

Article

Cooperation of Thin-Airfoil Theory and Deep Learning for a Compact Airfoil Shape Parameterization

Jianmiao Yi and Feng Deng * 

College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; yi_jianmiao@nuaa.edu.cn

* Correspondence: fdeng@nuaa.edu.cn

Abstract: An airfoil shape parameterization that can generate a compact design space is highly desirable in practice. In this paper, a compact airfoil parameterization is proposed by incorporating deep learning into the PAERO parameterization method based on the thin-airfoil theory. Following the PAERO parameterization, the mean camber line is represented by a number of aerodynamic performance parameters, which can be used to narrow down the design space according to the thin-airfoil theory. In order to further reduce the design space, the airfoil thickness distribution is represented by data-driven generative models, which are trained by the thickness distributions of existing airfoils. The trained models can automatically filter out the physically unreasonable airfoil shapes, resulting in a highly compact design space. The test results show that the proposed method is significantly more efficient and more robust than the widely used CST parameterization method for airfoil optimization.

Keywords: aerodynamic shape optimization; airfoil parameterization; thin-airfoil theory; deep learning; generative model



Citation: Yi, J.; Deng, F. Cooperation of Thin-Airfoil Theory and Deep Learning for a Compact Airfoil Shape Parameterization. *Aerospace* **2023**, *10*, 650. <https://doi.org/10.3390/aerospace10070650>

Received: 28 June 2023

Revised: 13 July 2023

Accepted: 18 July 2023

Published: 20 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Airfoils play a critical role in the aerodynamic design of aircraft, wind turbines and gas turbines. Airfoil optimization design includes three components, namely airfoil parameterization, Computational Fluid Dynamics (CFD) simulation and optimization search. It is well known that during optimization, the number of CFD evaluations increases rapidly with the increase in design space, which is known as the “curse of dimensionality” [1,2]. Although the gradient-based methods such as the adjoint method [3,4] can make the number of CFD evaluations almost independent of design variables, the gradients are usually either not available or difficult to calculate. For the surrogate-based methods [5] and the gradient-free methods [6], an efficient parameterization method that can generate a compact design space is still highly desirable [7].

Traditionally, the airfoil parameterization methods can be classified into two categories. The first category consists of the direct construction methods such as parametric section (PARSEC) [8], class function/shape function transformation (CST) [9] and B-spline curves [10], and the second category consists of the perturbation deformation methods such as Hicks–Henne bump functions [11] and the free-form deformation (FFD) method [12]. In recent years, the data-driven methods have gradually been proved to be able to reduce the number of design variables in aerodynamic shape optimization. Masters et al. [13] studied a number of parameterization methods, concluding that data-driven singular value decomposition (SVD) parameterization can minimize the number of design variables representing airfoils. The idea of SVD is that a number of orthogonal modes can be extracted from the existing airfoil geometries, and then, the linear combination of these modes can be used to construct new airfoils. Li et al. [14] extended this idea by extracting the orthogonal modes from the airfoil thickness and the airfoil camber, respectively. More recently, Chen et al. [15]

proposed Bézier Generative Adversarial Networks (Bézier-GANs) for airfoil shape parameterization. By learning from the existing UIUC [16] airfoil database, a Bézier-GAN can ensure a physically reasonable airfoil geometry, resulting in a compact design space for airfoil optimization. Du et al. [17,18] incorporated a surrogate model into a B-spline-based GAN to further speed up the airfoil design. In addition, Li et al. [19,20] adopted the UIUC airfoil database and a number of randomly deformed airfoils to train a discriminative model for unrealistic airfoils, resulting in a further speedup.

Although the above methods are proven to be useful in some applications, their design variables are usually independent of aerodynamic performances. In optimization design, aerodynamic performances often appear as goals or constraints. Therefore, incorporating aerodynamic performances into shape parameterization can provide extra guidance for optimal design. Recently, a number of studies introduced aerodynamic performances into shape parameterization by using deep learning technology. Achour et al. [21] trained a Conditional Generative Adversarial Network (CGAN) that can generate an airfoil geometry directly by a given lift–drag ratio and a given shape area. The results showed that about 75% of the generated airfoils maintained the given aerodynamic performances. Yilmaz et al. [22] adopted the airfoil stall performance and the drag pole curve as given design parameters to train a CGAN. Sekar et al. [23] proposed a deep learning model that can map a given pressure distribution to an airfoil geometry. Wang et al. [24] proposed an inverse design method for supercritical airfoils based on the Conditional Variational AutoEncoder (CVAE) and Wasserstein Generative Adversarial Networks (WGANs). This method first generates a Mach number distribution with given features, and then, the geometric coordinates are obtained from this Mach number distribution by a neural network model built for nonlinear mapping. In addition, Lei et al. [25] combined the generative model and the optimization algorithm to find a pressure coefficient distribution with given features and then mapped it to the corresponding airfoil geometry.

It is worth noting that the above trained models are often limited by the given training data, and they cannot provide a general shape parameterization method. Inspired by the classical thin-airfoil theory [26], Deng et al. [27] proposed a novel general parameterization method, named PAERO, which uses four aerodynamic performance parameters, namely the Mach number, the angle of attack, the lift coefficient and the pitching moment coefficient, and a number of optional parameters to represent the mean camber line of the airfoil. In the original method, the airfoil thickness distribution is represented by either the CST method or B-spline curves. In addition to being a general flexible shape parameterization, this method provides four aerodynamic performance parameters to manipulate the design space directly based on the thin-airfoil theory.

Following the previous work [26], this paper aims to extend the PAERO method by absorbing the advantages of deep learning technology. The airfoil thickness distribution will be represented by a data-driven generative model, which is built from the standard dataset of UIUC. The trained model can ensure a physically reasonable geometry due to its learning ability. The proposed parameterization method is expected to have the following advantages: (1) it can generate airfoil shapes with given aerodynamic performance parameters; (2) the aerodynamic performance parameters can be used as design variables, providing an efficient way to narrow down the design space; (3) the compact thickness representation can avoid a physically unreasonable geometry and reduce the design space as well.

2. Methodology

2.1. Problem Definition

The airfoil shape and coordinate system of interest are shown in Figure 1. It has a round nose with a radius equal to r_{LE} and a trailing-edge thickness equal to $\Delta z_{TE}^{(t)}$. The coordinate system origin lies on its leading edge. Its chord line coincides with the x axis, with the chord length being c . In order to parameterize this airfoil, its shape was decomposed into two basic components, namely the mean camber line and the thickness distribution, denoted as

$z^{(s)}(x)$ and $z^{(t)}(x)$, respectively. For brevity, the coordinates are normalized by the chord length, introduced as $X = \frac{x}{c}$, $Z^{(s)}(X) = \frac{z^{(s)}(X)}{c}$ and $Z^{(t)}(X) = \frac{z^{(t)}(X)}{c}$. Given the mean camber line and the thickness distribution, the upper surface of this airfoil can be written as

$$\frac{z_u}{c} = Z^{(s)}(X) + \frac{1}{2}Z^{(t)}(X), \tag{1}$$

and the lower surface of this airfoil can be written as

$$\frac{z_l}{c} = Z^{(s)}(X) - \frac{1}{2}Z^{(t)}(X). \tag{2}$$

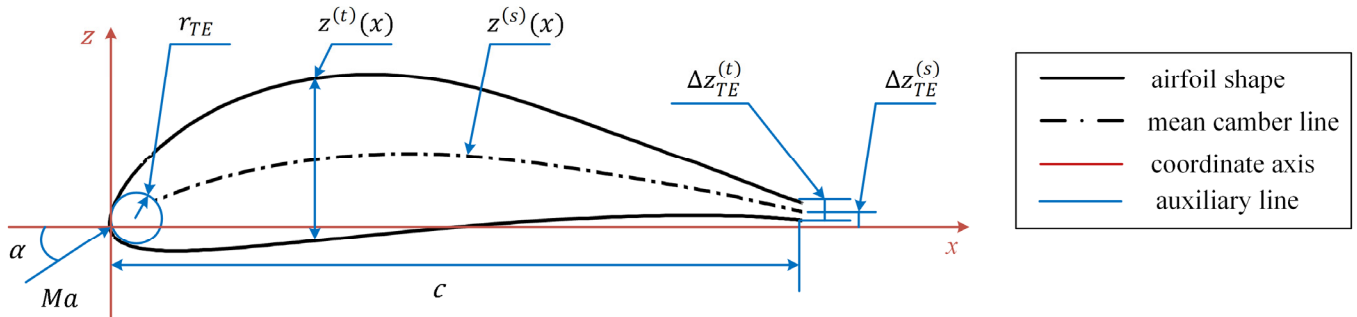


Figure 1. Airfoil geometry.

The following sections will complete the defining equations for the mean camber line $Z^{(s)}(X)$ and the thickness distribution $Z^{(t)}(X)$.

2.2. Representation of Mean Camber Line

According to the thin-airfoil theory, the mean camber line of an airfoil is related to its aerodynamic performance parameters, including the lift coefficient and the pitching moment. The idea of the PAERO method [26] is to use these aerodynamic performance parameters as design variables. The mean camber line, $Z^{(s)}(X)$, can be expressed as follows:

$$Z^{(s)}(X) = (\alpha - A_0)X - \frac{1}{\pi} \int_0^1 \left[\sum_{n=1}^N A_n \sin(n \times \arccos(2X' - 1)) \right] \ln \left| \frac{X - X'}{X} \right| dX', \tag{3}$$

where the first three coefficients, namely A_0 , A_1 and A_2 , can be calculated as follows:

$$A_0 = 3 \left[\alpha - \frac{\Delta z_{TE}^{(s)}}{c} - \frac{1}{3\pi} C_L - \frac{4}{3\pi} C_M + \sum_{i=2}^{N/2} \frac{A_{2i}}{(2i-1)(2i+1)} \right], \tag{4}$$

$$A_1 = \frac{1}{\pi} C_L - 2A_0, \tag{5}$$

$$A_2 = -\frac{1}{\pi} C_L - \frac{4}{\pi} C_M + 2A_0, \tag{6}$$

where the angle of attack, α , the lift coefficient, C_L , and the pitching moment coefficient around the quarter-chord point, C_M , are design variables. In addition, $\Delta z_{TE}^{(s)}$ is the height of the camber line on the trailing edge, and c is the chord length.

When the freestream Mach number, denoted as Ma , is less than one, the Prandtl Glauert rule can be used to take the compressibility effect into account, and the effective aerodynamic coefficients are written as follows:

$$C_L = C_{L, Ma} \times \sqrt{1 - Ma^2}, \tag{7}$$

$$C_M = C_{M, Ma} \times \sqrt{1 - Ma^2}. \tag{8}$$

Finally, the design variables for the mean camber line can be expressed as follows:

$$DV^{(s)} = \left(Ma_{DV}, \alpha_{DV}, C_{L,DV}, C_{M,DV}, A_3, \dots, A_N, \Delta z_{TE}^{(s)} \right), \tag{9}$$

where the subscripts “DV” are added to avoid confusion with the actual aerodynamic performance parameters. The design Mach number, Ma_{DV} , can be given according to the design requirements. Because the design lift coefficient, $C_{L,DV}$, depends on both the angle of attack, α_{DV} , and the mean camber line, either $C_{L,DV}$ or α_{DV} can be fixed in the optimization design.

2.3. Representation of Airfoil Thickness Distribution

2.3.1. CST Method

The CST method is a widely used parameterization method. It combines a class function and a shape function. The surface coordinates of an airfoil with a trailing-edge thickness of $\Delta z_{TE}^{(t)}$ can be expressed as

$$Z^{(t)}(X) = 2 \times \left[\left(X^{0.5} \times (1 - X) \right) \times S(X) + \frac{1}{2} \times X \times \frac{\Delta z_{TE}^{(t)}}{c} \right], \tag{10}$$

where $S(X)$ is the shape function defined as the linear combination of Bernstein polynomials, written as follows:

$$S(X) = \sum_{i=0}^n a_i B_{i,n}(X), \tag{11}$$

$$B_{i,n}(X) = \frac{n!}{i!(n-i)!} \times X^i \times (1 - X)^{n-i}, \tag{12}$$

where $B_{i,n}(X)$ is a Bernstein polynomial of degree n .

2.3.2. SVD Method

The idea of SVD is to extract the orthogonal basic modes through a proper orthogonal decomposition of the chosen dataset. A matrix is first assembled by using the Z coordinates of the airfoil thickness distribution, while the corresponding X coordinates are kept the same for all airfoils. The matrix, H , is written as follows:

$$H = \begin{bmatrix} Z_{11}^{(t)} & \dots & Z_{m1}^{(t)} \\ \vdots & \ddots & \vdots \\ Z_{1n}^{(t)} & \dots & Z_{m1}^{(t)} \end{bmatrix}, \tag{13}$$

where n is the number of coordinate points and m is the number of samples. Then, this matrix is further decomposed by singular value decomposition as follows:

$$H = U \Sigma V^T, \tag{14}$$

where the columns of U , namely $(U_1, U_2, \dots, U_{\min(m,n)})$, represent the orthogonal thickness modes. A new thickness distribution can be expressed as a linear combination of these modes, written as follows:

$$Z^{(t)}(X) = \sum_{i=1}^k a_i U_i(X) + X \times \frac{\Delta z_{TE}^{(t)}}{c}. \tag{15}$$

Figure 2 shows the first ten thickness modes extracted from the UIUC dataset. Note that the number of samples, m , is equal to 1433 and the number of coordinate points, n , is equal to 121. The coordinate distribution is expressed as follows:

$$X_i = \frac{1}{2} \left(\cos \frac{2\pi(i + 119)}{240} + 1 \right), i = 1, 2, \dots, 121. \tag{16}$$

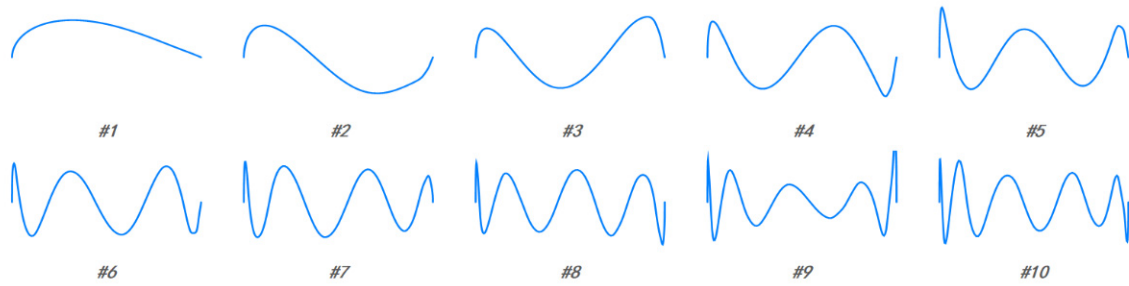


Figure 2. The first ten airfoil thickness modes extracted from the UIUC database (# indicates No.).

2.3.3. Deep Learning Method

The Generative Adversarial Network (GAN) [28] is a widely used model that can learn the main features of data and generate a dataset similar to the original data. The architecture of the standard GAN is shown in Figure 3a, where S is the training dataset.

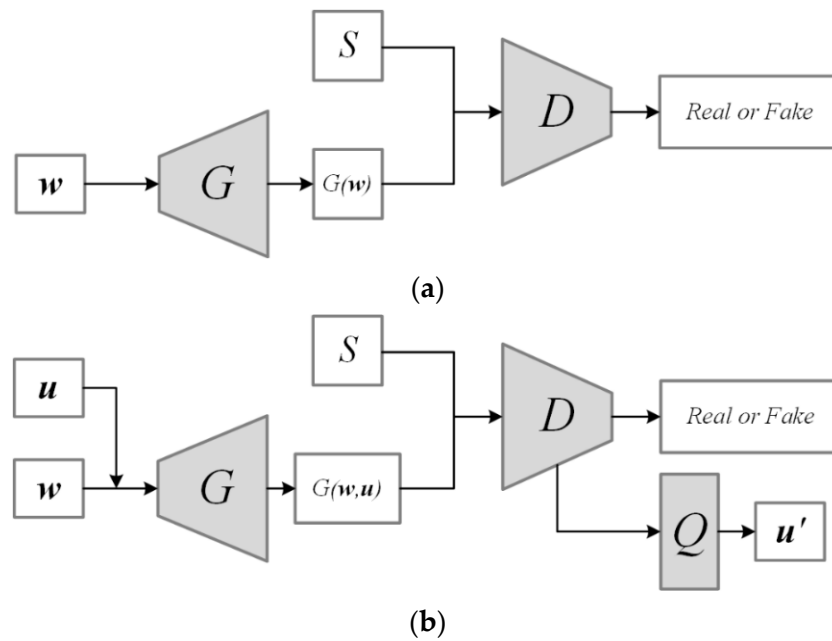


Figure 3. Architectures of (a) GAN and (b) InfoGAN.

It is composed of two players, namely the generator, G , and the discriminator, D . G obtains the random noise, w , from the random distribution, P_w , and generates the data, $G(w)$. D judges whether the input data come from the real distribution, P_{data} , or are generated by G . G constantly adjusts parameters to cheat D , making it unable to determine the source. The two players play games with each other and finally reach the Nash equilibrium, so that the generative distribution, P_G , of G is the same as the distribution of the real data, P_{data} . Generally, the training loss function of GAN can be expressed as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{s \sim P_{data}} [\log D(s)] + \mathbb{E}_{w \sim P_w} [\log(1 - D(G(w)))] \tag{17}$$

In fact, GAN is very easy to collapse during training, with the result that the generator, G , can only generate a small part of the training data to completely cheat the discriminator, D . Therefore, the WGAN [29] model was proposed to solve this problem by improving the activation function and the training process. The loss function of WGAN can be expressed as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{s \sim P_{data}} [D(s)] - \mathbb{E}_{w \sim P_w} [D(G(w))], \tag{18}$$

$$\| \partial D / \partial (v^D, b^D) \| \leq \gamma, \tag{19}$$

where v^D and b^D are the training parameters of D and γ is the weight gradient clipping boundary used to ensure the stability of model training.

Because the noise variables, w , of the standard GAN are uninterpretable, the InfoGAN [30] model was proposed to add a number of latent variables that are interpretable. The architecture of the InfoGAN model is shown in Figure 3b. Like GAN, this model also has a G and a D that play games with each other and ultimately enable the G to synthesize data similar to the training data. Unlike GAN, the input variables of G include the latent variables, u . InfoGAN adds the auxiliary distribution, Q , to restore the u inputted to the G . Both Q and D can share the convolution layers. InfoGAN maximizes the mutual information, $I(u; G(w, u))$, of the latent variables, u , and the generated samples, $G(w, u)$. By training, the main features of the data can be extracted into u .

In practice, it is difficult to calculate the mutual information directly, so the lower bound, L_1 , of the mutual information is generally calculated as follows:

$$L_1(G, D) = \mathbb{E}_{s \sim G(w, u)} \left[\mathbb{E}_{u' \sim P(u|s)} [\log Q(u'|s)] \right] + H(u) \leq I(u; G(w, u)), \tag{20}$$

where $H(u)$ is the entropy of u . If the distribution of u is fixed, $H(u)$ becomes a constant. Combined with the mutual information, the loss function for the InfoGAN training can be expressed as follows:

$$\min_{G, Q} \max_D V_I(D, G, Q) = \mathbb{E}_{s \sim P_{data}} [\log D(s)] + \mathbb{E}_{u \sim P_u, w \sim P_w} [\log(1 - D(G(w, u)))] - \lambda L_1(G, Q), \tag{21}$$

where λ is introduced to balance the loss function and P_u is the prior probability distribution of the latent variables.

The dataset S used for training the generative model is from the UIUC airfoil database. The thickness distribution of an airfoil can be regarded as showing a symmetric airfoil. The generator, G , obtains the latent variables, u , and the noise variables, w , from the random distribution and then synthesizes the airfoils. The discriminator, D , determines whether the airfoil is correct. Q attempts to restore the latent variables, u' , by extracting the features. Eventually, G can synthesize a reasonable symmetric airfoil through the latent variables and the noise variables, and almost all geometric information is extracted into the latent variables. Therefore, the latent variables, u , of InfoGAN can be used for the airfoil thickness representation.

For a symmetric airfoil with a trailing-edge thickness of $\Delta z_{TE}^{(t)}$, its coordinates, $Z^{(t)}(X)$, can be expressed as follows:

$$Z^{(t)}(X) = G(X; u) + X \times \frac{\Delta z_{TE}^{(t)}}{c}. \tag{22}$$

Finally, the design variables for the airfoil thickness distribution can be expressed as follows:

$$DV^{(t)} = (u, \Delta z_{TE}^{(t)}) = (u_0, u_1, u_2, \dots, u_N, \Delta z_{TE}^{(t)}). \tag{23}$$

2.4. Summary

The schematic diagram of the proposed airfoil parameterization is shown in Figure 4. The PAERO method is used to represent the mean camber line, and the generative model is used to represent the airfoil thickness distribution. The parameters used for the camber representation include two types, namely the aerodynamic performance parameters and the optional parameters. The parameters used for the thickness representation are the latent variables of the generator, G .

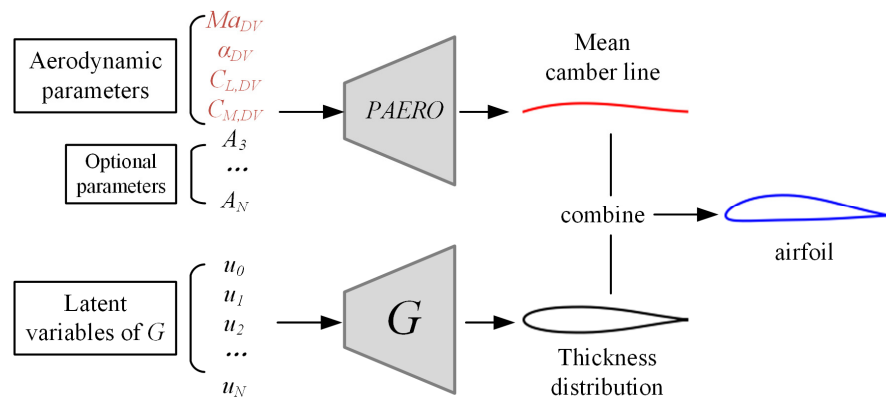


Figure 4. The schematic diagram of the proposed parameterization method.

For an airfoil with a trailing-edge height of $\Delta z_{TE}^{(s)}$ and a trailing-edge thickness of $\Delta z_{TE}^{(t)}$, the design variables can be expressed as follows:

$$DV = (DV^{(s)}, DV^{(t)}) = (Ma_{DV}, \alpha_{DV}, C_{L,DV}, C_{M,DV}, A_3, \dots, A_N, \Delta z_{TE}^{(s)}, u_0, u_1, u_2, \dots, u_N, \Delta z_{TE}^{(t)}). \tag{24}$$

Note that when the method is used to generate airfoil families, four aerodynamic performance parameters need to be specified; when it is used for aerodynamic shape optimization, only two aerodynamic performance parameters are needed, due to the fact that the lift coefficient is dependent on the angle of attack. The optional parameters for the mean camber line can be selected according to the design requirements. The number of latent variables needed to generate the airfoil thickness distribution will be discussed later.

In summary, the coordinates of the upper surface, z_u , and the coordinates of the lower surface, z_l , for an airfoil can be expressed as follows:

$$\frac{z_u}{c} = Z^{(s)}(X; DV^{(s)}) + \frac{1}{2}Z^{(t)}(X; DV^{(t)}), \tag{25}$$

$$\frac{z_l}{c} = Z^{(s)}(X; DV^{(s)}) - \frac{1}{2}Z^{(t)}(X; DV^{(t)}). \tag{26}$$

It should be pointed out that the chosen formulation does not exactly follow the theory of camber and thickness, as it neglects the X component of the normal to the camber line when adding the thickness. We adopt this common simplification to avoid negative X coordinates at the leading edge.

3. Results and Discussion

3.1. Model Training and Airfoil Thickness Synthesis

3.1.1. Data Preparation

The UIUC airfoil database includes most airfoils used in a wide range of scenarios, including aircraft, gas turbines, wind turbines, etc. Many studies [15,17,19,23] have adopted this database as the training basis for a neural network model. However, some of the airfoil data in the database are incomplete or impractical. Therefore, the airfoil database is filtered and normalized by using the method suggested by Li et al. [31]. The process is divided into the following four steps:

- (1) Remove the airfoils with incomplete data, and order the airfoil coordinates in a counterclockwise direction starting from the trailing edge, as shown in Figure 5.
- (2) Convert a blunt trailing edge into a sharp trailing edge, adjust the angle of attack to 0 degree, and synchronize the x coordinates.
- (3) Remove the airfoils with more than one local maximum thickness.

- (4) Remove the airfoil mean cambers.

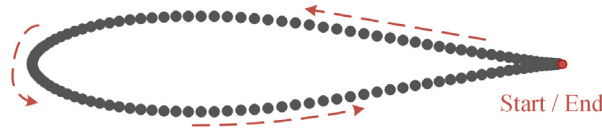


Figure 5. Demonstration of airfoil data preprocessing.

Finally, 1433 symmetrical airfoils were selected from the UIUC database with a total of 1550 existing airfoils. Typical samples obtained from the database are shown in Figure 6. The database only stores the normalized Z coordinates of the airfoils, and the corresponding normalized X coordinate distribution is as follows:

$$X_i = \frac{1}{2} \left(\cos \frac{2\pi(i-1)}{240} + 1 \right), i = 1, 2, \dots, 241. \tag{27}$$

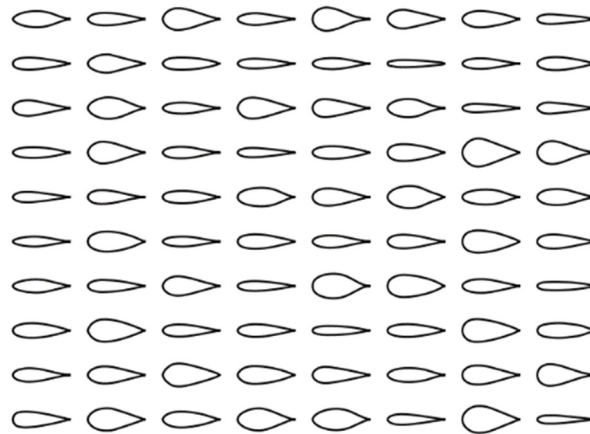


Figure 6. Dataset for symmetrical airfoils.

3.1.2. Model Training

A symmetrical airfoil can be defined by a number of key geometric features, such as the leading edge radius, the airfoil thickness, and the location of the maximum thickness. An InfoGAN model is established to extract these interpretable features as design variables for airfoil parameterization. The schematic diagram of the InfoGAN model is shown in Figure 7.

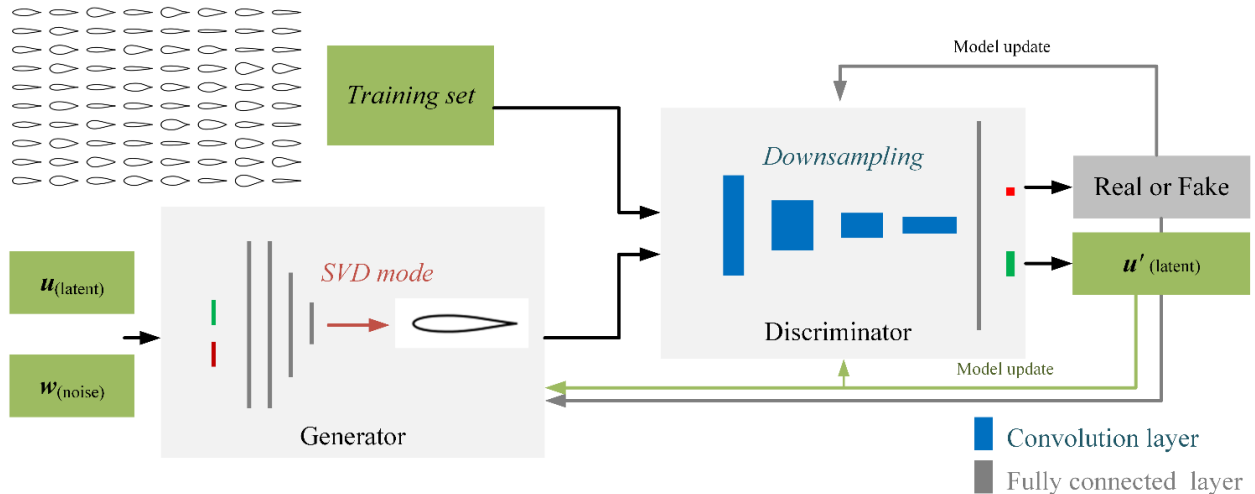


Figure 7. The neural network architecture for training the generative model.

The training set contains a total of 1433 symmetrical airfoils, each of which is represented as a vector in the format of 241×1 . The latent variables, \mathbf{u} , and the noise variables, \mathbf{w} , are used as the input of the generator, G , where \mathbf{u} follows a uniform distribution in $[0, 1]$, and \mathbf{w} follows a normal distribution with a mean of 0 and a variance of 1. G includes the multi-layer full connected layers and a mode layer. The modes come from the SVD decomposition of symmetric airfoils. The research by Li et al. [14] showed that the first 20 modes were already sufficient for fitting the airfoils. In addition, using the modes as the last layer of G is helpful for ensuring a smooth surface and a symmetrical geometry. The discriminator, D , is composed of a number of convolution layers and a full connected layer. The convolution layers extract the features of the airfoil, and the full connected layer integrates them for judgment. An independent full connected layer, Q , is set behind the convolution layers of D to extract the features of the generated airfoil and tries to restore the input information, \mathbf{u} .

The parameters for the neural network were chosen by experience [32]. The final network parameters of the generator and the discriminator are shown in Table 1, where FC layer means a full connected layer and C layer means a convolution layer. The models were implemented in the open-source software PyTorch [33], and the final model parameters were determined by trial and error [32]. The parameters related to D 's convolution kernel are set as follows: kernel size = 7, step size = 2 and padding = 3. The training parameters are set as follows: batch size = 64, learning rate = 0.0001 and training round epochs = 500.

Table 1. Neural network parameters for the generator and the discriminator.

Generator			Discriminator		
Layer	Neurons	AF	Layer	Filters	AF
FC layer	128	ReLU	C layer	10	LeakyReLU
FC layer	128	ReLU	C layer	20	LeakyReLU
FC layer	64	ReLU	C layer	40	LeakyReLU
FC layer	20	ReLU	C layer	80	LeakyReLU
SVD mode layer			FC layer	1	
			FC layer(Q)	3	Sigmoid

3.1.3. Symmetrical Airfoil Synthesis

The input variables of symmetrical airfoil generator include the latent variables, \mathbf{u} , and the noise variables, \mathbf{w} . Here, we chose $d_{latent} = 3$ and $d_{noise} = 10$ as an example to study the generated samples, as shown in Figure 8.

Figure 8a shows the results of airfoil synthesis by changing each latent variable, while the noise variables, \mathbf{w} , remain 0. It can be seen that each component of \mathbf{u} is related to one of the geometric features. u_1 is related to the leading edge radius, u_2 is related to the airfoil thickness, and u_3 is related to the location of the maximum thickness. Figure 8b shows the results of airfoil synthesis by changing the first three components of \mathbf{w} , while the latent variables, \mathbf{u} , remain 0.5. It can be seen that there is no significant difference between the generated airfoils, indicating that the features of symmetric airfoils have been mainly extracted into the latent variables, \mathbf{u} .

Figure 9 shows the symmetric airfoils randomly generated by the InfoGAN method and the traditional parameterization methods. The InfoGAN method adopts a generator with six latent variables. The CST method adopts a 5th-order polynomial, so that there are six design variables. The SVD method adopts the first six modes. The variable range of the generative model is $[0, 1]$, and the variable range of CST and SVD comes from the fitting results of the dataset. It can be seen that all of the symmetric airfoils generated by the InfoGAN generator maintain the features of the existing airfoils, and a number of the symmetric airfoils generated by the CST and SVD methods are actually physically unreasonable.

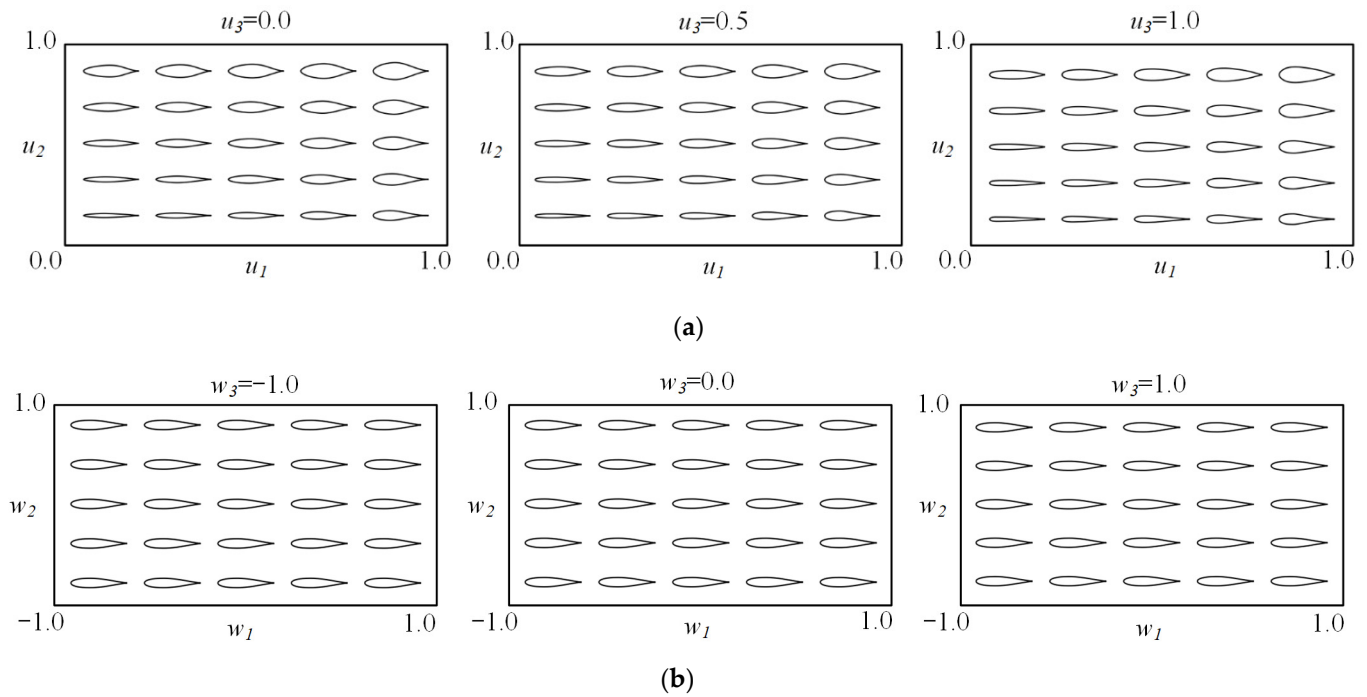


Figure 8. Symmetric airfoils generated by the trained model ($d_{latent} = 3$ and $d_{noise} = 10$): (a) design space of latent variables; (b) design space of noise variables (first three dimensions).

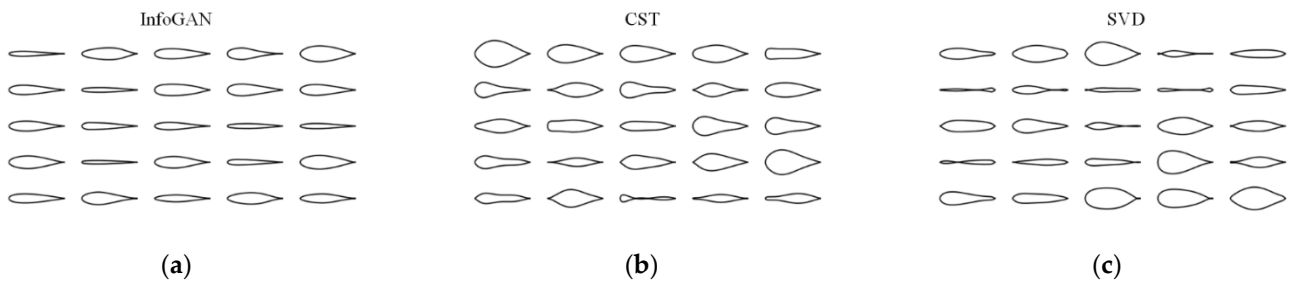


Figure 9. Symmetric airfoils generated by different parameterization methods: (a) InfoGAN; (b) CST method; (c) SVD method.

In order to test the accuracy of the trained model, the existing airfoils in the database have been fitted. Fitting an existing airfoil can be turned into an unconstrained optimization problem. The latent variables of the generator are taken as the design variables, and Powell's BOBYQA [6] algorithm is used to minimize the root mean square error (RMSE), which is expressed as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Z_{i,G}(X) - Z_{i,data}(X))^2}, \quad (28)$$

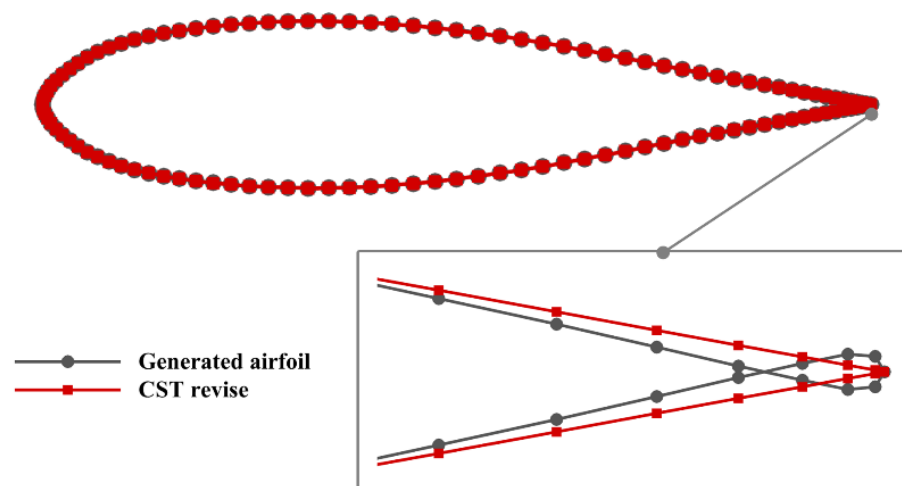
where n , equal to 241, is the total number of the coordinate points, and $Z_{i,G}(x)$ and $Z_{i,data}(x)$ are the normalized Z coordinates of the generated airfoil and the airfoil to be fitted, respectively.

While keeping the noise variables, w , at $\mathbf{0}$, the optimization algorithm searches the design space spanned by the latent variables, u , to minimize the RMSE. The average fitting errors of all the symmetric airfoils in the database are shown in Table 2. It can be seen that when the number of latent variables is greater than 8, the RMSE is less than 5×10^{-4} , which is the tolerance for wind tunnel geometries [34].

Table 2. Errors of fitting symmetric airfoil database.

d_{latent}	RMSE ($\times 10^{-4}$)
2	33.11
3	12.01
4	7.253
6	6.234
8	5.067
10	3.367

Note that in some cases, the upper surface and the lower surface will cross each other near the trailing edge, as shown in Figure 10. This phenomenon was also observed by Chen et al. [15], who suggested adding a regularization term to the loss function to avoid this risk. We find that this problem can be easily solved by fitting the airfoil to a high-order CST polynomial, as shown in Figure 10.

**Figure 10.** CST revision for airfoil surface intersection near the trailing edge.

3.2. Verification of Aerodynamic Performance Parameters

This section will test the accuracy of aerodynamic performance parameters. CFL3D [35], an open-source CFD solver based on multi-block structured grids, was used for high-fidelity aerodynamic performance calculations. The CFD solver was first validated by the wind tunnel test of the airfoil E387 [36]. The flow conditions are $Ma = 0.1$, $\alpha = 0^\circ$ and $Re = 0.2 \times 10^6$. Note that the wind tunnel test data were measured by allowing a free transition. The Spalart–Allmaras one-equation turbulence model coupled with an e^N empirical method for the transition prediction was adopted. Three types of grids were used to calculate the aerodynamic performances. The farfield is 15 times the chord length, and the height of the first layer is set to 1×10^{-5} to ensure $y^+ < 1.0$. As shown in Table 3, the results of grid No. 2 are close to those of the densest grid, No. 3. Considering the computational cost, grid No. 2, shown in Figure 11, was adopted in the simulations. The comparison of the numerical simulation and the wind tunnel test is shown in Figure 12, indicating a satisfying accuracy.

Table 3. Grid sensitivity study for the E387 airfoil ($Ma = 0.1$, $\alpha = 0^\circ$ and $Re = 0.2 \times 10^6$).

No.	Grid	C_L	C_D	C_M
1	143×49	0.3773	0.01012	−0.07740
2	257×97	0.3817	0.009681	−0.07792
3	451×97	0.3876	0.009594	−0.07860

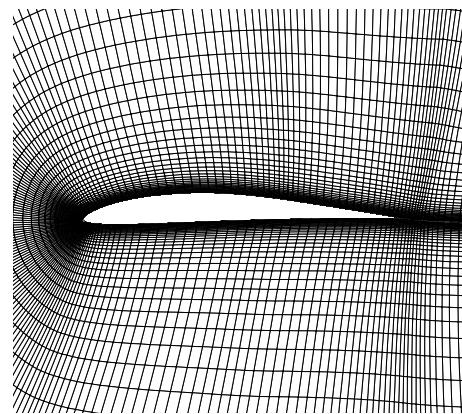


Figure 11. Grid No. 2 for E387 airfoil.

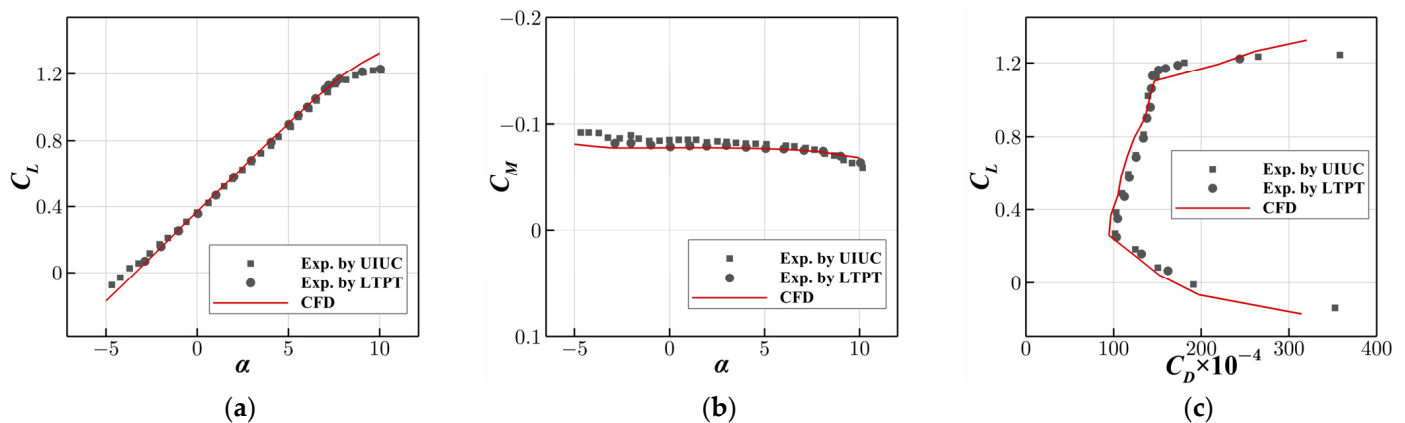


Figure 12. The comparison of the numerical simulation and the wind tunnel test for the E387 airfoil: (a) lift coefficient; (b) drag coefficient; (c) drag polar.

The airfoil performance parameters at a low Mach number were first tested. The aerodynamic performance parameters were set as follows: $Ma_{DV} = 0.2$, $\alpha_{DV} = 2.0$, $C_{L,DV} = 0.40$ and $C_{M,DV} = -0.05$. The optional parameters are not considered here. The generator, G , synthesized the thickness distributions based on the latent variables, \mathbf{u} , of a random distribution. The dimension of \mathbf{u} is three, and the value range is $[0, 1]$. High-fidelity CFD simulations were used to verify the accuracy of the aerodynamic performance parameters. The simulation conditions were $Ma = 0.2$, $\alpha = 2.0$ and $Re = 4 \times 10^6$. The simulations were performed assuming a fully turbulent flow. A total of 50 airfoils were randomly generated, and the test results are shown in Figure 13. The given aerodynamic performance parameters are represented by a star, and the airfoils are displayed on their corresponding aerodynamic performance coordinates. It can be seen that due to the viscous effect, $C_L < C_{L,DV}$ and $C_M > C_{M,DV}$. However, the aerodynamic performances of most airfoils are close to the given parameters.

Then, the airfoil performance parameters at a high Mach number were tested. The aerodynamic performance parameters were set as follows: $Ma_{DV} = 0.6$, $\alpha_{DV} = 2.0$, $C_{L,DV} = 0.50$ and $C_{M,DV} = -0.07$. In addition, the optional parameters, A_3 and A_4 , were considered as well. Following the work of [26], the recommended range of A_3 and A_4 is $[-0.02, 0.02]$. The flow simulation conditions were $Ma = 0.6$, $\alpha = 2.0$ and $Re = 6.5 \times 10^6$. The simulation results of 50 randomly generated airfoils are shown in Figure 14. It can be seen that for most of the generated airfoils, $C_L < C_{L,DV}$, $C_M > C_{M,DV}$, and their aerodynamic performances are still close to the given parameters. In summary, it can be concluded that the aerodynamic performance parameters can significantly determine the aerodynamic performances of the airfoil family even at a high Mach number. In contrast, the optional parameter A_i does not significantly affect the aerodynamic performances.

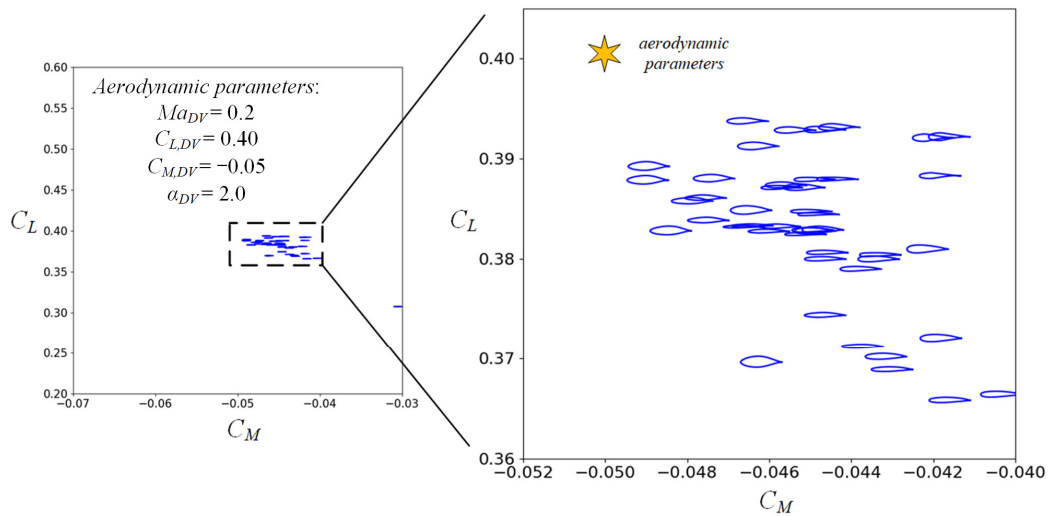


Figure 13. Test results for the airfoil family with given aerodynamic performance parameters ($Ma_{DV} = 0.2$).

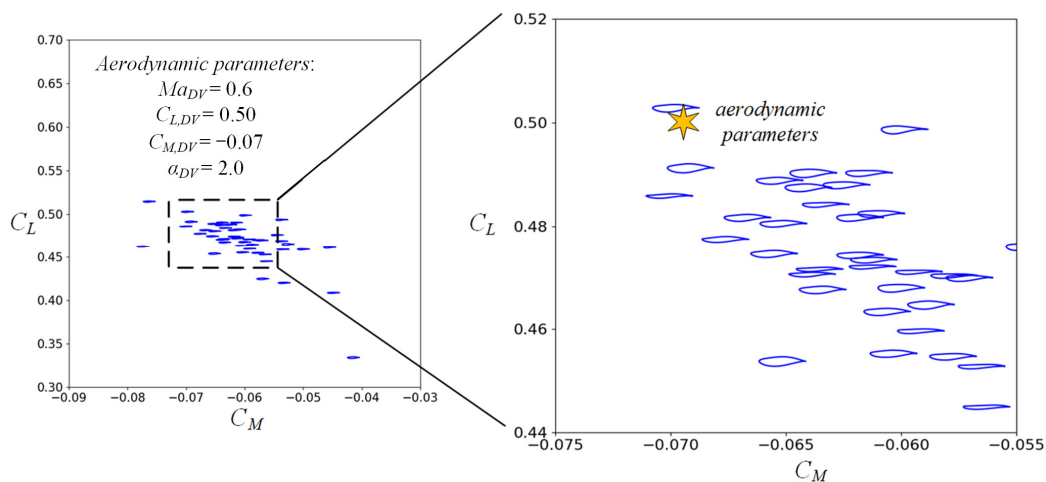


Figure 14. Test results for the airfoil family with given aerodynamic performance parameters ($Ma_{DV} = 0.6$).

3.3. Airfoil Shape Optimization

In this section, the capabilities of the proposed parameterization method will be tested in the context of airfoil shape optimization. Three parameterization methods are adopted for comparison, namely the combination of PAERO and InfoGAN, the combination of PAERO and CST, and CST for the entire airfoil surface. Note that the PAERO method is only adopted for the camber representation, the InfoGAN generative model is only adopted for the thickness representation, and the CST method is a general parameterization method, which can be used for representing either the airfoil thickness distribution or the airfoil surface.

The problem of airfoil drag minimization is considered here. The objective function, the design variables and the constraints are described in Table 4. The design conditions are $Ma = 0.6$, $C_L = 0.5$ and $Re = 6.5 \times 10^6$. The chosen optimization algorithm is Powell's BOBYQA [6], a gradient-free algorithm based on quadratic approximation. The convergence is thought to be reached when the change in the objective function is less than a tolerance of 1×10^{-5} . The maximum number of iterations is set to 500. The constraints are as follows: the airfoil area is not less than 0.07, and the pitching moment coefficient around the quarter-chord point is not less than -0.07 . The aerodynamic constraints and

the geometric constraints are added to the objective function as the penalty functions, so the objective function is expressed as follows:

$$\text{Obj} = \lambda_0 C_D + \lambda_1 \max\left(1 + \frac{0.07}{C_M}, 0\right) + \lambda_2 \max\left(1 - \frac{\text{area}}{0.07}, 0\right), \lambda_0 = 100, \lambda_{1,2} = 10, \quad (29)$$

where λ_0 , λ_1 and λ_2 are the Lagrange multipliers.

Table 4. Statement of airfoil optimization for drag minimization.

	Functions	Description	Quantity
Minimize	C_D	Drag coefficient	1
With respect to	DV	Design variables	12
	α	Angle of attack	1
Subject to	$C_L = 0.5$	Lift coefficient	1
	$C_M \geq -0.07$	Quarter-chord pitching moment	1
	$\text{area} \geq 0.07$	Area constraint	1

The value range of design variables is shown in Table 5. Ma_{DV} and $C_{L,DV}$ are fixed, and α_{DV} and $C_{M,DV}$ are allowed to be changed. The optional parameters, namely A_3 , A_4 , A_5 and A_6 , are added as well. The thickness representation maintains the same degree of freedom as the camber representation. The thickness distribution is generated by either the generative model controlled by six latent variables or the CST method with a 5th-order polynomial. The value range for the CST method is derived from the fitting results of the UIUC symmetric airfoil database, and all design variables are assumed to be positive. For representing the entire airfoil surface, the CST method adopts two 5th-order polynomials for the upper surface and the lower surface, respectively, and the value ranges come from the fitting results of the UIUC airfoil database as well. The value range for the upper surface is [0.01, 0.4], and the value range for the lower surface is [−0.4, −0.01].

Table 5. Design variables and their value ranges.

Variable Type	Symbol	Value Range
Camber representation	Ma_{DV}	0.6
	$C_{L,DV}$	0.5
	α_{DV}	[1.8, 2.2]
	$C_{M,DV}$	[−0.06, −0.08]
Thickness representation (InfoGAN)	A_3, A_4, A_5, A_6	[−0.02, 0.02]
Thickness representation (InfoGAN)	u_0, u_1, \dots, u_n	[0, 1]
Thickness representation (CST)	a_0, a_1, \dots, a_n	[0.01, 0.4]

The optimization tests were repeated 10 times for each method by using random initial design variables. The convergence histories are shown in Figure 15. The shaded regions in the figure show the convergence histories of different tests, and the solid lines are the average convergence histories. It can be seen from the convergence histories that the method of using the InfoGAN model for thickness representation has the best performance. It converged near the minimum point after about 30 iterations, far fewer than the other two methods. Note that the large initial objective function shown in the figure is due to the fact that the area of the initial shape does not satisfy the design requirements and a large penalty function is imposed. The final designs of airfoils are shown in Figure 16. The optimized results of the InfoGAN model maintain the features of the existing airfoils and are all physically reasonable. On the contrary, the optimized results of the CST method are divergent, indicating a less compact and bumpier design space. Thus, it can be concluded that the proposed method is significantly more efficient and more robust than the traditional CST method.

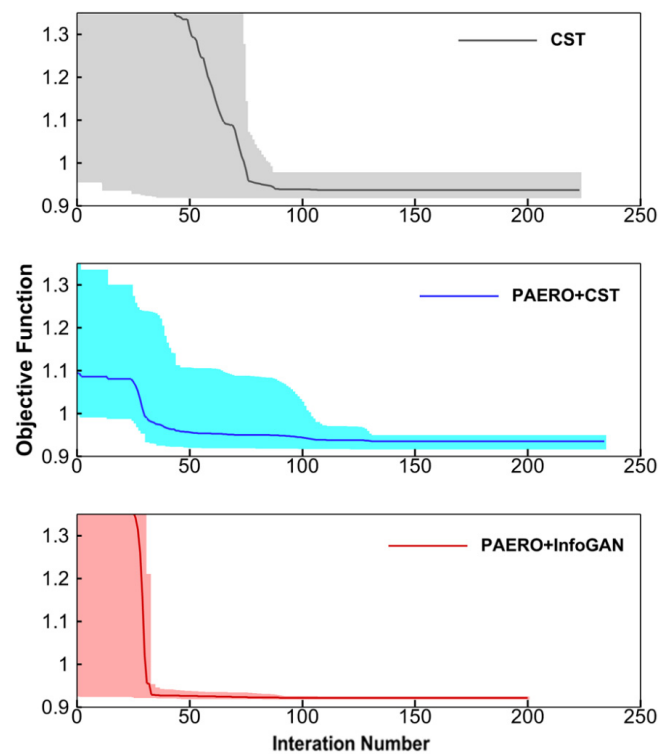


Figure 15. Convergence histories resulting from using different parameterization methods. Note that the shaded regions represent the ranges of convergence histories and the solid lines represent the average values.

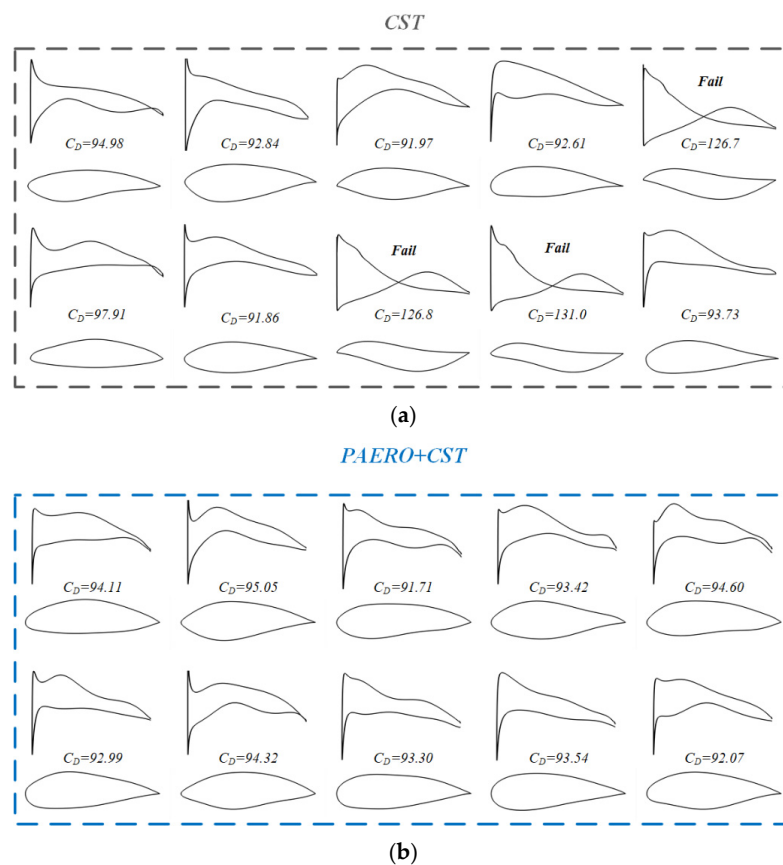


Figure 16. Cont.

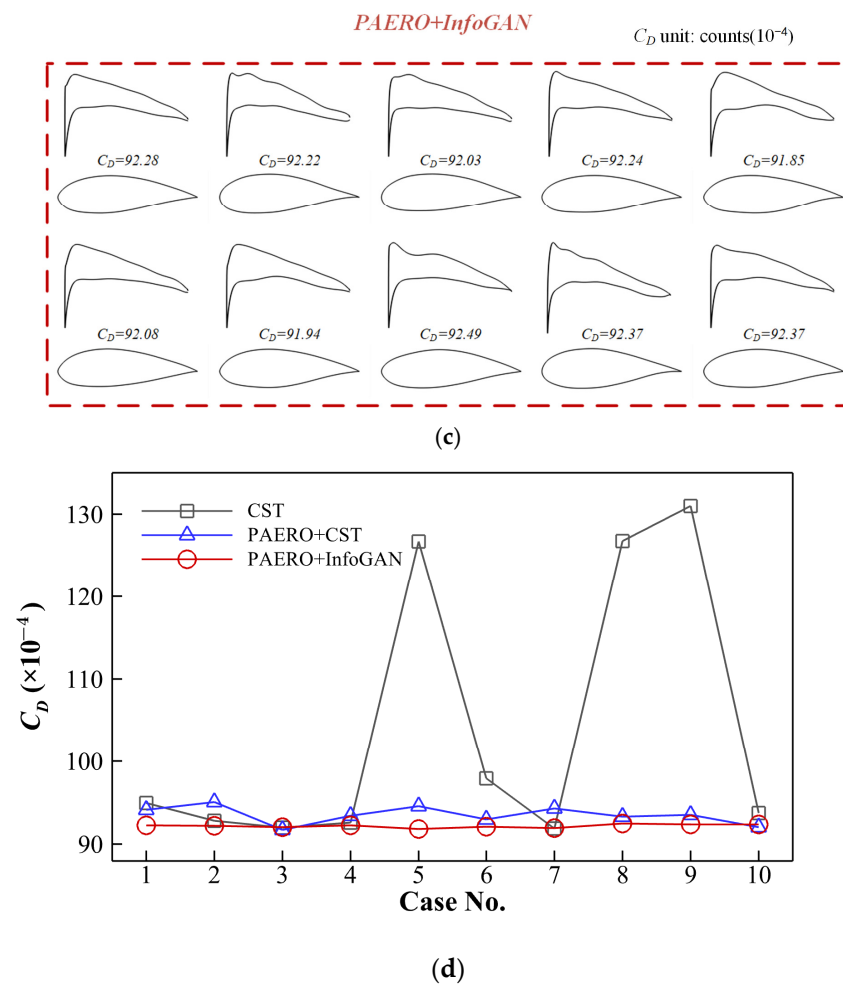


Figure 16. Comparison of the optimized results of using different parameterization methods: (a) CST; (b) PAERO + CST; (c) PAERO + InfoGAN; (d) drag coefficients.

4. Conclusions

In this paper, a compact airfoil shape parameterization is proposed by adopting the recent PAERO method for the camber representation and the data-driven InfoGAN model for the thickness representation. The InfoGAN model shows an excellent ability to learn the thickness distributions of existing airfoils. Its generator can compactly represent the thickness distribution and relate some of the key geometric features to the latent variables. Moreover, it can filter the unrealistic thickness distribution, resulting in a highly compact design space for airfoil design. In addition, it is confirmed that the aerodynamic performance parameters can significantly determine the aerodynamic performances of the airfoil family at subsonic flows and the optional parameters do not significantly affect the aerodynamic performances. In the test of airfoil shape optimization, it is found that the proposed method is significantly more efficient and more robust than the traditional CST method.

Author Contributions: Conceptualization, F.D.; methodology, J.Y.; software, J.Y.; validation, J.Y. and F.D.; formal analysis, J.Y.; investigation, J.Y.; resources, F.D.; data curation, J.Y.; writing—original draft preparation, F.D. and J.Y.; writing—review and editing, F.D.; visualization, J.Y.; supervision, F.D.; project administration, F.D.; funding acquisition, F.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (No. 12032011).

Data Availability Statement: Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

A_i	i th Fourier coefficient
C_D	drag coefficient
C_L	lift coefficient
$C_{L,DV}$	design variable corresponding to lift coefficient
C_M	pitching moment coefficient around the quarter-chord point
$C_{M,DV}$	design variable corresponding to pitching moment coefficient
D	discriminator
DV	design vector
d_{latent}	latent dimension
d_{noise}	noise dimension
G	generator
Ma	freestream Mach number
Ma_{DV}	design variable corresponding to freestream Mach number
P_u	prior distribution of latent variables
P_{data}	data distribution
P_G	generative distribution
P_w	prior distribution of noise variables
Q	auxiliary distribution
Re	Reynolds number
u	latent variable for generative model
w	noise variable for generative model
$Z^{(s)}$	normalized mean camber curve
$Z^{(t)}$	normalized thickness distribution
α	angle of attack
α_{DV}	design variable corresponding to angle of attack
λ_i	i th Lagrange multiplier
$\Delta z_{TE}^{(s)}$	ordinate of the mean camber curve on the trailing edge
$\Delta z_{TE}^{(t)}$	trailing-edge thickness

References

1. Shan, S.; Wang, G.G. Survey of Modeling and Optimization Strategies to Solve High-Dimensional Design Problems with Computationally-Expensive Black-Box Functions. *Struct. Multidiscip. Optim.* **2010**, *41*, 219–241.
2. Forrester, A.; Keane, A.J. Recent Advances in Surrogate-Based Optimization. *Prog. Aerosp. Sci.* **2009**, *45*, 50–79. [[CrossRef](#)]
3. Kenway, G.; Mader, C.A.; He, P.; Martins, J.R.R.A. Effective Adjoint Approaches for Computational Fluid Dynamics. *Prog. Aerosp. Sci.* **2019**, *110*, 100524. [[CrossRef](#)]
4. Jameson, A. Aerodynamic Design via Control Theory. *J. Sci. Comput.* **1988**, *3*, 233–260. [[CrossRef](#)]
5. Han, Z.H.; Xu, C.Z.; Qiao, J.L.; Liu, P. Recent Progress of Efficient Global Aerodynamic Shape Optimization Using Surrogate-Based Approach. *Acta Aeronaut. Astronaut. Sin.* **2020**, *41*, 623344.
6. Powell, M.J.D. *The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives*; Department of Applied Mathematics and Theoretical Physics, University of Cambridge: Cambridge, UK, 2009.
7. Li, J.; Du, X.; Martins, J. Machine Learning in Aerodynamic Shape Optimization. *Prog. Aerosp. Sci.* **2022**, *134*, 100849. [[CrossRef](#)]
8. Sobieczky, H. Parametric Airfoils and Wings. In *Recent Development of Aerodynamic Design Methodologies. Notes on Numerical Fluid Mechanics (NNFM)*; Fujii, K., Dulikravich, G.S., Eds.; Vieweg+Teubner Verlag: Wiesbaden, Germany, 1999; pp. 71–87. ISBN 978-3-322-89954-5.
9. Kulfan, B.M. Universal Parametric Geometry Representation Method. *J. Aircr.* **2008**, *45*, 142–158. [[CrossRef](#)]
10. Rajnarayan, D.; Ning, A.; Mehr, J.A. Universal Airfoil Parametrization Using B-Splines. In *AIAA Paper 2018–3949, Proceedings of the 2018 Applied Aerodynamics Conference, Atlanta, GA, USA, 25–29 June 2018*; American Institute of Aeronautics and Astronautics: Atlanta, GA, USA, 2018.
11. Hicks, R.M.; Henne, P.A. Wing Design by Numerical Optimization. *J. Aircr.* **1978**, *15*, 407–412. [[CrossRef](#)]
12. Sederberg, T.W.; Parry, S.R. Free-Form Deformation of Solid Geometric Models. *ACM SIGGRAPH Comput. Graph.* **1986**, *20*, 151–160. [[CrossRef](#)]

13. Masters, D.A.; Taylor, N.J.; Rendall, T.C.S.; Allen, C.B.; Poole, D.J. Geometric Comparison of Aerofoil Shape Parameterization Methods. *AIAA J.* **2017**, *55*, 1575–1589. [[CrossRef](#)]
14. Li, J.; Bouhlef, M.A.; Martins, J. Data-Based Approach for Fast Airfoil Analysis and Optimization. *AIAA J.* **2019**, *57*, 581–596. [[CrossRef](#)]
15. Chen, W.; Chiu, K.; Fuge, M.D. Airfoil Design Parameterization and Optimization Using Bézier Generative Adversarial Networks. *AIAA J.* **2020**, *58*, 4723–4735. [[CrossRef](#)]
16. Selig, M.S. UIUC Airfoil Coordinates Database. Available online: https://m-selig.ae.illinois.edu/ads/coord_database.html (accessed on 17 July 2023).
17. Du, X.; He, P.; Martins, J. A B-Spline-Based Generative Adversarial Network Model for Fast Interactive Airfoil Aerodynamic Optimization. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6 January 2020.
18. Du, X.; He, P.; Martins, J.R.R.A. Rapid Airfoil Design Optimization via Neural Networks-Based Parameterization and Surrogate Modeling. *Aerosp. Sci. Technol.* **2021**, *113*, 106701. [[CrossRef](#)]
19. Li, J.; Zhang, M.; Martins, J.; Shu, C. Efficient Aerodynamic Shape Optimization with Deep-Learning-Based Geometric Filtering. *AIAA J.* **2020**, *58*, 4243–4259. [[CrossRef](#)]
20. Li, J.; Zhang, M. On Deep-Learning-Based Geometric Filtering in Aerodynamic Shape Optimization. *Aerosp. Sci. Technol.* **2021**, *112*, 106603. [[CrossRef](#)]
21. Achour, G.; Sung, W.J.; Pinon-Fischer, O.J.; Mavris, D.N. Development of a Conditional Generative Adversarial Network for Airfoil Shape Optimization. In *AIAA Paper 2020–2261, Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6 January 2020*; American Institute of Aeronautics and Astronautics: Orlando, FL, USA, 2020.
22. Yilmaz, E.; German, B. Conditional Generative Adversarial Network Framework for Airfoil Inverse Design. In *AIAA Paper 2020–3185, Proceedings of the AIAA Aviation 2020 Forum, Virtual Event, 15–19 June 2020*; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2020.
23. Sekar, V.; Zhang, M.; Shu, C.; Khoo, B.C. Inverse Design of Airfoil Using a Deep Convolutional Neural Network. *AIAA J.* **2019**, *57*, 993–1003. [[CrossRef](#)]
24. Wang, J.; Li, R.; He, C.; Chen, H.; Cheng, R.; Zhai, C.; Zhang, M. An Inverse Design Method for Supercritical Airfoil Based on Conditional Generative Models. *Chin. J. Aeronaut.* **2022**, *35*, 62–74. [[CrossRef](#)]
25. Lei, R.; Bai, J.; Wang, H.; Zhou, B.; Zhang, M. Deep Learning Based Multistage Method for Inverse Design of Supercritical Airfoil. *Aerosp. Sci. Technol.* **2021**, *119*, 107101. [[CrossRef](#)]
26. Abbott, I.H.; Doenhoff, A.E. *Theory of Wing Sections*; Dover Publications Inc.: New York, NY, USA, 1959; p. x + 693.
27. Deng, F.; Xue, C.; Qin, N. Parameterizing Airfoil Shape Using Aerodynamic Performance Parameters. *AIAA J.* **2022**. [[CrossRef](#)]
28. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M. Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
29. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 214–223.
30. Chen, X.; Duan, Y.; Houthoofd, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *Adv. Neural Inf. Process. Syst.* **2016**.
31. Li, J.; He, S.; Martins, J. Data-Driven Constraint Approach to Ensure Low-Speed Performance in Transonic Aerodynamic Shape Optimization. *Aerosp. Sci. Technol.* **2019**, *92*, 536–550. [[CrossRef](#)]
32. Deng, F.; Yi, J. Fast Inverse Design of Transonic Airfoils by Combining Deep Learning and Efficient Global Optimization. *Aerospace* **2023**, *10*, 125. [[CrossRef](#)]
33. PyTorch. Available online: <https://Github.Com/Pytorch/Pytorch> (accessed on 17 July 2023).
34. Kulfan, B.; Bussoletti, J. “Fundamental” Parametric Geometry Representations for Aircraft Component Shapes. In *AIAA Paper 2006–6948, Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA, USA, 6–8 September 2006*; American Institute of Aeronautics and Astronautics: Portsmouth, VA, USA, 2006.
35. NASA CFL3D. Available online: <https://cfl3d.larc.nasa.gov/> (accessed on 17 July 2023).
36. Selig, M.S.; McGranahan, B.D. Wind Tunnel Aerodynamic Tests of Six Airfoils for Use on Small Wind Turbines. *J. Sol. Energy Eng.* **2004**, *126*, 986–1001. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.