

Benchmark solutions

A deep learning approach for efficiently and accurately evaluating the flow field of supercritical airfoils

Haizhou Wu^{a,c,d,b}, Xuejun Liu^{a,d,*}, Wei An^e, Songcan Chen^{a,d}, Hongqiang Lyu^e^a College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing 211106, China^b Key Laboratory of Aerodynamic Noise Control, Mianyang, China^c State Key Laboratory of Aerodynamics, Mianyang, China^d Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China^e College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

ARTICLE INFO

Article history:

Received 13 July 2019

Revised 4 November 2019

Accepted 17 November 2019

Available online 18 November 2019

Keywords:

Flow field

Supercritical airfoil

Computation fluid dynamics (CFD)

Generative adversarial network (GAN)

ABSTRACT

The efficient and accurate access to the aerodynamic performance is important for the design and optimization of supercritical airfoils. The aerodynamic performance is usually obtained by using computational fluid dynamics (CFD) methods or wind-tunnel experiments. But the computations of CFD are very time intensive and expensive, and the prior knowledge in wind-tunnel experiments plays a decisive role in engineering. Though many surrogate methods were proposed to alleviate the costs of these traditional approaches, most of them can only calculate the low-dimensional aerodynamic performance, and is not able to provide the accurate prediction of transonic flow fields for supercritical airfoils. Since the flow fields are equipped with its own discipline as a physical system in fluid dynamics, it is therefore possible to learn this discipline via data-driven machine learning approaches. Deep learning is witness to expansive growth into diverse applications due to its immense ability to extract essential features from complicated physical systems. Generative adversarial networks (GANs) as a recent popular method in deep learning are capable of efficiently capturing the distribution of training data. In this work, we proposed a surrogate model, ffsGAN, which leverage the property of GANs combined with convolution neural networks (CNNs) to directly establish a one-to-one mapping from a parameterized supercritical airfoil to its corresponding transonic flow field profile over the parametric space. Compared with the most existing surrogate models, the ffsGAN is superior in efficiently and accurately predicting the high-dimensional flow field rather than the low-dimensional aerodynamic characteristics. The ffsGAN method is first trained using 500 airfoils that sampled based on RAE2822. The flow fields are then predicted for unseen airfoils to evaluate the generalization of the model in terms of prediction accuracy. An investigation of the effects of various hyper-parameters in the network architectures and loss functions is performed. The experimental results show that ffsGAN is a promising tool for rapid evaluation of detailed aerodynamic performance. The elaborate flow field predicted by ffsGAN is possible to be considered in airfoil design to further improve the design and optimization quality in the future.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Aerodynamic performance, as the primary characteristics in wing design, is related to the geometry of airfoil profile, velocity, atmospheric density, flight attitude and other operating conditions. Supercritical airfoils are especially useful for improving the aerodynamic performance in transonic range, reducing drag and improving position control [1]. However, the aerodynamic performance

of supercritical airfoils is extremely sensitive to the shape of airfoil and the operating conditions. Therefore, the effective access to aerodynamic performance of supercritical airfoils is crucial in wing design.

Traditionally, wind tunnels are used to evaluate the aerodynamic performance of airfoils, but the results strongly depend on prior knowledge of designers and only part of the aerodynamic characteristics is considered on the surveys [2–6]. Furthermore, the design process is time-consuming and costly. With the development of computer technologies and numerical simulation methods, high-fidelity computation fluid dynamics (CFD) is applied to solve

* Corresponding author.

E-mail address: xuejun.liu@nuaa.edu.cn (X. Liu).

the governing equations of aerodynamics and significantly reduces the dependency on wind tunnel experiments in wing design. However, the methods of CFD need abundant computing resources for extensive numerical computation. In engineering, surrogate models, also called response surface models or meta-models, were proposed to increase the efficiency in the evaluation of aerodynamic performance for airfoils as an auxiliary of the costly CFD computation. The existing surrogate models can be divided into three categories [43]: multi-fidelity models, reduced-order models (ROMs) and data-driven models [7]. The first two categories are proposed based on physical models and are precise and efficient under certain conditions. Nevertheless, they are sensitive to model parameters which may lead to bad performance in robustness. By contrast, data-driven models are able to capture the hidden characteristics of data via the learning process involving machine learning algorithms. This process is highly flexible and can be further generalized to unseen data. Therefore, the data-driven models are widely utilized in the surrogate-based optimization design of airfoils. Our work in this paper mainly focuses on data-driven surrogate models.

At present, data-driven models primarily rely on regression methods that modeling the relationship between the shape of airfoil profiles and the aerodynamic characteristics. The commonly used data-driven surrogate models, such as polynomial response surface model [8], artificial neural network [9,10], radial basis function [11], support vector regression [12], Kriging model [13,14,49], and multiple output GP (MOGP) [15], etc., have been widely used in optimization design for rapid prediction of aerodynamic characteristics. However, most of the current surrogate models are limited to predict the low-dimensional physical quantities, such as lift coefficient, drag coefficient, moment coefficient and pressure distribution, which reflect the average aerodynamic performance for a single characteristic in the flow field and cannot depict the precise and complete flow field structures. Considering the flow field structures, including vortex, boundary layer, wake, shock wave and so on, are expected to wholly enhance the performance of optimization design of airfoils, we aim to build a surrogate to rapidly predict these detailed physical characteristics of flow field structures. In general, accurate and efficient prediction of the detailed flow field structures plays an important role in improving the quality of airfoil design. However, the structures of flow field are most available with the costly wind-tunnel experiments and CFD simulations, and can not be obtained efficiently and accurately from most surrogate models. Although some approaches combining proper orthogonal decomposition (POD) and data-driven surrogates are able to precisely predict smooth flow field, they are incapable of handling flow field with shocks [49] and cannot be considered in the design and optimization of supercritical airfoils.

From the viewpoint of machine learning, most current surrogate models are “shallow” models, which are only able to fit a small subset in the original function space and are limited in fitting the whole complex function space. “Shallow” models are therefore lack of generalization for the data far away from the training data. For this reason, large numbers of data are usually needed in the training process to enhance the model generalization for complex problems. Deep learning approaches provide a possibility to handle these problems in a “deep” way. The successful applications in the area of computer vision have recently drawn researchers’ attention to deep learning methods in the field of structure mechanics and fluid mechanics [16,17]. Due to the ability of convolutional neural network (CNN) in automatically extracting features from images, Zhang et al. utilized CNN to extract the geometric features of the two-dimensional airfoil profiles. Then the features consisting of the Mach and Reynolds are fed into a fully connected neural network to predict aerodynamic characteristics [18]. Miyanawala and Jaiman proposed an efficient CNN

model to predict the coefficients of cylindrical wake flow field under different two-dimensional geometric shapes [19]. Sekar et al. used CNN to extract the characteristics of pressure coefficient distribution in the inverse design and predict its corresponding airfoil shape [20]. The studies above have shown that the deep learning models are superior in solving non-linear problems. Nevertheless, only the average aerodynamic characteristic is predicted in these methods, and the high-dimensional structure of flow field has not been considered yet.

Generative adversarial networks (GANs) [21] have been proposed to generate various types of data, and innovated in theory and model structure [22,23,28–37]. Many variants of GANs have been widely used in image processing, including text-to-image synthesis [38,39], image-to-image translation [35,40,41] and image super-resolution [42]. In light of the superiority of GANs in modeling the generation of images, the whole flow fields displayed in the form of images can be modeled by GANs. In this work, we aim to construct a deep conditional generative adversarial network to model the one-to-one mapping between the shape of airfoil profile and the image of elaborate flow field structure based on the aerodynamic simulation data.

In this work, we devise a generative adversarial network combined with CNN for the efficient prediction of flow field structure, which is called ffsGAN for abbreviation. This model is able to automatically generate precise and high resolute images to elaborate the whole structures of transonic flow fields over a specific range of supercritical airfoils. We investigate the selection of loss function among several options during the learning of the model. The kernel size of the embedded CNN is also examined for optimal performance. Using CFD simulation data, we demonstrate that our model achieves the accurate prediction of the flow field structures for given airfoil profiles. Taking the place of the time-consuming CFD simulation and the expensive wind-tunnel experiments, our generative model can be taken as a surrogate model for rapidly predicting the detailed flow field structures for given airfoil profiles, rather than the average aerodynamic performance of several characteristics. The generated images, which contain abundant physical characteristics in the structure of flow field, can be set to accurately access the aerodynamic performance. Therefore, we also discuss the potential application of our model to the elaborate design of airfoils.

The structure of this paper is organized as follows. A brief description of the problem is given in Section 2, and following with our surrogate model in which considering the related methods including multilayer perceptron, convolution neural network and generative adversarial network. In Section 3, a series of experiments are conducted and the effects of the hyper-parameters in CNN are discussed. The extensive analyses on the objective function are also considered. Concluding remarks are provided in Section 4.

2. Methodology

2.1. Formulation of the problem

We work on evaluating steady flow field structures over a range of supercritical airfoils given a defined Mach and Reynolds number. The Hicks-Henne (HH) method [25] is used for airfoil parameterization due to its capability of constructing a smooth airfoil with a few design variables. The upper and lower surfaces of an airfoil are designed as a 14-dimensional vector. It commonly takes one hour to obtain the numerical results of the flow field around one airfoil using CFD simulation. The computed results can then be explicitly shown in the form of images using the post-processing software Tecplot. In contrast, our surrogate model is aim to take the place of this time expensive process and realize an efficient one-to-one

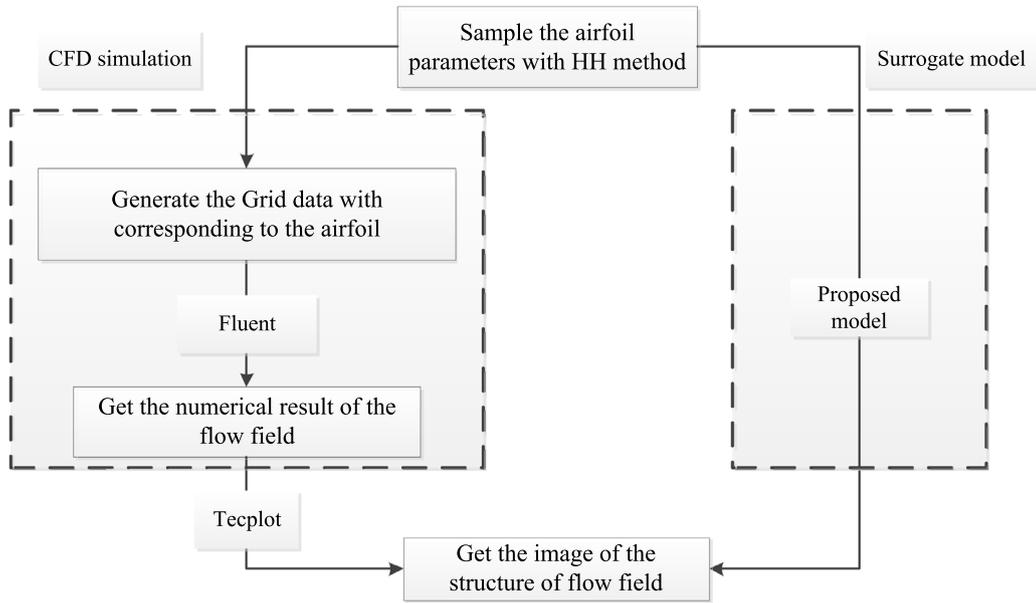


Fig. 1. The flowchart for the comparison between CFD simulation and the proposed surrogate.

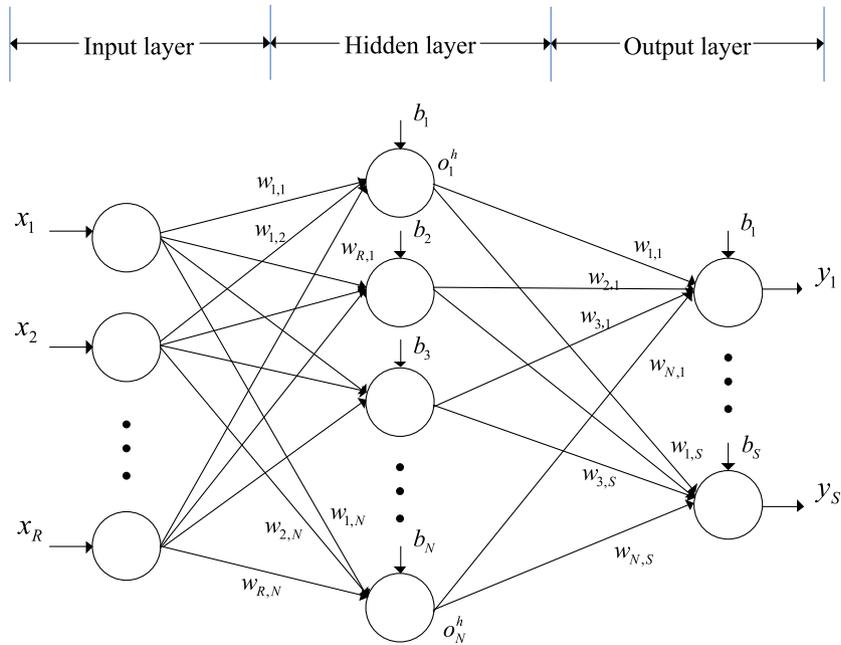


Fig. 2. A MLP with one hidden layer.

mapping from a given supercritical airfoil to its corresponding flow field structure shown in images. The flowchart for the comparison between CFD simulation and our surrogate is shown in Fig. 1. The construction of the proposed surrogate model is elaborated in the subsequent sections.

2.2. Multilayer perceptron

Multi-layer perceptron (MLP) is the most typical model of artificial neural networks. It is made up of neurons, which are connected together in a complex manner to form a network. Neurons are the basic elements of the MLP, performing mapping from the input to the output defined in the regression problem. An example of the model is shown in Fig. 2. For the neurons in one layer, the sum of the weighted inputs is computed at the presence of a bias,

and is then transferred through an activation function (e.g. sigmoid function) to obtain the output. The calculation from the weighted inputs to the hidden neuron h_j can be denoted as

$$h_j = \sum_{i=1}^R w_{i,j}x_i + b_j \tag{1}$$

where x_i is the corresponding input data, $w_{i,j}$ is the weight connecting the input layer neuron $i = (1, 2, \dots, R)$ and the hidden layer neuron $j = (1, 2, \dots, N)$, b_j is the bias in the hidden layer, and R and N are the total numbers of neurons in input and hidden layers, respectively.

Therefore, the output is obtained by transferring the input signal through a nonlinear activation function $\sigma(\cdot)$. The sigmoid function is a commonly used activation function as shown below,

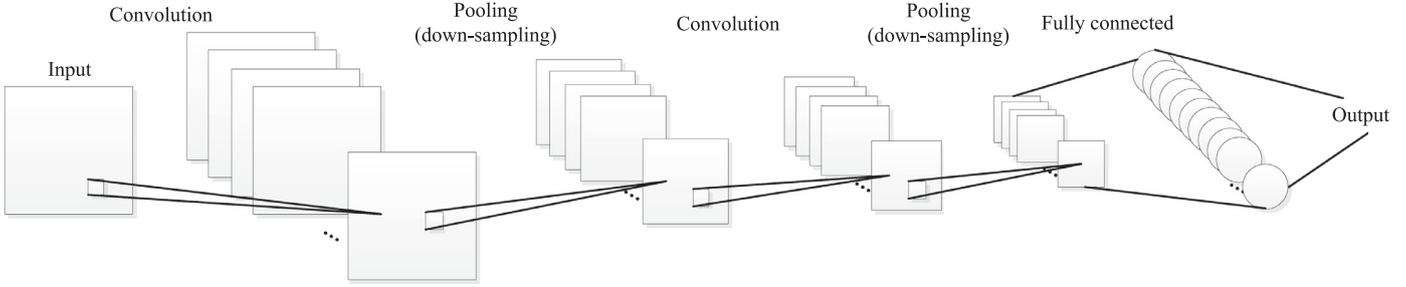


Fig. 3. Typical schematic of convolution neural networks.

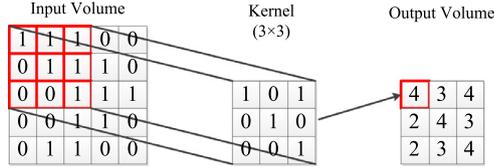


Fig. 4. Typical convolution operation in the CNN.

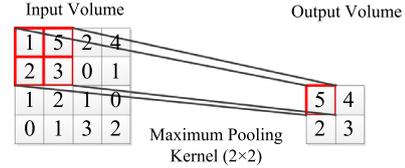


Fig. 5. Typical pooling operation in the CNN.

$$o_j^h = \sigma(h_j) = \frac{1}{(1 + e^{-h_j})} \quad (2)$$

In the output layer, the output of the neuron is obtained as following,

$$y_k = \sigma\left(\sum_{j=1}^N w_{j,k} o_j^h + b_k\right) \quad (3)$$

where $w_{j,k}$ is the weight connecting the hidden layer neuron $j = (1, 2, \dots, N)$ and the output layer neuron $k = (1, 2, \dots, S)$, and b_k and S are the bias and the total number of neurons in the output layer, respectively. The weights and biases are the parameters of MLP and are usually optimized by SGD (Stochastic Gradient Descent), Adam (Adaptive moments) and so on.

2.3. Convolution neural network

The MLPs have been widely used in various fields, such as pattern recognition [44], classification [45], function approximation [46], signal processing [47] and so on. However, they are limited in images processing in the field of computer vision for the reason that MLPs have too many parameters (the weights and biases) to estimate. Hence, Lecun et al. [27] proposed the convolutional neural network with significantly fewer parameters to alleviate this difficulty by introducing convolutional kernels. CNNs normally comprise several types of layers, such as the convolutional layer, the pooling layer and the fully connected layer. A typical schematic of the CNN architecture is shown in Fig. 3.

In convolutional layer, the convolution operation leverages two ideas that play an important role in CNN: sparse interactions and parameter sharing. The sparse interactions are accomplished by making the kernels (also known as filters) smaller than the input in spatial dimensions, typically in processing the images. The parameters in kernels are shared at every position in one layer and can be revisited.

Similar to MLPs, the weights of the kernels make up an element-wise scalar product whose area is connected to the input volume, known as the receptive field. In each convolution layer, the scalar product operation performed for one kernel is shown in Fig. 4. The scalar product mentioned above is then transferred by the non-linear activation maps $\sigma(\cdot)$. The two operations aforementioned form one convolutional layer and the corresponding output $o_{i,j}$ for

one kernel can be calculated as follows:

$$o_{i,j} = \sigma\left(\sum_{m=0}^{l_1} \sum_{n=0}^{l_2} \sum_{c=1}^C w_{m,n,c} \cdot I_{i+m,j+n,c}\right) \quad (4)$$

where w is the kernel with a size of $l_1 \times l_2$ and I is the input image with length L , height H and channel C . The spatial dimensionality of the output volume in the convolution layer can be altered using different sizes of the kernels with two operations: the stride and the padding.

In addition to the convolutional layer, the pooling layer is another importance module of CNNs. The pooling layer makes a spatial reduction of the dimensions for a given input, called down-sampling. In all cases, pooling helps to make the representation approximately invariant to small translations of the input. The pooling layer generally follows a convolution layer and the output of convolution layer is scaled in dimensionality by the specified pooling operation: maximum pooling or average pooling. A maximum pooling operation with the kernel size of 2×2 is shown in Fig. 5. Fully connected layer is exactly the same as the network in MLP in which the neurons are full connected between neighboring layers. Similar to the mechanism in the operations of convolution and pooling, deconvolution and unpooling perform the corresponding inverse operations.

2.4. Generative adversarial network

Generative adversarial network (GAN) is a novel generative model [21] and can be viewed as the following two-player minimax game. One of them is called the generator which creates samples that are intended to come from the same distribution as that of the real data. The other player is the discriminator that determines whether the samples are from the generator or not. The process is shown in Fig. 6:

The loss of GAN expressed in Fig. 6 can be calculated as follows,

$$\min_{\theta^G} \max_{\theta^D} = \mathbb{E}_{x \sim \mathbb{P}_{data}} [\log(D(x))] + \mathbb{E}_z [\log(1 - D(G(z)))] \quad (5)$$

where \mathbb{P}_{data} is the underlying distribution of the real data x and z is a variable sampled from some simple prior distributions, such as Gaussian or uniform distributions. Since GAN is a minimax game, the discriminator and the generator work iteratively to carry out minimization and maximization on cross-entropy respectively.

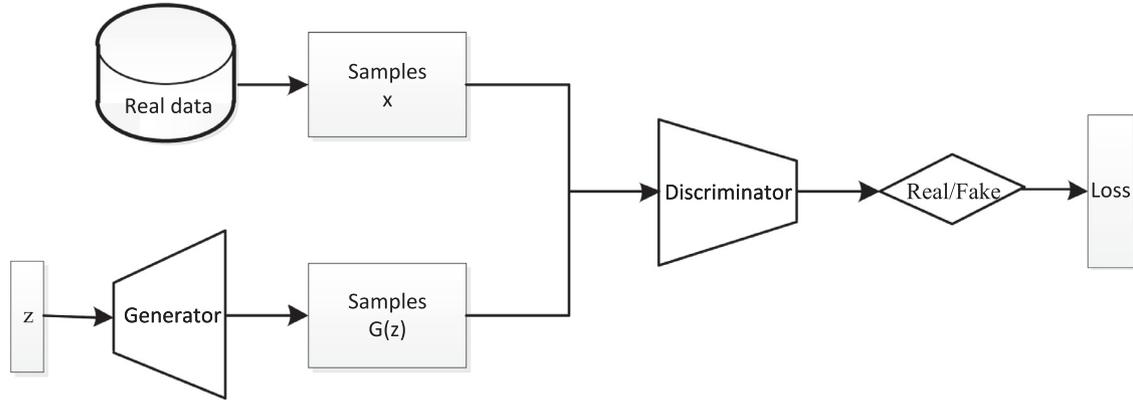


Fig. 6. The flowchart of GAN.

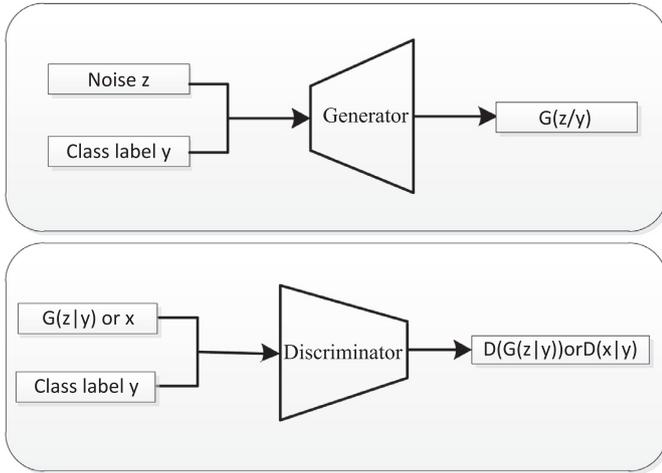


Fig. 7. Conditional generative adversarial network.

Both of the generator and the discriminator can be represented by deep neural networks. The generator network is defined by a function G that takes z as input and owns its parameters θ^G . The distribution of the generated samples $\mathbb{P}_{G(z)}$ is expected to be similar to \mathbb{P}_{data} in the desired setting of generator network. Similarly, the discriminator network with parameter θ^D is implemented as a function D that is fed with x or $G(z)$. The discriminator is designed to maximize the probability of assigning the correct label to both generated samples and samples from the training data. The goal of the minimax game is to find the Nash Equilibrium by minimizing the function in Eq. (5) with respect to G and maximizing the function with respect to D alternately. If both the generator and discriminator networks are of sufficient capacity, we have $G(z) \sim \mathbb{P}_{data}$ with given random input z . In other word, the output of discriminator network is 1/2 when deciding whether the input is x or $G(z)$.

Following the original GAN, two effective variants of GAN were proposed. DCGAN combined the GAN framework with deep convolutional neural networks for generating high quality images [22]. It has been shown that DCGAN is superior in the problems involving image processing. A conditional version of generative adversarial network (cGAN) was introduced to generate images conditioned on class labels [23]. The cGAN extended the original GAN from an unsupervised method to a supervised one. The structure of cGAN is shown in Fig. 7, in which the noise z and the class label y are fed to the generator and the input of the discriminator is also conditioned on the class label y . Thus the objective function of cGAN

is

$$\min_{\theta^G} \max_{\theta^D} = \mathbb{E}_{x \sim \mathbb{P}_{data}} [\log(D(x|y))] + \mathbb{E}_z [\log(1 - D(G(z|y)))] \quad (6)$$

2.5. ffsGAN

In this work, we combine cGAN with the deep structure of CNN and present a variant of GAN to build an efficient one-to-one mapping from a given supercritical airfoil to its corresponding flow field structure. The construction of our surrogate includes two phases, the training phase and the test phase. In the training phase, the parameterized profiles and the corresponding images of flow field structure are jointed to form the training data. For learning the model, the parameterized airfoil profiles represented as vectors are fed to the generator, while the generated images and the corresponding real images conditioned on their parameterized vectors are jointed as the input of the discriminator. In the test phase, the images of flow filed structures are generated by the learned generator. The detailed flowchart is shown in Fig. 8.

Some previous works combined auto-encoders (AEs) with CNNs to predict the flow field structure around airfoil [24,50,51]. Both AE and GAN belong to generative models. The AEs generate data by assuming low dimensional latent variables which capture the real features of data. In these works, AEs were solved by only considering the reconstruction errors. In contrast, the mechanism in GANs is to fulfill the process that the generated data match the distribution of the original images. Therefore, in ffsGAN we construct the loss function combing the minimax objective function in GANs and the reconstruction error, and this is expected to achieve more accurate results.

For a given data point (x, y) , it contains a parameterized supercritical airfoil represented as a 14-dimensional vector y , and its corresponding pressure profile illustrated as image x . [40] demonstrated that omitting the adversarial loss generate the related target images, while the details are hard to recognize. On the other hand, omitting the L1 loss (or identity loss) gives realistic images, but unrelated to the given source image. So here we define the loss function as below,

$$\min_G \max_D \text{Loss}(G, D) + \frac{\lambda}{MN} \|G(y) - x\|_1 \quad (7)$$

which consists of both the adversarial loss and the L1 loss.

In cGAN, the noise z and the one-hot labels y are concatenated as the input to the generator and the generation process is thought of as a one-to-many mapping. The noise z models the variability among samples within the same class. The label y is encoded in the one-hot label in classification problem. In contrast, our model

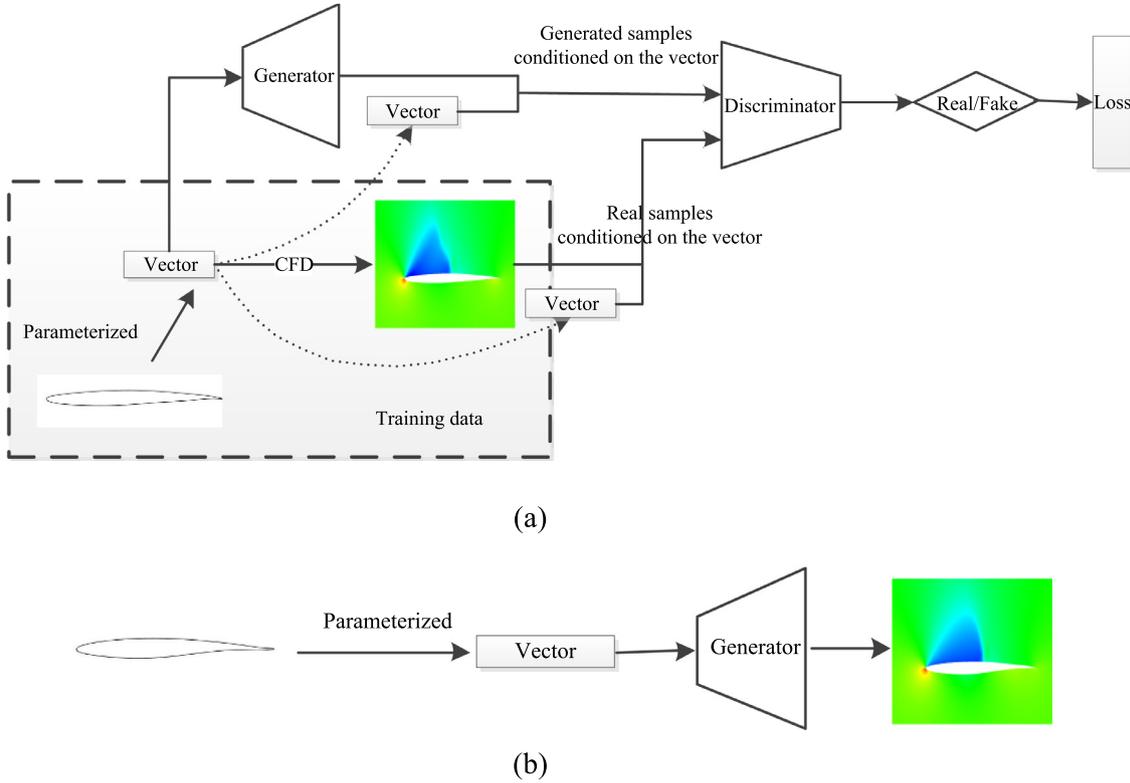


Fig. 8. The detailed flowchart of our surrogate. (a)-The training phase, in which the parameterized airfoil profile shown as a vector is the input of the generator. The generated and the real images of flow field structure, and the vector are jointed as the input of the discriminator. (b)-The test phase, in which the parameterized airfoil profile is fed to the trained generator, the image of the detailed flow field structure is obtained finally.

is aimed to constructing a one-to-one mapping from an airfoil profile to its corresponding flow field structure, and the noise z is therefore removed entirely. The optimization of minimax function can be performed with respect to the generator and the discriminator iteratively. At the step for optimizing the generator, the loss function is

$$Loss_G = \mathbb{E}_y[-\log(D(G(y)|y))] + \frac{\lambda}{MN} \mathbb{E}_{(x,y) \sim \mathbb{P}_r(x,y)} \|G(y) - x\|_1 \quad (8)$$

where y is a uniformly distributed variable whose upper and lower bounds depend on the design space of the airfoil. And $\mathbb{P}_r(x, y)$ denotes the joint distribution of x and y . M and N are the length and height of the image, respectively. The first term in Eq. (8) is a cross-entropy between the distribution of the generated and the real images. This loss function enables the networks to extract the underlying features of images in an unsupervised manner and tries to fool the discriminator network by generating a well predicted flow field structure. The second term is the scaled L1 loss (or identity loss) over the whole image, and is used to measure the difference between generated and the real images. λ is the hyper-parameter that balances the cross-entropy and L1 loss, and needs to be finely tuned. The term of MN is to average the L1 loss for each pixel.

The loss function for optimizing the discriminator is defined as follows:

$$Loss_D = -\mathbb{E}_{(x,y) \sim \mathbb{P}_r(x,y)} [\log(D(x|y))] - \mathbb{E}_y [\log(1 - D(G(y)|y))] \quad (9)$$

Minimizing Eq. (9) with respect to the discriminator is to distinguish the real flow field image from the generated flow field image. The loss functions of Eq. (8) and Eq. (9) are optimized iteratively with respect to the generator and the discriminator respectively. The effect of the two terms in Eq. (7) will be investigated in Section 3.

2.6. Configuration of neural networks

In light of the consideration above, we aim to build ffsGAN combined with CNNs to predict the detailed flow field structure from a given airfoil profile. The representation of the airfoil is crucial in airfoil design. It not only determines the resource consumption and computational efficiency during the design, but also impacts upon the smoothness and validity of the airfoil profile. Importantly, airfoil parameterization decides whether there is a meaningful optimum scheme in the design space and whether the optimum can be found by the optimization algorithm. Although many previous works successfully applied CNN to the airfoil profile images for automatic features extraction, we think this introduced more parameters in the model and led to more computations for working out the model. Also, using images to represent airfoil makes it difficult to search for the optimum scheme in airfoil design due to the high dimensionality of the design space. For all these reasons, we adopt the widely used Hicks-Henne (HH) method to parameterize the airfoil profile. The airfoil is represented in a 14-dimensional vector which is taken as the input of our model, and the corresponding output is the image of the flow field structure obtained from the CFD simulation and the post-processing tool Tecplot.

Under this data representation strategy, the input of generator is the 14-dimensional vector, y , and the output is the corresponding image. The structure of the discriminator is the same as the original cGAN with two inputs x and y . In Fig. 9 we show the overview architecture of the generator and the discriminator for the ffsGAN. The generator and the discriminator both contain 7 convolution/ deconvolution layers without the pooling layers or the fullyconnected layers. The convolution layer in the discriminator extracts the underlying features of the image and the deconvolution layer in the generator fulfills the transformation from an airfoil to its corresponding flow field image.

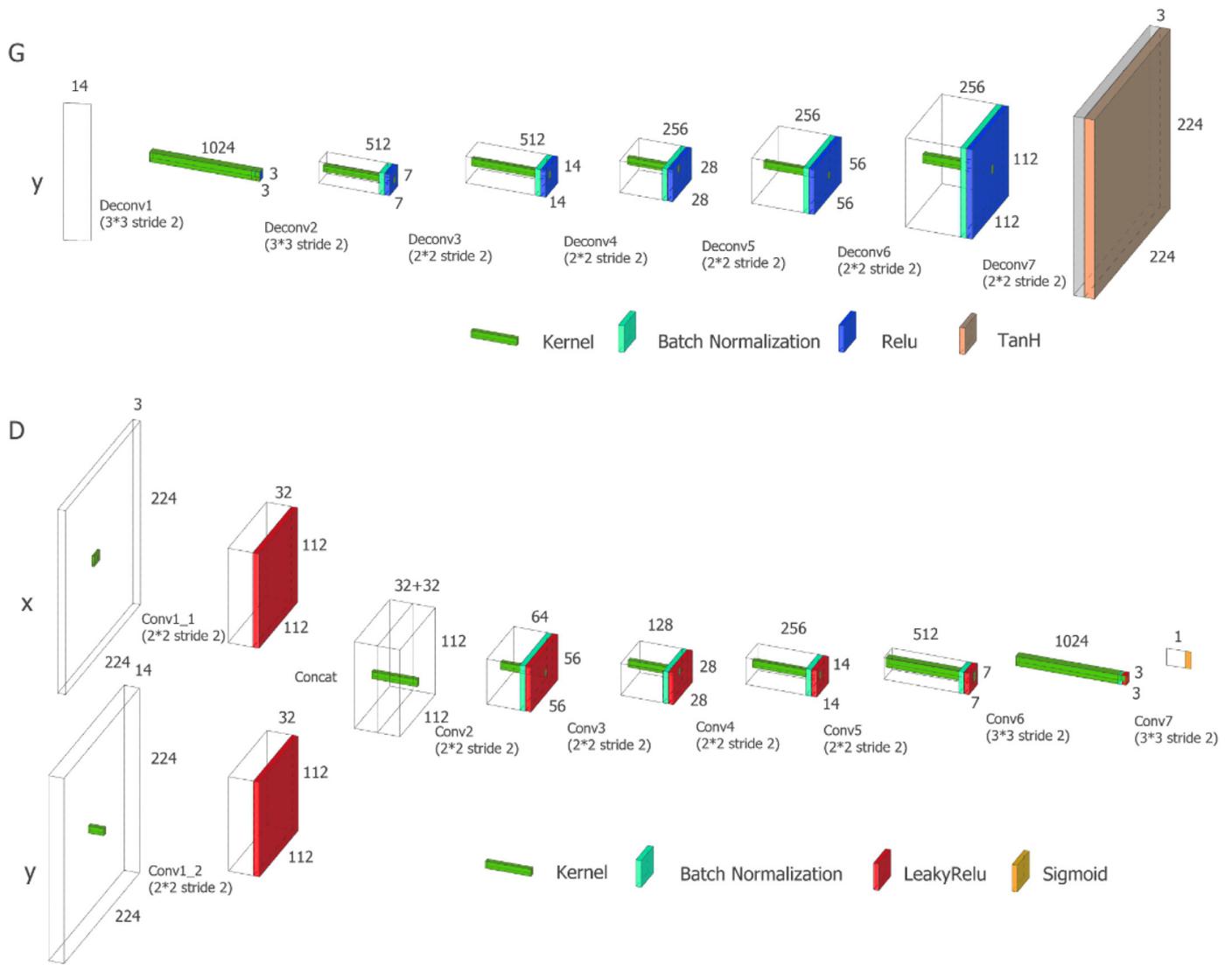


Fig. 9. Schematic diagram of the overall architecture of ffsGAN.

The property of the invariance to local translation in pooling layer is useful if we care more about whether some feature is present than exactly where it is. However, the locations of features are importance as well for modeling the elaborate flow field structure. Thus, the pooling operation is not appropriate here anymore. Additionally, if the features are transferred to the fullyconnected layer, the structure information may be damaged. We therefore remove the fully connected layers as well. The kernel size in the convolution layers and deconvolution layers is set as 2×2 or 3×3 which is the general setting in the field of computer vision (CV). Compared with the larger kernel size, such as 7×7 , the small ker-

nel size of 3×3 will get the same scale of receptive field with a deeper network, which is helpful for enhancing the model capacity and model depth. At the same time, the number of parameters in the whole network is reduced. For the sake of efficiently extracting the spatial features in images, the filters (or channels) that represent the number of kernels need to be large enough. To demonstrate the goodness of the chosen network structure, we also investigate the influence of the various kernel sizes in the following sections.

For a more detailed description of the model settings see Table 1.

Table 1 Detailed description of the generator and discriminator architectures.

Generator						Discriminator					
Kernel	Filters	Stride	Padding	BN	Activation	Kernel	Filters	Stride	Padding	BN	Activation
3×3	1024	2	0	Yes	Relu	2×2	32	2	0	Yes	LeakyRelu(0.2)
3×3	512	2	0	Yes	Relu	2×2	32	2	0	Yes	LeakyRelu(0.2)
2×2	512	2	0	Yes	Relu	2×2	64	2	0	Yes	LeakyRelu(0.2)
2×2	256	2	0	Yes	Relu	2×2	128	2	0	Yes	LeakyRelu(0.2)
2×2	256	2	0	Yes	Relu	2×2	256	2	0	Yes	LeakyRelu(0.2)
2×2	256	2	0	Yes	Relu	2×2	512	2	0	Yes	LeakyRelu(0.2)
2×2	256	2	0	Yes	Relu	3×3	1024	2	0	Yes	LeakyRelu(0.2)
2×2	3	2	0	No	Tanh	3×3	1	2	0	No	Sigmoid

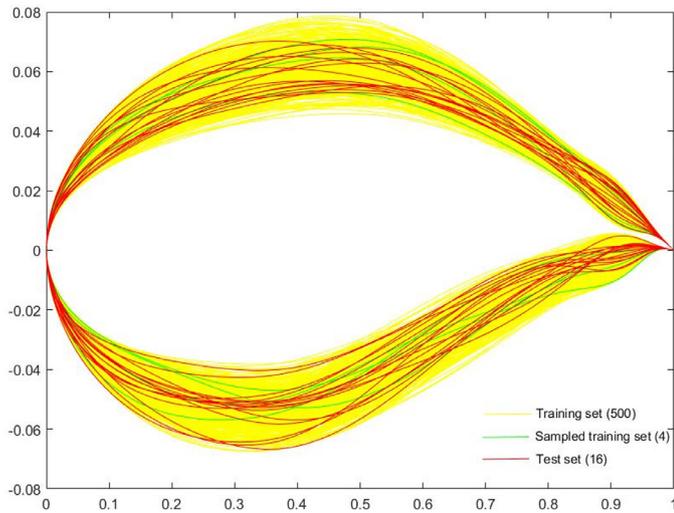


Fig. 10. Sampled training and test airfoils.

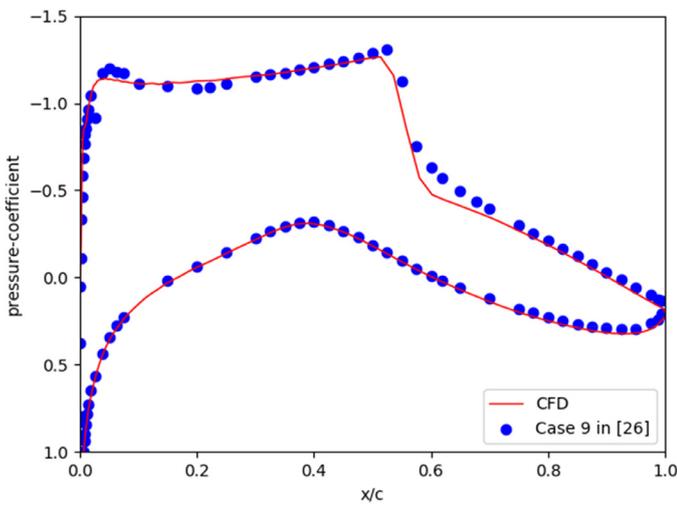


Fig. 11. Comparison of the pressure coefficient distributions for the RAE 2822 airfoil, $Ma = 0.73$, $\alpha = 2.79^\circ$, $Re = 6.5 \times 10^6$, between the experimental data of Case 9 in [26] and CFD values.

3. Experiment and analysis

3.1. Datasets

In order to train the proposed generative model, we sample 516 airfoils with HH method from the modified RAE2822. The training and test airfoils are presented in Fig. 10 including 500 training airfoils in yellow and 16 test airfoils in red. Especially, 4 airfoils in green are randomly selected from the training airfoils to demonstrate the fitting capacity of the model. Since the test airfoils have not been seen by the model during the training phrase, they are used to validate the generalization of the model.

The corresponding pressure profiles for the sampled airfoils (shown in Fig. 10) are generated under the fixed physical condition in which the Reynolds number $Re = 6.5 \times 10^6$, the Mach number $Ma = 0.73$ and the angle of attack $\alpha = 2.79^\circ$. The mesh plays an important role to assess the aerodynamic performance in the resolution and accuracy. According to the experiments in ref [48], we chose the same mesh with 53,756 elements. The validation for the deterministic case was performed by comparing CFD results with the experimental data for Case 9 from Cook et al. in [26]. The surface pressure-coefficient showed good agreement with the computational data (from CFD) in Fig. 11, with the small discrepancies occurring around the shock wave. In this condition, the corresponding value of y^+ is calculated as 3. The flow fields are obtained by solving the Reynolds Averaged Navier–Stokes (RANS) equations utilizing the finite volume method, for which the Spalart–Allmaras turbulence model is employed. The obtained flow field structures are transferred to images using Tecplot. Thus, the training sets contain the input (profile parameters of the supercritical airfoils) and the output (images of pressure profiles). The resolution of the raw pressure profiles images are all of 1045×929 . For efficient computation and clear display, the images are resized to 224×224 by cutting out the uninformative parts and keeping the parts of most interests. One example of the process for image resizing is shown in Fig. 12.

3.2. Results and discussions

The implementation of ffsGAN is based on the Pytorch code of cGAN. The deep learning models are sensitive to parameters, such as number of layers, filters size, and so on. In model training,

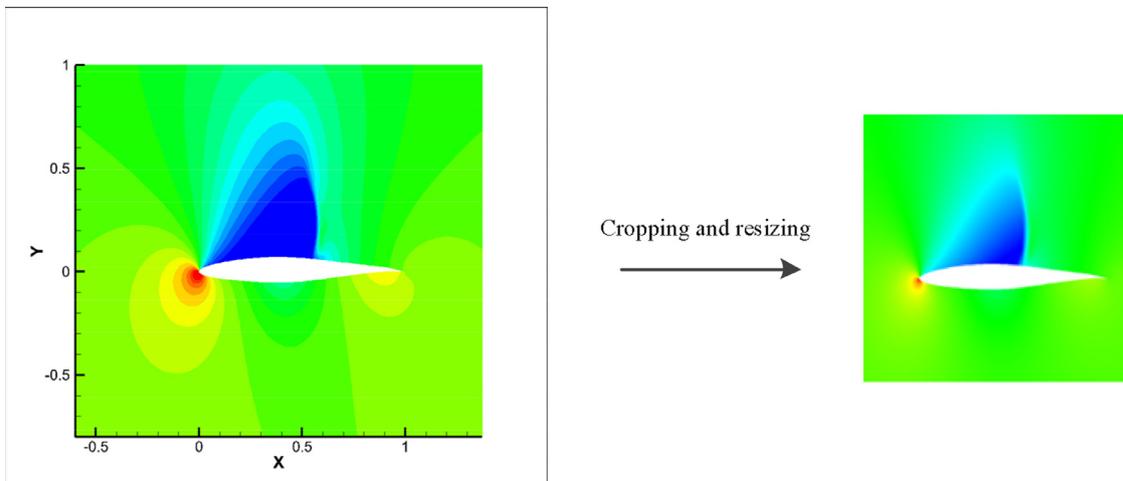


Fig. 12. Example of resizing the pressure profile image.

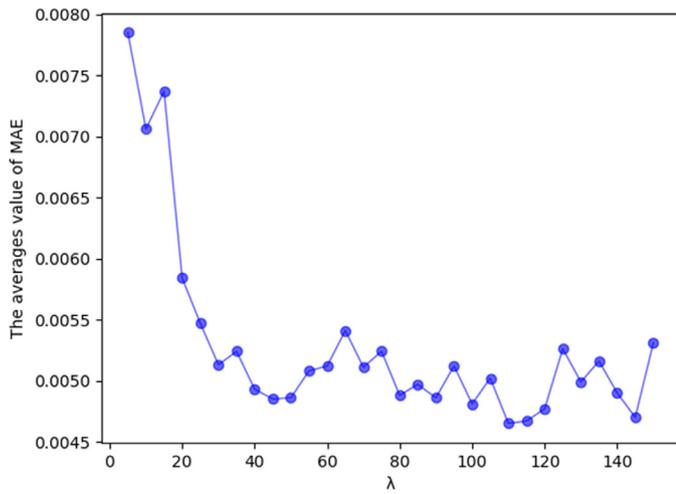


Fig. 13. The average values of the MAE between the generated and real images in training set.

the influences of these parameters usually need to be investigated in order to select the appropriate network with high accuracy. In this paper, we especially examine the choices of loss function, the hyper-parameter λ , the kernel size and the number of CNN layers.

3.2.1. The influence of the hyper-parameter λ

Mean absolute error (MAE) is commonly used to quantify the difference between the predicted and the true values in accuracy validation. The lower value of MAE, the better performance the method achieves. The metric of MAE in our work is set to calculate the average difference of the pixel points between the generated

and real images. The computation of MAE is as follows,

$$MAE = \frac{1}{m} \sum_{i=1}^m |(y_i - \hat{y}_i)| \tag{10}$$

where, y_i and \hat{y}_i represent the i th pixel intensities in the generated and the real image, respectively and m is the number of pixels of the image.

To demonstrate the influence of the hyper-parameters λ in Eq. (7), we apply the MAE metric to quantitatively evaluate the model performance for different settings of λ . The hyper-parameter λ is the scaling factor which is to balance the importance of the two terms, cross-entropy and L1 loss. Thus, it needs to be finely tuned. We choose λ among [5, 150] with interval of 5 and then the MAEs are calculated over the training set to quantitatively evaluate the performance for the 30 different values of λ . The results are shown in Fig. 13, the highest accuracy is reached at $\lambda=110$. However, the values of accuracy do not show significant difference actually when λ is beyond 25 since there is no visual difference among results obtained from these λ . In practice, when we look for the optimal value of λ , we generally test λ increasingly from lower values and we may first find 50 and stop testing since the accuracy is acceptable. For this consideration, we choose the value of 50 for λ in the following studies.

3.2.2. The choice of the loss function

To demonstrate which components of the loss function in Eq. (7) are important to the performance, we run ablation studies to isolate the effect of the L1 term and the adversarial term. The predicted results are shown in Figs. 14 and 15 where two training samples and two test samples are displayed, respectively. For each sample, we show the prediction for three types of loss functions, adversarial+L1, L1 and adversarial. It can be seen from these figures that using L1 or adversarial loss alone gives inaccurate results. L1 loss leads to blurry and incomplete images. Especially, the im-

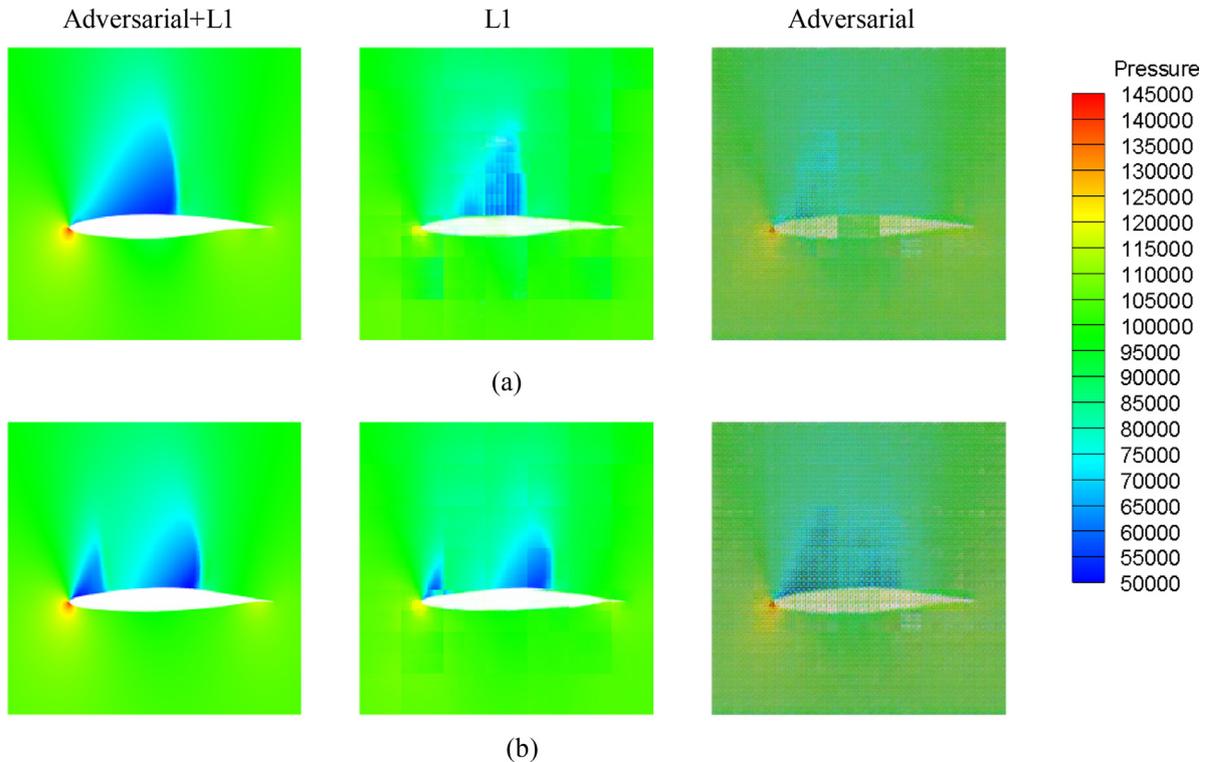


Fig. 14. Comparison of the images of pressure profiles from training set. (a)-row and (b)-row represent the generated images from two different airfoil profiles respectively. The images from the left to right are generated with three different loss functions: adversarial + L1, L1 and adversarial, respectively.

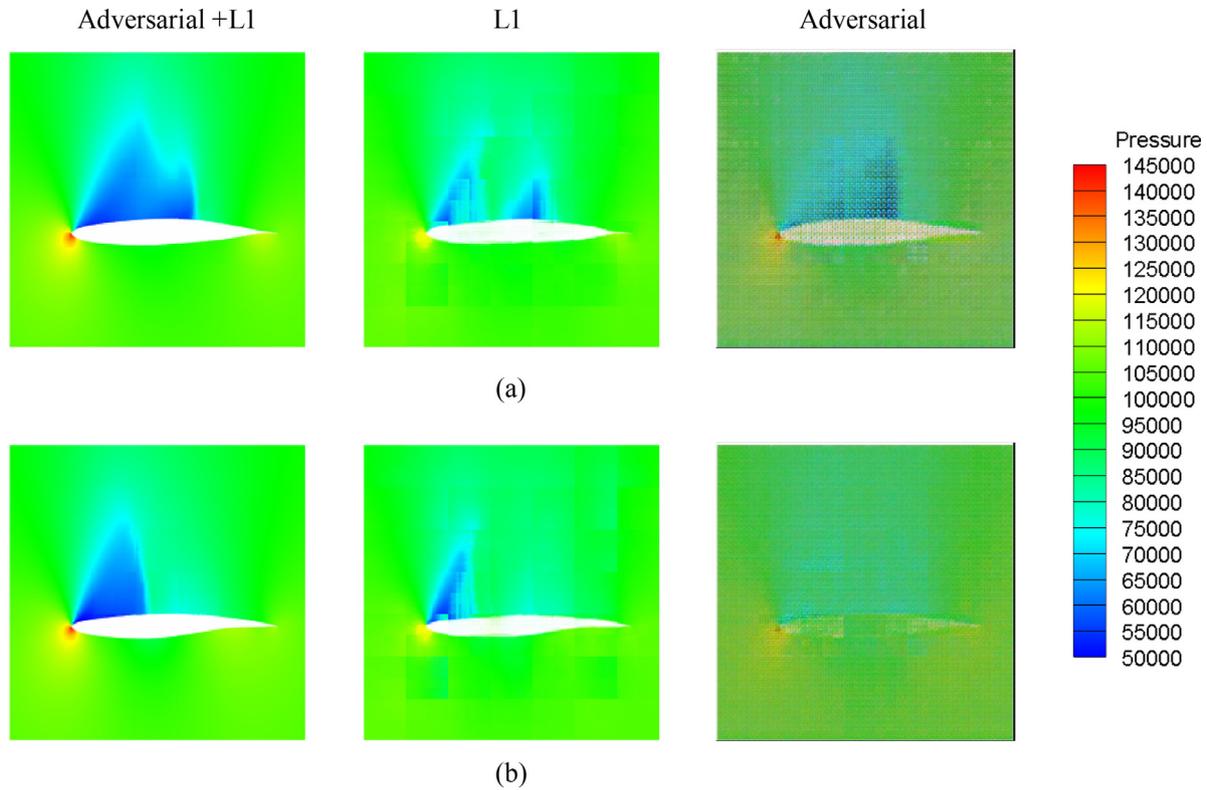


Fig. 15. Comparison of the images of pressure profiles from test set. (a)-row and (b)-row represent the two generated images from different airfoil profiles respectively. The images from the left to right are generated with three different loss functions: adversarial + L1, L1 and adversarial, respectively.

portant features (high pressure) around the leading edge of the airfoil profile get lost in prediction. Results with the adversarial term loss alone (setting $\lambda=0$) depict the whole features, but give much sharper images. In this case, since the loss does not penalize the deviation between the generated and the real images, it only cares that the generated image looks realistic. In our surrogate, the objective function that combines the two terms with the weight λ addresses these artifacts (shown in the leftmost column). Clearly, the loss measures the quality of the generated image in terms of two aspects: firstly, the adversarial term helps to match the distribution between the generated and the real images; secondly, the L1 term encourages the generated image to be close to the real image.

3.2.3. The influence of the elements in networks

In this section, we compare our surrogate with the following four more networks of different structures to demonstrate the generalization of our model.

Case 1: medium (256) consists of 6 layers with medium kernels (4×4 , 5×5 , 6×6) and the maximum number of kernels is 256.

Case 2: medium (1024) consists of 6 layers with medium kernels (4×4 , 5×5 , 6×6) and the maximum number of kernels is 1024.

Case 3: large (256) consists of 5 layers with large kernels (7×7 , 8×8 , 9×9) and the maximum number of kernels is 256.

Case 4: large (1024) consists of 5 layers with large kernels (7×7 , 8×8 , 9×9) and the maximum number of kernels is 1024.

Cases 1 and 2 use medium-sized kernels while Cases 3 and 4 use large-sized kernels. The number of kernels in Cases 1 and 3 are less than those in Cases 2 and 4. The detailed structures of the networks for the four cases are exhibited in appendix.

For a test sample, Fig. 16 shows the real image compared with the images generated from the four networks and our model respectively. As shown in Fig. 16, the number of kernels affects the quality of generation. The first two generated images (in Cases 1 and 3) in upper panel are more blurry than the others. However, the first two generated images (in Cases 2 and 4) in lower panel are more elaborate compared with the images in Cases 1 and 3. This owes to the less parameter of networks in Cases 1 and 3 than that in Cases 2 and 4. So, the more parameters equip in network, the better quality of the generation is. However, with the increase of the number of parameters, more storage space is required. Compared with the alternative four cases, the network in our surrogate, with 7 layers, smaller kernels (2×2 , 3×3) and more kernels (1024), has moderate number of parameters. At the same time, the training time is also a critical factor for model selection. As shown in Table 2, though the images generated in Cases 2 and 4 are smoother than that in our surrogate, the training time in Cases 2 and 4 is 6 and 9 h, respectively, which are longer than that of our surrogate, 3 h. Furthermore, the prediction accuracy is calculated to demonstrate the generalization of the models. The average values of the MAE for the 16 test samples are shown in Fig. 17. Notice that the MAE in Case 3 is much higher than the others and there is no significant difference among Case 2, Case 4 and our surrogate. This indicates that our surrogate achieves great generalization compared to the models with more parameters. Considering the training time, storage space and the accuracy, our model is superior to the four alternatives.

3.2.4. Overall validation of ffsGAN

In this section, we present the overall validation of our surrogate by examining the convergence process and the prediction accuracy in both training and test data sets. With the aim to visualize the iteration process, the convergence curve for model training is shown in Fig. 18. It can be seen that the process converges at

Table 2
The training time for the four cases and our surrogate.

Case	Medium (256)	Medium (1024)	Large (256)	Large (1024)	Our surrogate (small kernel size)
time	3 h	6 h	3.5 h	9 h	3 h

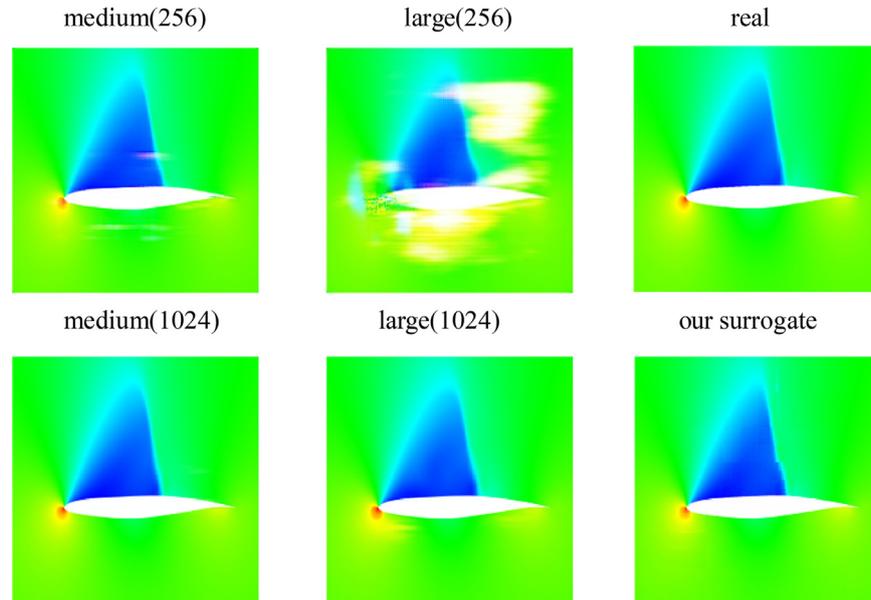


Fig. 16. Comparison of the five generated images of pressure profiles for the structures of networks in the four cases and our surrogate. The real image is exhibited for comparison.

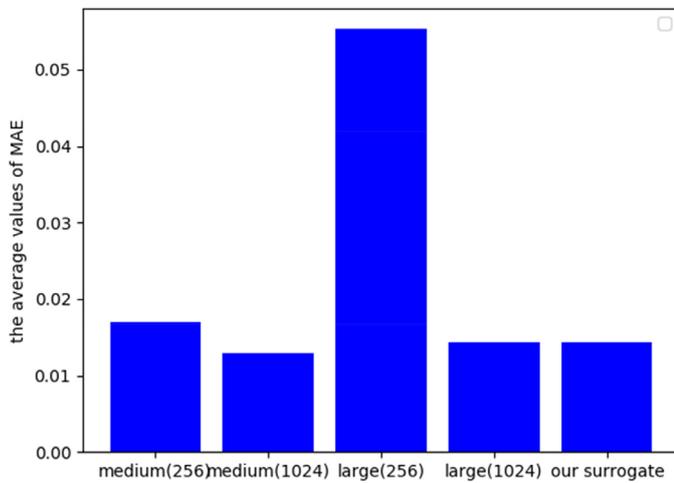


Fig. 17. The average values of MAE for the 16 test samples. Our surrogate is compared with the alternative four cases.

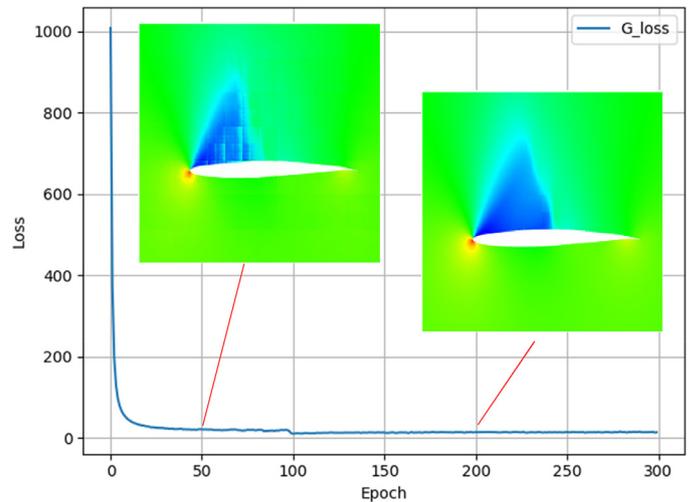


Fig. 18. The convergence curve for the training set.

about epoch 50 and the descending of loss at epoch 100 is due to the decay of learning rate. Although the generated image shown in epoch 50 is blurry, the overall flow structure is almost obtained. At epoch 200, the generated image is closed to the real image and the images are finely tuned in the latter iterations.

We investigate the fitting capacity and the generalization ability of the surrogate in the following part. Four airfoils are randomly selected from the training samples to access the fitting capacity. The generated and the real images are shown in Fig. 19 (a) and (b). As seen in the figure, the generated are close to the real images demonstrating the high fitting capacity of our model. In order to show the generalization of our method, 8 airfoils are randomly sampled from the test set. The validation results are shown

in Fig. 20 (a, b, e and f). It can be seen that our model also achieves high accuracy on the test data even though the generator network have never seen the sampled airfoils in the training process. The magnitude of pressure variables in the generated images has a great consistency with the real images. In addition, the absolute pressure error between the generated and the original images are shown in Figs. 19 (c) and 20 (c, g) for the training and the test samples, respectively. The generated images of the flow fields show a good match with the real images in spite of the slight discrepancies occurring around the shock wave. For an elaborate comparison of the spatial structure between the real and generated, the pressure contours of the images are shown in Fig. 19 (d) and Fig. 20 (d, h) for the training and test samples respectively. It is obvious that

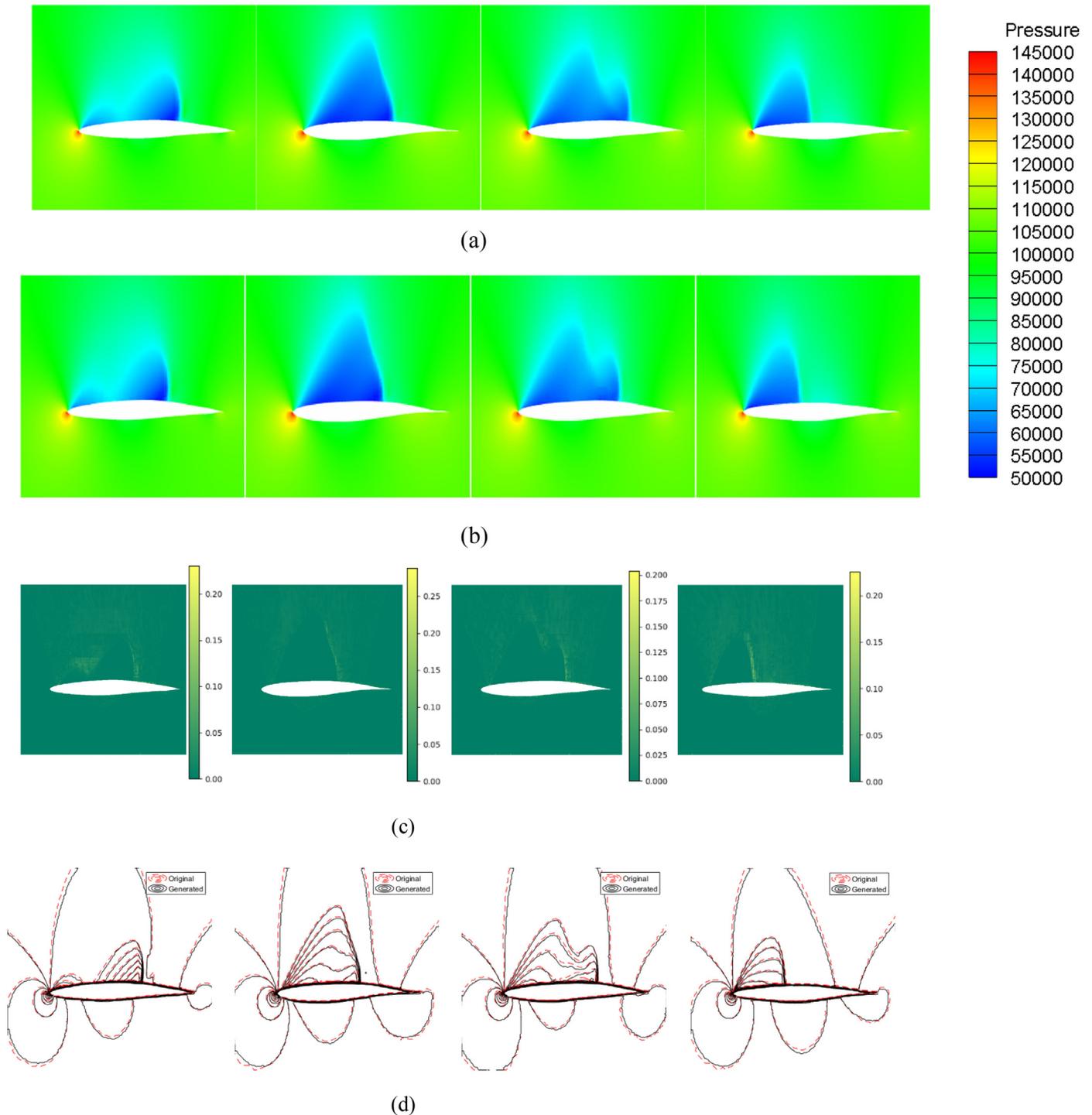


Fig. 19. Comparison of the images of pressure profiles for 4 randomly-chosen airfoils from training set. (a)-row: real images; (b)-row: generated images by ffsGAN; (c)-row: the absolute pressure error between the generated and the original images; (d)-row: the comparison of the pressure contours between the original images and the generated images.

the contours in generated images show great agreement with the real images over the entire domain of the flow field. Especially, our method obtains accurate prediction results over the shock area showing the superior ability to cope with highly non-linear problems.

In addition, the comparison of the pressure coefficient distribution over the airfoil surface between the generated image and the reference is shown in Fig. 21 to verify the accuracy of the predicted flow structure. The pressure coefficient distribution is

drawn according to the predicted image of pressure profile shown in Figs. 19 (a, b) and 20 (a, b). It is clear from the results that the pressure field prediction near the boundary also keeps good accuracy in comparison with the CFD results. More results are demonstrated in Figs. 23 and 24 in appendix. This shows that once we obtain the accurate flow field structure, all the low dimensional aerodynamic quantities, such as the distribution of pressure coefficient, can be easily calculated. Due to the low pixel precision of the flow structure images used in this paper, the roughness of the

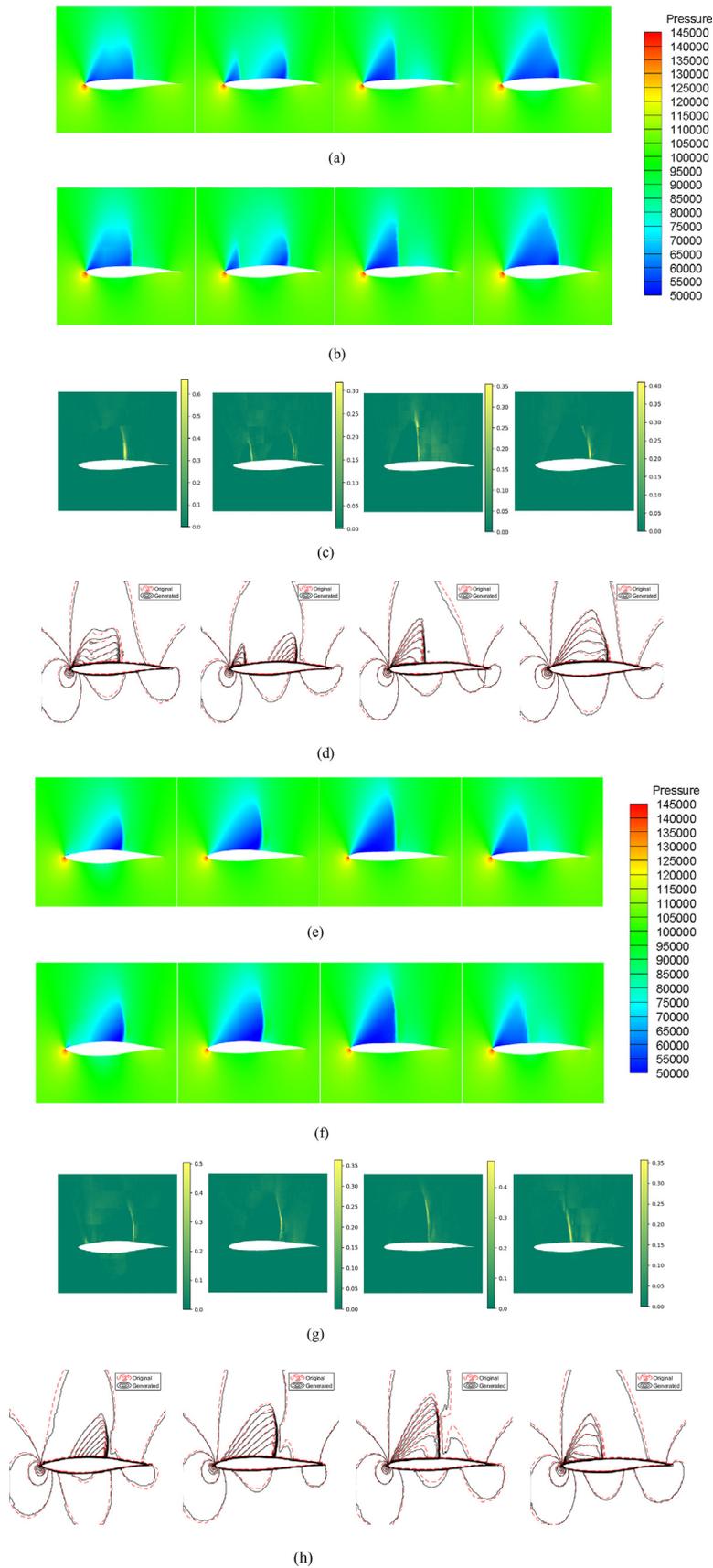


Fig. 20. Comparison of the images of pressure profiles for 8 randomly-chosen airfoils from test set. (a) (e)-row: real images; (b) (f)-row: generated images by ffsGAN; (c) (g)-row: the absolute pressure error between the generated and the original images; (d) (h)-row: the comparison of the pressure contours between the original images and the generated images.

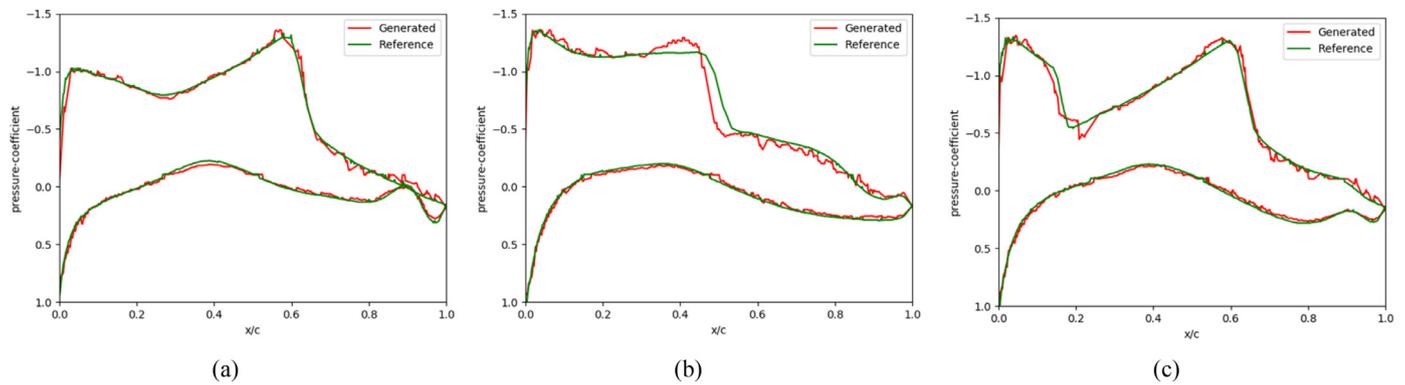


Fig. 21. The comparison of the pressure coefficient distribution over the airfoil surface between the generated image and the reference. (a) shows the comparison for one training sample; (b) and (d) show the comparison for two test samples.

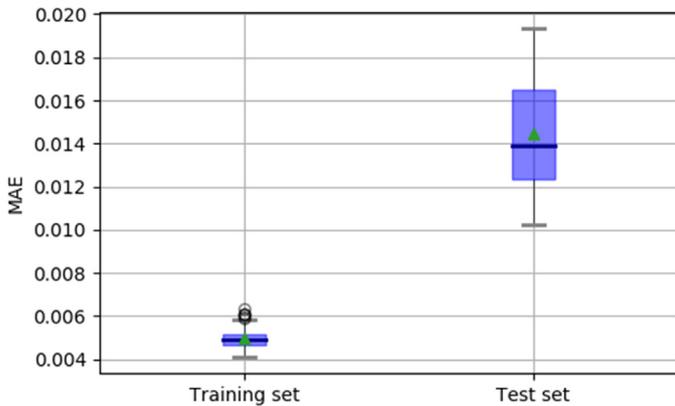


Fig. 22. The boxplots of the MAEs for the training and test sets.

pressure distribution can be observed. For realistic use of the pressure distribution in airfoil optimization, images with higher pixel precision or post-processing of the pressure distribution will be needed. For a further quantitative analysis of the model accuracy, we draw the boxplots of the MAEs for all samples in training and test sets as shown in Fig. 22. The low mean (marked as black line) and median (marked as blue dot) MAE values for the training set demonstrate the superior capacity of model fitting in the training phase. It is not surprising that the MAEs of the test samples are larger than that of the training samples, since these samples have not been seen by the model during the training phase. Although the relatively large MAEs are obtained for test samples, no obvious differences between the generated and real images of the pressure profiles can be visually observed in Fig. 20 (a, b, e and f). Therefore, it can be concluded that ffsGAN is capable of rapidly predicting the precise detailed aerodynamic characteristics under steady flow field over a given space of supercritical airfoils.

3.3. Further analysis

3.3.1. The cost-effectiveness of the proposed model

The low efficiency of obtaining the detailed field flow structures is exempted in our model. Meanwhile, the computation speed boost in several orders of magnitude qualitatively. Different from CFD simulations and wind-tunnel experiments, the process of training the generator and discriminator networks in our model costs a few hours, but it takes only a few seconds to generate the image of detailed flow field for any given supercritical airfoil. In contrast, numerous hours are required to generate precise flow field structures for CFD simulations. Therefore, our work has demonstrated that it is possible to rapidly and accurately eval-

uate the flow field for unseen airfoil by carefully establishing the relationship between the airfoil profile and the elaborate flow field structure using deep learning technique.

3.3.2. Compared to the related works

In some wing design problems, the lift and drag coefficients, and several other finite aerodynamic characteristics are insufficient to denote detailed aerodynamic performance of an airfoil. So, the elaborate flow field structure which is obtained from our surrogate can provide more abundant and complete characteristics of flow field. One potential application of our work is that the predicted flow field structure can be considered in the optimization objective in airfoil optimization. In [24], one of the deep neural networks, convolutional auto-encoder (CAE), was previously adopted to extract the features of flow field structure which was subsequently used to predict the drag coefficient for an airfoil profile in the airfoil optimization design [24]. Even though the flow field structure was considered in the modeling, the optimizer only took the drag parameter into consideration, and other general physical characteristics were ignored. Besides, some other work modeled the relationship between the aerodynamic parameters and the corresponding image of airfoil profile by using convolutional neural network, and obtained high prediction precision [18,20]. It is deserved to mention that additional operations for feature extraction need to be carried out, and this leads to more complexity and computational cost of the model.

Comparing with the above work, all the general physical characteristics are displayed in the flow field image which is the input of the discriminator network of our model. The discriminator is applied to fulfill the feature extraction of the flow field images and realize the dimension reduction. In the process of generation, the parameterized airfoils are fed into the generator network to reconstruct the flow field image. Taking the generator as the surrogate, our method can easily predict the high-dimensional flow field for parameterized airfoils. This enables the surrogate to be applied in airfoil design and optimization where detailed information of flow field is considered and other areas where the evaluation of the whole flow field structure is essential but expensive. For example, we may want to eliminate vortex in some area, for which detailed flow field structure is needed to help detecting vortex. In this study, we used the flow field images with the size of 224×224 and the accuracy of this data has been validated in the previous sections. To use more accurate data, the pixel precision of the images can be further increased.

4. Conclusion

Deep learning models are capable of modeling the highly nonlinear function and have the generalization ability in unseen cases.

In this work, we leveraged this property and first attempted to apply GANs with CNN structure to predict the flow field for supercritical airfoils. The numerical simulation in fluid dynamic was converted into computer vision (CV) problem and was modeled by the proposed deep learning model, ffsGAN. The input of the generator network is the parameterized airfoil profiles. The outputs (the generated images) present the elaborate flow field which includes the entire characteristics of the aerodynamic performance. We sampled 500 modified RAE2822 airfoils and calculated their corresponding images of pressure profiles to train ffsGAN. We investigated the influence of the model parameters including hyper-parameter λ , loss function, filter size and the number of CNN layers. By making appropriate choices of all these parameters, all training and test airfoils were used to demonstrate the overall effectiveness of our model. It can be concluded that ffsGAN is suitable for predicting accurate flow field structure for a given range of supercritical airfoils. It is therefore likely that the expensive CFD simulations and wind-tunnel experiments in some cases can be replaced by a trained generator network and the image of flow field can be precisely evaluated in a few seconds for a given parameterized airfoil profile.

In our experiments, the Mach and Reynolds number and other conditions were fixed. In the future work, we will investigate to model the changeable flow field conditions. Furthermore, since the generated images from our method include more detailed physical characteristics of flow field rather than aerodynamic coefficients or pressure distributions on solid surface, such as the information of vortex shedding and boundary layer interaction with shocks which may be hints of buffeting, we will study the application of ffs-

GAN to the improved airfoil optimization where detailed flow field structure is considered in the optimization objectives.

Declaration of Competing Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

CRediT authorship contribution statement

Haizhou Wu: Conceptualization, Methodology, Software, Validation, Investigation, Writing - original draft, Writing - review & editing, Visualization. **Xuejun Liu:** Conceptualization, Investigation, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Wei An:** Resources, Data curation, Formal analysis, Validation. **Songcan Chen:** Software, Validation, Funding acquisition, Supervision. **Hongqiang Lyu:** Data curation, Writing - review & editing, Funding acquisition, Supervision.

Acknowledgments

This work was supported by the funding of the Key Laboratory of Aerodynamic Noise Control (ANCL20190103), the State Key Laboratory of Aerodynamics (SKLA20180102), the [Aeronautical Science Foundation of China \(2018ZA52002\)](#), the [National Natural Science Foundation of China \(No. 61672281, 61732006\)](#). Besides, we thank to the suggestions of the reviewers and the helps from Prof Haixin Chen of Tsinghua University of China.

Appendix

Tables 3–6.

Table 3
The structures of the network for medium (256).

Generator						Discriminator					
Kernel	Filters	Stride	Padding	BN	Activation	Kernel	Filters	Stride	Padding	BN	Activation
4 × 4	256	2	0	Yes	Relu	6 × 6	16	2	0	Yes	LeakyReLu(0.2)
						6 × 6	16	2	0	Yes	LeakyReLu(0.2)
5 × 5	128	2	0	Yes	Relu	6 × 6	32	2	0	Yes	LeakyReLu(0.2)
5 × 5	128	2	0	Yes	Relu	5 × 5	64	2	0	Yes	LeakyReLu(0.2)
5 × 5	64	2	0	Yes	Relu	5 × 5	64	2	0	Yes	LeakyReLu(0.2)
6 × 6	64	2	0	Yes	Relu	25 × 5	128	2	0	Yes	LeakyReLu(0.2)
6 × 6	3	2	0	No	Tanh	4 × 4	1	2	0	No	Sigmoid

Table 4
The structures of the network for medium (1024).

Generator						Discriminator					
Kernel	Filters	Stride	Padding	BN	Activation	Kernel	Filters	Stride	Padding	BN	Activation
4 × 4	1024	2	0	Yes	Relu	6 × 6	64	2	0	Yes	LeakyReLu(0.2)
						6 × 6	64	2	0	Yes	LeakyReLu(0.2)
5 × 5	512	2	0	Yes	Relu	6 × 6	128	2	0	Yes	LeakyReLu(0.2)
5 × 5	512	2	0	Yes	Relu	5 × 5	256	2	0	Yes	LeakyReLu(0.2)
5 × 5	256	2	0	Yes	Relu	5 × 5	256	2	0	Yes	LeakyReLu(0.2)
6 × 6	256	2	0	Yes	Relu	25 × 5	512	2	0	Yes	LeakyReLu(0.2)
6 × 6	3	2	0	No	Tanh	4 × 4	1	2	0	No	Sigmoid

Table 5
The structures of the network for large (256).

Generator						Discriminator					
Kernel	Filters	Stride	Padding	BN	Activation	Kernel	Filters	Stride	Padding	BN	Activation
9 × 9	256	2	0	Yes	Relu	6 × 6	16	2	0	Yes	LeakyReLu(0.2)
						6 × 6	16	2	0	Yes	LeakyReLu(0.2)
7 × 7	128	2	0	Yes	Relu	6 × 6	32	2	0	Yes	LeakyReLu(0.2)
8 × 8	128	2	0	Yes	Relu	5 × 5	64	2	0	Yes	LeakyReLu(0.2)
7 × 7	64	2	0	Yes	Relu	5 × 5	128	2	0	Yes	LeakyReLu(0.2)
8 × 8	3	2	0	No	Tanh	4 × 4	1	2	0	No	Sigmoid

Table 6
The structures of the network for large (1024).

Generator						Discriminator					
Kernel	Filters	Stride	Padding	BN	Activation	Kernel	Filters	Stride	Padding	BN	Activation
9×9	1024	2	0	Yes	Relu	6×6	64	2	0	Yes	LeakyRelu(0.2)
7×7	512	2	0	Yes	Relu	6×6	64	2	0	Yes	LeakyRelu(0.2)
8×8	512	2	0	Yes	Relu	5×5	256	2	0	Yes	LeakyRelu(0.2)
7×7	256	2	0	Yes	Relu	5×5	512	2	0	Yes	LeakyRelu(0.2)
8×8	3	2	0	No	Tanh	4×4	1	2	0	No	Sigmoid

Figs. 23 and 24.

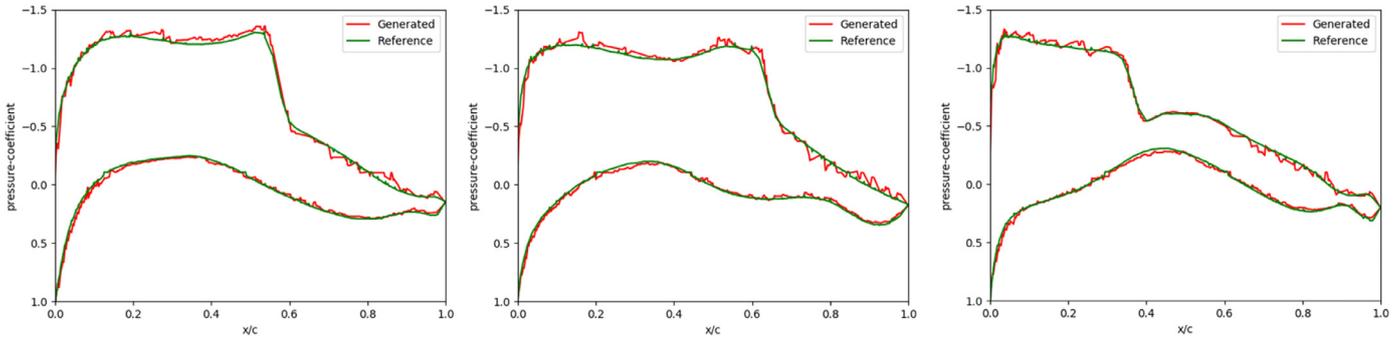


Fig. 23. The comparison of the pressure coefficient distribution over the airfoil surface between the generated image and the reference for three training samples.

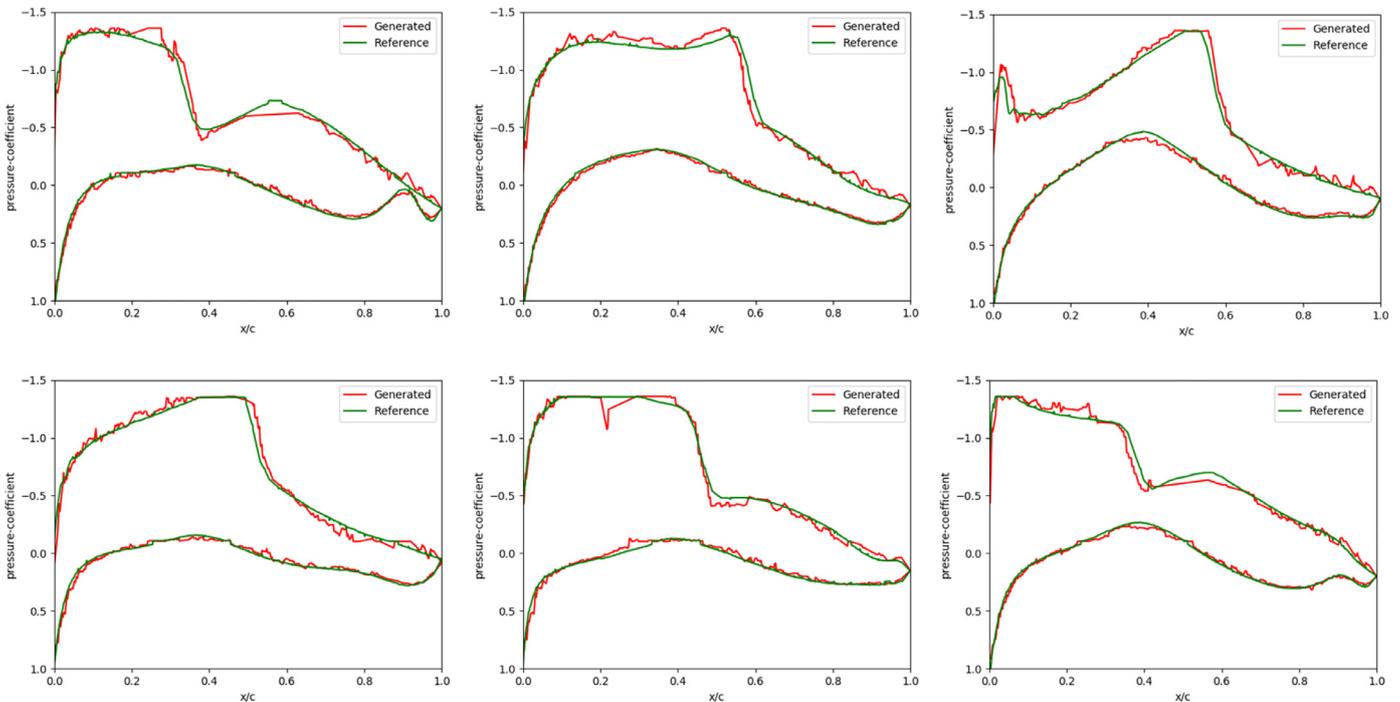


Fig. 24. The comparison of the pressure coefficient distribution over the airfoil surface between the generated image and the reference for six test samples.

References

- [1] Sobieczky H, Seebass AR. Supercritical Airfoil and Wing Design. *Annu Rev Fluid Mech* 1984;16(1):337–63.
- [2] Nakayama A. Characteristics of the Flow around Conventional and Supercritical Airfoils. *J Fluid Mech* 2006;160(160):155–79.
- [3] Ramaswamy MA. Characteristics of a Typical Lifting Symmetric Supercritical Airfoil. *Sadhana* 1987;10(3–4):445–58.
- [4] Roos FW. Surface Pressure and Wake Flow Fluctuations in a Supercritical Airfoil Flow field. *AIAA J* 1975;202(3):666–74.
- [5] Hurley FX, Spaid FW, Roos FW, Stivers LS Jr, Bandettini A. Detailed Transonic Flow Field Measurements about a Supercritical Airfoil Section. NASA TM X-3244, July 1975.
- [6] Alshabu A, Olivier H, Klioutchnikov I. Investigation of Upstream Moving Pressure Waves on a Supercritical Airfoil. *Aerosp Sci Technol* 2006;10(6):465–73.
- [7] March A, Willcox K. Provably Convergent Multifidelity Optimization Algorithm not Requiring High-Fidelity Derivatives. *AIAA J* 2012;50(5):1079–89.
- [8] Hill WJ, Hunter WG. A Review of Response Surface Methodology: a Literature Survey. *Technometrics* 1966;8:571–90.

- [9] Rai MM, Madavan NK. Aerodynamic Design using Neural Network. *AIAA J* 2000;38(1):173–82.
- [10] Hacıoglu A. Fast Evolutionary Algorithm for Airfoil Design via Neural Networks. *AIAA J* 2007;45(9):2196–203.
- [11] Mullur AA, Messac A. Extended Radial Basis Functions: More Flexible and Effective Metamodeling. *AIAA J* 2005;43(6):1306–15.
- [12] Clarke SM, Griebisch JH, Simpson TW. Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses. *J Mech Des* 2005;127:1077–87.
- [13] Han Z, Görtz S. Hierarchical Kriging Model for Variable-Fidelity Surrogate Modeling for structural engineering. *Archives of Computational Methods in Engineering* 2012;50(9):1885–96.
- [14] Laurenceau J, Sagaut P. Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging. *AIAA J* 2008;46(2):498–507.
- [15] Liu X, Zhu Q, Lu H. Modeling Multiresponse Surfaces for Airfoil Design with Multiple-Output-Gaussian-Process Regression. *J Aircr* 2014;51(3):740–7.
- [16] Lee S, Ha J, Zokhirova M, Moon H, Lee J. Background information of deep learning for structural engineering. *Archives of Computational Methods in Engineering* 2017;25(1):121–9. doi:10.1007/s11831-017-9237-0.
- [17] Nathan KJ. Deep Learning in Fluid Dynamics. *J Fluid Mech* 2017;814:4.
- [18] Zhang Y, Sung WJ, Mavris DN. Application of convolutional neural network to predict airfoil lift coefficient. 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference; 2018. p. 1903.
- [19] Miyanawala TP, Jaiman RK. An efficient deep learning technique for the Navier-Stokes equations: Application to unsteady wake flow dynamics. arXiv preprint arXiv:1710.09099, 2017.
- [20] Sekar V, Zhang M, Shu C, Khoo BC. Inverse Design of Airfoil using a Deep Convolutional Neural Network. *AIAA J* 2019. doi:10.2514/1.J057894.
- [21] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Networks. *Adv Neural Inf Process Syst* 2014;3:2672–80.
- [22] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations 2016 (ICLR 2016)* 2016:1511.06434. <http://arxiv.org/licenses/nonexclusive-distrib/1.0/>.
- [23] Mirza M, Osindero S. Conditional Generative Adversarial Nets. arXiv:1411.1784, 2014.
- [24] Deng KW, Chen HX, Zhang YF. Flow Structure Oriented Optimization Aided by Deep Neural Network. In: *Proceedings of the tenth international conference on computational fluid dynamics (ICCFD10)*, Barcelona, Spain; 2018.
- [25] Hicks RM, Henne PA. Wing Design by Numerical Optimization. *J Aircr* 1978;15(7):407–12.
- [26] Cook PH, Firmin MCP, McDonald MA. Aerofoil rae 2822: pressure distributions, and boundary layer and wake measurements. AGARD AR 138. Research and Technology Organisation, Neuilly-Sur-Seine 1979.
- [27] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. Back-propagation Applied to Handwritten Zip Code Recognition. *Neural Comput* 1989;1(4):541–51.
- [28] Arjovsky M, Chintala S, Bottou L. Wasserstein GAN. arXiv preprint arXiv:1701.07875, 2017.
- [29] Zhao J, Mathieu M, Lecun Y. Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126, 2016.
- [30] Mao X, Li Q, Xie H, Y.K.Lau R, Wang Z, Smolley SP. Least Squares Generative Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*. 2017: 2794–2802.
- [31] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC. Improved Training of Wasserstein GANs. *Advances in neural information processing systems*. 2017: 5767–5777.
- [32] Miyato T, Kataoka T, Koyama M, Yoshida Y. Spectral Normalization for Generative Adversarial Networks. arXiv preprint arXiv:1802.05957, 2018.
- [33] Che T, Li Y, Jacob A P, Bengio Y, Li W. Mode Regularized Generative Adversarial Networks. arXiv preprint arXiv:1612.02136, 2016.
- [34] Arjovsky M, Bottou, Léon. Towards Principled Methods for Training Generative Adversarial Networks. arXiv preprint arXiv:1701.04862.
- [35] Zhu JY, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE international conference on computer vision*. 2017: 2223–2232.
- [36] Yu L, Zhang W, Wang J, Yu Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.
- [37] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X. Improved techniques for training GANs. *Advances in neural information processing systems*. 2016: 2234–2242.
- [38] Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H. Generative Adversarial Text to Image Synthesis. arXiv preprint arXiv:1605.05396, 2016.
- [39] Zhang H, Xu T, Li H, Zhang S, Wang X, Huang X, Metaxas DN. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*. 2017: 5907–5915.
- [40] Isola P, Zhu JY, Zhou T, Efros AA. Image-to-Image Translation with Conditional Adversarial Networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 1125–1134.
- [41] Liu MY, Breuel T, Kautz J. Unsupervised Image-to-Image Translation Networks. *Advances in neural information processing systems*. 2017: 700–708.
- [42] Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 4681–4690.
- [43] Eldred M, Dunlavy D. Formulations for Surrogate-Based Optimization with Data Fit, Multifidelity, and Reduced-Order Models. In: *Proceedings of the 11th AIAA/ISSMO multidisciplinary analysis and optimization conference*; 2006. p. 7117.
- [44] Isa NAM, Mamat WMFW. Clustered-Hybrid Multilayer Perceptron Network for Pattern Recognition Application. *Appl Soft Comput* 2011;11(1):1457–66.
- [45] Lung SY. Efficient Text Independent Speaker Recognition with Wavelet Feature Selection Based Multilayered Neural Network using Supervised Learning Algorithm. *Pattern Recognit* 2007;40(12):3616–20.
- [46] Malakooti B, Zhou YQ. Approximating Polynomial Functions by Feedforward Artificial Neural Networks: Capacity Analysis and Design. *Appl Math Comput* 1998;90(1):27–51.
- [47] Cochoki A, Unbehauen R. *Neural Networks for Optimization and Signal Processing*. New York, USA: John Wiley & Sons, Inc.; 1993. ISBN: 0471930105.
- [48] Zijng L, Xuejun L, Xinye C. A New Hybrid Aerodynamic Optimization Framework Based on Differential Evolution and Invasive Weed Optimization. *Chin J Aeronaut* 2018 S1000936118301596.
- [49] Qiu Y, Bai J. Stationary Flow Fields Prediction of Variable Physical Domain Based on Proper Orthogonal Decomposition and Kriging Surrogate Model. *Chin J Aeronaut* 2015;28(1):44–56.
- [50] Bhatnagar S, Afshar Y, Pan S, Duraisamy K, Kaushik S. Prediction of aerodynamic flow fields using convolutional neural networks, 2019: 1–21.
- [51] Sekar V, Jiang Q, Shu C, Khoo BC. Fast Flow Field Prediction over Airfoils using Deep Learning Approach. *Phys Fluids* 2019;31(5):057103.