

RESEARCH

Open Access



High-dimensional aerodynamic data modeling using a machine learning method based on a convolutional neural network

Bo-Wen Zan¹, Zhong-Hua Han^{1*}, Chen-Zhou Xu¹, Ming-Qi Liu¹ and Wen-Zheng Wang²

*Correspondence:
hanzh@nwpu.edu.cn

¹ Institute of Aerodynamic and Multidisciplinary Design Optimization, National Key Laboratory of Science and Technology On Aerodynamic Design and Research, School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China

² Aircraft Swarm Intelligent Sensing and Cooperative Control Key Laboratory of Sichuan Province, School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu 611731, China

Abstract

Modeling high-dimensional aerodynamic data presents a significant challenge in aero-loads prediction, aerodynamic shape optimization, flight control, and simulation. This article develops a machine learning approach based on a convolutional neural network (CNN) to address this problem. A CNN can implicitly distill features underlying the data. The number of parameters to be trained can be significantly reduced because of its local connectivity and parameter-sharing properties, which is favorable for solving high-dimensional problems in which the training cost can be prohibitive. A hypersonic wing similar to the Sanger aerospace plane carrier wing is employed as the test case to demonstrate the CNN-based modeling method. First, the wing is parameterized by the free-form deformation method, and 109 variables incorporating flight status and aerodynamic shape variables are defined as model input. Second, more than 7000 sample points generated by the Latin hypercube sampling method are evaluated by performing computational fluid dynamics simulations using a Reynolds-averaged Navier–Stokes flow solver to obtain an aerodynamic database, and a CNN model is built based on the observed data. Finally, the well-trained CNN model considering both flight status and shape variables is applied to aerodynamic shape optimization to demonstrate its capability to achieve fast optimization at multiple flight statuses.

Keywords: Aerodynamic data modeling, High-dimensional problem, Machine learning, Convolutional neural network, Computational fluid dynamics

1 Introduction

Aircraft aerodynamic data modeling is conducted to establish an explicit or implicit relationship between input variables and output responses through a trained physics-informed or data-driven model. It provides a rapid evaluation and prediction of aerodynamic characteristics for an aircraft instead of conducting flight tests, wind tunnel experiments, or numerical simulations such as computational fluid dynamics (CFD).

Although sufficient samples are invariably required to build an adequate model with respect to the independent variables, such as flight status variables (or aircraft shape variables) and corresponding aerodynamic characteristics, it still can significantly reduce the cost of building a database covering the entire flight envelope compared with these

traditional approaches. The increase in the number of aerodynamic modeling independent variables resulting from the need for more high-dimensional designs leads to increased sample points and training costs to obtain desired model accuracy.

Generally, aerodynamic modeling methods can be divided into physics-informed modeling and data-driven modeling methods. A physics-informed model's physical laws or flow mechanisms determine its structure or terms, and the mapping between the input variables and output aerodynamic functions can be approximated from limited data. Physics-informed models include aerodynamic derivative models [1], linear incremental models [2], linear superposition aerodynamic models [3, 4], triangular series models [5, 6], and reduced-order models (ROM) [7]. These models are physically interpretable because they are closely related to the aerodynamic configuration. However, the accuracy of a physics-informed model is frequently insufficient, especially when the linear hypothesis at small angles of attack or control surface deflections is no longer valid. By contrast, data-driven models can achieve higher accuracy in fitting linear and non-linear relationships between input variables and their corresponding responses when the model parameters are well-tuned. Data-driven models are represented by machine learning (ML) models, such as kriging [8–11], radial basis function (RBF) neural networks [12], support vector machine (SVM) [13], and artificial neural networks (ANNs) [14–20].

Machine learning approaches have recently been increasingly applied to aerodynamic modeling with the rapid development of data science and artificial intelligence techniques [21, 22]. Bouhlel [23] proposed a modified Sobolev training for an artificial neural network (mSANN) to model airfoil aerodynamic force coefficients in the subsonic and transonic regimes. About 42,000 training sample points and 22,000 validation sample points, obtained by solving the Reynolds-averaged Navier–Stokes (RANS) equations with different shapes parameterized by 14 modes and flight statuses incorporating different Mach numbers and angles of attack, were used.

Du [20] adopted a combination of a multi-layer perceptron (MLP), a recurrent neural network (RNN), and mixture of experts (MoE) to predict airfoil lift and drag coefficients for various shapes defined by 26 B-spline curve variables and two flight status variables. Compared with airfoil aerodynamic modeling, more design variables are required for wings or aircraft. Secco [24] used an MLP to predict the lift and drag coefficients of a wing–fuselage aircraft configuration with different wing planar variables, shape variables in three airfoil profiles (10 variables describe each airfoil profile), and flight statuses, totaling 40 input variables.

Barnhart [25] proposed several ML methods to predict the lift and pitching moment coefficient for a blown wing configuration with 20 design variables. Karali [26] used an MLP model trained by 94,500 samples that can predict the aerodynamic characteristics of unmanned aerial vehicle (UAV) configurations with 22 input variables (21 geometric variables and the angle of attack). Li [27] chose 60 design variables, including the Mach number, altitude, angle of attack, seven twist angles, and 50 wing modes. The model was trained by 135,108 samples and further verified in multiple single-point, multi-point, and multi-objective wing design optimization problems.

In terms of combining the meta-modeling of CFD and ML techniques, most of the existing interdisciplinary work employs the learning architecture belonging to the

multi-layer perceptron category. The trade-off between the size and learning capability of standard MLP is highly quantified, with more complex problems requiring larger scale data and more complex network structures, yielding wasteful connectivity and a high tendency to overfitting. It is noted that no more than 60 independent input variables are introduced into existing aerodynamic modeling using an MLP, limiting more refined designs.

Convolutional neural networks have rapidly replaced standard MLP techniques in many challenging ML tasks (e.g., image recognition [28, 29]) because of their local connectivity and parameter-sharing scheme. Aerodynamic modeling research using airfoil images as input shows that it is advantageous for high-dimensional aerodynamic modeling. Thuerey [30] used a CNN to predict the airfoil flow fields for different Reynolds numbers ranging from 5×10^5 to 5×10^6 and angles of attack ranging from -22.5° to 22.5° , comparing their results with those calculated from the RANS equations. The prediction error of pressure and velocity contours was less than 3% when the number of training sample points was 12,800.

Zhang [19] proposed a CNN-based prediction method for airfoil lift coefficients for various shapes at multiple free-stream Mach numbers, Reynolds numbers, and angles of attack. After data augmentation, about 80,000 sample points of the airfoil coordinates were fed to a CNN instead of shape variables from parameterization. Yu [31] proposed a feature-enhanced image approach to perform aerodynamic modeling of an SC1095 airfoil with a CNN trained with 11,550 pairs of normal input/output training data. Chen [32] also adopted a graphical prediction method for multiple airfoil aerodynamic coefficients based on CNN trained by 3360 samples and tested on 840 samples. CNNs have also been used to predict the aerodynamic characteristics of iced airfoils. He [33] proposed a prediction method for the aerodynamic characteristics of iced airfoils based on a CNN, using 11,200 training samples to realize rapid prediction from ice images to aerodynamic characteristics with prediction errors below 8%. The relevant aerodynamic modeling studies using ML-based methods for aerodynamic coefficient and flow field prediction are listed in Table 1.

It is noted that most aerodynamic modeling methods using CNNs adopt the idea of using airfoil graphics as input, which is limited to processing two-dimensional structured

Table 1 Statistics of some related studies

Reference	Aerodynamic shape	Number of independent variables	Model
Bouhlel [23]	Airfoil	16	mSANN
Du [20]	Airfoil	28	MLP
Secco [24]	Aircraft	40	MLP
Barnhart [25]	Wing	20	MLP
Karali [26]	Aircraft	22	MLP
Li [27]	Wing	60	MLP
Thuerey [30]	Airfoil	Pixels of airfoil image	CNN
Zhang [19]	Airfoil		CNN
Yu [31]	Airfoil		CNN
Chen [32]	Airfoil		CNN
He [33]	Airfoil		CNN

image data. For the three-dimensional aerodynamic shape, there is no research on modeling considering more than 100 parameterized independent variables, motivating the research described in this article.

The purpose of this study is to develop a CNN-based method for high-dimensional aerodynamic modeling to alleviate the “curse of dimensionality” [34] and apply it to surrogate-based aerodynamic shape optimization [35–37]. The convolution operation implicitly extracts feature information underlying the aerodynamic data and slightly squeezes the tensor’s dimensionality. Usually, pooling layers are added after the convolutional layers to further reduce the number of neural network parameters (weights and biases) to be trained. With the assistance of local connectivity and parameter sharing, CNNs have better resistance to overfitting.

The remainder of this article is organized as follows. Section 2 presents a detailed introduction to multi-layer perceptrons and convolutional neural networks. Then, the CNN-based aerodynamic data modeling process is described in Section 3. In Section 4, a well-trained CNN is built by investigating the influence of the hyperparameters and applying it to fast aerodynamic shape optimization for a wing. Concluding remarks are provided in Section 5.

2 Fundamentals of convolutional neural networks

A convolutional neural network (CNN) is a supervised deep learning algorithm developed based on the multi-layer perceptron (MLP), and its basic theory is outlined in this section.

2.1 Multi-layer perceptron

The MLP is a machine learning model known as a fully-connected feedforward neural network. The typical MLP, shown in Fig. 1, receives an input (a single vector) and transforms it through a series of hidden layers. Each hidden layer consists of a set of neurons, where each neuron is fully connected to all neurons in the previous layer and where neurons in a single layer function independently and do not share any connections.

For the forward propagation of an MLP, the output of the k^{th} layer, $\mathbf{H}^{(k)}$, can be expressed as follows:

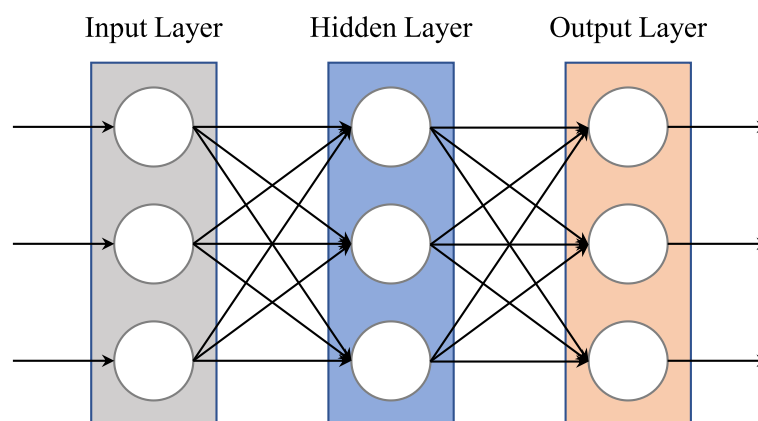


Fig. 1 Typical multi-layer perceptron

$$\begin{aligned}
 h_i^{(k)} &= \sum_{j=1}^{m_{k-1}} w_{i,j}^{(k)} x_j^{(k-1)} + b_i^{(k)}, \quad (1 \leq i \leq m_k), \\
 \mathbf{H}^{(k)} &= [h_1^{(k)}, h_2^{(k)}, \dots, h_{m_k}^{(k)}]^T = \mathbf{W}^{(k)} \mathbf{X}^{(k-1)} + \mathbf{b}^{(k)},
 \end{aligned}$$

where $w_{i,j}^{(k)}$ is a scalar of weights to be determined between the i^{th} neuron on the k^{th} layer and the j^{th} neuron on the previous layer, $x_j^{(k-1)}$ is the input of this layer, i.e., the output of the j^{th} neuron on the $k - 1$ layer, and $b_i^{(k)}$ denotes a bias scalar of the i^{th} neuron on the k^{th} layer.

2.2 Convolutional neural network

A CNN is a type of ANN specifically designed for large-scale structured data such as images. Its structure makes the implementation more efficient and significantly reduces the number of parameters in the network through convolution and pooling operations [38, 39]. The CNN architecture mainly consists of convolutional, pooling, and fully-connected layers. A schematic of a typical CNN architecture for handwritten digit recognition is shown in Fig. 2.

Unlike a fully-connected neural network, the number of a CNN's free parameters describing their shared weights does not depend on the input dimensionality, which avoids the pressure caused by the surge of high-dimensional input parameters in a neural network. The following section focuses on the convolutional and pooling layers. The structure of the fully-connected layer is consistent with that of an MLP mentioned above.

Inspired by the conventional two-dimensional CNN, we propose a modified structure (Fig. 3) that takes the one-dimensional tensor for parameterized variables. These variables taken as model inputs include shape variables (defined by the free-form deformation method, class-function shape-function transformation, and related methods), flight status variables (such as Mach number, angle of attack, Reynolds number, and flight altitude), and wing control surface deflection variables. The outputs are the predicted aerodynamic characteristics such as, \hat{C}_L , \hat{C}_D and \hat{C}_M .

2.2.1 Convolutional layer

The convolution layer processes the input data using a convolution filter and distills the local and global information. If two-dimensional input data are taken as the input of a

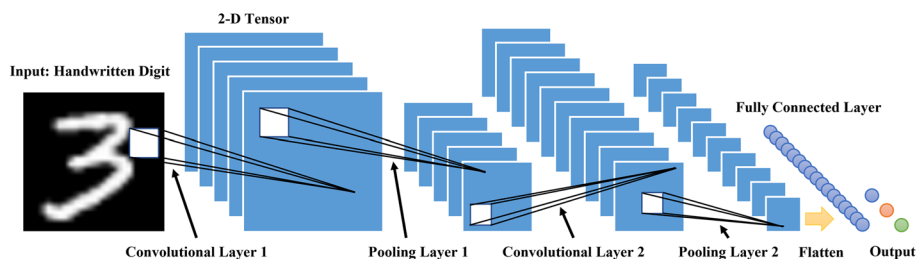


Fig. 2 Conventional CNN architecture applied to a handwritten digit recognition task

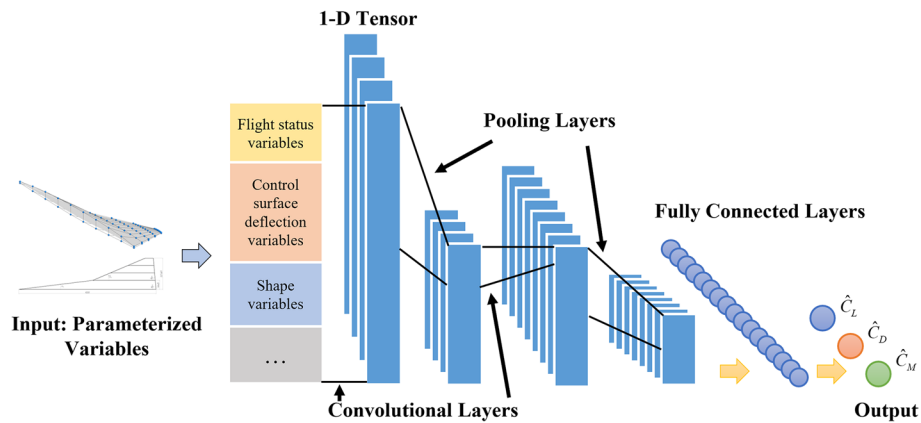


Fig. 3 Modified CNN architecture for aerodynamic data modeling

convolution layer (I with coordinates (m,n)), the convolution filter is a two-dimensional matrix (K) , and the output data is a two-dimensional matrix S with coordinates (i,j) . Therefore, the convolution process can be expressed by the following formula:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i - m,j - n). \tag{2}$$

Unlike the full connectivity of an MLP, only the dynamic connection between the convolutional filters and the neurons covered by the receptive field is established, significantly reducing the number of parameters to be trained in the CNN and contributing to the alleviation of overfitting.

2.2.2 Pooling layer

It is common to periodically insert a pooling layer between successive convolutional layers in a CNN architecture. A pooling layer’s function is to progressively reduce the representation’s spatial size to reduce the number of parameters and computations in the network and control overfitting [40, 41]. Common pooling operations are maximum pooling and average pooling. The pooling layer keeps the maximum or average value in the pooling filter’s receptive field and transfers it to the next layer while discarding the other values and moving the filter with a given stride to the next local region to perform the same operation.

2.2.3 Fully-connected layer

The last pooling layer is usually followed by several fully-connected layers whose structure is the same as the MLP. The depth and width of the fully-connected layers are defined according to the problem’s complexity and the data size.

3 CNN-based aerodynamic data modeling method

The steps of the aerodynamic data modeling method using CNN, shown in Fig. 4, can be described as follows:

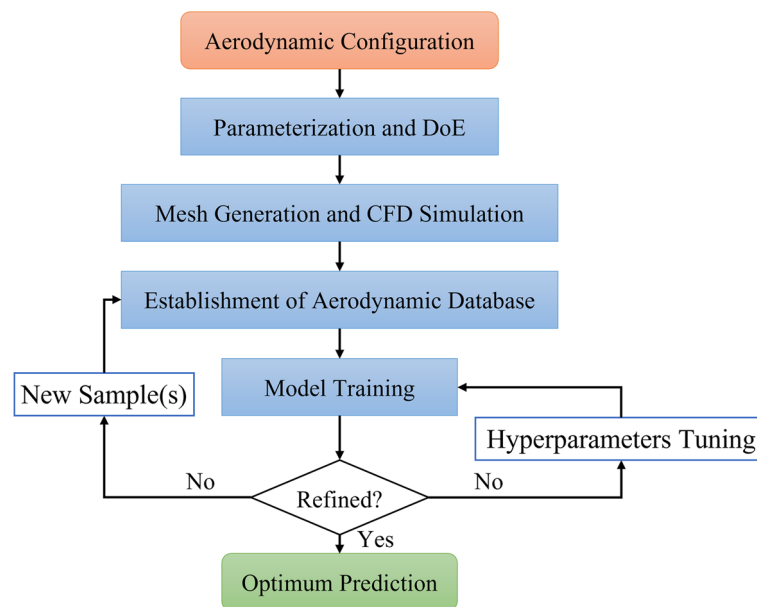


Fig. 4 Aerodynamic data modeling flowchart

- a) Parameterization and design of experiments (DoE): Specify the parameter space by defining the input variables and their range. Generate samples based on DoE theory.
- b) Mesh generation and CFD simulation: Set up the flow field computational block and meshing. Obtain the aerodynamic data at the sample points by conducting CFD simulations.
- c) Establishment of aerodynamic database preparation: Organize the design space and its corresponding aerodynamic characteristics into a database.
- d) Model training: Update model parameters using an optimization algorithm iteratively.
- e) Optimum prediction: Output the optimum prediction if achieving the required accuracy. Otherwise, refine the model through sample augmentation and hyperparameter adjustment.

Sufficient samples are required to build a data-driven model with reasonable accuracy. To obtain enough samples to train the model, we use CFD simulations to compute the aerodynamic force coefficients, such as lift coefficient, drag coefficient, and pitching moment coefficient, corresponding to different flight statuses (with a given range for the free-stream Mach number and angle of attack), and different shapes described by the wing planar variables and profile variables.

The Sanger aerospace plane carrier wing (the aircraft's configuration is shown in Fig. 7) is employed as a test case to validate the prediction capability of the CNN model. The following sections detail the five procedures for obtaining the aerodynamic data and building the CNN.

3.1 Validation of RANS flow solver

The flow solver must be validated before using the CFD simulation to obtain aerodynamic data. The RANS solver is validated by simulating the hypersonic flow over the

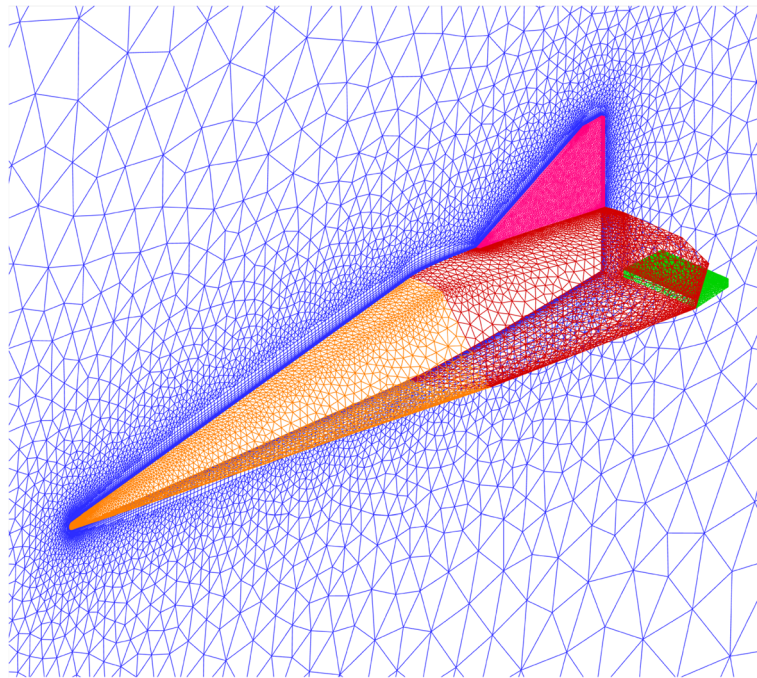


Fig. 5 Sketch of the computational mesh for the FDL-5A configuration

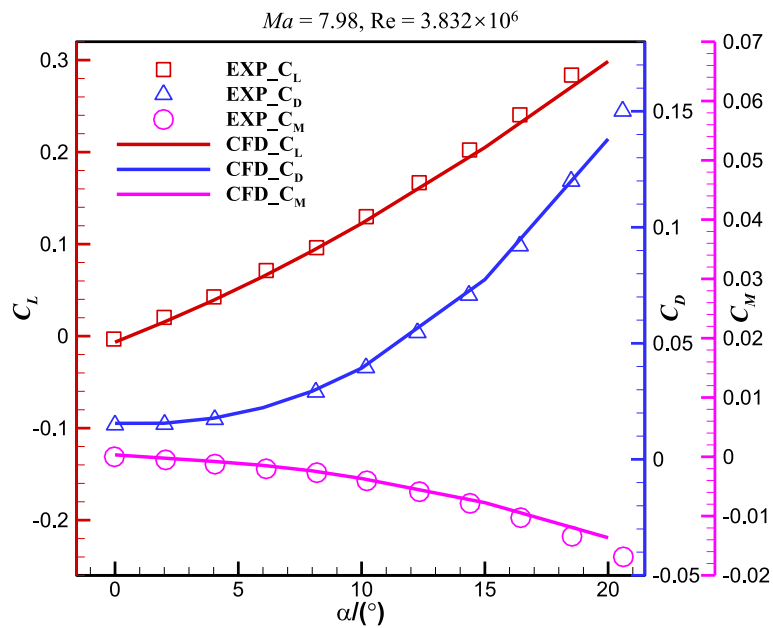


Fig. 6 Comparison of computed force coefficients with experimental data for the FDL-5A configuration

FDL-5A configuration. An AUSM (advection upstream splitting method) scheme is used for the spatial discretization, and a $k-\omega$ SST turbulence model is adopted for turbulence closure. An image showing the unstructured computational mesh is provided in Fig. 5, and the number of computational cells in the mesh is 0.48 million.

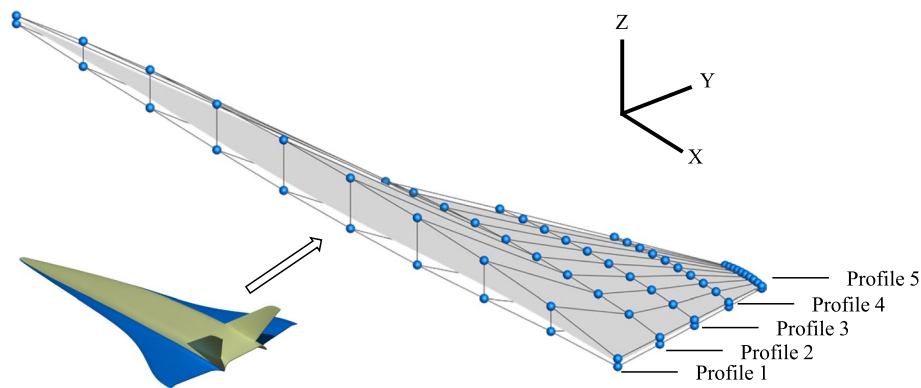


Fig. 7 Schematics of FFD control points in five wing profiles of the Sanger aerospace plane carrier wing

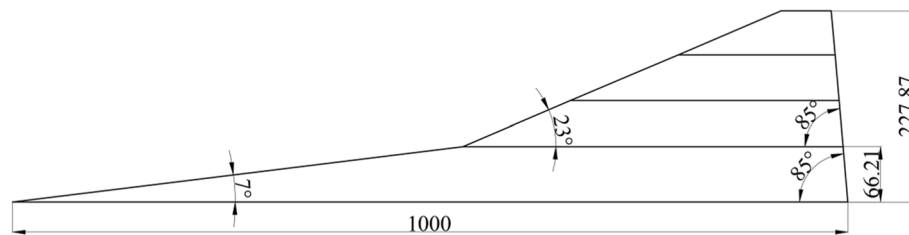


Fig. 8 Schematic of parameterization for the planar shape of the Sanger aerospace plane carrier wing

The hypersonic flow over the FDL-5A is simulated at $Ma = 7.98$, $Re = 3.832 \times 10^6$. Figure 6 compares the computed force coefficients with the corresponding experimental data. The lift, drag, and pitch moment coefficients accurately match the experimental data at different angles of attack. Although the calculated Mach number for the flow solver verified here is 7.98, above that of data for aerodynamic modeling, they are both in the hypersonic regime. Consequently, it makes sense to validate the flow solver.

3.2 Parameterization

The wing is parameterized into 100 control point variables for five profiles at spanwise locations by the free-form deformation (FFD) method and planar shape variables for seven wing configuration variables. As shown in Fig. 7, five wing profiles are parameterized by the two-dimensional FFD method. New wings are obtained by independently perturbing the planar shape variables and the FFD control point variables, and the five perturbed wing profiles are assembled on the corresponding spanwise locations. Thus, the FFD control point variables are decoupled from the planar shape variables.

Figure 8 shows the seven design variables used to parameterize the planar shape of the wing. These design variables are the root chord, leading-edge sweep angle of the inner wing segment, leading-edge sweep angle of the outer wing segment, trailing-edge sweep angle of the inner wing segment, trailing-edge sweep angle of the outer wing segment, span of the inner wing segment, and wingspan.

In total, there are 107 variables describing this configuration and two flight status variables (free-stream Mach number and angle of attack). The boundaries of these variable values are presented in Table 2. In particular, the 100 wing profile variables (Index ϵ

Table 2 Upper and lower boundaries of the input variables for the CNN

Type	Variable	Index	Lower boundary	Upper boundary
Flight status	Angle of attack	1	0°	5°
	Mach number	2	5	6
Wing planar	Root chord	3	900	1100
	Span of the inner wing segment	4	59.5854	72.8266
	Wing span	5	205.083	250.657
	Leading edge sweep angle of the inner wing segment	6	80.9°	85.1°
	Leading edge sweep angle of the outer wing segment	7	18.4°	73.7°
	Trailing edge sweep angle of the inner wing segment	8	4.5°	5.5°
	Trailing edge sweep angle of the outer wing segment	9	4.5°	5.5°
Wing profile	FFD control points in profile 1	10–29	Z-coordinates varying from -10% to +10% on the baseline	
	FFD control points in profile 2	30–49		
	FFD control points in profile 3	50–69		
	FFD control points in profile 4	70–89		
	FFD control points in profile 5	90–109		

[10,109]) represent the Z-coordinates of the FFD control points, varying from -10% to 10% on the benchmark of the wing.

3.3 Design of experiments

Latin hypercube sampling (LHS) is used as the design of experiments (DoE) method to establish the distribution of input variables. LHS is a method of approximate random sampling from a multivariate parameter distribution belonging to hierarchical sampling technology and is often used in DoE. Samples x_j^i obtained using the LHS method can be expressed as follows:

$$x_j^i = \frac{\pi_j^i + U_j^i}{N}, \quad 1 \leq j \leq d, 1 \leq i \leq N, \quad (3)$$

where i denotes the i^{th} sample, j denotes the j^{th} design variable, U denotes a random number in $[0,1]$, and π_j denotes a random permutation in $\{0, 1, \dots, N-1\}$. An example in which the LHS method selects 20 sample points in a DoE problem of two-dimensional input is shown in Fig. 9.

3.4 Mesh generation and CFD simulation

This work uses the mesh reconstruction method to generate meshes for different shapes. A mesh independence study [42, 43] is performed to ensure sample data accuracy. As shown in Figs. 10 and 11, three meshes are generated: coarse, medium, and fine meshes having 0.41 million, 1.34 million, and 2.86 million cells, respectively. The results for the three meshes are in good agreement.

The variation in force coefficients with mesh size is shown in Fig. 12. The lift coefficient, drag coefficient and moment coefficient calculated by the coarse mesh differ from those calculated by the fine mesh by 0.00016, 1 count (1 count = 0.0001) and 0.000012, respectively.

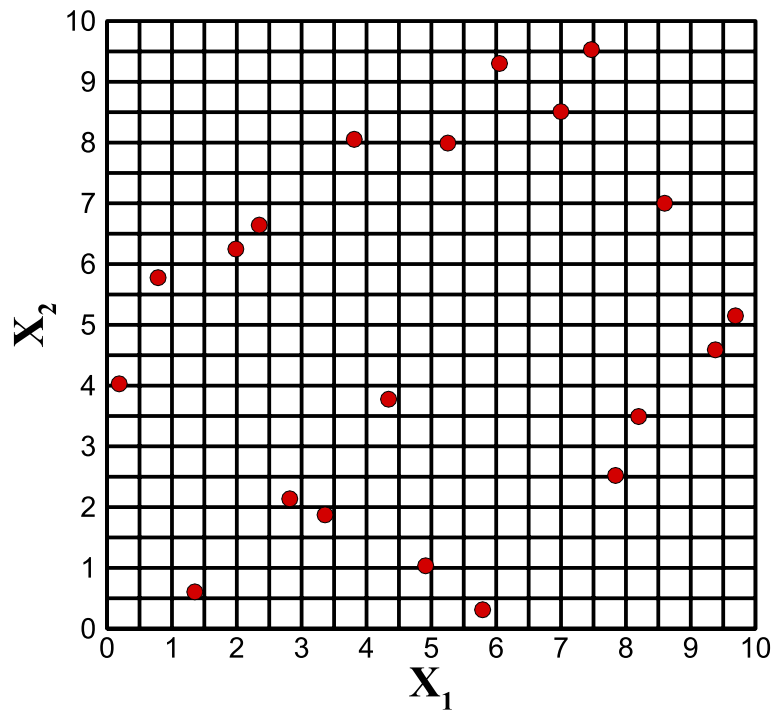


Fig. 9 Schematic of 20 sample points selected by Latin hypercube sampling

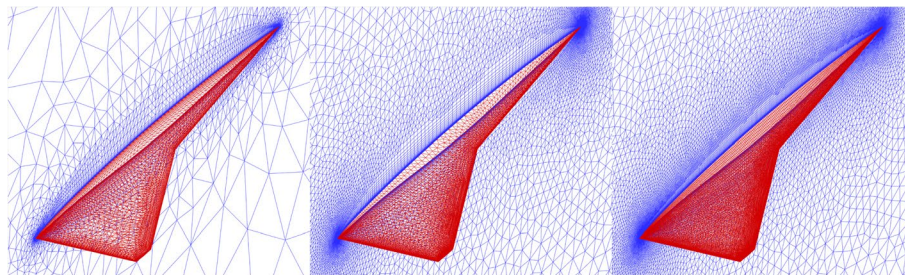


Fig. 10 The surfaces of the coarse (left), medium (middle), and fine (right) meshes

The coarse mesh is chosen (Fig. 13) to build the aerodynamic dataset at a minimal computational cost. The same topology and mesh parameters are used for all sample points to keep the physical problem consistent. CFD simulations are conducted using the RANS equations, and a two-equation $k-\omega$ SST turbulence model is adopted for turbulence closure. An aerodynamic dataset consisting of 7431 sample points is constructed.

3.5 CNN Training

The CNN network weights and biases are optimized using the backpropagation algorithm [44]. For the regression problem, the mean square error (MSE) loss function of the model is expressed as:

$$L = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \tag{4}$$

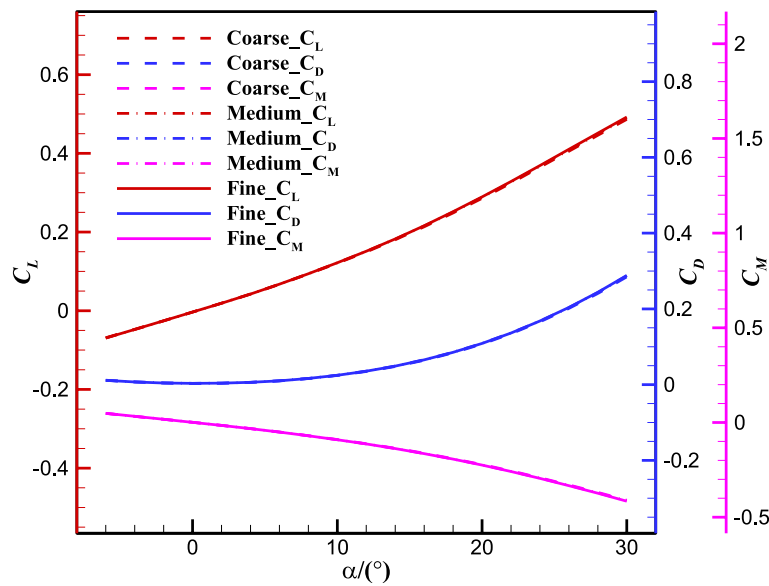


Fig. 11 Comparison of aerodynamic force coefficients calculated using the three meshes

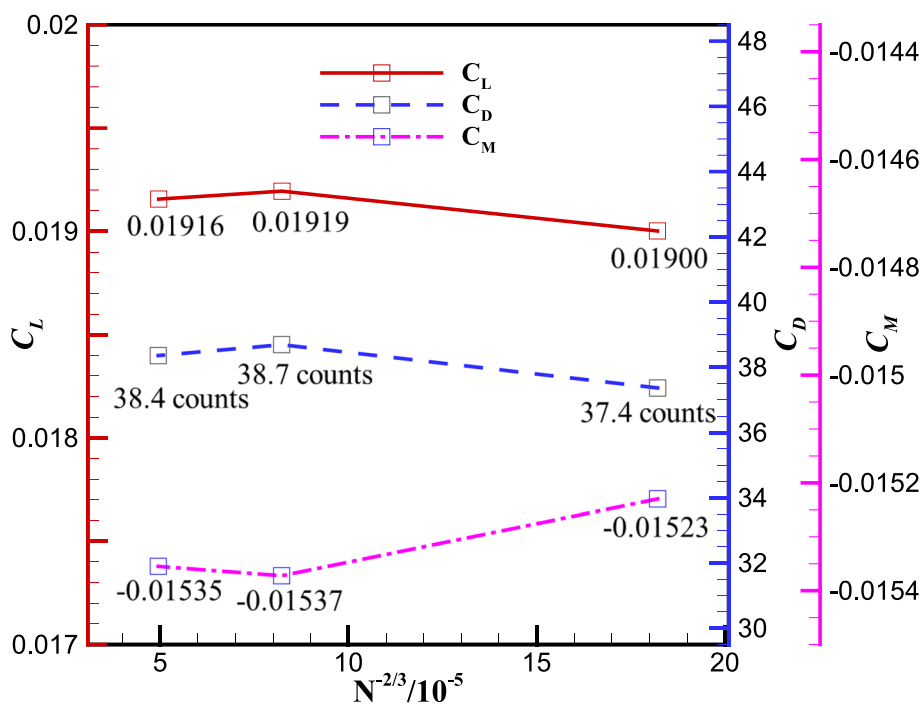


Fig. 12 Variation of the force coefficients with the mesh size

where N denotes the number of sample points in the training dataset, y_i is the numerical simulation value calculated by the CFD simulation, and \hat{y}_i denotes the prediction value.

The dataset is divided into training and test datasets in a 19:1 ratio. The reason why the validation dataset is not used is that this article does not use techniques such as

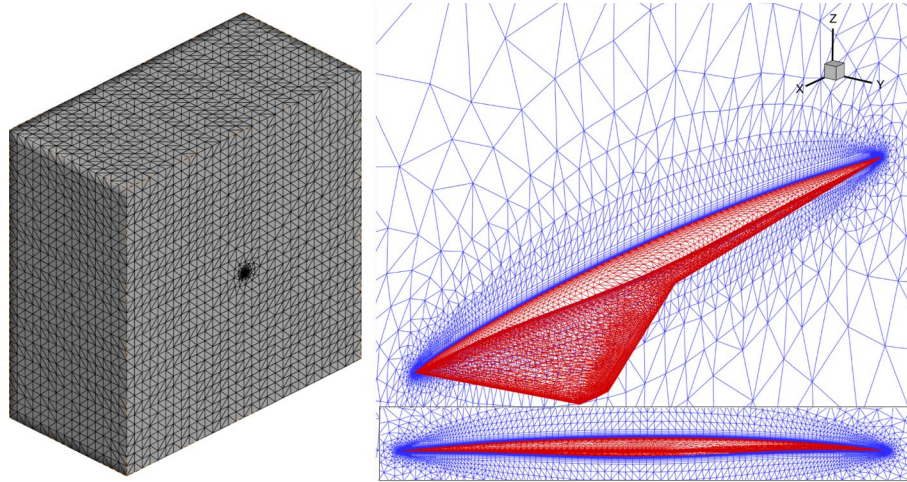


Fig. 13 Sketch of the computational mesh for the Sanger aerospace plane carrier wing

Table 3 CNN architecture with its layers and parameters

Layer type	CNN-1	CNN-2	CNN-3	CNN-4
Input	109×1	109×1	109×1	109×1
First convolution	$16, 3 \times 1, 2 \times 1$	$16, 3 \times 1, 2 \times 2$	$16, 3 \times 1, 2 \times 1$	$16, 3 \times 1, 2 \times 1$
Second convolution	-	$32, 3 \times 1, 2 \times 2$	$32, 3 \times 1, 2 \times 1$	$32, 3 \times 1, 2 \times 1$
Third convolution	-	-	$64, 3 \times 1, 2 \times 1$	$64, 3 \times 1, 2 \times 1$
Fourth convolution	-	-	-	$128, 3 \times 1, 2 \times 1$
Fully-connected	3×128	3×128	3×128	3×128
Output	3×1	3×1	3×1	3×1

early stopping and dynamic learning rate adjustment. Hence, there are 7059 wing sample points used to tune the parameters of the network and 372 sample points used to test the predicted accuracy of the CNN. The Adam [45] method optimizes the model to approximate the input data's underlying mapping. The initial learning rate is set at 0.0001, and the initial batch size is set to 128. The training procedure is performed on a GPU (NVIDIA RTX 3080).

4 Results and discussion

4.1 Influence of CNN hyperparameters

The hyperparameters directly affect the training and predictive performance of neural networks. In this section, we investigate several CNN hyperparameters. Better hyperparameters can be found by observing the convergence performance of the loss function on the training and test datasets.

4.1.1 Number of convolutional layers

Four CNN models (CNN-1, CNN-2, CNN-3, and CNN-4) were set up, having 1, 2, 3, and 4 convolutional layers, respectively. The network structures and parameters are shown in Table 3. In the first convolutional layer of CNN-1, 16 indicates the number

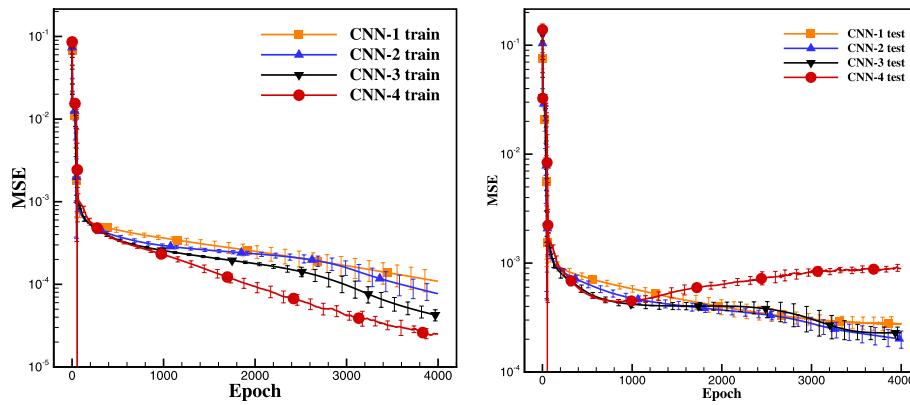


Fig. 14 MSE convergence histories of CNN-1, CNN-2, CNN-3, and CNN-4 in training dataset (left) and test dataset (right)

Table 4 Variation of CNN-2 with the number of convolutional filters

Layer type	CNN-2-NCF8	CNN-2-NCF16	CNN-2-NCF32	CNN-2-NCF64
Input	109×1	109×1	109×1	109×1
First convolution	8, 3×1, 2×1	16, 5×1, 2×1	32, 6×1, 2×1	64, 8×1, 2×1
Second convolution	16, 3×1, 2×1	32, 5×1, 2×1	64, 6×1, 2×1	128, 8×1, 2×1
Output	3×1	3×1	3×1	3×1

of filters, 3×1 indicates the convolutional filter size, and 2×1 indicates the pooling filter size. In the fully-connected layer of CNN-1, 3×128 indicates 3 layers with 128 neurons.

Without loss of generality, each CNN structure is trained 10 times to exclude the influence of random processes on the training results. The average values and standard deviation of the MSE are taken for comparison. Training is implemented for all four architectures, and the MSE convergences of the CNNs are shown in Fig. 14. The training histories of the four CNN structures with a learning rate of 0.00001 are given here. Although CNN-4 converges rapidly and performs better on the training dataset, significant overfitting occurs, i.e., the MSE on the test dataset is larger, while the MSE on the test dataset increases as the MSE on the training set decreases. Because it has the minimum MSE, CNN-2 is selected as the benchmark architecture for further parameter studies.

4.1.2 Number of convolutional filters

The influence of the number of filters is investigated by increasing them from 8 (NCF8) to 64 (NCF64) for CNN-2’s first convolutional layer and from 16 to 128 for its second convolutional layer (see Table 4). Figure 15 shows the effect of the number of filters on the MSE convergence of the model. An increase in the number of filters significantly reduces the training MSE and also accelerates the MSE convergence. However, the MSEs of NCF8 and NCF64 are slightly higher than those of NCF16 and NCF32 on the test dataset. The MSE of NCF16 is the lowest and tends to decrease continuously. Therefore, NCF16 is chosen as the next setting for the study.

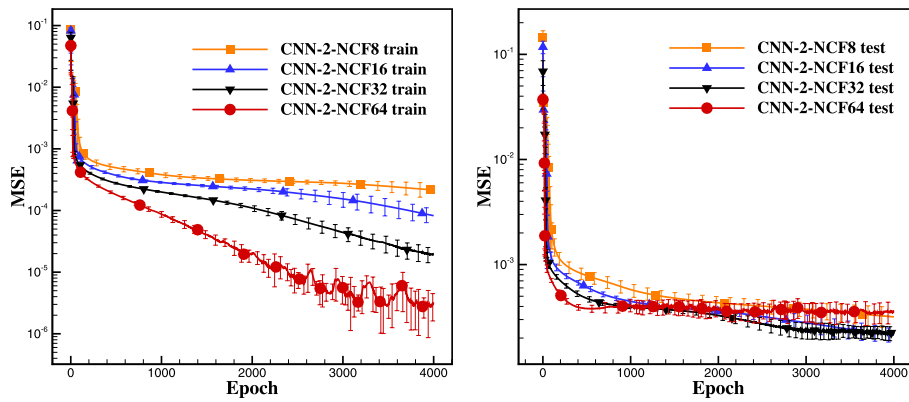


Fig. 15 MSE convergence histories with different numbers of convolutional filters in training dataset (left) and test dataset (right)

4.1.3 Number of fully-connected layers

CNN-2 is modified to have one, two, and four fully-connected layers (CNN-2-FC1, CNN-2-FC3, and CNN-2-FC4) to investigate the effect of the number of fully-connected layers, as shown in Table 5. The MSE convergence histories are given in Fig. 16. More fully-connected layers improve the MSE convergence, and CNN-2-FC4 has the best training MSE convergence. However, CNN-2-FC3 finally achieves the minimum MSE on the test dataset. In addition, CNN-2-FC3 tends to decrease even further, which is better than CNN-2-FC4.

Table 5 Variation of CNN-2 with different numbers of fully-connected layers

Model	Fully-connected layers, depth×neurons
CNN-2-FC1	1×128
CNN-2-FC2	2×128
CNN-2-FC3	3×128
CNN-2-FC4	4×128

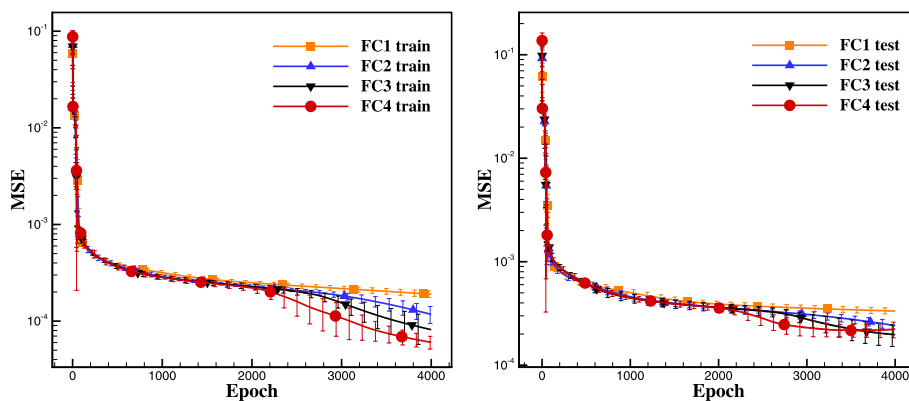


Fig. 16 MSE convergence histories of CNN-2 with different numbers of fully-connected layers in training dataset (left) and test dataset (right) (e.g., FC2 indicates two fully-connected layers)

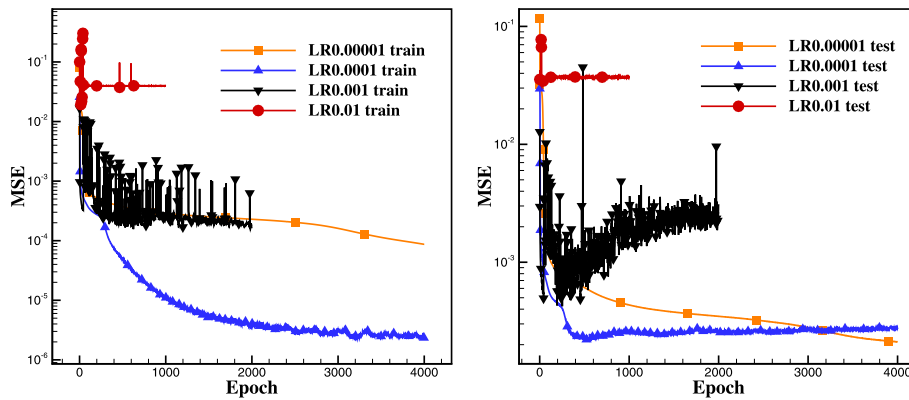


Fig. 17 MSE convergence histories of CNN-2 with different learning rates in training dataset (left) and test dataset (right)

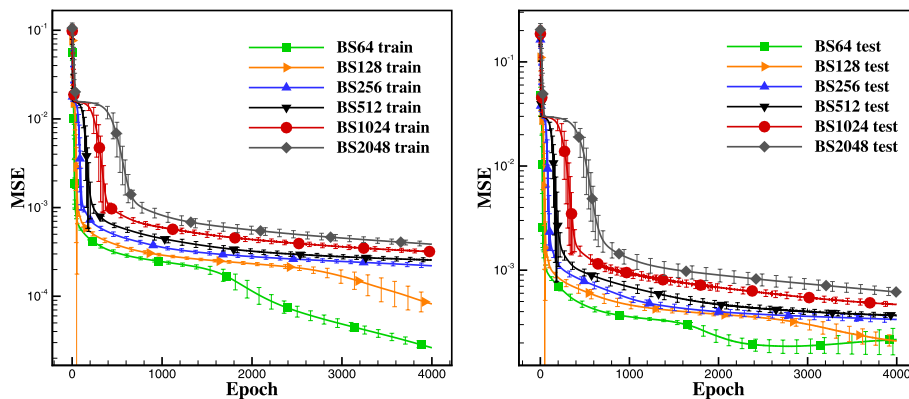


Fig. 18 MSE convergence histories of CNN-2 with different batch sizes in training dataset (left) and test dataset (right)

4.1.4 Learning rate

As a vital hyperparameter in CNN training, the learning rate (LR) largely determines the model convergence efficiency and final convergence level. In general, a high, inappropriate learning rate leads to model convergence failure, while a low, unsuitable learning rate results in higher training time costs and the risk of falling into a local optimum.

The MSE convergence histories with different learning rates (1×10^{-2} , 1×10^{-3} , 1×10^{-4} , and 1×10^{-5}) are shown in Fig. 17. These results indicate that LR0.01 and LR0.001 exhibit poor MSE convergence and severe oscillations. This may be because the optimization algorithm skips the optimal path because of a large initial learning rate [46]. Compared with LR0.00001, LR0.0001 achieves faster convergence, and then overfitting occurs. Ultimately, we consider LR0.00001, which converges smoothly and retains a downward trend, to be the superior learning rate, despite its longest training time.

4.1.5 Batch size

A study on the effect of batch size (BS) on MSE convergence is carried out for a series of batch sizes decreasing from 2048 (BS2048) to 64 (BS64) by factors of 2. The BS mainly impacts the amount of computation, i.e., a larger BS produces a faster computation but requires access to more samples to achieve the same error because there are fewer updates

per epoch [47]. From Fig. 18, it is observed that a smaller batch size leads to a smaller MSE in the training dataset. Because BS128 has the minimum MSE in the test dataset and retains the descent gradient, we use a minibatch of 128 as the optimum for the training.

The effect of the convolutional filter size on CNN training is also investigated. These results show that its influence is very small, so the commonly used 3×3 filter size is used.

4.2 Validation of CNN prediction performance

Based on the preceding investigations, we adopt a CNN containing two convolutional layers with a 3 × 1 filter size, two pooling layers with a 2 × 1 filter size, and three fully-connected hidden layers with 128 neurons per layer as the model structure. There are 16 convolutional filters in the first convolutional layer and 32 convolutional filters in the second convolutional layer. The learning rate and batch size are set to 0.00001 and 128, respectively. For full convergence of the model, the epoch is set to 6000.

The relative error ε , coefficient of determination R^2 , relative root-mean-square error (*RRMSE*), and relative maximum absolute error (*RMAE*) are employed as the metrics to validate the CNN’s predictive performance, and their equations are

$$\begin{aligned}
 \varepsilon &= \sum_{i=1}^N \frac{\|y_i - \hat{y}_i\|_2}{N \times \|y_i\|_2} \times 100\%, \\
 R^2 &= 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y}_i)^2}, R^2 \in (-\infty, 1], \\
 RRMSE &= \frac{1}{STD} \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}, \\
 RMAE &= \frac{1}{STD} \max |y_i - \hat{y}_i|, \\
 STD &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y}_i)^2}, i = 1, 2, \dots, N,
 \end{aligned}
 \tag{5}$$

where y_i is the simulation value of the i^{th} test sample calculated by CFD simulation, \hat{y}_i denotes the predicted value of the i^{th} test sample, and N is the number of test samples. The model is perfectly accurate when $R^2 = 1.0$, whereas $R^2 = 0.0$ indicates an extremely poor approximation. The *RRMSE* reflects the model’s global accuracy, and the *RMAE* is a criterion indicating the local prediction performance.

Figure 19 shows the MSE convergence history of the CNN training. The MSEs of both the training and test datasets decrease smoothly, with no oscillation or over-fitting. Figure 20 compares predicted versus CFD simulation values for the three aerodynamic coefficients in the test dataset. Nearly all the sample points of the test dataset are clustered near the 45° line, demonstrating the CNN’s very reliable prediction error. Among them, the predictive performance for C_L and C_M is better but is relatively poor for C_D . The distribution plot of the absolute error is shown in Fig. 21. The y -axis indicates the number of samples corresponding to the absolute error in the x -axis. It is apparent that the error distribution is Gaussian. From Fig. 21, the prediction errors for C_D in the test dataset are less than 10 counts for most of the samples, and the prediction errors for C_L and C_M in the testing dataset are less than 0.002 for most of the samples.

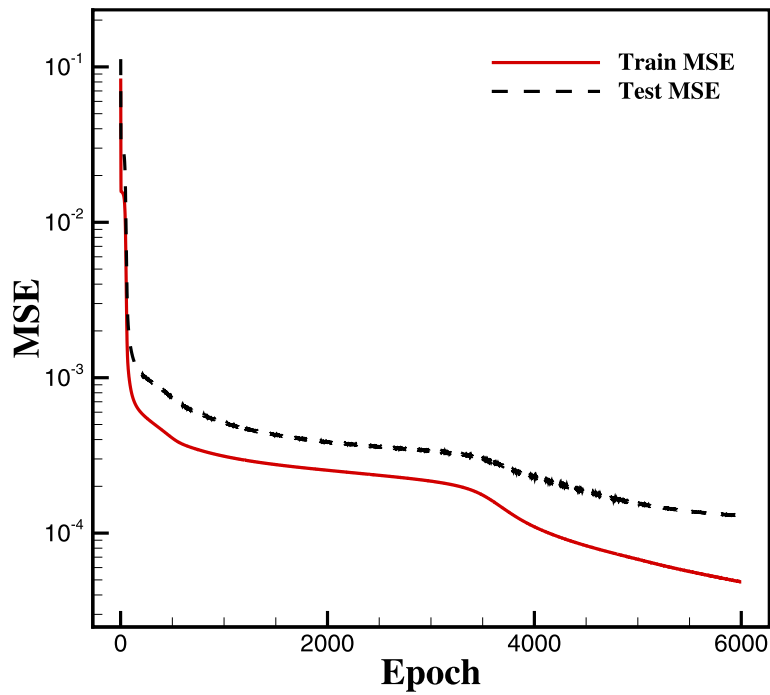


Fig. 19 Convergence history of MSE for the CNN

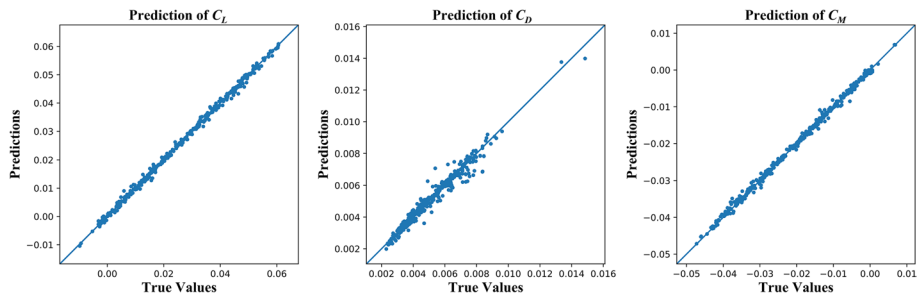


Fig. 20 Predicted versus CFD simulation values of C_L , C_D , and C_M

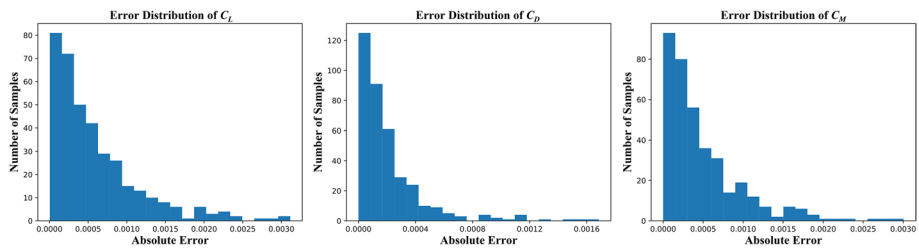


Fig. 21 Absolute error distribution of the CNN prediction for C_L , C_D , and C_M

As described in Table 1, the two-dimensional convolution operation is used in most previous works to process the pixel points of airfoils. The proposed architecture (CNN_1D shown in Fig. 3) in this article is also compared here with CNN_2D (using a

Table 6 Architectures of MLP, CNN_2D, and CNN_1D

Model	MLP		CNN_2D		CNN_1D	
	Layer type	Parameter	Layer type	Parameter	Layer type	Parameter
Input	-	109×1	-	10×10, 9×1	-	109×1
1st layer	Fully-connected	128	Conv_2D	16, 3×3, 2×2	Conv_1D	16, 3×1, 2×1
2nd layer	Fully-connected	128	Conv_2D	32, 3×3, 2×2	Conv_1D	32, 3×1, 2×1
3rd layer	Fully-connected	128	Fully-connected	128	Fully-connected	128
4th layer	Fully-connected	128	Fully-connected	128	Fully-connected	128
5th layer	Fully-connected	128	Fully-connected	128	Fully-connected	128
Output	-	3×1	-	3×1	-	3×1

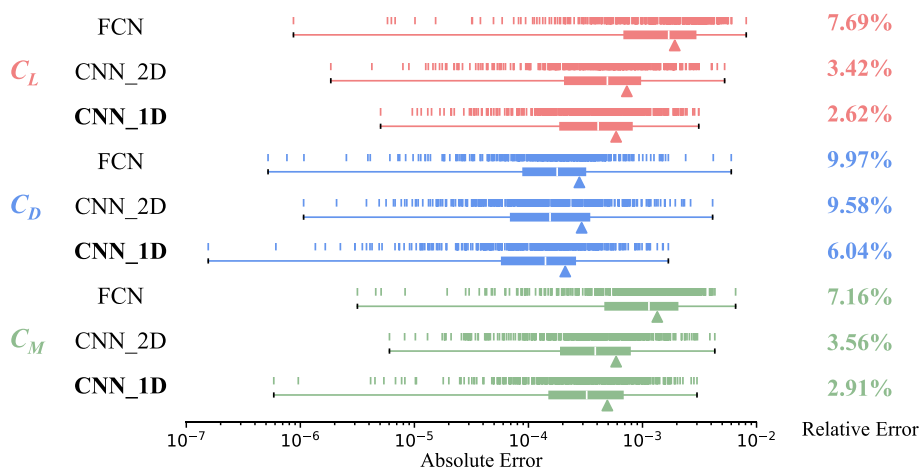


Fig. 22 Comparison of the box plots and absolute errors of the predictions obtained by MLP, CNN_2D, and CNN_1D

two-dimensional convolution operation for the FFD control point variables and the rest of the variables directly as inputs to the fully-connected layer) and MLP.

The three architectures are shown in Table 6. “Conv_2D” represents a two-dimensional convolution layer, “10 × 10” denotes a 10-row, 10-column matrix including 100 FFD control point variables, and “9 × 1” denotes the remaining 9 variables. The “128” refers to the number of neurons in the fully-connected layer, “16, 3 × 3, 2 × 2” denotes that the layer contains 16 convolutional filters with a 3 × 3 filter size and a 2 × 2 pooling filter size. The hyperparameters of CNN_2D are the same as those of CNN_1D. The commonly used hyperparameters with a 0.001 learning rate, batch size of 128, and epoch of 6000 are adopted for MLP.

Figure 22 shows the scatter plots of the absolute error and the corresponding box plots. The scatter plots at the top provide the absolute errors of the predicted force coefficients for each wing configuration and flight status, and the box plots at the bottom provide the statistics for these absolute errors. The white lines in the box plots indicate the median of these errors. The right and left edges of the box are the upper and lower quartiles, respectively. The triangle’s location below the box refers to the average of these errors.

Specifically, the largest CNN error for C_L is 0.0031. The CNN predicts 82.80% of the sample points with an error of less than 0.001. For C_D , the largest CNN error is 17

Table 7 Comparison of accuracy metrics for predictions of aerodynamic performance by MLP, CNN_2D, and CNN_1D in high-dimensional aerodynamic modeling for flight status and wing shape variables

Object	Model	Relative error ϵ	R^2	RRMSE	RMAE
C_L	MLP	7.69%	0.9813	0.1366	0.4638
	CNN_2D	3.42%	0.9963	0.0606	0.2998
	CNN_1D	2.62%	0.9978	0.0466	0.1779
C_D	MLP	9.97%	0.9076	0.3036	3.3977
	CNN_2D	9.58%	0.9147	0.2917	2.3320
	CNN_1D	6.04%	0.9661	0.1840	0.9539
C_M	MLP	7.16%	0.9815	0.1357	0.5212
	CNN_2D	3.56%	0.9954	0.0676	0.3425
	CNN_1D	2.91%	0.9970	0.0551	0.2391

counts. The percentages of test sample points with errors below 1, 5, and 10 counts are 38.44%, 91.13%, and 97.58%, respectively. For C_M , the largest CNN error is 0.0030, and 86.83% of test sample points have an error below 0.001. Three other accuracy metrics are computed for a more detailed comparison (Table 7), from which it is noted that CNN_1D performs significantly better than MLP and slightly better than CNN_2D.

The ability of the trained MLP and CNN models to predict the aerodynamic coefficients of the baseline configuration at different flight statuses, such as Mach number and angle of attack, is also evaluated. 121 additional sample points obtained by the uniform sampling method are calculated by conducting CFD simulation. These test sample points are plotted as white spheres in Fig. 23, with Mach numbers ranging from 5 to 6 and angles of attack in the range of 0 to 5 degrees. The response surfaces predicted by MLP and CNN are shown in Fig. 23(a), (d), and (g), respectively, from which several curves are sliced to obtain the variations of lift coefficient, drag coefficient, and pitch moment coefficient with respect to Mach number or angle of attack. CNN produces smoother response surfaces that conform to the flow mechanism and are a better match to the CFD simulation values than MLP, although the prediction error for the drag coefficient at $\alpha = 1^\circ$ is relatively large.

Figure 24 shows contour plots of the absolute error between the CFD simulation values and values predicted by MLP and CNN for the three aerodynamic coefficients. The CNN's absolute error is significantly smaller than that of MLP. Four accuracy metrics are computed for a more detailed comparison (Table 8), from which it is observed that the CNN outperforms the MLP.

(See figure on next page.)

Fig. 23 Comparison of predicted aerodynamic coefficients using MLP and CNN as a function of Mach number and angle of attack. **(a)** Predicted C_L by MLP (left) and CNN (right), **(b)** Predicted C_L by MLP and CNN at $Ma = 5.2$, **(c)** Predicted C_L by MLP and CNN at $\alpha = 1^\circ$, **(d)** Predicted C_D by MLP (left) and CNN (right), **(e)** Predicted C_D by MLP and CNN at $Ma = 5.2$, **(f)** Predicted C_D by MLP and CNN at $\alpha = 1^\circ$, **(g)** Predicted C_M by MLP (left) and CNN (right), **(h)** Predicted C_M by MLP and CNN at $Ma = 5.2$, **(i)** Predicted C_M by MLP and CNN at $\alpha = 1^\circ$

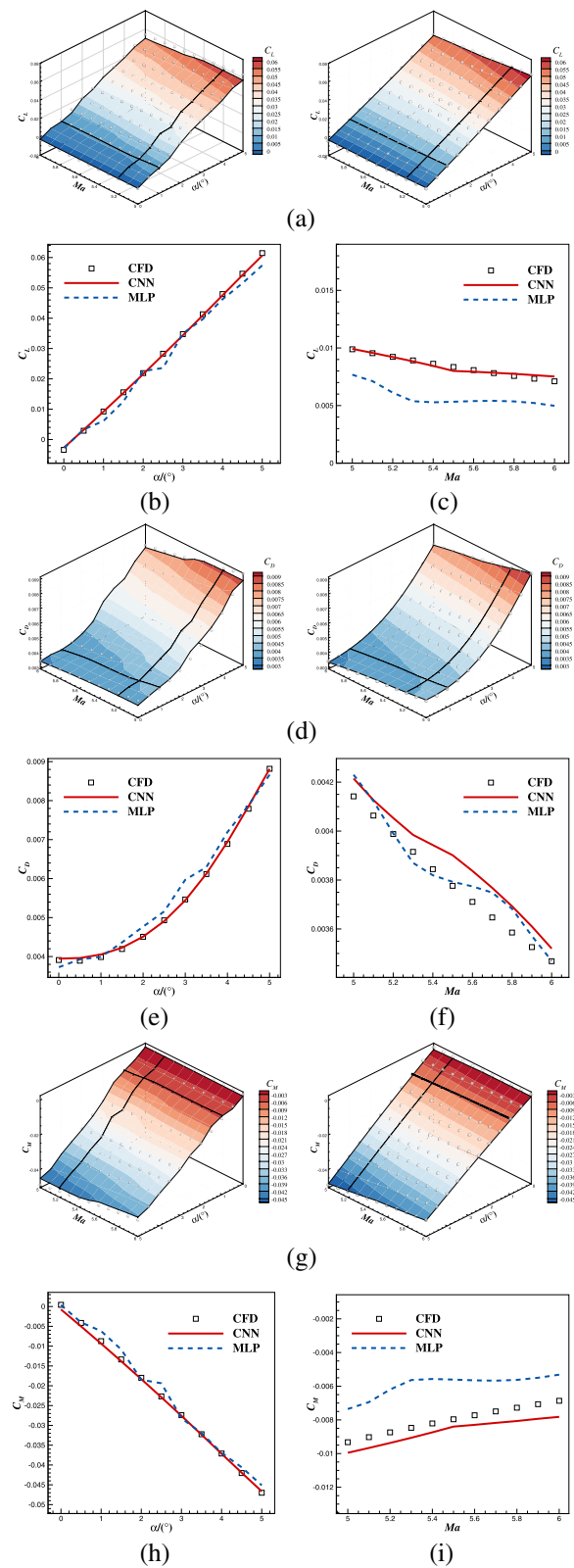


Fig. 23 (See legend on previous page.)

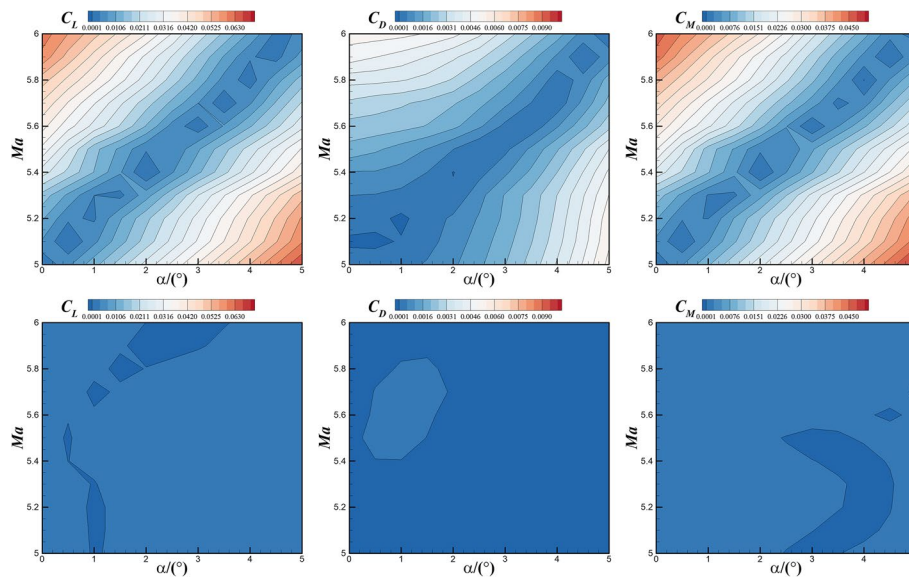


Fig. 24 Comparison of absolute error contours of C_L (left), C_D (middle), and C_M (right) for MLP and CNN predictions

Table 8 Comparison of accuracy metrics for aerodynamic coefficient predictions of the baseline wing by MLP and CNN at different Mach numbers and angles of attack

Object	Model	Relative error ϵ	R^2	RRMSE	RMAE
C_L	MLP	7.10%	0.9855	0.1198	0.3143
	CNN	1.55%	0.9993	0.0262	0.0699
C_D	MLP	4.61%	0.9745	0.1591	0.4046
	CNN	1.16%	0.9984	0.0402	0.0798
C_M	MLP	5.74%	0.9893	0.1029	0.2550
	CNN	2.23%	0.9984	0.0399	0.0913

Table 9 Design cases for wing profile optimization

Optimization case	Optimization problem	Mach number	Altitude (km)	Reynolds number	Angle of attack (α)
1	$\max f_{obj}(x) = C_L/C_D$ s.t. $ t - t_0 < 0.04t_0$	5.2	25	4.29×10^6	1°
2	$\max f_{obj}(x) = C_L/C_D$ s.t. $ t - t_0 < 0.04t_0$	5.2	25	4.29×10^6	2°
3	$\min f_{obj}(x) = C_D$ s.t. $ t - t_0 < 0.04t_0$ $C_L \geq C_{L,0}$	5.2	25	4.29×10^6	1°

4.3 Preliminary application to fast aerodynamic shape optimization

4.3.1 Optimization problem statement

The modeling approach proposed in this paper is now used with a genetic algorithm (GA) implemented on the well-trained CNN to optimize the five wing profiles to support fast aerodynamic shape optimization. In addition, three aerodynamic shape optimization cases are performed to verify the efficiency and convenience

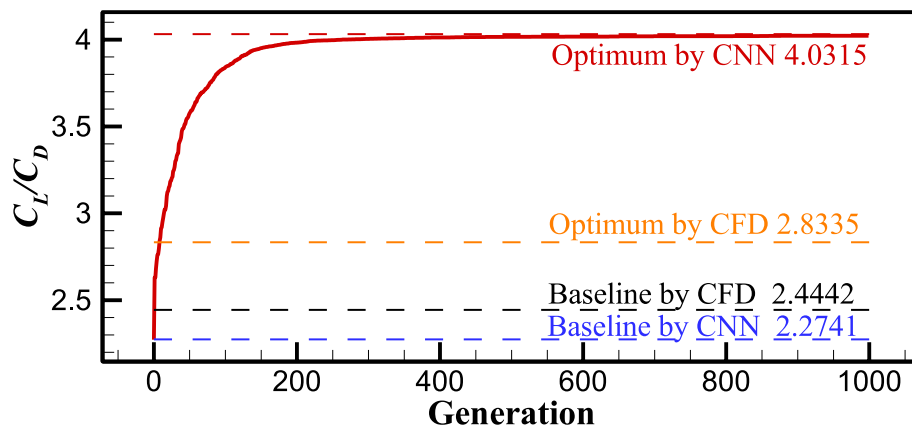


Fig. 25 Convergence history of aerodynamic shape optimization for the wing (optimization case 1)

Table 10 Comparison of aerodynamic performance obtained from the CNN prediction and the CFD validation for the baseline and optimized wings (optimization case 1)

		C_L	C_D	C_L/C_D	C_M
CNN prediction	Baseline wing	0.00921	0.00405	2.2741	-0.00937
	Optimized wing	0.01186	0.00294	4.0315	-0.01048
	Δ	+28.77%	-27.41%	+77.28%	+11.85%
CFD validation	Baseline wing	0.00952	0.00389	2.4442	-0.00897
	Optimized wing	0.01068	0.00377	2.8335	-0.00960
	Δ	+12.18%	-3.08%	+15.93%	+7.02%

advantages of the modeling approach, considering both flight status and shape variables for multiple flow statuses and optimization objectives. The details of the three optimization cases are shown in Table 9, where t represents the maximum profile thickness. All notations with subscript “0” indicate the baseline configuration, and the maximum profile thickness is also constrained by the structural requirements. An additional 5290 samples are added to the training dataset to further improve the model accuracy of the validation in this section.

Scaling the flow conditions from a three-dimensional swept wing to a two-dimensional profile is not considered here because the scaling rule for small- or medium-swept subsonic or transonic wings may not apply to low aspect ratios and highly swept hypersonic wings. Although it is necessary to scale the flow conditions in a real-world design, it is still sensible to validate and demonstrate the applicability of the CNN-based modeling method for efficient aerodynamic shape optimization.

Note that the five wing profiles selected as the baseline for optimization already have good aerodynamic characteristics in the hypersonic regime, so we use adequate design variables incorporating 100 FFD control points for a more refined optimization to further improve the aerodynamic performance.

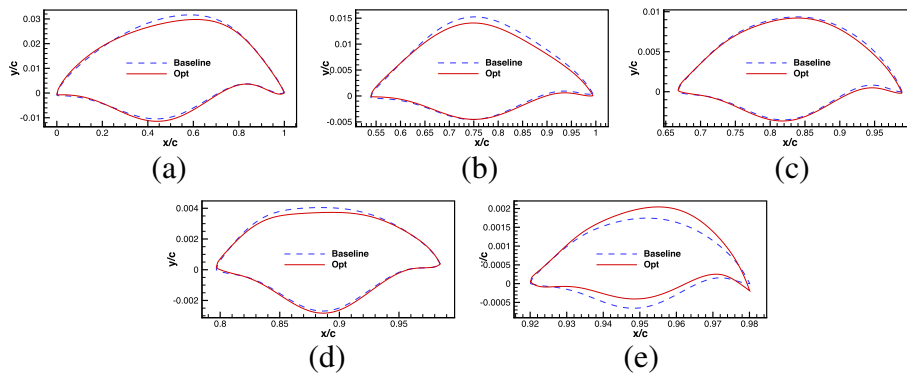


Fig. 26 Comparison of five baseline and optimized wing profiles (optimization case 1). (a) Profile 1, (b) Profile 2, (c) Profile 3, (d) Profile 4, (e) Profile 5

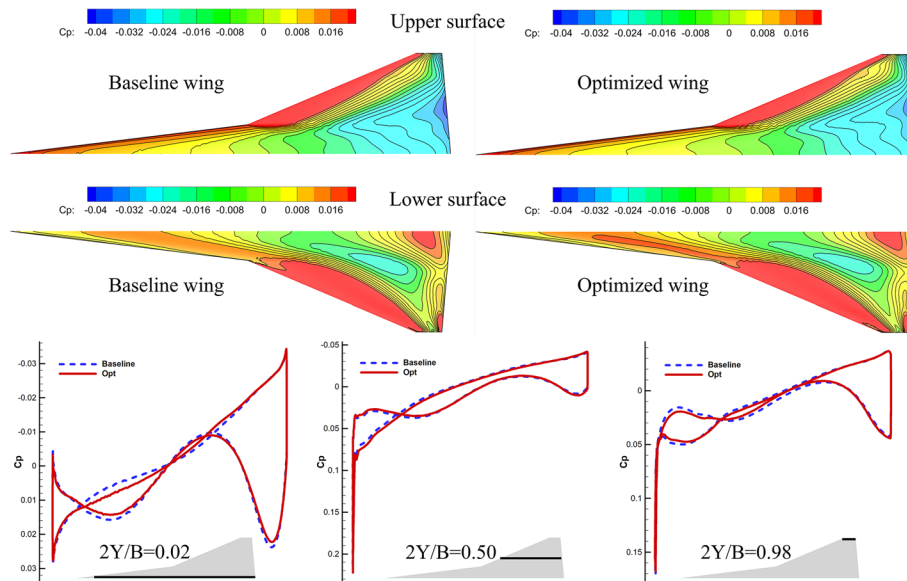


Fig. 27 Comparison of pressure coefficient contours and sectional pressure coefficient distributions at three wing spanwise locations for the baseline and optimized profiles (optimization case 1)

4.3.2 Optimization results

As seen from Fig. 25 and Table 10, with the support of the CNN providing the correct optimization direction, $C_L/C_D = 2.8335$, which is 15.93% larger than for the baseline wing, although the C_D prediction is not very accurate, resulting in relatively poor C_L/C_D predictions.

Figure 26 shows a comparison of the baselines and optimized geometric profiles. The maximum thickness of all profiles satisfies the constraints. The pressure coefficient contours and pressure distributions at three wing spanwise locations using the

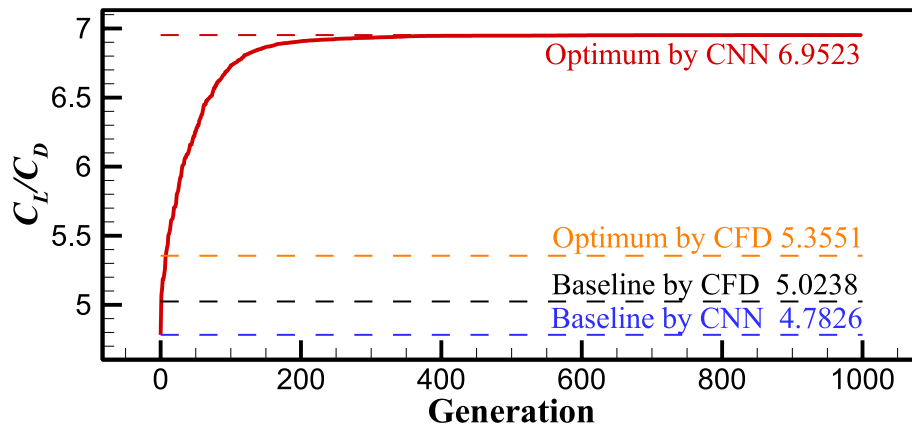


Fig. 28 Convergence history of aerodynamic shape optimization for the wing (optimization case 2)

Table 11 Comparison of aerodynamic performance obtained from the CNN prediction and the CFD validation for the baseline and optimized wings (optimization case 2)

		C_L	C_D	C_L/C_D	C_M
CNN prediction	Baseline wing	0.02157	0.00451	4.7826	-0.01832
	Optimized wing	0.02374	0.00341	6.9523	-0.01925
	Δ	+10.06%	-24.39%	+45.37%	+5.08%
CFD validation	Baseline wing	0.02221	0.00442	5.0238	-0.01825
	Optimized wing	0.02292	0.00428	5.3551	-0.01852
	Δ	+3.20%	-3.17%	+6.59%	+1.48%

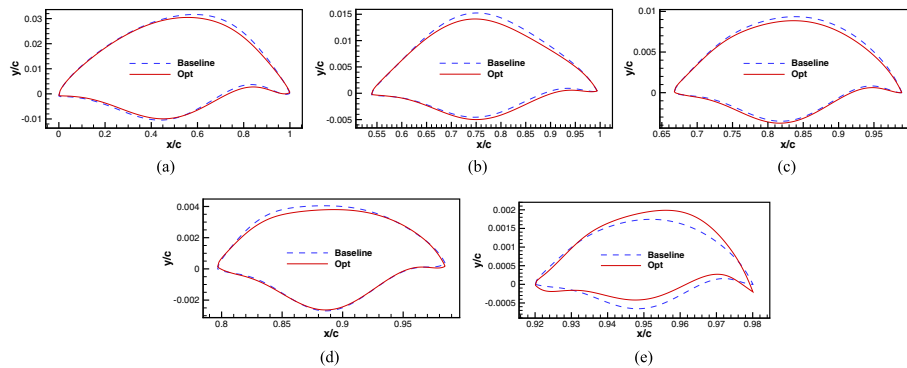


Fig. 29 Comparison of five baseline and optimized wing profiles (optimization case 2). (a) Profile 1, (b) Profile 2, (c) Profile 3, (d) Profile 4, (e) Profile 5

baseline and optimized wing profiles are shown in Fig. 27. A larger high-pressure regime is observed on the wing’s lower surface using the optimized profiles, and a slightly larger low-pressure regime is observed on the upper surface.

As seen from Fig. 28 and Table 11 for optimization case 2, $C_L/C_D = 5.3551$, which is 6.59% larger than for the baseline wing. Figure 29 compares the five baseline and optimized wing profiles. Pressure coefficient contours and sectional pressure distributions are presented in Fig. 30.

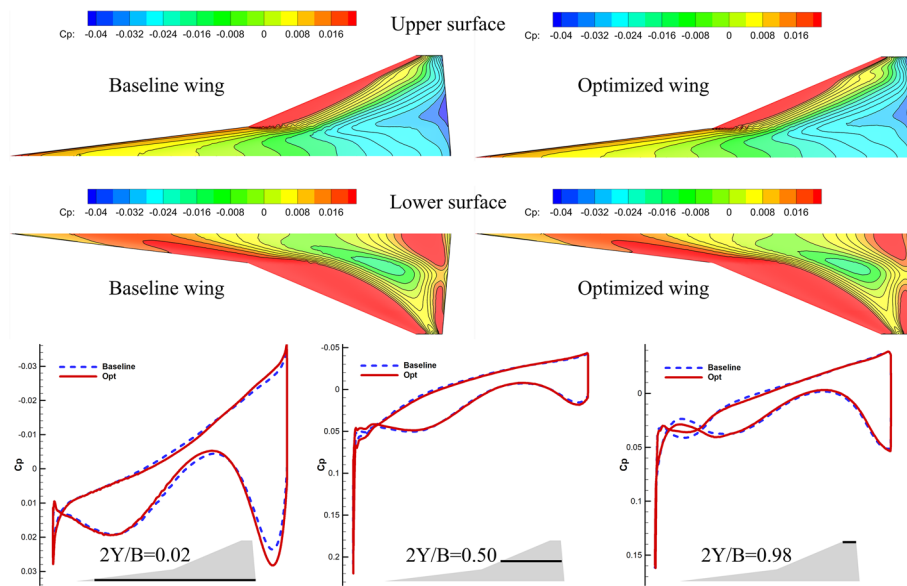


Fig. 30 Comparison of pressure coefficient contours and sectional pressure coefficient distributions at three wing spanwise locations for the baseline and optimized profiles (optimization case 2)

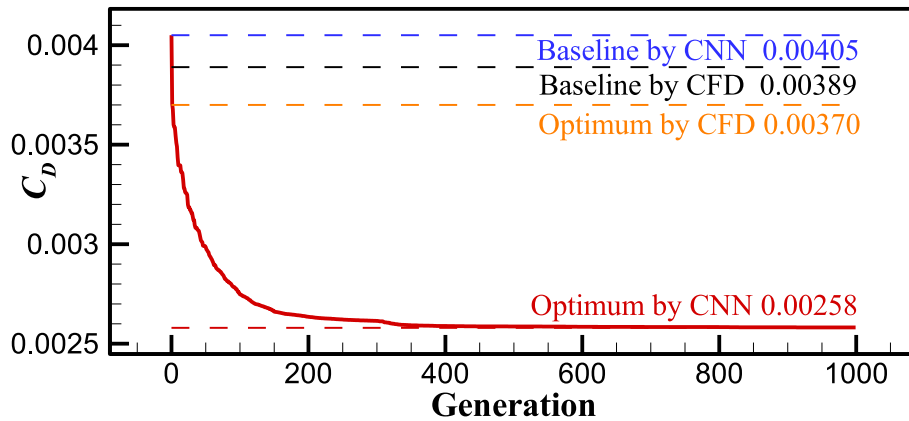


Fig. 31 Convergence history of aerodynamic shape optimization for the wing (optimization case 3)

Table 12 Comparison of aerodynamic performance obtained from the CNN prediction and the CFD validation for the baseline and optimized wings (optimization case 3)

		C_L	C_D	C_L/C_D	C_M
CNN prediction	Baseline wing	0.00921	0.00405	2.2741	-0.00937
	Optimized wing	0.01001	0.00258	3.8769	-0.00939
	Δ	+8.65%	-36.30%	+70.48%	+0.21%
CFD validation	Baseline wing	0.00952	0.00389	2.4442	-0.00897
	Optimized wing	0.00978	0.00370	2.6434	-0.00896
	Δ	+2.73%	-4.88%	+8.15%	-0.11%

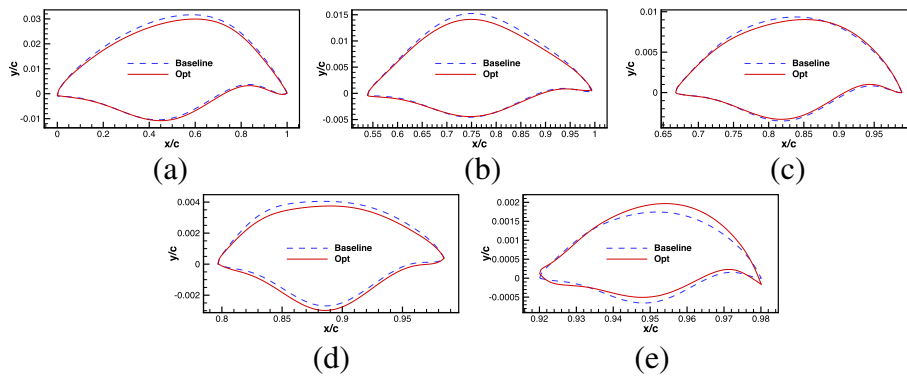


Fig. 32 Comparison of five baseline and optimized wing profiles (optimization case 3). (a) Profile 1, (b) Profile 2, (c) Profile 3, (d) Profile 4, (e) Profile 5

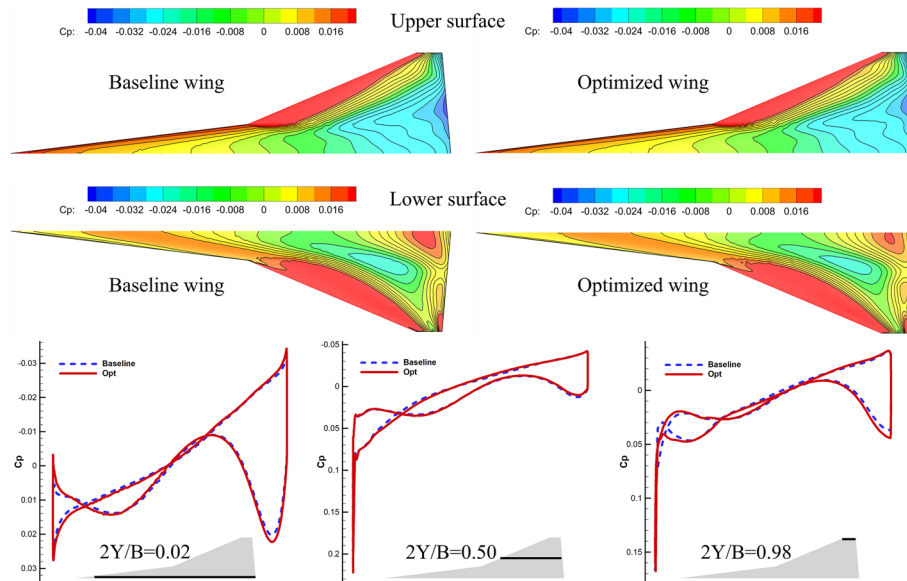


Fig. 33 Comparison of pressure coefficient contours and sectional pressure coefficient distributions at three wing spanwise locations using baselines and optimized profiles (optimization case 3)

Table 13 GA parameters and time cost for the three optimization cases

Optimization case	Population	Generation	Evaluation	Total time used (Intel core i9 11900 K@3.50 Hz)
1	500	1000	500,000	40 min 57 s
2	500	1000	500,000	40 min 24 s
3	500	1000	500,000	41 min 10 s

For optimization case 3, $C_D = 0.00370$ from Fig. 31 and Table 12, which is 4.88% less than for the baseline wing, while C_L is also improved. A comparison of the five baseline and optimized wing profiles is given in Fig. 32. The thicknesses of the five wing profiles are decreased by varying degrees. Figure 33 shows the pressure coefficient contours and the sectional pressure coefficient distributions.

5 Discussion

Table 13 shows the GA parameters and the time cost for the three wing optimization cases. It is observed that the CNN-based high-dimensional aerodynamic modeling method considering the flight status and shape variables can quickly yield better aerodynamic shapes for all three cases, i.e., different design points and different optimization problems.

From the results of the three optimization cases, the aerodynamic coefficient prediction errors of the CNN for the optimized solutions are somewhat larger than the global error, reflecting the fact that the current database size used to model a design space containing up to 100 variables is not sufficient. Therefore, the model cannot provide very accurate global predictions but can give the correct optimization orientation. Hence, it can still demonstrate the advantages of the proposed modeling approach in high-dimensional aerodynamic modeling and the ability to achieve fast aerodynamic shape optimizations for multiple flight statuses.

6 Conclusions

This article proposes a CNN-based machine learning approach for high-dimensional aerodynamic data modeling to provide fast and reliable aerodynamic performance predictions. This modeling approach is demonstrated by an aerodynamic modeling case similar to the Sanger aerospace plane carrier wing with a 109-dimensional input incorporating both the flight status and aerodynamic shape variables. The following conclusions can be drawn:

- a) The MLP adapts to the augmentation of samples by expanding the network scale when dealing with high-dimensional problems, resulting in an overfitting trend. The CNN, having the advantages of sparse connection and weight sharing, can alleviate this problem.
- b) The network's convergence depends largely on the CNN's hyperparameters. A learning rate (LR) between 1×10^{-5} and 1×10^{-4} is a good choice, with a higher LR leading to oscillation (or even a failure to converge) and a lower LR leading to increased training costs. Increasing the number of convolutional layers enhances the ability to distill information, but networks that are too deep are prone to overfitting. A smaller batch size gives rise to faster MSE convergence and greater computational intensity for the same epoch.
- c) Compared with the MLP, the CNN-based modeling method is dramatically more accurate, not only for high-dimensional modeling problems with respect to both aerodynamic shape variables and flight status variables but also for response surface prediction at different Mach numbers and angles of attack.

- d) A surrogate model considering both the flight status and the shape enables fast aerodynamic shape optimization for multiple flight statuses without the need to conduct expensive CFD simulations to build additional surrogate models.

The CNN-based aerodynamic modeling approach emerges as a gradient-free, fast, and accurate tool that complements other traditional methods in aerodynamic research and provides a novel way to conduct practical design optimization. Furthermore, the targeted augmentation of sample points during modeling can further improve the predictive accuracy and is conducive to enhancing the quality of the optimization solution, which will be the subject of future research work.

Acknowledgements

The work was carried out at the National Supercomputer Center in Tianjin, and the calculations were performed on the TianHe-1A supercomputer.

Authors' contributions

This research is the result of a joint effort. All authors have read and approved the final manuscript.

Funding

This work is supported by the National Numerical Wind Tunnel Project (grant No. NNW2019ZT6-A12), the Science Fund for Distinguished Young Scholars of Shaanxi Province of China (grant No. 2020JC-31), and the Natural Science Foundation of Shaanxi Province (grant No. 2020JM-127).

Availability of data and materials

The datasets generated during the current study are available from the corresponding author upon reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 12 July 2022 Accepted: 13 October 2022

Published online: 22 December 2022

References

1. Bryan GH (1911) *Stability in aviation*. Macmillan & Co. Ltd, London
2. Russell WR (1978) *Aerodynamic design data book*. Vol I, Orbiter Vehicle. Rockwell International Space Division, Report No. SD72-SH-0060-1L
3. Pamadi BN, Brauckmann GJ, Ruth MJ et al (2001) Aerodynamic characteristics, database development, and flight simulation of the X-34 vehicle. *J Spacecr Rockets* 38(3):334–344. <https://doi.org/10.2514/2.3706>
4. Keshmiri S, Colgren R, Mirmirani M (2005) Development of an aerodynamic database for a generic hypersonic air vehicle. Paper presented at the AIAA guidance, navigation, and control conference and exhibit, San Francisco, 15–18 August 2005. <https://doi.org/10.2514/6.2005-6257>
5. Fu JM (2005) Three-dimensional aerodynamic mathematical model for tactical missiles with jet steering. *Aerospace Shanghai* 22(4):13–18 (in Chinese)
6. He KF, Wang WZ, Qian WQ (2004) Mathematic modeling for the missile aerodynamics with tail-wing according to wind-tunnel test results. *Exp Meas Fluid Mech* 18(4):62–66 (in Chinese)
7. Ghoreyshi M, Cummings RM, Da Ronch A et al (2013) Transonic aerodynamic load modeling of X-31 aircraft pitching motions. *AIAA J* 51(10):2447–2464. <https://doi.org/10.2514/1.J052309>
8. Krige DG (1951) A statistical approach to some basic mine valuation problems on the Witwatersrand. *J Chem Metall Min Soc South Afr* 52(6):119–139
9. Han ZH, Görtz S (2012) Hierarchical kriging model for variable-fidelity surrogate modeling. *AIAA J* 50(9):1885–1896. <https://doi.org/10.2514/1.J051354>
10. Han ZH, Görtz S, Zimmermann R (2013) Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerosp Sci Technol* 25(1):177–189. <https://doi.org/10.1016/j.ast.2012.01.006>
11. Liu J, Song WP, Han ZH et al (2017) Efficient aerodynamic shape optimization of transonic wings using a parallel infilling strategy and surrogate models. *Struct Multidisc Optim* 55(3):925–943. <https://doi.org/10.1007/s00158-016-1546-7>
12. Shi ZW, Wang ZH, Li JC (2012) The research of RBFNN in modeling of nonlinear unsteady aerodynamics. *Acta Aerodyn Sin* 30(1):108–112, 119 (in Chinese)
13. Wu C, Yao H, Peng XZ et al (2013) Application of support vector regression for aerodynamic modeling. *Comput Simul* 30(10):128–132 (in Chinese)

14. Santos M, Mattos B, Girardi R (2008) Aerodynamic coefficient prediction of airfoils using neural networks. Paper presented at the 46th AIAA aerospace sciences meeting and exhibit. Reno, 7–10 January 2008. <https://doi.org/10.2514/6.2008-887>
15. Wang C, Wang GD, Bai P (2019) Machine learning method for aerodynamic modeling based on flight simulation data. *Acta Aerodyn Sin* 37(3):488–497. <https://doi.org/10.7638/kqdlxxb-2019.0024> (in Chinese)
16. Zhu L, Gao ZH (2007) Aerodynamic optimization design of airfoil based on neural networks. *Aeronautical Computing Technique* 37(3):33–36 (in Chinese)
17. Fu JQ, Shi ZW, Chen K et al (2018) Applications of real-time recurrent neural network based on extended Kalman filter in unsteady aerodynamics modeling. *Acta Aerodyn Sin* 36(4):658–663. <https://doi.org/10.7638/kqdlxxb-2016.0131> (in Chinese)
18. Shi ZW, Ming X (2005) The application of FNN in unsteady aerodynamics modeling based on fuzzy clustering. *Acta Aerodyn Sin* 23(1):21–24 (in Chinese)
19. Zhang Y, Sung WJ, Mavris DN (2018) Application of convolutional neural network to predict airfoil lift coefficient. Paper presented at the 2018 AIAA/ASCE/AHS/ASC structures, structural dynamics, and materials conference. Kissimmee, 8–12 January 2018. <https://doi.org/10.2514/6.2018-1903>
20. Du X, He P, Martins JRRR (2021) Rapid airfoil design optimization via neural networks-based parameterization and surrogate modeling. *Aerosp Sci Technol* 113:106701. <https://doi.org/10.1016/j.ast.2021.106701>
21. Punjani A, Abbeel P (2015) Deep learning helicopter dynamics models. Paper presented at the 2015 IEEE international conference on robotics and automation (ICRA). Seattle, 26–30 May 2015. <https://doi.org/10.1109/ICRA.2015.7139643>
22. Wang Q, Qian WQ, Ding D (2016) A review of unsteady aerodynamic modeling of aircrafts at high angles of attack. *Acta Aeronaut Astronaut Sin* 37(8):2331–2347. <https://doi.org/10.7527/S1000-6893.2016.0072> (in Chinese)
23. Bouhleih MA, He S, Martins JRRR (2020) Scalable gradient-enhanced artificial neural networks for airfoil shape design in the subsonic and transonic regimes. *Struct Multidisc Optim* 61(4):1363–1376. <https://doi.org/10.1007/s00158-020-02488-5>
24. Secco NR, de Mattos BS (2017) Artificial neural networks to predict aerodynamic coefficients of transport airplanes. *Aircr Eng Aerosp Tec* 89(2):211–230. <https://doi.org/10.1108/AEAT-05-2014-0069>
25. Barnhart SA, Narayanan B, Gunasekaran S (2021) Blown wing aerodynamic coefficient predictions using traditional machine learning and data science approaches. Paper presented at the AIAA Scitech 2021 Forum. Virtual Event, 11–15 & 19–21 January 2021. <https://doi.org/10.2514/6.2021-0616>
26. Karali H, Inalhan G, Umut Demirezen MU et al (2021) A new nonlinear lifting line method for aerodynamic analysis and deep learning modeling of small unmanned aerial vehicles. *Int J Micro Air Veh* 13:17568293211016816. <https://doi.org/10.1177/17568293211016817>
27. Li J, Zhang M (2021) Data-based approach for wing shape design optimization. *Aerosp Sci Technol* 112:106639. <https://doi.org/10.1016/j.ast.2021.106639>
28. LeCun Y, Bottou L, Bengio Y et al (1998) Gradient-based learning applied to document recognition. *P IEEE* 86(11):2278–2324. <https://doi.org/10.1109/5.726791>
29. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444. <https://doi.org/10.1038/nature14539>
30. Thuerey N, Weißbenow K, Prantl L et al (2020) Deep learning methods for Reynolds-averaged Navier-Stokes simulations of airfoil flows. *AIAA J* 58(1):25–36. <https://doi.org/10.2514/1.J058291>
31. Yu B, Xie L, Wang F (2020) An improved deep convolutional neural network to predict airfoil lift coefficient. In: Jing Z (eds) *Proceedings of the International Conference on Aerospace System Science and Engineering 2019. ICASSE 2019. Lecture Notes in Electrical Engineering*, vol 622. Springer, Singapore. https://doi.org/10.1007/978-981-15-1773-0_21
32. Chen H, He L, Qian W et al (2020) Multiple aerodynamic coefficient prediction of airfoils using a convolutional neural network. *Symmetry* 12(4):544. <https://doi.org/10.3390/sym12040544>
33. He L, Qian WQ, Dong KS et al (2022) Aerodynamic characteristics modeling of iced airfoil based on convolution neural networks. *Acta Aeronaut Astronaut Sin* 43(10):126434. <https://doi.org/10.7527/S1000-6893.2021.26434> (in Chinese)
34. Shan S, Wang GG (2010) Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Struct Multidisc Optim* 41(2):219–241. <https://doi.org/10.1007/s00158-009-0420-2>
35. Han ZH, Xu CZ, Zhang L et al (2020) Efficient aerodynamic shape optimization using variable-fidelity surrogate models and multilevel computational grids. *Chinese J Aeronaut* 33(1):31–47. <https://doi.org/10.1016/j.cja.2019.05.001>
36. Han ZH, Zhang Y, Song CX et al (2017) Weighted gradient-enhanced kriging for high-dimensional surrogate modeling and design optimization. *AIAA J* 55(12):4330–4346. <https://doi.org/10.2514/1.J055842>
37. Liu F, Han ZH, Zhang Y et al (2019) Surrogate-based aerodynamic shape optimization of hypersonic flows considering transonic performance. *Aerosp Sci Technol* 93:105345. <https://doi.org/10.1016/j.ast.2019.105345>
38. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. *Commun ACM* 60(6):84–90. <https://doi.org/10.1145/3065386>
39. Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. Paper presented at the 2017 International conference on engineering and technology (ICET). Antalya, 21–23 August 2017. <https://doi.org/10.1109/ICEngTechnol.2017.8308186>
40. Narasinga Rao MR, Venkatesh Prasad V, Sai Teja P et al (2018) A survey on prevention of overfitting in convolution neural networks using machine learning techniques. *Int J Eng Technol* 7(2.32):177–180. <https://doi.org/10.14419/ijet.v7i2.32.15399>
41. Xiao M, Wu Y, Zuo G et al (2021) Addressing overfitting problem in deep learning-based solutions for next generation data-driven networks. *Wirel Commun Mob Comput* 2021:8493795. <https://doi.org/10.1155/2021/8493795>

42. Xu CZ, Qiao JL, Nie H et al (2019) Numerical investigation on aerodynamic performance of a standard model CHN-T1 using an unstructured flow solver. *Acta Aerodyn Sin* 37(2):291–300. <https://doi.org/10.7638/kqdlxxb-2018.0198> (in Chinese)
43. Zhang YB, Tang J, Chen JT et al (2019) Aerodynamic characteristics prediction of CHN-T1 standard model with unstructured grid. *Acta Aerodyn Sin* 37(2):262–271. <https://doi.org/10.7638/kqdlxxb-2018.0201> (in Chinese)
44. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323(6088):533–536. <https://doi.org/10.1038/323533a0>
45. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. <https://doi.org/10.48550/arXiv.1412.6980>
46. Sekar V, Zhang M, Shu C et al (2019) Inverse design of airfoil using a deep convolutional neural network. *AIAA J* 57(3):993–1003. <https://doi.org/10.2514/1.J057894>
47. Bengio Y (2012) Practical recommendations for gradient-based training of deep architectures. In: Montavon G, Orr GB, Müller KR (eds) *Neural Networks: Tricks of the Trade*. Lecture Notes in Computer Science, vol 7700. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-35289-8_26

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
