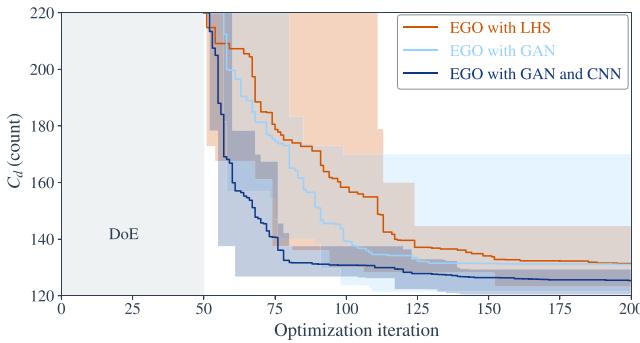


Fig. 15 Statistical data show that GAN and CNN double the efficiency of EGO.

implemented in pyGeo [44], which is a Python module for manipulating geometries (<https://github.com/mdolab/pygeo>). The optimization starts with a DoE process, where initial samples are generated by LHS or DCGAN. The volume meshes are deformed using IDWarp (<https://github.com/mdolab/idwarp>), which uses an efficient analytic inverse-distance method. Flow solutions of these samples are solved by ADflow [45], which is a finite-volume structured multiblock RANS code (<https://github.com/mdolab/adflow>) with an approximate Newton–Krylov solver [46]. Then, pyAeroEGO uses the kriging model implemented in the Surrogate Modeling Toolbox (SMT; <https://github.com/SMTorg/smt>) [47] to construct surrogate models of aerodynamic coefficients. Surrogate models trained by the initial samples are usually inaccurate, and they are refined via an infill sampling process. The infill process is performed by solving suboptimization problems, which consist of an objective minimization and an EI maximization.

Based on the results by Li et al. [38] and also on the findings of Haftka [48], gradient-based algorithms are more efficient and effective than gradient-free algorithms in solving suboptimizations of EGO. pyAeroEGO uses the sequential least-squares programming (SLSQP) algorithm implemented in pyOptSparse (<https://github.com/mdolab/pyoptsparse>) [49] coupled with a multistart strategy to solve these suboptimizations. pyGeo is used to ensure that all new infill samples satisfy the thickness or volume constraints. pyGeo also supports other geometric constraints like the fixed leading-edge point constraint, if required.

The CNN discriminator is coupled with pyGeo in pyAeroEGO as another geometric constraint to ensure the discriminative score of infill airfoil or wing sections larger than the lower bound. The new infill samples (solutions of suboptimizations) are further solved by ADflow. The surrogate models are updated until the convergence criterion is satisfied. The current version of pyAeroEGO

Table 4 Optimization problem statement in the wing case

	Function	Quantity	Description
Minimize with respect to	C_D	1	Drag coefficient
	α	1	Angle of attack
	$\mathbf{x}_{\text{shape}}$	120	y perturbation of FFD control points
subject to	$C_L \geq 0.5$	1	Lift constraint
	$V \geq V_{\text{initial}}$	1	Volume constraint
	$S_{\text{validity}} \geq 0.9$	5	Discriminative score constraints

uses a fixed iteration convergence criterion, which is a commonly used approach for comparing efficiency in surrogate-based optimization [4,38].

The XDSM (eXtended Design Structure Matrix) diagram [50] of pyAeroEGO is shown in Fig. 11. DCGAN and CNN models are used as the sampling method and geometric constraints, respectively. The training of DCGAN and CNN discriminator is fulfilled in advance before these deep-learning models are embedded in pyAeroEGO to perform aerodynamic design optimization. To show the advantage of the proposed method, we conduct three optimizations for each case:

EGO with LHS: EGO with initial surrogate models trained by LHS samples

EGO with GAN: EGO with initial surrogate models trained by DCGAN samples

EGO with GAN and CNN: EGO with initial surrogate models trained by DCGAN samples and suboptimizations subject to the CNN discriminator constraint

B. Airfoil Design Optimization

In this airfoil design optimization case, the baseline airfoil is RAE2822, the Mach number is 0.734, and the Reynolds number is 6.5×10^6 . The angle of attack is a design variable and can vary among the range of $[1.5^\circ, 4.5^\circ]$ in the optimization. We use a 10×2 FFD volume to control the airfoil shape. Figure 6 shows the FFD volume and the bounds of each FFD control point. The variation of the two FFD points around the leading and trailing edges is coupled (in the opposite direction) to ensure that the y coordinates of the leading and trailing edges are approximately zero. Thus there are 18 independent shape variables in this case. This optimization problem is summarized in Table 3. The discriminative score constraint applies only to EGO with GAN and CNN.

We first compare the accuracy of the surrogate models trained using LHS airfoils with those using GAN airfoils to verify our

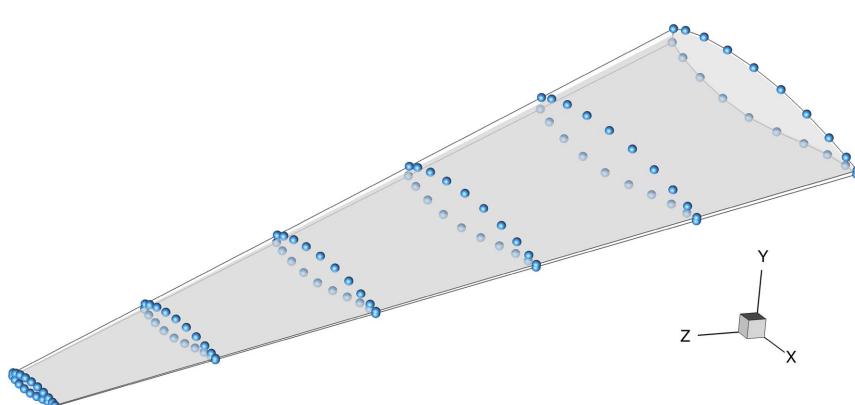
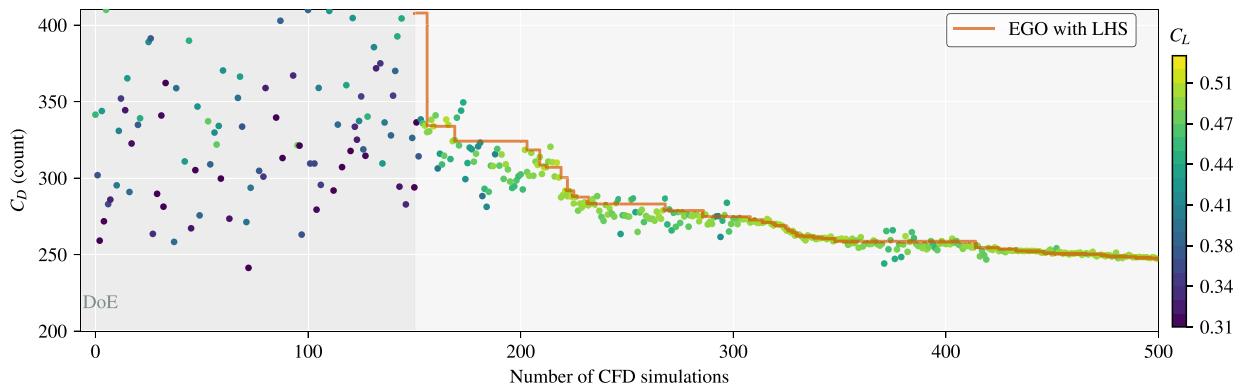
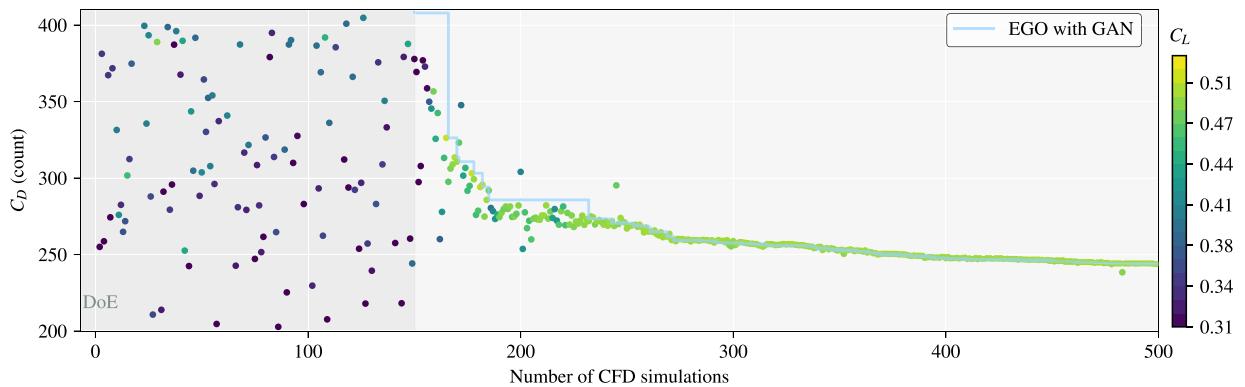


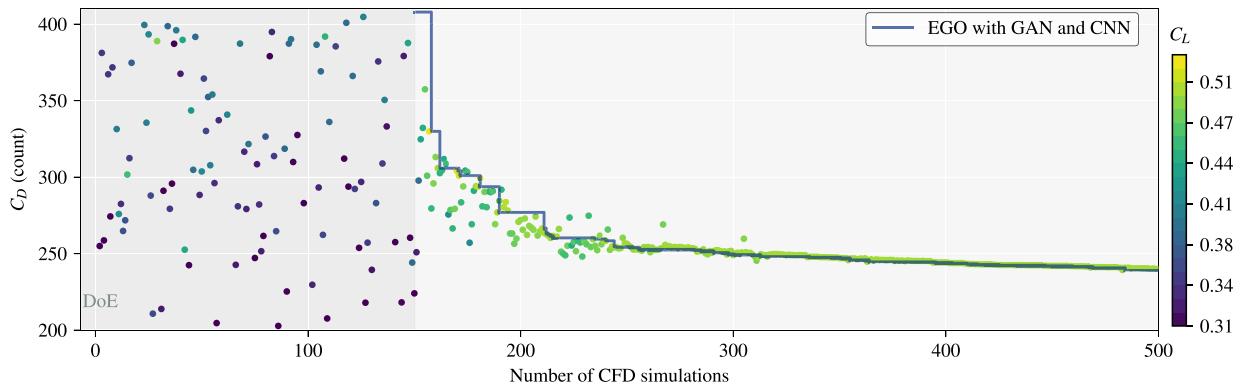
Fig. 16 Wing geometry and the FFD control points.



a) EGO with initial surrogate models trained by LHS wings



b) EGO with initial surrogate models trained by GAN wings



c) EGO with CNN constraint with initial surrogate models trained by GAN wings

Fig. 17 EGO with GAN and CNN makes aerodynamic design optimization of the wing converge efficiently.

hypothesis. Figure 12 shows the errors of surrogate models trained by different sets of samples and each set consists of 50 LHS or GAN airfoils. The errors shown in Fig. 12 are reported in counts (one C_L count equals 0.001 and one C_d count equals 0.0001). The scatter plot shows the error of each validation point, and the box plot shows the statistics for these errors. The white line in the box plot indicates the median of the errors. As expected, sampling with GAN significantly improves the accuracy of surrogate models, which implies that the design space of GAN samples is more representative and thus more efficient in the construction of accurate surrogate models.

We conduct airfoil design optimizations using pyAeroEGO. Each DoE consists of 50 samples. The optimization results and convergence histories are shown in Figs. 13 and 14. The surrogate models

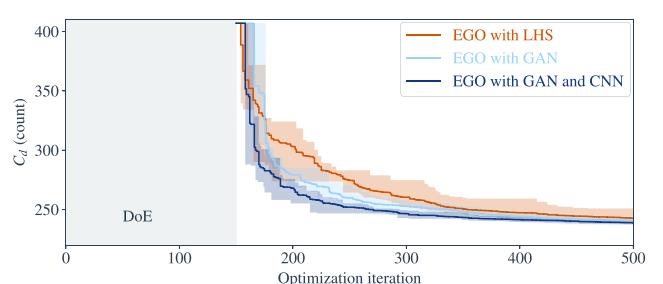


Fig. 18 Multiple tests show the improvement of EGO efficiency by GAN and CNN.

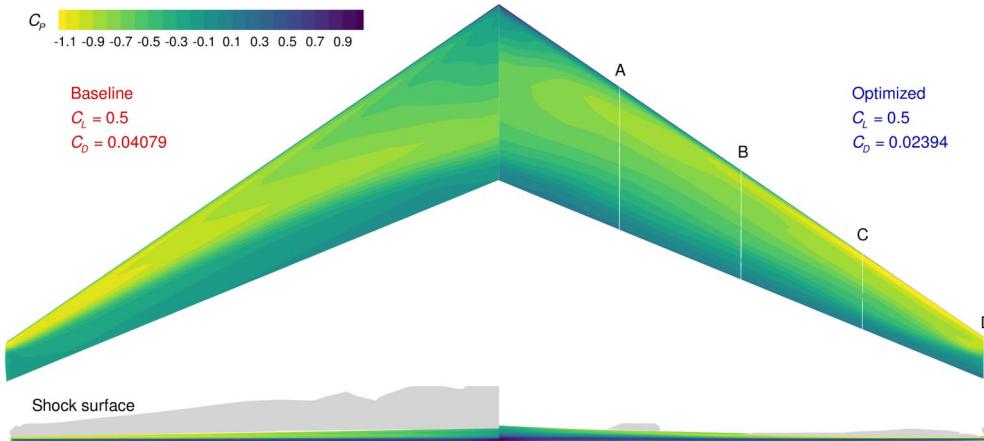


Fig. 19 The wing optimized using EGO with GAN and CNN has a much lower drag.

get more accurate with increasing infill samples. Thus, the minimum drag of samples satisfying the lift constraint is getting smaller, which is shown by the convergence lines in Fig. 14. In addition to the convergence line, we also show the airfoil shape of each sample in Fig. 14. These airfoil shapes are drawn using a color that represents their design lift coefficients.

EGO with GAN is much more efficient than EGO with LHS. However, some infill airfoils of EGO with GAN are still abnormal. After adding the CNN discriminator as a constraint in EGO with GAN and CNN, all the infill airfoils become realistic. As a result, EGO with GAN and CNN achieves the highest optimization efficiency and the best airfoil design for the same computational budget.

To reduce the influence from a particular DoE, we run 10 sets of optimization. Each set consists of three optimization settings: EGO with LHS, EGO with GAN, and EGO with GAN and CNN. Figure 15 shows the convergence histories, where the lines and margins are the average and ranges of convergence histories with each setting, respectively. On average, EGO with GAN and CNN obtains a design with $C_d < 135$ counts using only 75 CFD simulations, whereas EGO with LHS uses about 150 CFD simulations. EGO with GAN also improves the efficiency, which reaches $C_d < 135$ after about 100 CFD simulations. However, it is not shown to be more effective than EGO with LHS after 200 CFD simulations. The optimized drag coefficients for both settings are about 130 counts, which is larger than the optimized C_d in EGO with GAN and CNN (125 counts). Therefore, we conclude that EGO with GAN and CNN does not only improve the optimization efficiency but also makes the design optimization more effective.

C. Wing Design Optimization

We now perform the wing shape optimization for a transonic commercial aircraft. The wing planform is based on publicly available Boeing 787-800 drawings and initial airfoil shapes correspond to the RAE 2822 airfoil. The mean chord of the wing is 1.35 m, and the sweep angle at the quarter-chord line is 32.2° . The wing span is 6.1 m. The FFD in Fig. 16 is used to control the wing shape, where the shape design variables are the y displacements of the 120 FFD control points. We impose CNN discriminative constraints at five spanwise sections ($z = 0.1, z = 1.5, z = 3.0, z = 4.5$, and $z = 6.0$), which only apply to EGO with GAN and CNN. This optimization problem is summarized in Table 4.

To reduce the computational cost, we use a coarse CFD mesh with 151,200 volume cells in the wing design optimization. Figure 17 shows the EGO convergence histories with different settings, where each optimization starts with 150 initial samples. Similarly to the airfoil optimization results, EGO with GAN and CNN is the most efficient. Within the computational budget of 500 CFD simulations, EGO with LHS, EGO with GAN, and EGO with GAN and CNN reduce the drag coefficient from 408 counts of the baseline wing to 247.5 counts, 243.9 counts, and 239.4 counts, respectively. Compared with EGO with LHS, EGO with GAN improves the optimization efficiency by using more realistic initial samples. This benefit does not always generate a better optimization performance in each iteration. For example, the best design of EGO with LHS is slightly better than that of EGO with GAN in the 220th CFD simulation. After adding the geometric constraints with the CNN discriminator, the optimization efficiency improves significantly. The design of the 215th CFD simulation in EGO with GAN and CNN has a similar C_d with the design of the 420th CFD simulation in EGO with LHS. From this

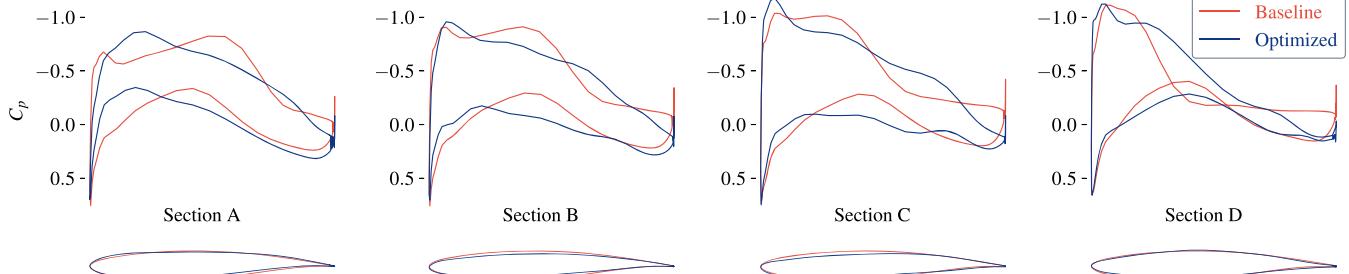


Fig. 20 Airfoils optimized by EGO with GAN and CNN.

point of view, EGO with GAN and CNN almost doubles the optimization efficiency of a regular EGO with LHS. Figure 18 shows the results of multiple tests, which more credibly demonstrate the efficiency of EGO with GAN and CNN.

Figure 19 compares the baseline wing to the wing optimized using EGO with GAN and CNN. The optimization significantly reduces the drag by weakening shock waves on the upper surface. Similarly to previous transonic wing shape optimization subject to a fixed volume constraint [1,29,51], EGO with GAN and CNN reduces the drag by increasing thickness of the wing root, which is increased to 14.3% from 12.1%. Thickness of some sections in the outer wing is reduced to a minimum of 9.3%, but the volume constraint is still satisfied. Figure 20 shows the optimized sectional airfoil shapes and pressure distributions. EGO with GAN and CNN makes the optimized wing almost shock-free.

IV. Conclusions

In this work, a DCGAN-based sampling method is proposed to generate realistic airfoils and wings for DoE. With realistic airfoils generated using DCGAN and abnormal airfoils generated using LHS as the training data, a CNN-based discriminative model is developed to efficiently detect geometric abnormal airfoils and wing sections. A surrogate-based optimization framework (pyAeroEGO) that combines these models with CFD to perform aerodynamic shape optimization is developed. Airfoil and wing shape optimizations show that these machine learning models significantly improve the optimization efficiency of EGO.

As the fundamental model in this work, DCGAN not only influences the performance of the sampling method, but also determines whether the CNN discriminator can distinguish abnormal airfoils from realistic ones. The parameters in the DCGAN model are studied, and it is found that its performance is robust and stable. The proposed airfoil normalization method helps address mode collapse issues. DCGAN can always generate synthetic airfoils similar to realistic airfoils unless the size or depth of convolutional filters in DCGAN is too small.

It is also found that the performance of DCGAN is robust to the number of training airfoils. A reasonable DCGAN model can be trained using a small database of only dozens of airfoils, and synthetic airfoils of this model reserve the specific geometric features of training airfoils. From this perspective, our proposed framework is of high potential for aircraft manufacturers to further improve optimization efficiency by training a specific DCGAN model using their own airfoil database.

Previous surrogate-based optimization methods tend to suffer from inaccurate surrogate models. Our work addresses this issue by using deep-learning models. Realistic samples of DCGAN span a simpler design space, which helps to train more accurate surrogate models. Constraints enforced by the CNN discriminator exclude the abnormal shapes from the infill samples. These machine learning models improve the accuracy of surrogate models and make the aerodynamic shape optimization much more efficient. Nevertheless, because the deep-learning models are trained by previous designs, future work could investigate whether these models would prevent the discovery of innovative shapes.

This work showcases the superiority of applying deep-learning algorithms in numerical aircraft design. This approach can also be applied to the design of other aircraft components or other engineering design problems, which is helpful to shorten the design process and reduce costs.

Appendix A: Airfoil DCGAN

We set the x coordinates for all UIUC airfoils to be

$$x_i = \frac{1}{2} \left(\cos \frac{2\pi(i-1)}{250} + 1 \right), \quad i = 1, 2, \dots, 251 \quad (\text{A1})$$

As proposed by Li et al. [6], camber and thickness mode shapes can be used to represent airfoils. For each airfoil, the coordinates of the camber and thickness lines can be recorded as (x_i, y_{c_i}, y_{t_i}) , where $i = 1, \dots, n_f$ ($n_f = 126$). Similarly to the full-airfoil mode approach, the y coordinate values of the camber and thickness lines are assembled using snapshot matrices

$$\mathbf{A}_c = \begin{bmatrix} y_{c_{11}} & y_{c_{21}} & \cdots & y_{c_{m1}} \\ y_{c_{12}} & y_{c_{22}} & \cdots & y_{c_{m2}} \\ \vdots & \vdots & \ddots & \vdots \\ y_{c_{1n_f}} & y_{c_{2n_f}} & \cdots & y_{c_{mn_f}} \end{bmatrix}, \quad (\text{A2})$$

$$\mathbf{A}_t = \begin{bmatrix} y_{t_{11}} & y_{t_{21}} & \cdots & y_{t_{m1}} \\ y_{t_{12}} & y_{t_{22}} & \cdots & y_{t_{m2}} \\ \vdots & \vdots & \ddots & \vdots \\ y_{t_{1n_f}} & y_{t_{2n_f}} & \cdots & y_{t_{mn_f}} \end{bmatrix}$$

Then, the modes of camber and thickness lines, \mathbf{U}_c and \mathbf{U}_t , are obtained by SVD:

$$\mathbf{A}_c = \mathbf{U}_c \Sigma_c \mathbf{V}_c^T, \quad \mathbf{A}_t = \mathbf{U}_t \Sigma_t \mathbf{V}_t^T$$

We use Φ_c and Φ_t to denote the first n_c camber modes and the first n_t thickness modes, respectively. The first column of Φ_c is the first camber mode, which dominates the airfoil camber and is denoted as \mathbf{C} in Algorithm 1. For a given airfoil \mathbf{y} , we compute the dominant camber coefficient $c_1 = \mathbf{C}^T \mathbf{y}_{\text{camber}}$ with the camber line solved by $\mathbf{y}_{\text{camber}} = f_c(\mathbf{y})$.

Figure A1 shows the comparison of airfoils before and after the normalization stated in Algorithm 1.

Figure A2 shows the training processes of the DCGAN model (with $n_{\text{size}} = 7$ and $n_{\text{depth}} = 10$) with different initial weights. All of these models converge to a similar IG(G) with reasonable synthetic airfoils, which demonstrates the stability of DCGAN with different weight initializations.

Figure A3 shows that the discriminative model cannot distinguish abnormal LHS airfoils from realistic ones (UIUC and DCGAN synthetic airfoils), and thus an independent discriminative model is required.

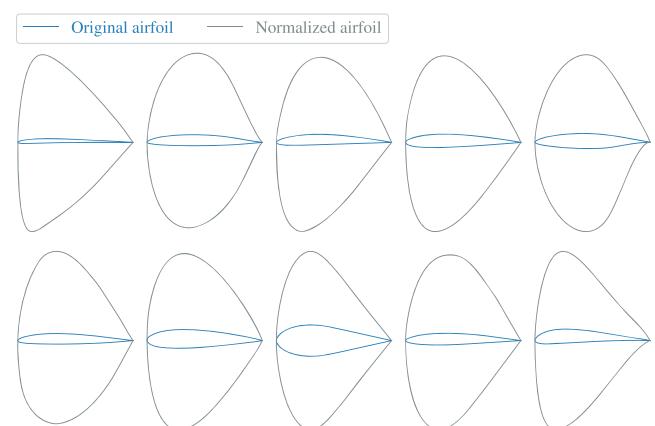


Fig. A1 Airfoils before and after the normalization.

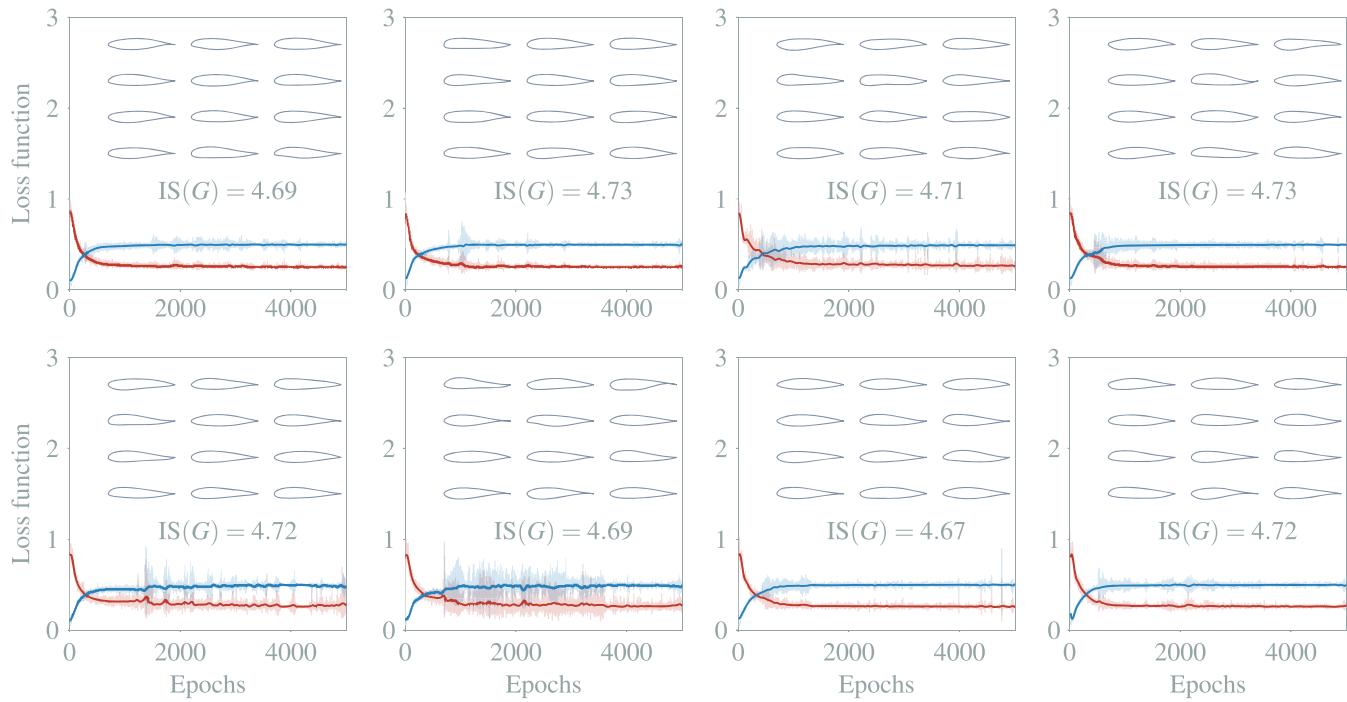


Fig. A2 Training DCGAN with different initial weights shows robustness of the selected model.

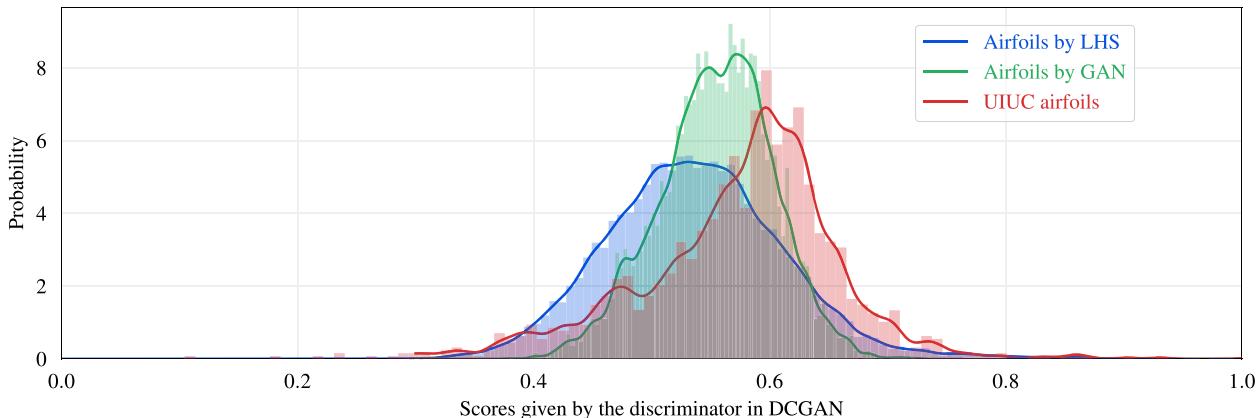


Fig. A3 The discriminative model in DCGAN cannot distinguish abnormal airfoils from realistic ones.

Acknowledgments

The computational resources of the National Supercomputing Centre, Singapore (<https://www.nscc.sg>), and National University of Singapore (NUS) Information Technology are acknowledged. Mengqi Zhang acknowledges the Tier 1 grant from the Ministry of Education, Singapore (No. R-265-000-654-114).

References

- [1] Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., "Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark," *AIAA Journal*, Vol. 53, No. 4, 2015, pp. 968–985. <https://doi.org/10.2514/1.J053318>
- [2] He, X., Li, J., Mader, C. A., Yildirim, A., and Martins, J. R. R. A., "Robust Aerodynamic Shape Optimization—From a Circle to an Airfoil," *Aerospace Science and Technology*, Vol. 87, April 2019, pp. 48–61. <https://doi.org/10.1016/j.ast.2019.01.051>
- [3] 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. <https://doi.org/10.1023/A:1008306431147>
- [4] Li, J., Cai, J., and Qu, K., "Surrogate-Based Aerodynamic Shape Optimization with the Active Subspace Method," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 2, 2019, pp. 403–419. <https://doi.org/10.1007/s00158-018-2073-5>
- [5] Sun, G., and Wang, S., "A Review of the Artificial Neural Network Surrogate Modeling in Aerodynamic Design," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 16, 2019, pp. 5863–5872. <https://doi.org/10.1177/0954410019864485>
- [6] Li, J., Bouhlel, M. A., and Martins, J. R. R. A., "Data-Based Approach for Fast Airfoil Analysis and Optimization," *AIAA Journal*, Vol. 57, No. 2, 2019, pp. 581–596. <https://doi.org/10.2514/1.J057129>
- [7] Sekar, V., Zhang, M., Shu, C., and Khoo, B. C., "Inverse Design of Airfoil Using a Deep Convolutional Neural Network," *AIAA Journal*, Vol. 57, No. 3, 2019, pp. 993–1003. <https://doi.org/10.2514/1.J057894>
- [8] Wang, S., Sun, G., Chen, W., and Zhong, Y., "Database Self-Expansion Based on Artificial Neural Network: An Approach in Aircraft Design," *Aerospace Science and Technology*, Vol. 72, Jan. 2018, pp. 77–83. <https://doi.org/10.1016/j.ast.2017.10.037>
- [9] Li, J., Cai, J., and Qu, K., "Adjoint-Based Two-Step Optimization Method Using Proper Orthogonal Decomposition and Domain Decomposition," *AIAA Journal*, Vol. 56, No. 3, 2018, pp. 1133–1145. <https://doi.org/10.2514/1.J055773>
- [10] Li, J., and Cai, J., "Massively Multipoint Aerodynamic Shape Design via Surrogate-Assisted Gradient-Based Optimization," *AIAA Journal*,

- Vol. 58, No. 5, 2020, pp. 1949–1963.
<https://doi.org/10.2514/1.J058491>
- [11] Yondo, R., Andrés, E., and Valero, E., “A Review on Design of Experiments and Surrogate Models in Aircraft Real-Time and Many-Query Aerodynamic Analyses,” *Progress in Aerospace Sciences*, Vol. 96, Jan. 2018, pp. 23–61.
<https://doi.org/10.1016/j.paerosci.2017.11.003>
- [12] Koziel, S., and Leifsson, L., “Surrogate-Based Aerodynamic Shape Optimization by Variable-Resolution Models,” *AIAA Journal*, Vol. 51, No. 1, 2013, pp. 94–106.
<https://doi.org/10.2514/1.J051583>
- [13] Han, Z.-H., Abu-Zurayk, M., Görtz, S., and Ilic, C., “Surrogate-Based Aerodynamic Shape Optimization of a Wing-Body Transport Aircraft Configuration,” *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, Springer International Publishing, New York, 2018, pp. 257–282.
https://doi.org/10.1007/978-3-319-72020-3_16
- [14] Zuo, Y., Chen, P., Fu, L., Gao, Z., and Chen, G., “Advanced Aerostructural Optimization Techniques for Aircraft Design,” *Mathematical Problems in Engineering*, Vol. 2015, Oct. 2015, pp. 1–12.
<https://doi.org/10.1155/2015/753042>
- [15] Krige, D. G., “A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand,” *Journal of the Chemical, Metallurgical and Mining Society*, Vol. 52, No. 6, 1951, pp. 119–139.
- [16] Han, Z.-H., Zhang, Y., Song, C.-X., and Zhang, K.-S., “Weighted Gradient-Enhanced Kriging for High-Dimensional Surrogate Modeling and Design Optimization,” *AIAA Journal*, Vol. 55, No. 12, 2017, pp. 4330–4346.
<https://doi.org/10.2514/1.J055842>
- [17] Bouhlel, M. A., and Martins, J. R. R. A., “Gradient-Enhanced Kriging for High-Dimensional Problems,” *Engineering with Computers*, Vol. 35, No. 1, 2019, pp. 157–173.
<https://doi.org/10.1007/s00366-018-0590-x>
- [18] Bouhlel, M. A., Bartoli, N., Otsmane, A., and Morlier, J., “Improving Kriging Surrogates of High-Dimensional Design Models by Partial Least Squares Dimension Reduction,” *Structural and Multidisciplinary Optimization*, Vol. 53, No. 5, 2016, pp. 935–952.
<https://doi.org/10.1007/s00158-015-1395-9>
- [19] Liem, R. P., Mader, C. A., and Martins, J. R. R. A., “Surrogate Models and Mixtures of Experts in Aerodynamic Performance Prediction for Aircraft Mission Analysis,” *Aerospace Science and Technology*, Vol. 43, June 2015, pp. 126–151.
<https://doi.org/10.1016/j.ast.2015.02.019>
- [20] Robinson, G. M., and Keane, A. J., “Concise Orthogonal Representation of Supercritical Airfoils,” *Journal of Aircraft*, Vol. 38, No. 3, 2001, pp. 580–583.
<https://doi.org/10.2514/2.2803>
- [21] Poole, D. J., Allen, C. B., and Rendall, T. C. S., “Metric-Based Mathematical Derivation of Efficient Airfoil Design Variables,” *AIAA Journal*, Vol. 53, No. 5, 2015, pp. 1349–1361.
<https://doi.org/10.2514/1.J053427>
- [22] Masters, D. A., Taylor, N. J., Rendall, T. C. S., Allen, C. B., and Poole, D. J., “Geometric Comparison of Aerofoil Shape Parameterization Methods,” *AIAA Journal*, Vol. 55, No. 5, 2017, pp. 1575–1589.
<https://doi.org/10.2514/1.J054943>
- [23] Chen, W., Chiu, K., and Fuge, M., “Aerodynamic Design Optimization and Shape Exploration Using Generative Adversarial Networks,” *AIAA SciTech 2019 Forum*, AIAA Paper 2019-2351, 2019.
<https://doi.org/10.2514/6.2019-2351>
- [24] 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, New York, 2014, pp. 2672–2680.
- [25] McKay, M. D., Beckman, R. J., and Conover, W. J., “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code,” *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.
- [26] Radford, A., Metz, L., and Chintala, S., “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” arXiv preprint arXiv:1511.06434, 2015.
- [27] Wu, H., Zheng, S., Zhang, J., and Huang, K., “GP-GAN: Towards Realistic High-Resolution Image Blending,” *Proceedings of the 27th ACM International Conference on Multimedia (MM 19)*, Assoc. for Computing Machinery, New York, 2019.
<https://doi.org/10.1145/3343031.3350944>
- [28] Kingma, D. P., and Ba, J., “Adam: A Method for Stochastic Optimization,” arXiv preprint arXiv:1412.6980, 2014.
- [29] Li, J., He, S., and Martins, J. R. R. A., “Data-Driven Constraint Approach to Ensure Low-Speed Performance in Transonic Aerodynamic Shape Optimization,” *Aerospace Science and Technology*, Vol. 92, Sept. 2019, pp. 536–550.
<https://doi.org/10.1016/j.ast.2019.06.008>
- [30] Lipton, Z. C., and Tripathi, S., “Precise Recovery of Latent Vectors from Generative Adversarial Networks,” arXiv preprint arXiv:1702.04782, 2017.
- [31] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M., “Striving for Simplicity: The All Convolutional Net,” arXiv preprint arXiv:1412.6806, 2014.
- [32] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X., “Improved Techniques for Training Gans,” *Advances in Neural Information Processing Systems*, Curran Associates, Inc., New York, 2016, pp. 2234–2242.
- [33] Arjovsky, M., Chintala, S., and Bottou, L., “Wasserstein Generative Adversarial Networks,” *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70, 2017, pp. 214–223.
- [34] Papamakarios, G., Pavlakou, T., and Murray, I., “Masked Autoregressive Flow for Density Estimation,” *Advances in Neural Information Processing Systems*, Curran Associates, Inc., New York, 2017, pp. 2338–2347.
- [35] Grathwohl, W., Chen, R. T. Q., Bettencourt, J., and Duvenaud, D., “FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models,” *International Conference on Learning Representations*, 2018.
- [36] Zhou, Z., Cai, H., Rong, S., Song, Y., Ren, K., Zhang, W., Yu, Y., and Wang, J., “Activation Maximization Generative Adversarial Nets,” arXiv preprint arXiv:1703.02000, 2017.
- [37] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A., “A Kernel Two-Sample Test,” *Journal of Machine Learning Research*, Vol. 13, March 2012, pp. 723–773.
- [38] Li, J., Cai, J., and Qu, K., “Drag Reduction of Transonic Wings with Surrogate-Based Optimization,” *Proceedings of the 2018 Asia-Pacific International Symposium on Aerospace Technology (APISAT 2018)*, 2019, pp. 1065–1080.
https://doi.org/10.1007/978-981-13-3305-7_85
- [39] Bartoli, N., Meliani, M., Morlier, J., Lefebvre, T., Bouhlel, M.-A., and Martins, J. R. R. A., “Multi-Fidelity Efficient Global Optimization: Methodology and Application to Airfoil Shape Design,” *AIAA Aviation Forum*, AIAA Paper 2019-3236, 2019.
<https://doi.org/10.2514/6.2019-3236>
- [40] Bui-Thanh, T., Willcox, K., and Ghattas, O., “Parametric Reduced-Order Models for Probabilistic Analysis of Unsteady Aerodynamic Applications,” *AIAA Journal*, Vol. 46, No. 10, 2008, pp. 2520–2529.
<https://doi.org/10.2514/1.35850>
- [41] Li, J., Qu, K., Cai, J., and Cao, C., “Adjoint Approach Based on Reduced-Order Model for Steady PDE Systems,” *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA Paper 2016-3668, 2016.
<https://doi.org/10.2514/6.2016-3668>
- [42] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., “Machine Learning for Fluid Mechanics,” *Annual Review of Fluid Mechanics*, Vol. 52, Jan. 2020, pp. 477–508.
<https://doi.org/10.1146/annurev-fluid-010719-060214>
- [43] Liu, F., Han, Z.-H., Zhang, Y., Song, K., Song, W.-P., Gui, F., and Tang, J.-B., “Surrogate-Based Aerodynamic Shape Optimization of Hypersonic Flows Considering Transonic Performance,” *Aerospace Science and Technology*, Vol. 93, Oct. 2019, Paper 105345.
<https://doi.org/10.1016/j.ast.2019.105345>
- [44] Kenway, G. K., Kennedy, G. J., and Martins, J. R. R. A., “A CAD-Free Approach to High-Fidelity Aerostructural Optimization,” *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, AIAA Paper 2010-9231, 2010.
<https://doi.org/10.2514/6.2010-9231>
- [45] Kenway, G. K. W., Mader, C. A., He, P., and Martins, J. R. R. A., “Effective Adjoint Approaches for Computational Fluid Dynamics,” *Progress in Aerospace Sciences*, Vol. 110, Sept. 2019, Paper 100542.
<https://doi.org/10.1016/j.paerosci.2019.05.002>
- [46] Yıldırım, A., Kenway, G. K. W., Mader, C. A., and Martins, J. R. R. A., “A Jacobian-Free Approximate Newton–Krylov Startup Strategy for RANS Simulations,” *Journal of Computational Physics*, Vol. 397, Nov. 2019, Paper 108741.
<https://doi.org/10.1016/j.jcp.2019.06.018>
- [47] Bouhlel, M. A., Hwang, J. T., Bartoli, N., Lafage, R., Morlier, J., and Martins, J. R. R. A., “A Python Surrogate Modeling Framework with Derivatives,” *Advances in Engineering Software*, Vol. 135, Sept. 2019, Paper 102662.
<https://doi.org/10.1016/j.advengsoft.2019.03.005>

- [48] Haftka, R. T., "Requirements for Papers Focusing on New or Improved Global Optimization Algorithms," *Structural and Multidisciplinary Optimization*, Vol. 54, No. 1, 2016, p. 1.
<https://doi.org/10.1007/s00158-016-1491-5>
- [49] Perez, R. E., Jansen, P. W., and Martins, J. R. R. A., "pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization," *Structural and Multidisciplinary Optimization*, Vol. 45, No. 1, 2012, pp. 101–118.
<https://doi.org/10.1007/s00158-011-0666-3>
- [50] Lambe, A. B., and Martins, J. R. R. A., "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," *Structural and Multidisciplinary Optimization*, Vol. 46, No. 2, 2012, pp. 273–284.
<https://doi.org/10.1007/s00158-012-0763-y>
- [51] Kenway, G. K. W., and Martins, J. R. R. A., "Multipoint Aerodynamic Shape Optimization Investigations of the Common Research Model Wing," *AIAA Journal*, Vol. 54, No. 1, 2016, pp. 113–128.
<https://doi.org/10.2514/1.J054154>

D. ZHAO
Associate Editor