Chinese Journal of Aeronautics, (2022), 35(3): 62–74

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

An inverse design method for supercritical airfoil based on conditional generative models

Jing WANG [a], Runze LI [b], Cheng HE [c], Haixin CHEN [b], Ran CHENG [c], Chen ZHAI [a], Miao ZHANG [a,*]

[a] *Shanghai Aircraft Design and Research Institute, Shanghai 200436, China*
[b] *School of Aerospace Engineering, Tsinghua University, Beijing 100084, China*
[c] *Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China*

Received 10 October 2020; revised 13 December 2020; accepted 28 December 2020
Available online 20 March 2021

KEYWORDS

Conditional Variational AutoEncoder (CVAE);
Deep learning;
Generative Adversarial Networks (GAN);
Generative models;
Inverse design;
Supercritical airfoil

Abstract Inverse design has long been an efficient and powerful design tool in the aircraft industry. In this paper, a novel inverse design method for supercritical airfoils is proposed based on generative models in deep learning. A Conditional Variational AutoEncoder (CVAE) and an integrated generative network CVAE-GAN that combines the CVAE with the Wasserstein Generative Adversarial Networks (WGAN), are conducted as generative models. They are used to generate target wall Mach distributions for the inverse design that matches specified features, such as locations of suction peak, shock and aft loading. Qualitative and quantitative results show that both adopted generative models can generate diverse and realistic wall Mach number distributions satisfying the given features. The CVAE-GAN model outperforms the CVAE model and achieves better reconstruction accuracies for all the samples in the dataset. Furthermore, a deep neural network for nonlinear mapping is adopted to obtain the airfoil shape corresponding to the target wall Mach number distribution. The performances of the designed deep neural network are fully demonstrated and a smoothness measurement is proposed to quantify small oscillations in the airfoil surface, proving the authenticity and accuracy of the generated airfoil shapes.

1. Introduction

In the field of aerodynamics, the inverse design of airfoils forms a class of efficient and powerful design tools in the aircraft industry[1] and has been a challenging task and subject of investigation for a long time, dating back to the beginning of airfoil research. Inverse design takes designers' experience and all available prior information into account, searching

* Corresponding author.
E-mail address: zhangm-168@163.com (M. ZHANG).

Peer review under responsibility of Editorial Committee of CJA.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.cja.2021.03.006
1000-9361

for a configuration that can realize the target velocity or pressure distribution on an airfoil. Moreover, inverse design is attractive from the theoretical point of view, as it allows the realization of specified targets, such as shock-free transonic flows, which cannot be realized by traditional trail-and-error.

Inverse design usually proceeds in two steps. First, target distributions are generated to reflect the design goals. Traditionally, the target distributions are specified by an experienced aerodynamicist. Beyond that, optimization methods, such as genetic algorithms, are applied to optimize the parameterized pressure distributions to obtain the target distributions,[2] yet the feasibility of the optimized distributions remains unguaranteed. Once the target distributions are specified, the next step in the design process is to find the configuration satisfying the target distribution. So far, many proven methods have been developed and widely used, such as iterative residual-correction methods,[3] error minimization methods based on adjoint methods,[4,5] evolutionary optimization strategies,[6] and proper orthogonal decomposition.[7] Recently, deep learning techniques have been applied due to significant developments in the field of machine learning and some encouraging results have been achieved. For instance, Kharal[8] and Sun et al.[9] used an artificial neural network along with airfoil parameterization techniques to directly provide an airfoil/wing that fits the required aerodynamic features, in which the model was trained with a relatively smaller airfoil database. Recently, Sekar et al. applied deep convolutional neural networks to obtain the airfoil shape from pressure distribution coefficients.[10]

Although much progress has been made for the inverse design, under a specific design requirement, even an experienced designer or an excellent optimizer can hardly set a proper pressure distribution. Moreover, the target pressure distribution applied by the designer or optimizer may be infeasible. Most commonly used methods for infeasible distributions tend to find the geometry that matches as close as possible to the target distribution, and thus a slight local change in the geometry can have a significant impact on the pressure residual, and thus an enormous number of evaluations to obtain the geometry are inevitable. Consequently, generating a physically-existing distribution that corresponds to a unique and continuous airfoil is still hard to realize and proves to be an excessive demand for airfoil design. Recently, emerging and developing generative models based on deep learning techniques offer a viable and promising solution for generating feasible target distributions, due to their ability to model a data distribution and sample required data from the learned distribution. In addition, conditional generative models[11,12] can impose certain conditions on generative models, which introduce the possibility of altering generated data by conditioning the model on additional information.

One of the most popular generative methods is the so-called Generative Adversarial Networks (GANs) invented by Goodfellow et al.[13] Here, a generator competes against a discriminative model. Specifically, the generator tries to generate synthetic data by sampling a noise space, whereas the discriminator tries to distinguish data as synthetic or real. Due to its high efficiency in generaing realistic samples using only a small set of dataset, GANs have been applied in the field of aerodynamic design. Chen et al. proposed a BézierGAN for smooth shape representation that can synthesize aerodynamic shapes from interpretable low-dimensional latent codes,[14] and then applied this approach to the airfoil shape parameterization of aerodynamic optimization[15,16]. Du et al. developed the B-spline based Generative Adversarial Networks (BSpline-GAN) as a parameterization method to automatically infer the design space with sufficient shape variability, and thus realized interactive airfoil aerodynamic optimization.[17] Achour et al. employed a Conditional Generative Adversarial Network (CGAN)[11] to create a model that can generate airfoil shapes corresponding to a given lift-to-drag ratio and area.[18] Yilmaz and German applied the CGAN to create new airfoil shapes based on a vector set of conditional data indicating the desired drag polar.[19] It is seen that GANs in the current studies are mainly utilized to directly generate airfoil geometries with aerodynamic performances as conditional data in some applications.

The Variational AutoEncoder (VAE)[20] is another type of popular generative model that encodes the input data to a probabilistic latent space and regularizes the hidden representation by a standard Gaussian prior. The VAE is relatively easy to implement and stable to train, and it allows for efficient inference. The applications of the VAE in the field of aerodynamic design is still in its early stage. Bertrand et al. first applied the Conditional Variational AutoEncoder (CVAE)[12] to generate the pressure distribution of an airfoil that matches the targeted lift and pitching moment coefficients.[21] However, the generated pressure distributions deviate by a few percent from the target coefficients, which could be caused by a limited variety of training samples or inadequate numbers of latent variables.

The focus of the present research is to develop an approach for the inverse design of supercritical airfoils that can realize the generation of physically-existing target distribution and obtain an airfoil that fits the target distribution. For supercritical airfoils, the suction peak, shock wave, and aft loading intensity are sensitively connected to the geometry due to the transonic nonlinearity. Many explicit rules of hint experiences on the pressure distribution of supercritical airfoils have been adopted in industrial design. For instance, the shock should locate properly to obtain good robustness, and the aft loading should not be too large so that the compromise between lift generation and a mild pitching moment can be achieved.[22,23] To realize these explicit requirements,[24,25] generative models are developed in our work to generate the target wall Mach number distributions, so that the requirements that affect the aerodynamic performance are satisfied. Two generative models are conducted and compared in this paper: (A) CVAE, and (B) an integrated generative network CVAE-GAN[26] that combines the CVAE with the Wasserstein Generative Adversarial Networks (WGAN).[27] These two models are selected since they are capable of generating high quality data while efficiently utilizing conditional information. In this way, wall Mach distributions satisfying ten target features related to the suction peak, shock and aft loading are produced. Furthermore, wall Mach distribution curves with features beyond the range of the dataset can be generated. After the target distribution is specified by generative models, the corresponding airfoil shape is obtained by a deep neural network with the target wall Mach distribution as input features.

In the following section, the dataset and theoretical descriptions of the mapping and generative models used in our work to realize the inverse design are firstly provided in Section 2. Then, the training details and prediction results of the mapping network used to obtain the airfoil shape are presented. The results of two generative models used to produce target distributions are also demonstrated and compared. Finally, the conclusions are drawn in Section 4.

2. Theoretical methods

The overall steps implemented in this work are presented in Fig. 1. First, the proposed method generates a dataset including airfoil shapes, wall Mach number distributions, and their features. Afterwards, the data are preprocessed for the deep learning models. Then the CVAE and CVAE-GAN methods are employed to generate target wall Mach number distributions for the inverse design. Finally, a deep neural network is adopted to realize the mapping from the wall Mach number to the airfoil shape.

2.1. Dataset

Dataset is one of the most important aspects for deep learning algorithms to have better training and generative performance. In order to ensure the availability and diversity of the dataset, an adaptive sampling strategy is employed to generate airfoils with various flow structures,[28] and several constraints are applied during the sampling, such as the leading edge radius must not be smaller than 0.007 and the drag coefficient should not exceed 0.1 to avoid impractical airfoils. In this paper, a total of 1500 supercritical airfoils from the dataset and a subset of samples in the dataset are depicted in Fig. 2. Each sample includes the following three types of data.

(1) x-y coordinates

Each airfoil includes 255 points on the surface. The coordinates are encoded as a tuple (x, y) starting from the trailing edge, following the lower surface to the leading edge, and back to the trailing edge along the upper surface. A spline interpolation approach is employed for the surface points to make the x coordinates for all the airfoils identical. Therefore, only the y coordinates are involved in training, which can reduce the dimensionality of the input or output of the deep learning network to accelerate the training procedure.

(2) Wall Mach number distributions

The wall Mach number Ma_w is defined as the equivalent Mach number of the surface pressure coefficient calculated by the isentropic relation based on the free stream Mach number.[29] For each airfoil, the wall Mach number distributions are evaluated at a fixed lift coefficient of 0.7 with Mach number of 0.76 and Reynolds number of 5.0 million. Zhang et al. investigated the drag, moment, and robustness of three categories of typical pressure distributions (shock-free, double shock, and weak shock) for supercritical airfoils, and concluded that the weak shock pressure distribution was evaluated as the best.[30] Therefore, only airfoils with single shock wave are considered in this paper.

(3) Features of wall Mach number distributions

Features embedded in the pressure or wall Mach number distributions can be of great value to aerodynamic design for supercritical airfoils. The features of the wall Mach number distributions with single shock are described and listed as follows (see Fig. 3):

Suction peak F_{sp}: The x and Ma_w values at the point with the highest wall Mach number near the leading edge on an airfoil. The suction peak should not be too high to avoid an excessive shock intensity or unreasonable leading edge radius.

The start of the shock wave F_{sw0}: The x and Ma_w values at the start of the shock wave.

The end of the shock wave F_{sw1}: The x and Ma_w values at the end of the shock wave. In order to improve the robustness of the aircraft, the shock wave development should be as stable as possible.

Aft loading F_{al}: The x and Ma_w values at the point with the maximum difference in the wall Mach number between the upper and lower surfaces near the trailing edge. An appropriate aft loading can reduce the supersonic zone and velocity on the upper surface, and thus improving the lift drag ratio and off design point characteristics.

Maximum wall Mach number of lower surface F_{lm}: The x and Ma_w values at the point with the maximum wall Mach number on the lower surface. It is used to characterize the maximum velocity of the lower surface to avoid the occurrence of a supersonic zone on the lower surface.

2.2. Generative models

Generative models achieve an unsupervised learning task that involves automatically discovering and learning the regularities or patterns in input data to generate new examples. In this paper, both the CVAE and the integrated generative network CVAE-GAN are employed to generate new target wall Mach number distribution curves for the inverse design of supercritical airfoils with the target features imposed by the condition vector.

2.2.1. CVAE

VAE is a directed graphical generative model based on the concept of auto-encoding,[31] which is among the state- of-the-

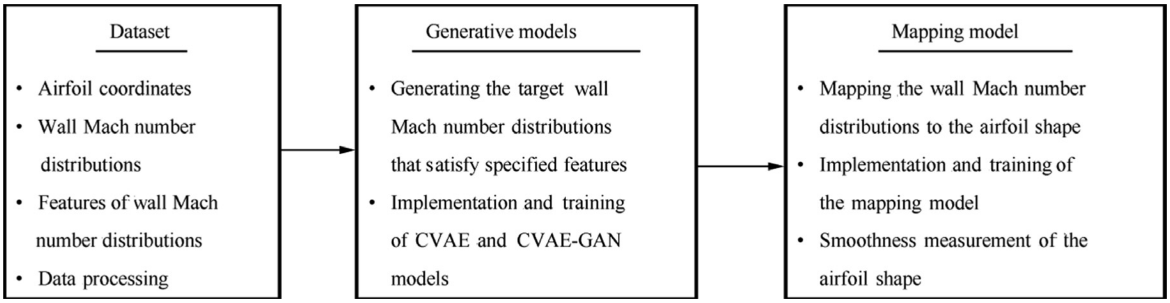

Dataset
• Airfoil coordinates
• Wall Mach number distributions
• Features of wall Mach number distributions
• Data processing
Generative models
• Generating the target wall Mach number distributions that satisfy specified features
• Implementation and training of CVAE and CVAE-GAN models
Mapping model
• Mapping the wall Mach number distributions to the airfoil shape
• Implementation and training of the mapping model
• Smoothness measurement of the airfoil shape

Fig. 1 Overall steps in our approach.

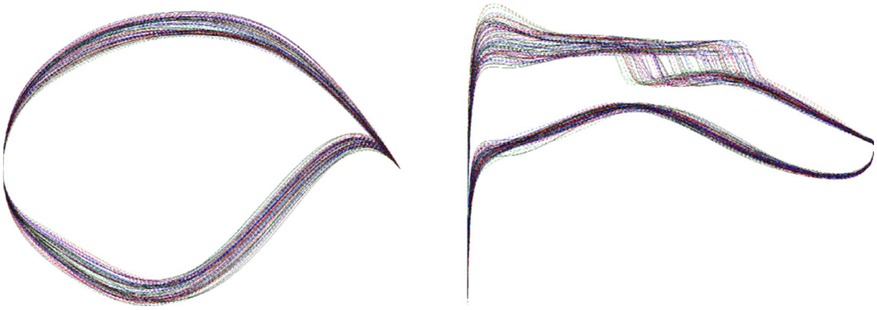


Fig. 2 A subset of samples in dataset.

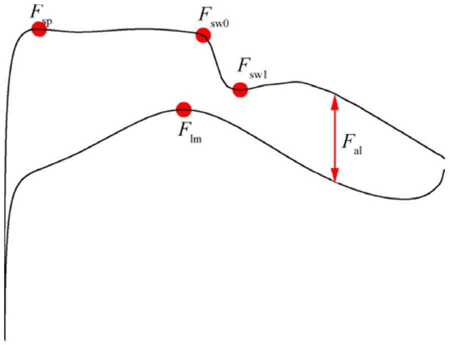


Fig. 3 Features of wall Mach number distributions.

art generative modeling approaches and has obtained excellent results. VAE assumes that the data are generated by some random process involving an unobserved continuous random variable. As an extension of the VAE, CVAE is one of the most popular generative models and is distinguished from the VAE in that it can impose certain conditions in the encoding and decoding processes. The framework of the CVAE consists of two parts:

The encoder network, which maps data sample x to a latent representation z through a learned distribution $P(z|x,c)$, where c is the given condition that the data satisfy.

The decoder network, which tries to generate x' under the given latent vector z and conditions c by sampling from a learned distribution $P(x|z,c)$.

Concretely, the network is trained by minimizing the loss function:

$$L_{\mathrm{CVAE}} = \mathrm{MSE}(x',x) + \mathrm{KL}(q(z|x,c)||p(z|x)) \quad (1)$$

where $\mathrm{MSE}(x',x)$ is the reconstruction loss between the generated data x' and real data x', and the Kullback-Leibler divergence $\mathrm{KL}(q(z|x,c)||p(z|x))$ denotes a regularization loss between the prior $p(z|x)$ and the auxiliary posterior $q(z|x,c)$.

Fig. 4 illustrates the CVAE architecture used in this paper, which is simplified for clarity purposes. In our case, each dataset consists of 255 points on the wall Mach distribution. More importantly, the features of the wall Mach number distribution curve listed in Section 2.1 are represented as the condition vector.

2.2.2. CVAE-GAN

To promote the generation of more realistic curves in our work, the WGAN discriminator based on the CVAE is employed to aid in realizing more accurate reconstruction. The WGAN is employed instead of traditional GANs since it improves the stability of learning, gets rid of problems such as mode collapse, provides meaningful learning curves, and generates high-quality images.[32] The VAE decoder and the GAN generator are collapsed into one by sharing their parameters and training them jointly. The framework of the CVAE-GAN consists of three parts:

The encoder network E: It maps the data sample x to a latent representation z through a learned distribution $P(z|x,c)$, where c is the given condition that the data satisfy.

The generative network G: It generates the x' under the given latent vector z and conditions c by sampling from a learned distribution $p(x|z,c)$. Then the generated data x' are passed to the discriminator network with the ground truth data x to promote the generated data absolutely real.

In the training stage, the network tries to minimize the loss:

$$L_{\mathrm{generator}} = \mathrm{MSE}(x',x) + \mathrm{KL}(q(z|x,c)||p(z|x)) + D(G(z)) \quad (2)$$

The discriminative network D: The function of this network is the same as that in the WGAN. A classification network is trained to distinguish between real and fake samples. The discriminator tries to maximize the difference between the real and fake instances. The Wasserstein-1 distance, or Earth Mover (EM) is used here to measure the two distributions. The final loss function is given as:

$$L_{\mathrm{discriminator}} = D(x) - D(G(z)) \quad (3)$$

These three parts are seamlessly cascaded together, and the whole pipeline is trained end-to-end. The whole architecture of the model is presented in Fig. 5. The input vector and condition vector for our case are the same as in CVAE. Besides, the entire dataset is used for training instead of being split into a training subset and a validation subset. Validation can be achieved by verifying the consistency between the labels of the generated data and the input, i.e., the requirements and constraints.

2.3. Mapping model

2.3.1. Deep neural network

The nonlinear mapping of the wall Mach number distribution to the airfoil shape is realized by a deep neural network, whose architecture is symbolically depicted in Fig. 6. The mapping function is obtained through making the output values for the samples as close as possible to the ground truth. In our case, the input consists of 255 points on the wall Mach distri-

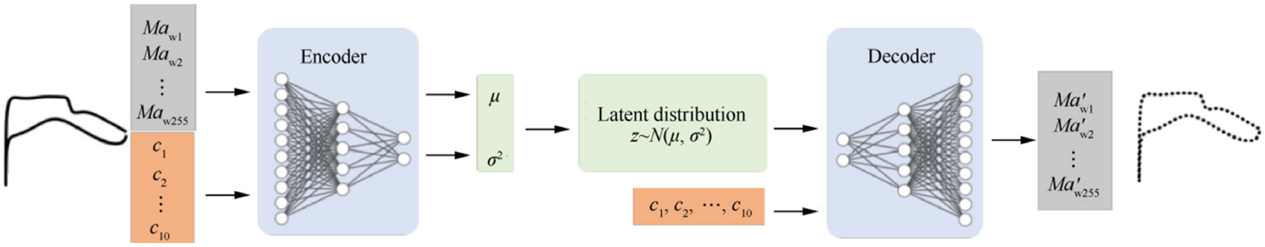


Fig. 4 Architecture of CVA.

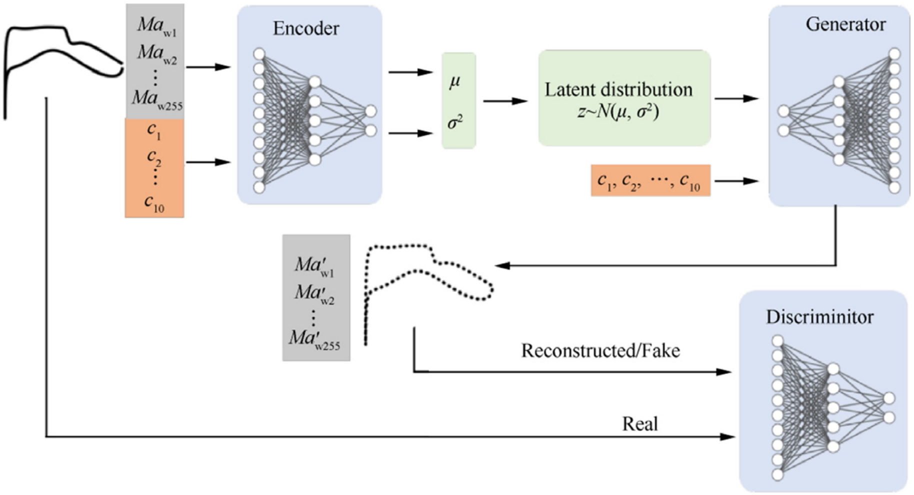


Fig. 5 Architecture of CVAE-GAN.

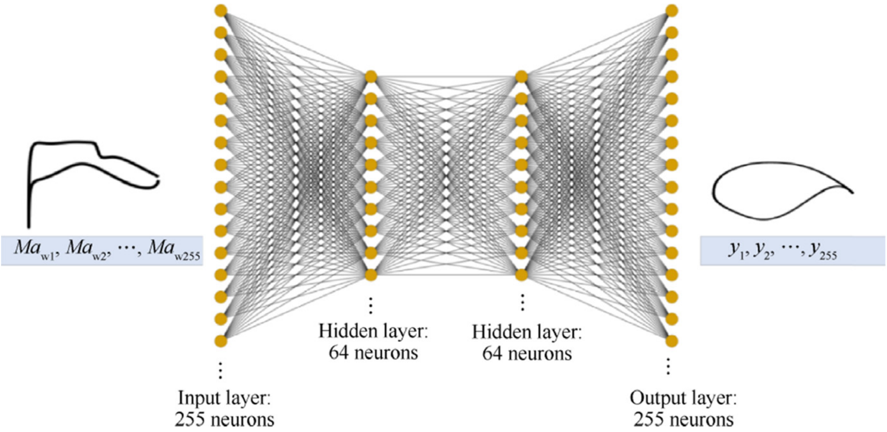


Fig. 6 Architecture of mapping model.

bution, while the output consists of 255 points on the y coordinates of the airfoil shape. The nonlinear mapping is comprised of a sequence of tensor operations and nonlinear activations of weight parameters. The objective of the learning is to obtain appropriate weight parameters to minimize a loss function. The optimization cost in this case is defined as the aggregation of the average sum of squared error terms, which evaluates the difference between the estimated output tensor and the ground truth or the desired output tensor. The overall cost to be minimized is given as follows:

$$L_{\mathrm{map}} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}||y' - y||^2\right), \tag{4}$$

where m denotes the number of datasets. The gradients of the cost are backpropogated through the entire network to train the parameters using an optimizer.

2.3.2. Smoothness measurement of airfoil shape

In our work, the airfoil shape is generated on the basis of the target wall Mach number distribution, which is generated by the generative models described in the last section. Inappropriate or unrealistic wall Mach number distributions may lead to abnormal airfoils with oscillations on the surface. In order to measure the quality of the airfoil shape based on the generated wall Mach number distributions, a smoothness measurement for airfoil shapes is conducted by the summation of distances from each point to the line connected by its two adjacent points, which is expressed by the following equation:

$$smoothness = \sum_{i=1}^{N-1} Distance_{P_n \perp |P_{n-1}P_{n+1}|} \tag{5}$$

where N denotes the number of points on the airfoil surface, which equals 255 in our case. The lower score indicates smoother curve. Fig. 7 presents part of the smooth and unsmooth airfoil shapes. The smooth curve is chosen from the dataset, whose smoothness score calculated using Eq.(5) is 0.0078. The unsmooth curve is mapped from an unrealistic wall Mach number distribution curve by the mapping model described in this section, whose smoothness score is 0.0208. It can be seen that the distances of each point on the smooth curve to the line connected by its two adjacent points are much smaller than those on the unsmooth curve, which illustrates that the smoothness measurement conducted in our work is rational.

3. Experiments and results

The implementation, training, and testing for both mapping and generative models proposed above are performed using PyTorch[33] in Python. PyTorch is a scientific computing framework with wide support for machine learning algorithms. It allows the use of GPUs to improve performance and contains a large ecosystem of community-driven packages. The training details and prediction results of the mapping network, CVAE, and CVAE-GAN are presented below.

3.1. Mapping wall Mach distribution to airfoil shape

The mapping network described in Section 2.3.1 features two hidden layers with 64 neurons per hidden layer. As for activation functions, the hyperbolic tangent (tanh) function is used. Both the input and output tensor are normalized. For the training stage, a random combination of 90% of the total number of airfoils are used for training, and the remaining 10% are used for validation. The loss function presented in Eq. (5) is optimized using the Adam optimizer[34] with a learning rate of 0.0003. The model uses a fixed batch size of 256 and is trained for 2000 epochs.

The convergence histories of the loss function defined by Eq. (4) in both the training and testing datasets are shown in Fig. 8. In order to reveal the reconstruction accuracy in detail, residuals for each point on 3 selected airfoil surfaces are shown in Fig. 9. The distribution statistics of the Root Mean Square Error (RMSE) and MAXimum Error (MAXE) for all the reconstructed airfoils are presented in Fig. 10. The formulas for the RMSE and MAXE are expressed as:

$$\begin{aligned} &\mathrm{RMSE} = \sqrt{\frac{1}{255}\sum_{\mathrm{i}=1}^{255}\left(|\mathrm{y}_\mathrm{i}' - \mathrm{y}_\mathrm{i}||^2\right)} \\ &\mathrm{MAXE} = \max\left(|\mathrm{y}_\mathrm{i}' - \mathrm{y}_\mathrm{i}|\right) \\ &i = 1, 2, \cdots, 255 \end{aligned} \tag{6}$$

It is seen that the training error converges after 2000 epochs, and the trained model in the testing datasets behaves as well as in the training datasets, which shows the generalization ability of the proposed model. For most of the dataset, the RMSE is below 1.0×10^{-5} and the MAXE is below 4.0×10^{-4}, which satisfies the tolerance requirement proposed by Sobieczky[35] (the tolerance zone in a wind tunnel is satisfied when the fitting error of airfoil parameterization is below 0.0007). Stated thus, the designed mapping network architecture is capable of accurately estimating the corresponding airfoil shape from the given wall Mach distribution.

To investigate the smoothness of the generated airfoil shape generated by the mapping model, the smoothness measurement of the airfoil shape proposed in Section 2.3.2 is applied to the following five datasets:

1. Original: 1500 airfoils in the dataset.
2. Reconstructed: 1500 airfoils generated by the mapping model from the original wall Mach number distribution curves in the dataset.
3. Modified1: 1500 airfoils generated by the mapping model from the original wall Mach number distribution curves in the dataset with each point increased by 0.01.
4. Modified2: 1500 airfoils generated by the mapping model from the original wall Mach number distribution curves in the dataset with specified points on the upper surface increased by 0.05.
5. Modified3: 1500 airfoils generated by the mapping model from the original wall Mach number distribution curves in the dataset with specified points on the lower surface increased by 0.05.

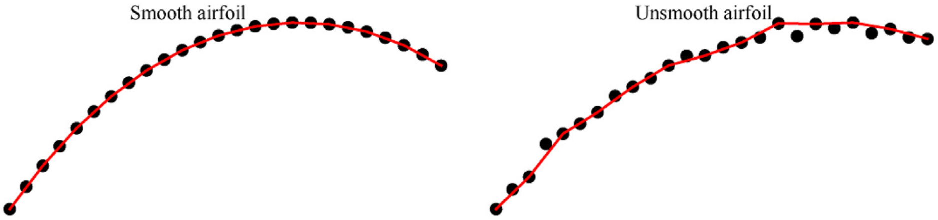


Fig. 7 Smooth and unsmooth airfoil shapes.

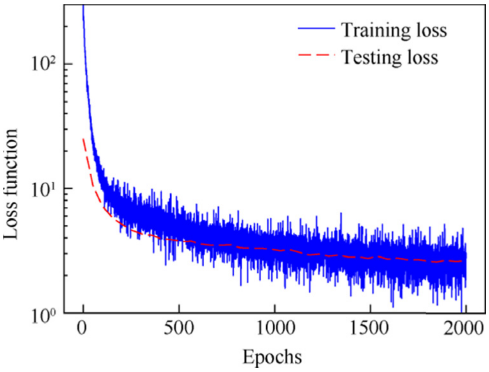


Fig. 8 Convergence history of loss function in mapping model.

Fig. 11 shows the detailed modification where "distance" in the third row denotes the distances of each point on the surface to the line connected by its two adjacent points. The statistical smoothness scores for the five datasets are quantitatively assessed and listed in Table 1. As shown, the disparity in the smoothness between the original and reconstructed airfoils by the mapping model is small and acceptable, which demonstrates the validity of the mapping model. While evident differences appear between the original and the three modified airfoils, small oscillations can be observed on the airfoil shapes in the three modified cases. Especially in the Modified1 case, it is hard to visually distinguish the original and modified wall Mach number distribution curves

however, the shapes and smoothness scores for the corresponding airfoils are quite different. Based on the above results, the smoothness measurement we proposed can be used to measure the authenticity and rationality of the wall Mach number distributions generated by the generative models. In our following work, a generated airfoil shape is determined to be smooth, and a wall Mach number distribution is further determined to be realistic if the smoothness score is less than the maximum smoothness score, i.e., 0.01095.

3.2. Generation of wall Mach distributions

In this section, we demonstrate and compare the effectiveness of the two generative models, the CVAE and CVAE-GAN

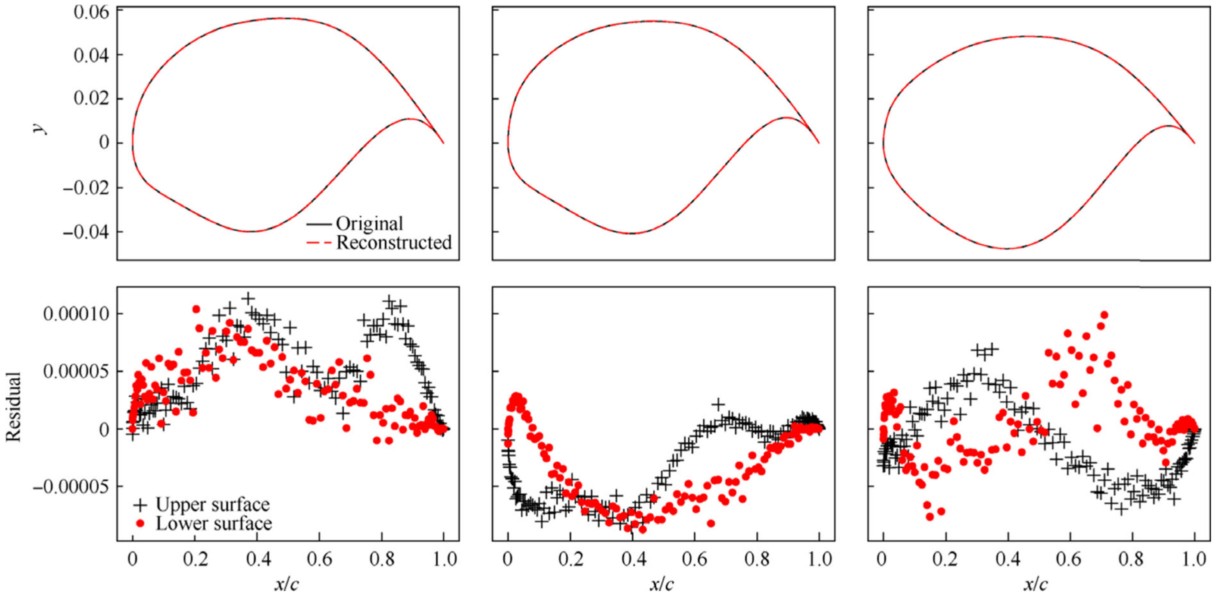


Fig. 9 Reconstruction residuals of 3 selected airfoil surfaces.

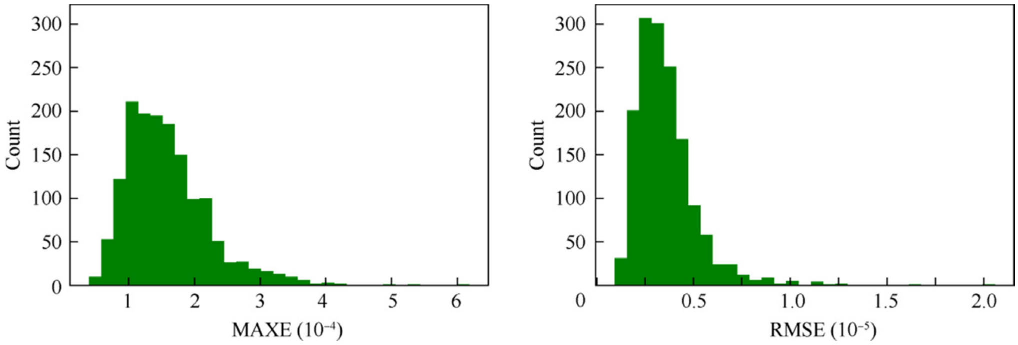


Fig. 10 Distribution statistics of RMSE and MAXE for all reconstructed airfoils.

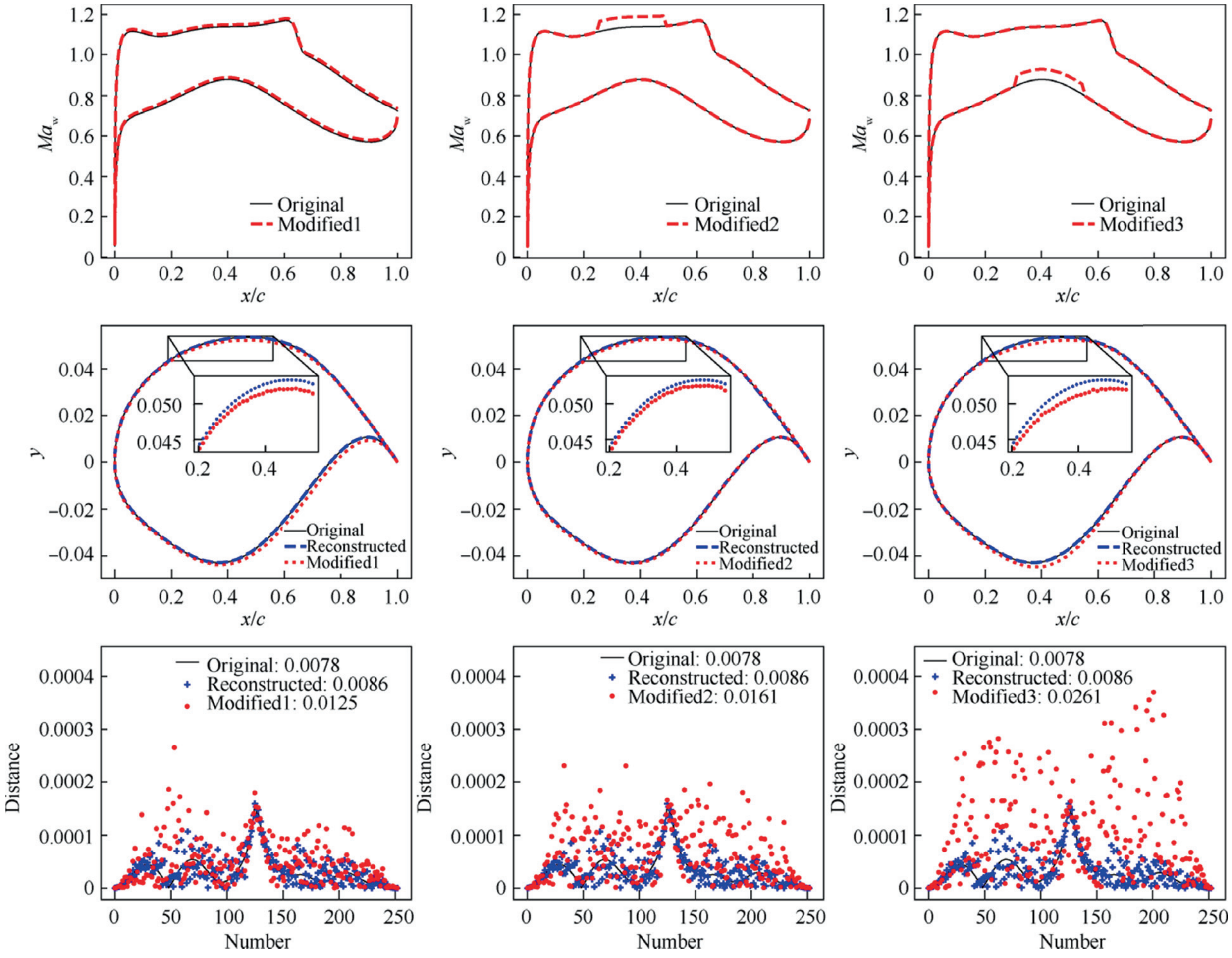


Fig. 11 Smoothness of five sample datasets.

Table 1 Statistical smoothness scores for five datasets.

Dataset	Original	Reconstructed	Modified1	Modified2	Modified3
Maximum	0.00824	0.01095	0.02995	0.02660	0.03350
Minimum	0.00736	0.00761	0.02436	0.01435	0.02499
Average	0.00782	0.00828	0.02746	0.02055	0.02964

introduced in Section 2.2 in generating target wall Mach number distributions for the inverse design.

The settings for the network structures in the two models are described as follows. The encoder in both two methods consists of 4 Fully-Connected layers (FCs) with 255, 128, 64, and 32 units, respectively. The decoder in the CVAE as well as the generator in the CVAE-GAN consist of 4 FCs with 32, 64, 128, and 255 units, respectively. The discriminator in the CVAE-GAN also consists of 4 FCs with 255, 128, 64, and 32 units, respectively. The dimensions of the latent vector and the conditional vector in the two methods are set to 10. A fixed learning rate of 0.0002 and the hyperbolic tangent (tanh) function are employed for all the above networks. An Adam optimizer is used in the above networks to realize the training except for the discriminator using an RMSProp optimizer.[36] Both the inputs and condition vectors are normalized. For the training stage of the CVAE, a random combination of 90% of the total number of airfoils are used for training, and the remaining 10% are used for validation.

The convergence histories of the reconstruction errors $\mathrm{MSE}(x', x)$ for the CVAE and CVAE-GAN with a fixed batch size of 256 are shown in Fig. 12. The reconstructed loss in the CVAE converges to 0.00388 for the training dataset and 0.00431 for the testing dataset, which shows the portability of the trained model on unseen data. The loss in the CVAE-GAN converges to 0.00294, which is much smaller than that in the CVAE and indicates better reconstructions for the original curve. Table 2 presents the statistical smoothness scores for airfoil shapes mapped from the original and reconstructed wall Mach number curves. In the last section, we determine

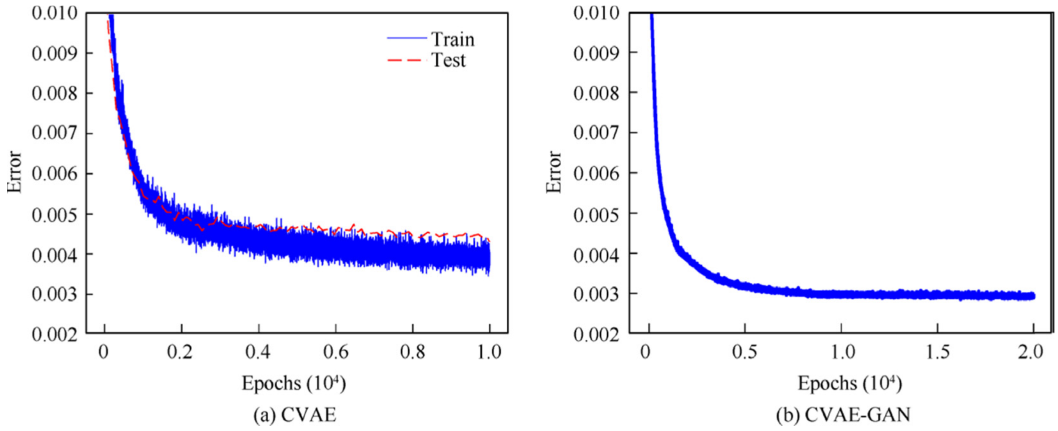


Fig. 12 Convergence histories of reconstruction errors for CVAE and CVAE-GAN.

Table 2 Statistical smoothness scores for airfoil shapes mapped from generated wall Mach number curves.

Dataset	Reconstructed	CVAE	CVAE-GAN
Maximum	0.01095	0.01042	0.01051
Minimum	0.00761	0.00765	0.00761
Average	0.00828	0.00821	0.00823

that a generated airfoil shape and further a wall Mach number distribution are realistic if the smoothness score of the mapped airfoil shape is less than 0.01095. It is obvious that both methods satisfy the threshold and the wall Mach number distributions generated by both methods are realistic. To visualize and compare the training process between the two methods, the developments of the reconstructed shapes and the errors during the training process for the two methods are further illustrated in Fig. 13. The results show that as the number of iterations increases, more realistic shapes are generated by both methods with smaller reconstruction errors.

In order to make detailed comparisons of the reconstructive and generative abilities between two methods, the distribution statistics of the RMSE and the MAXE defined in Eq. (6) for the reconstructed wall Mach number distributions in the whole dataset are presented in Fig. 14. It is seen that the RMSEs of the CVAE and CVAE-GAN are below 6.0×10^{-4} and 4.0×10^{-4}, respectively. The MAXEs are below 6.0×10^{-2} and 4.0×10^{-2}, respectively. We can conclude that the accuracies of the wall Mach number distributions reconstructed by the CVAE-GAN are better than those reconstructed by the

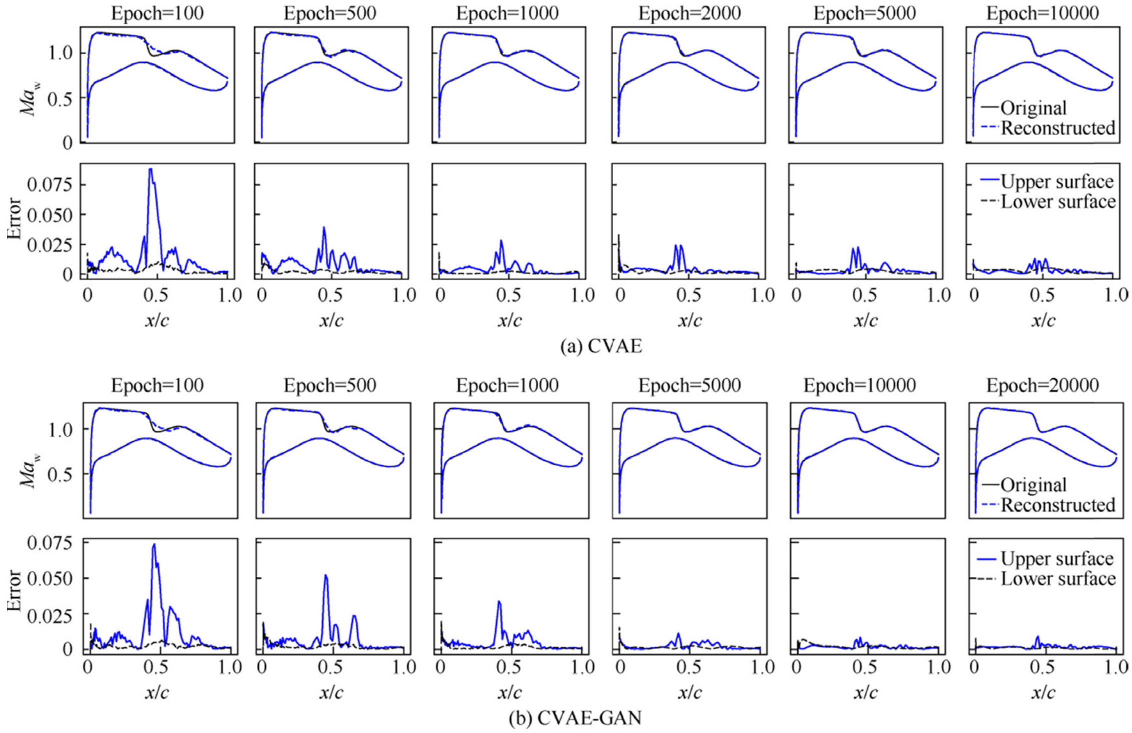
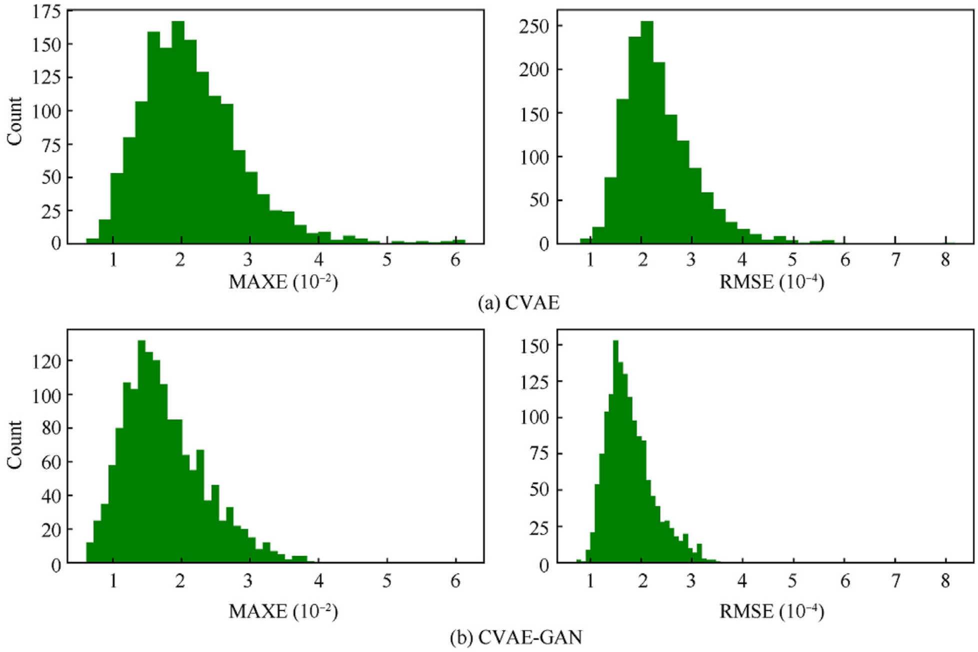


Fig. 13 Training processes for CVAE and CVAE-GAN.



(a) CVAE

(b) CVAE-GAN

Fig. 14 Distribution statistics of RMSE and MAXE for all wall Mach number distributions reconstructed by CVAE and CVAE-GAN.

CVAE. This could be due to the inclusion of the discriminator by the WGAN promoting the model to generate more realistic wall Mach number distributions.

Fig. 15 presents the statistical results for the relative errors between the features of the original and reconstructed wall Mach number distributions in the whole dataset for the CVAE and CVAE-GAN. Interestingly, the performance of the two models varies significantly from feature to feature. For the suction peak related features (Features 1–2), errors for the Ma_{w} are less than 0.5% in most of the samples. For the aft loading related features (Features 3–4), the errors are less than 5% for the x values and less than 2% for the Ma_{w} values in nearly all the samples. For the shock wave related features (Features 5–8), the results are inferior to those of other features but the rel-

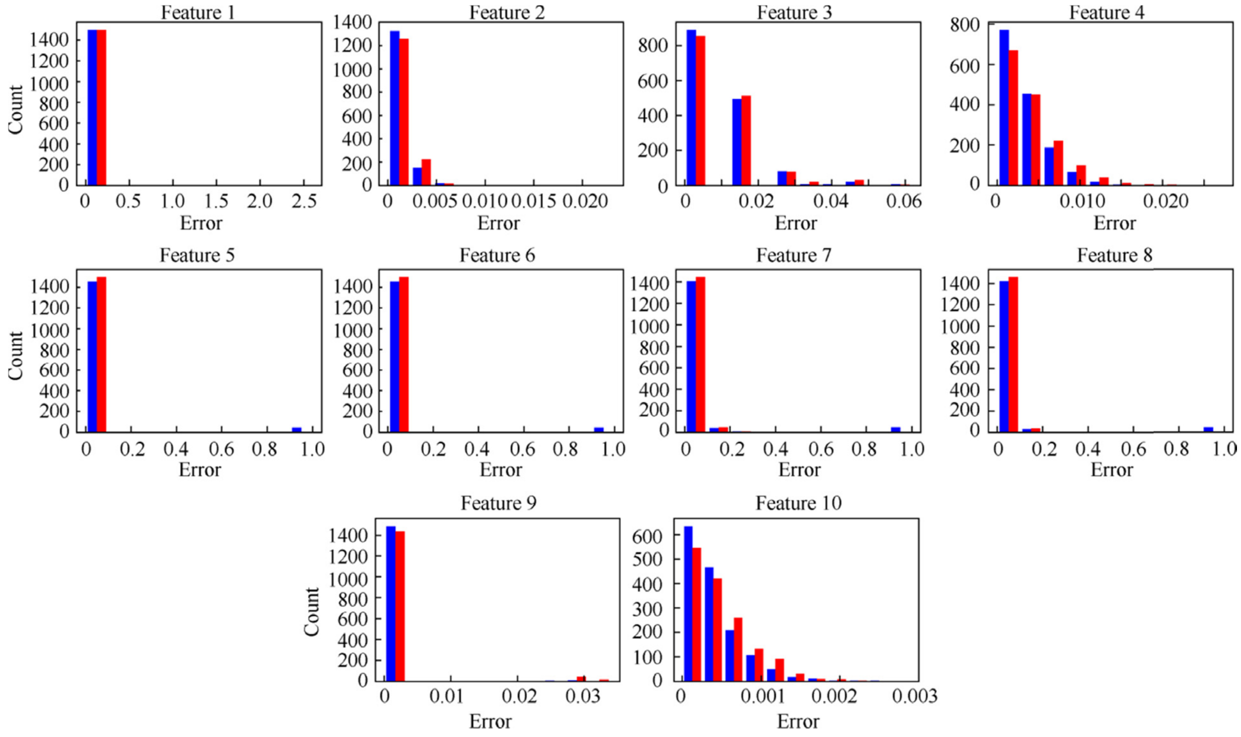


Fig. 15 Statistical results for relative errors between features of original and reconstructed wall Mach number distributions for CVAE (blue) and CVAE-GAN (red).

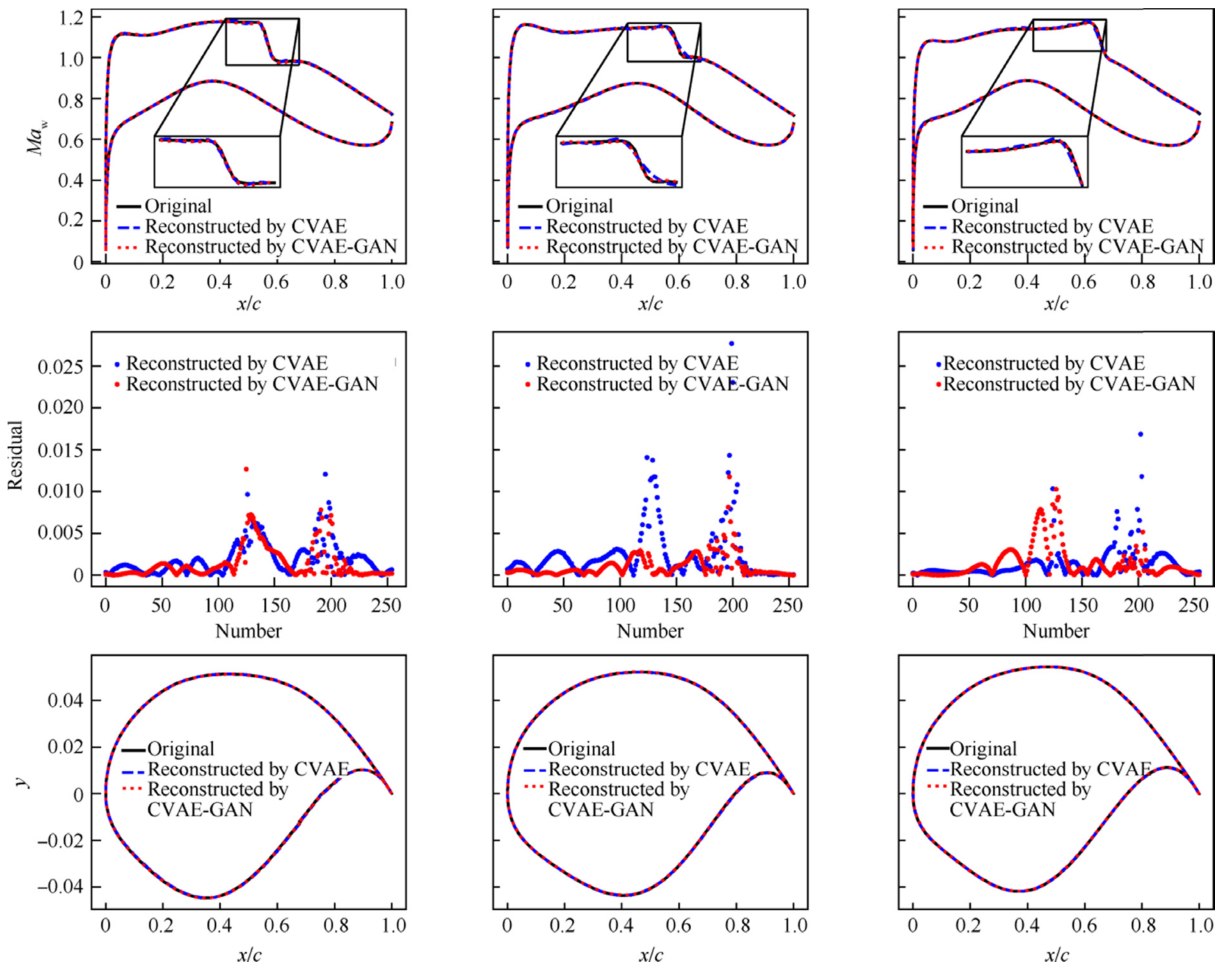


Fig. 16 Reconstructed results for CVAE and CVAE-GAN on 3 selected samples.

Table 3 Features of original and reconstructed wall Mach number distribution curve.

Item	F_{sp}		F_{al}		F_{sw0}		F_{sw1}		F_{lm}	
	x	Ma_w	x	Ma_w	x	Ma_w	x	Ma_w	x	Ma_w
Original	0.044	1.279	0.764	0.304	0.358	1.147	0.419	0.992	0.457	0.897
CVAE	0.044	1.280	0.754	0.302	0.354	1.149	0.425	0.989	0. 457	0. 897
CVAE-GAN	0.044	1.276	0.754	0.304	0.357	1.147	0.421	0.992	0. 457	0. 897
Original	0.066	1.154	0.710	0.328	0.611	1.116	0.800	0.883	0.396	0.888
CVAE	0.066	1.155	0.710	0.329	0.602	1.117	0.791	0.892	0.396	0.887
CVAE-GAN	0.066	1.153	0.710	0.327	0.610	1.124	0.800	0.882	0.396	0.887
Original	0.086	1.153	0.743	0.306	0.532	1.161	0.601	0.987	0.408	0.884
CVAE	0.086	1.153	0.743	0.305	0.520	1.181	0.605	0.979	0.408	0.884
CVAE-GAN	0.086	1.150	0.754	0.307	0.528	1.170	0.594	0.981	0.408	0.884

ative errors are less than 10% for the vast majority of the dataset. For the maximum wall Mach number of the lower surface related features (Features 9–10), the errors are less than 3% for the x values and less than 0.3% for the Ma_w values in all the samples. These two features are most satisfied compared to all features because the fitting errors on the lower surface are much smaller than those on the upper surface. From the above analysis, we can conclude that the errors for the x values are larger than those for the corresponding Ma_w values. This is expected because the Ma_w values are involved in the model directly as the input while the x values are obtained indirectly. Furthermore, for the shock wave related features, the CVAE-GAN performs better than the CVAE with a lower average percentage error. While for other features, the CVAE performs slightly than the CVAE-GAN.

In order to reveal the reconstructed and generative capabilities of the two methods, the original and reconstructed shapes and residuals on 3 selected samples are illustrated in Fig. 16.

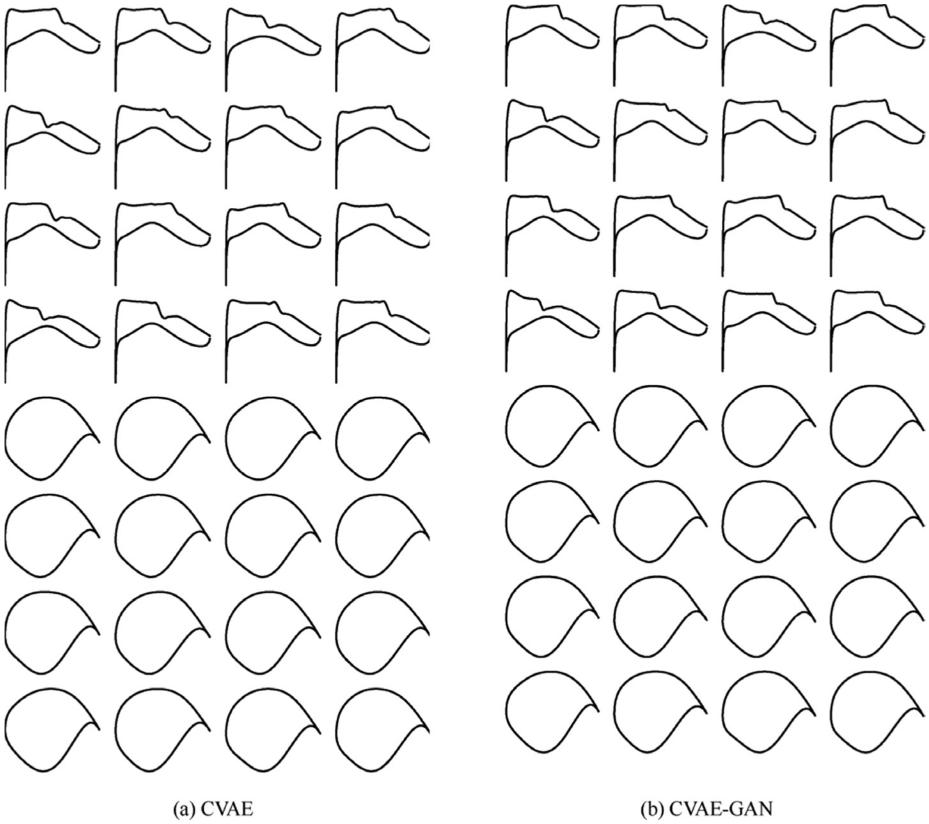


Fig. 17 Samples generated by CVAE and CVAE-GAN.

Additionally, detailed features of the wall Mach number distributions for the 3 samples are listed in Table 3. The table shows that the reconstructed wall Mach number distributions by the two methods fit very well with the original curves in most locations, and more evident differences can be observed at the locations near the suction peak and shock with a large change in curvature. At locations near the shock, the CVAE-GAN achieves smoother and more realistic curves with smaller residuals than the CVAE. Consequently, the shock wave related features of the CVAE-GAN are much closer to the original/given features. At locations near the suction peak, the CVAE attains smaller residuals, which contributes to the suction peak related features of the CVAE closer to the original/given features than the CVAE-GAN. At locations on the lower surface, the distributions reconstructed by both methods agree well with the ground truth thus, the maximum wall Mach number of lower surface related features matches very well with the given features.

Finally, we generate 16 samples using the trained CVAE and CVAE-GAN under 16 given features beyond the range of the dataset, and the generated samples are presented in Fig. 17. These results suggest that both methods can generate diverse and realistic wall Mach number distributions and smooth airfoil shapes, provided that the given features are proper and rational. Additionally, similar results can be observed in that the curves generated near the shock wave by the CVAE-GAN are smoother than those generated by the CVAE.

4. Conclusions

(1) A designed mapping network based on the deep neural network for the inverse design process is conducted, and the model is capable of estimating the corresponding airfoil shape to the given wall Mach distribution. The predicted results show that the present approach has reasonable accuracy for both training and testing samples.

(2) In order to measure the quality of the airfoil shape based on the generated wall Mach number distributions, a smoothness measurement for the airfoil shape is proposed. The validity and accuracy of the measurement were demonstrated on five datasets. A threshold value is given based on the statistical results to determine whether the generated airfoil shape is smooth and the target wall Mach number distribution is further realistic.

(3) Two generative models, the CVAE and CVAE-GAN, for the inverse design process are proposed. The effectiveness of the two models is demonstrated and compared. The results show that both models can generate diverse and realistic wall Mach number distributions satisfying the given features, while the CVAE-GAN model outperforms the CVAE model and achieves better reconstruction accuracies for all the samples in the dataset.

Overall, the method developed in this article is a reliable and promising inverse design method for supercritical airfoils in transonic aircraft design. The utilization of the generative models not only achieves the classical patterns on the pressure distribution of supercritical airfoils, but also satisfies the explicit rules and requirements involved in the industrial design. Moreover, the designed mapping network and the proposed smoothness measurement ensure the feasibility of the generated airfoils in engineering tasks. Last but not the least, the method can be further applied in generating a variety of samples with good space-filling properties in the objective space.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This study was co-supported by the National Key Project of China (No. GJXM92579) and the National Natural Science Foundation of China (Nos. 92052203, 61903178 and 61906081).

References

1. Barrett TR, Bressloff NW, Keane AJ. Airfoil shape design and optimization using multifidelity analysis and embedded inverse design. *AIAA J* 2006;**44**(9):2051–60.
2. Obayashi S, Takanashi S. Genetic optimization of target pressure distributions for inverse design methods. *AIAA J* 1996;**34**(5):881–6.
3. Takanashi S. Iterative three-dimensional transonic wing design using integral equations. *J Aircr* 1985;**22**(8):655–60.
4. Nadarajah S, Jameson A. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. In: 38th Aerospace Sciences Meeting and Exhibit. Reston: AIAA; 2000
5. Kim S, Alonso J, Jameson A. Design optimization of high-lift configurations using a viscous continuous adjoint method. In: 40th AIAA aerospace sciences meeting & exhibit. Reston: AIAA; 2002
6. Obayashi S. Inverse optimization method for aerodynamic shape design. Recent development of aerodynamic design methodologies: Springer; 1999.
7. Bui-Thanh T, Damodaran M, Willcox K. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA J* 2004;**42**(8):1505–16.
8. Kharal A, Saleem A. Neural networks based airfoil generation for a given cp using Bezier–PARSEC parameterization. *Aerosp Sci Technol* 2012;**23**(1):330–44.
9. Sun G, Sun Y, Wang S. Artificial neural network based inverse design: Airfoils and wings. *Aerosp Sci Technol* 2015;**42**:415–28.
10. Sekar V, Zhang M, Shu C, Khoo BC. Inverse design of airfoil using a deep convolutional neural network. *AIAA J* 2019;**57**(3):993–1003.
11. Mirza M, Osindero S. Conditional generative adversarial nets. *arXiv preprint arXiv:14111784* 2014.
12. Kingma DP, Mohamed S, Rezende DJ, et al. Semi-supervised learning with deep generative models. Adv Neural Inf Process Syst; 2014. 3581–9.
13. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. Adv Neural Inf Process Syst; 2014: 2672–80.
14. Chen W, Fuge M. BézierGAN. Automatic generation of smooth curves from interpretable low-dimensional parameters. arXiv preprint arXiv:180808871; 2018.
15. Chen W, Chiu K, Fuge M. Aerodynamic design optimization and shape exploration using generative adversarial networks. AIAA scitech 2019 forum. Reston: AIAA; 2019
16. Chiu K, Fuge M. Airfoil design parameterization and optimization using Bézier generative adversarial networks. arXiv preprint arXiv:200612496; 2020.
17. Du X, He P, Martins J. A b-spline-based generative adversarial network model for fast interactive airfoil aerodynamic optimization. *AIAA scitech 2020 forum. Reston:AIAA* 2020.
18. Achour G, Sung WJ, Pinon-Fischer OJ, et al. Development of a conditional generative adversarial network for airfoil shape optimization. AIAA scitech 2020 forum. Reston: AIAA; 2020.
19. Yilmaz E, German B. Conditional generative adversarial network framework for airfoil inverse design. In: AIAA aviation 2020 forum. Reston: AIAA; 2020.
20. Kingma DP, Welling M. Auto-encoding variational bayes. *arXiv preprint arXiv:13126114* 2013.
21. Bertrand X, Tost F, Champagneux S. Wing airfoil pressure calibration with deep learning. In: AIAA aviation 2019 forum. Reston: AIAA; 2019.
22. Harris CD. NASA supercritical airfoils: A matrix of family-related airfoils. Washington, D.C.: NASA; 1990.
23. Kim HJ, Rho OH. Aerodynamic design of transonic wings using the target pressure optimization approach. *J Aircr* 1998;**35**(5):671–7.
24. Zhang Y, Fang X, Chen H, et al. Supercritical natural laminar flow airfoil optimization for regional aircraft wing design. Aerosp Sci Technol 2015;43:152–64.
25. Li YZ, Deng KW, Zhang Y, et al. Pressure distribution guided supercritical wing optimization. Chin J Aeronaut 2018;31(9):1842–54.
26. Bao J, Chen D, Wen F, et al. CVAE-GAN: fine-grained image generation through asymmetric training. *Proceedings of the IEEE international conference on computer vision*, 2017.p.2745–54..
27. Martin AS, Bottou L. Wasserstein generative adversarial networks. *Proceedings of the 34th international conference on machine learning*, 2017.
28. Li R, Zhang Y, Chen H. Design of experiment method in objective space for machine learning of flow structures. *8th European conference for aeronautics and space sciences*, 2019.
29. Délery J, Marvin JG, Reshotko E. Shock-wave boundary layer interactions. *Advisory Group For Aerospace Research And Development Neuilly-Sur-Seine* 1986.
30. Zhang Y, Chen H, Fu S, et al. A practical optimization design method for transport aircraft wing/nacelle integration. Acta Aeronautica et Astronautica Sinica 2012;33(11):1993–2001 [Chinese].
31. Rumelhart DE, McClelland JL, Group PR, et al. Parallel distributed processing, vol. 1. Foundations. Explorations in the microstructure of cognition. 1986.
32. He C, Huang S, Cheng R, et al. Evolutionary multiobjective optimization driven by generative adversarial networks (GANs). IEEE Trans Cybernet 2020.
33. Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 2019:8026–37.
34. Kingma DP, Adam BaJ:. A method for stochastic optimization. *preprint arXiv:14126980* 2014.
35. Sobieczky H. Parametric airfoils and wings*Recent development of aerodynamic design methodologies*. Springer; 1999. p. 71–87.
36. Hinton G. Neural networks for machine learning online course. [Internet]. [cited 2020 Oct 10]. Available from https://www.coursera.org/learn/ neural-networks/ home/welcome.