

High-Dimensional Aerodynamic Data Modeling Using Machine Learning Method Based on Convolutional Neural Network

Bo-Wen Zan (✉ bwzan@mail.nwpu.edu.cn)

Northwestern Polytechnical University <https://orcid.org/0000-0002-8350-927X>

ZhongHua Han

Northwestern Polytechnical University

Chen-Zhou Xu

Northwestern Polytechnic University

Ming-Qi Liu

Northwestern Polytechnical University

Wen-Zheng Wang

University of Electronic Science and Technology of China

Research Article

Keywords: aerodynamic data modeling, high-dimensional problem, machine learning, convolutional neural network, computational fluid dynamics

Posted Date: July 20th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1848322/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

1 High-Dimensional Aerodynamic Data Modeling Using Machine 2 Learning Method Based on Convolutional Neural Network

3 Bo-Wen Zan^a, Zhong-Hua Han^{a,*}, Chen-Zhou Xu^a, Ming-Qi Liu^a, Wen-Zheng Wang^b

4 *a. Institute of Aerodynamic and Multidisciplinary Design Optimization, National Key Laboratory of*
5 *Science and Technology on Aerodynamic Design and Research, School of Aeronautics, Northwestern*
6 *Polytechnical University, Xi'an 710072, China*

7 *b. Aircraft Swarm Intelligent Sensing and Cooperative Control Key Laboratory of Sichuan Province,*
8 *School of Aeronautics and Astronautics, University of Electronic Science and Technology of China,*
9 *Chengdu 611731, China*

11 **Abstract**

12 Modeling of high-dimensional aerodynamic data presents a major challenge in
13 the context of aero-loads prediction, aerodynamic shape optimization, flight control
14 and simulation, etc. In this article, a machine learning approach based on convolu-
15 tional neural network (CNN) is developed to address this problem. CNN is able to
16 implicitly distill features underlying the data, and the number of parameters to be
17 trained can be significantly reduced due to its local connectivity and parameter shar-
18 ing properties, which is favorable for solving high-dimensional problems in which the
19 training cost can be prohibitive. A hypersonic wing similar to the wing of Sanger aer-
20 ospace plane carrier is employed as the test case to demonstrate the CNN-based mod-
21 eling method. First, the wing is parameterized by free-form deformation method and
22 109 variables incorporating flight status and aerodynamic shape variables are defined
23 as model input. Second, 7431 sample points generated by Latin hypercube sampling
24 method are evaluated by performing computational fluid dynamics simulations based
25 on a Reynolds-Average Navier-Stokes flow solver to finally obtain an aerodynamic

*Corresponding author. *E-mail address:* hanzh@mwpw.edu.cn.

26 database, and a CNN model is built based on the observed data. Finally, the well-
27 trained CNN model considering both flight status and shape variables is applied to
28 aerodynamic shape optimization to demonstrate its capability to achieve fast optimi-
29 zation at multiple flight statuses.

30 Keywords: aerodynamic data modeling; high-dimensional problem; machine learning;
31 convolutional neural network; computational fluid dynamics.

32

33 **1. Introduction**

34 Modeling of aerodynamic data for an aircraft is to establish an explicit or implic-
35 it relationship between input variables and output responses through a trained physics-
36 informed model or data-driven model. It provides a rapid evaluation and prediction of
37 aerodynamic characteristics for an aircraft instead of conducting flight test, wind tun-
38 nel experiment, or numerical simulation such as computational fluid dynamics (CFD).
39 Although sufficient number of samples are invariably required to build an adequate
40 model with respect to independent variables, such as flight status variables (or aircraft
41 shape variables) and corresponding aerodynamic characteristics, it still can signifi-
42 cantly reduce the cost of building a database covering the entire flight envelope com-
43 pared with these traditional approaches of obtaining aerodynamic characteristics. The
44 increase in the number of independent variables of aerodynamic modeling brought by
45 the need for more high-dimensional designs leads to more sample points and training
46 cost to obtain desired model accuracy.

47 Generally, aerodynamic modeling methods can be divided into two categories:
48 physics-informed modeling method and data-driven modeling method. A physics-

49 informed model is a type of models whose structure or terms determined by physical
50 laws or flow mechanism, and the mapping between the input variables and output
51 aerodynamic functions can be approximated according to limited data. It usually in-
52 cludes aerodynamic derivative model^[1], linear incremental model^[2], linear superposi-
53 tion aerodynamic model^{[3]-[4]}, triangular series model^{[5]-[6]}, reduced-order model
54 (ROM)^[7], etc. These models are physically interpretable since it is closely related to
55 aerodynamic configuration. However, the accuracy of a physics-informed model is
56 frequently not sufficient especially when the linear hypothesis at small angle of attack
57 or control surface deflection is no longer valid. In contrast, a data-driven model can
58 achieve higher model accuracy in fitting both linear and nonlinear relationships be-
59 tween input variables and their corresponding responses when model parameters are
60 well tuned. Data-driven models are represented by machine learning (ML) models,
61 such as kriging^{[8]-[11]}, radial basis function (RBF) neural network^[12], support vector
62 machine (SVM)^[13], artificial neural network (ANN)^{[14]-[20]}, etc.

63 Recently, with the rapid development of data science and artificial intelligence
64 techniques, more ML approaches were applied to aerodynamic modeling^{[21]-[22]}.
65 Bouhlel^[23] proposed a modified Sobolev training for artificial neural network
66 (mSANN) to model aerodynamic force coefficients of airfoils in both subsonic and
67 transonic regimes using about 42,000 training sample points and 22,000 validation
68 sample points obtained by solving the Reynolds Average Navier-Stokes (RANS)
69 equations with different shapes parameterized by 14 modes and different flight status
70 incorporating Mach number and angle of attack. Du^[20] adopted a combination of mul-
71 ti-layer perceptron (MLP), recurrent neural network (RNN), and the mixture of ex-
72 perts (MoE) to predict the lift and drag coefficients of airfoils with a variety of shapes

73 defined by 26 B-spline curve variables and 2 flight status variables. Compare with
74 aerodynamic modeling of airfoil, more design variables are required for that of wing
75 or aircraft. Secco^[24] used an MLP to predict the lift and drag coefficients of a wing-
76 fuselage aircraft configuration with different wing planar variables, shapes variables
77 in three airfoil profiles (each airfoil profile is described by 10 variables), and flight
78 status, totally of 40 input variables. Barnhart^[25] proposed several ML methods to pre-
79 dict the lift and pitching moment coefficient for blown wing configuration with 20
80 different design variables. Karali^[26] used an MLP model, trained by 94,500 training
81 samples, that can predict the aerodynamic characteristic of unmanned aerial vehicles
82 (UAV) configurations concerning 22 input variables (21 geometric variables and the
83 angle of attack). Li^[27] chose 60 design variables containing the Mach number, the alti-
84 tude, the angle of attack, seven twist angles, and 50 wing modes in aerodynamic mod-
85 eling. The model was used to train by 135,108 training sample data and further veri-
86 fied in multiple single-point, multi-point, and multi-objective wing design optimiza-
87 tion problems.

88 Most of the existing interdisciplinary work, in terms of the combination of meta-
89 modeling of CFD and ML techniques, employs the learning architecture that belongs
90 to the category of multi-layer perceptron. The trade-off space between the size and
91 learning capability of standard MLP is highly quantified, with more complex prob-
92 lems requiring larger scale data and more complex network structures which yield
93 wasteful connectivity and a high tendency to overfitting. It is noted that no more than
94 60 independent input variables are introduced into existing work of aerodynamic
95 modeling using MLP, which limits the more refined design. Later, convolutional neu-
96 ral networks have rapidly replaced standard MLP techniques in a multitude of chal-

97 lenging ML tasks (e.g., image recognition^{[28]-[29]}) because of its local connectivity and
98 parameter sharing scheme. The research work of aerodynamic modeling method using
99 airfoil image as input shows its advantages for high-dimensional aerodynamic model-
100 ing. Thuerey^[30] used CNN to predict the information of airfoil flow field for different
101 Reynolds number ranging from 5×10^5 to 5×10^6 and angles of attack ranging from
102 -22.5° to 22.5° , and compared it with the results calculated by the RANS equations.
103 The prediction error of pressure contour and velocity contour is less than 3% when the
104 number of training sample points is up to 12,800. Zhang^[19] proposed a CNN-based
105 prediction method of airfoil lift coefficient with a variety of shapes in multiple free-
106 stream Mach number, Reynolds number, and angle of attack. About 80,000 sample
107 points of the airfoil coordinates, after data augment, were fed to CNN instead of shape
108 variables from parameterization. Yu^[31] proposed a feature-enhanced image approach
109 to complete the aerodynamic modeling of SC1095 airfoil by a CNN trained by 11,550
110 pairs of normal input/output training data. Chen^[32] also adopted a graphical prediction
111 method for multiple aerodynamic coefficients of airfoils based on CNN trained by
112 3360 samples and tested by 840 samples. Besides, CNN is also used to predict the
113 aerodynamic characteristics of iced airfoils. He^[33] proposed a prediction method for
114 the aerodynamic characteristics of iced airfoil based on CNN, which used 11,200
115 training samples to realize the rapid prediction from ice image to aerodynamic charac-
116 teristics with the prediction error less than 8%. The relevant studies in aerodynamic
117 modeling using ML-based methods for aerodynamic coefficient prediction as well as
118 flow field prediction are listed in Table 1.

119

120

Table 1 A 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		CNN
Zhang ^[19]	Airfoil		CNN
Yu ^[31]	Airfoil	Pixel of airfoil image	CNN
Chen ^[32]	Airfoil		CNN
He ^[33]	Airfoil		CNN

122 Please be noted that most of aerodynamic modeling methods using CNN adopt
 123 the idea taking airfoil graphics as input, which is limited to processing two-
 124 dimensional structured image data. For the three-dimensional aerodynamic shape, the
 125 research on modeling considering more than 100 parameterized independent variables
 126 is vacant, which motivates the research of this article.

127 The objective of this article is to develop a CNN-based method for high-
 128 dimensional aerodynamic modeling to alleviate the “curse-of-dimensionality”^[35] and
 129 apply it to surrogate-based aerodynamic shape optimization^{[36]-[38]}. The convolution
 130 operation of CNN is used to implicitly extract feature information underlying aerody-
 131 namic data and slightly squeeze the dimensionality of tensor. Usually, pooling layers
 132 are added after the convolutional layers to further reduce the number of neural net-
 133 work parameters (weights and biases) to be trained. With assistance of local connec-
 134 tivity and parameter sharing, CNN has higher training efficiency and better resistance
 135 to overfitting.

136 The rest of this article is organized as follows. In Sec. 2, a detailed introduction

137 of multi-layer perceptron and convolutional neural network is presented. Then, the
138 process of CNN-based aerodynamic data modeling method is given in Sec. 3. In Sec.
139 4, a well-trained CNN is built by investigating the influence of the hyperparameters
140 and apply it to fast aerodynamic shape optimization for a wing. Concluding remarks
141 are provided in Sec. 5.

142 **2. Fundamental of Convolutional Neural Network**

143 Machine learning methods are widely divided into supervised and unsupervised
144 learning. The scope of this article is limited to supervised machine learning methods
145 applied to regression problems, in which the machine learning model is trained using
146 labeled dataset to extract the underlying features from the data. CNN is the choice of
147 algorithms for the present work. Details of multi-layer perceptron (MLP) and CNN
148 will be explained briefly in this section.

149 *2.1. Multi-Layer Perceptron*

150 MLP is a machine learning model, also known as fully connected feedforward
151 neural network. The typical MLP, shown in Figure 1, receives an input (a single vec-
152 tor), and transforms it through a series of hidden layers. Each hidden layer is made up
153 of a set of neurons, where each neuron is fully connected to all neurons in the previ-
154 ous layer, and where neurons in a single layer function completely independently and
155 do not share any connections. Weights are established between neurons at each two
156 layers of the neural network for data forward propagation and error back propagation.
157 The last fully connected layer is called the “output layer”.

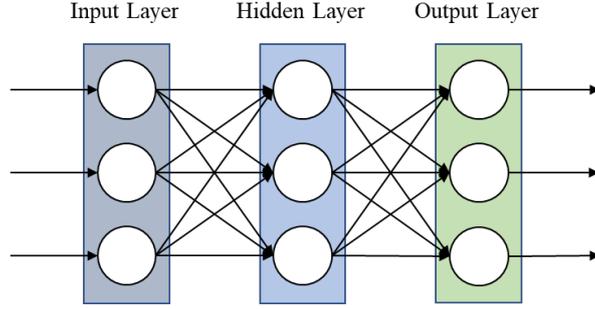


Figure 1 Typical Multi-Layer perceptron

158
159

160 For the forward propagation of MLP, the output of k -th layer $\mathbf{H}^{(k)}$ can be ex-
161 pressed as follows:

$$h_i^{(k)} = \sum_{j=1}^{m_{k-1}} w_{i,j}^{(k)} x_j^{(k-1)} + b_i^{(k)}, (1 \leq i \leq m_k) \quad (1)$$

$$\mathbf{H}^{(k)} = [h_1^{(k)}, h_2^{(k)}, \dots, h_{m_k}^{(k)}]^T = \mathbf{W}^{(k)} \mathbf{X}^{(k-1)} + \mathbf{b}^{(k)}$$

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

167 Defining the activation function as σ , we have the following activation opera-
168 tion:

$$\mathbf{Y}^{(k)} = \sigma(\mathbf{H}^{(k)}) \quad (2)$$

170 where $\mathbf{Y}^{(k)}$ is the activation vector on the k -th layer, i.e., the input of the next layer,
171 σ denotes the sigmoid activation function which is formulated as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (3)$$

173 In addition to the typical sigmoid activation function, the commonly used activa-
174 tion functions include ReLU, Leaky ReLU, and Tanh which is used in this article, and
175 its formula is shown below:

176

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (4)$$

177

The ability of the neural network to fit nonlinearity is continuously improved by iterative optimization of the network parameters. The weights and biases of the MLP are usually optimized using backpropagation algorithm^[34].

178

179

2.2. Convolutional Neural Network

180

181

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^{[39]-[40]}. The architecture of CNN mainly consists of convolutional layer, pooling layer and fully connected layers. A typical schematic of the CNN architecture for handwritten digit recognition is shown in Figure 2.

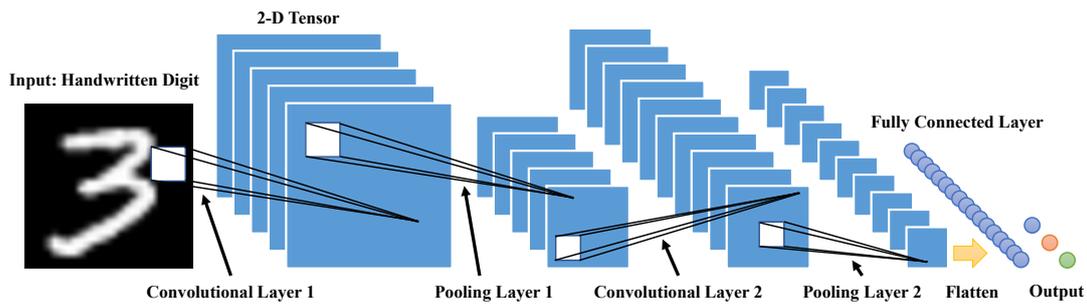
182

183

184

185

186



187

188

Figure 2 Conventional CNN architecture applied to handwritten digit recognition task

189

190

191

192

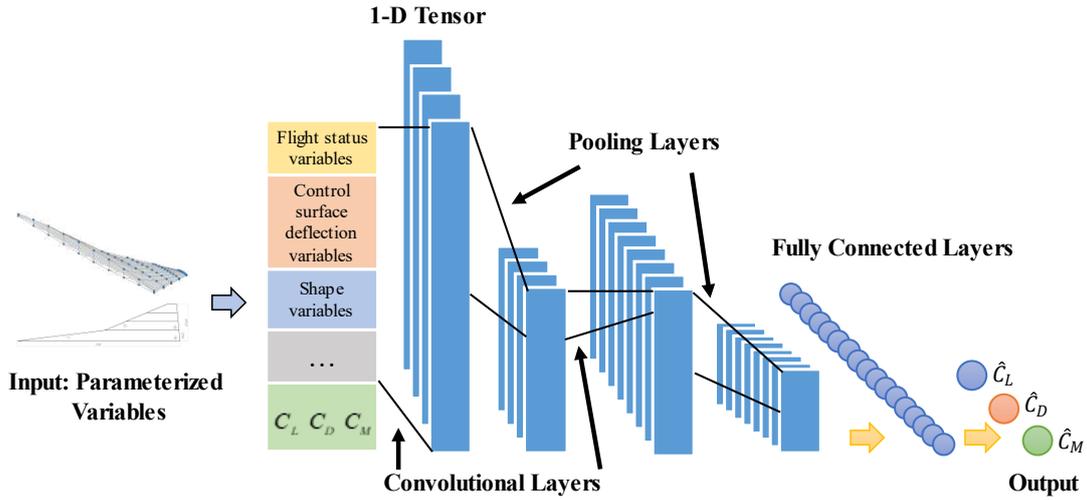
193

194

195

Different from fully connected neural network, CNN preserves the input's neighborhood relations and spatial locality in their latent higher-dimensional feature representations. Meanwhile, the number of free parameters describing their shared weights does not depend on the input dimensionality, which avoids the pressure caused by the surge of the parameters of high-dimensional input in neural network. The following section focuses on the convolutional layer and pooling layer. The structure of the fully-connected layer is consistent with that of an aforementioned MLP.

196 Inspired by the conventional 2-D CNN, we propose the modified structure
 197 (Figure 3) which takes the 1-D tensor for parameterized variables including shape var-
 198 iables (defined by free-form deformation method, class-function shape-function trans-
 199 formation, etc. methods), flight status variables (such as Mach number, angle of attack,
 200 Reynolds number, flight altitude, etc.), and control surface deflection variables of the
 201 wing as model inputs. The outputs are the prediction of aerodynamic characteristics
 202 such as \hat{C}_L , \hat{C}_D , and \hat{C}_M .



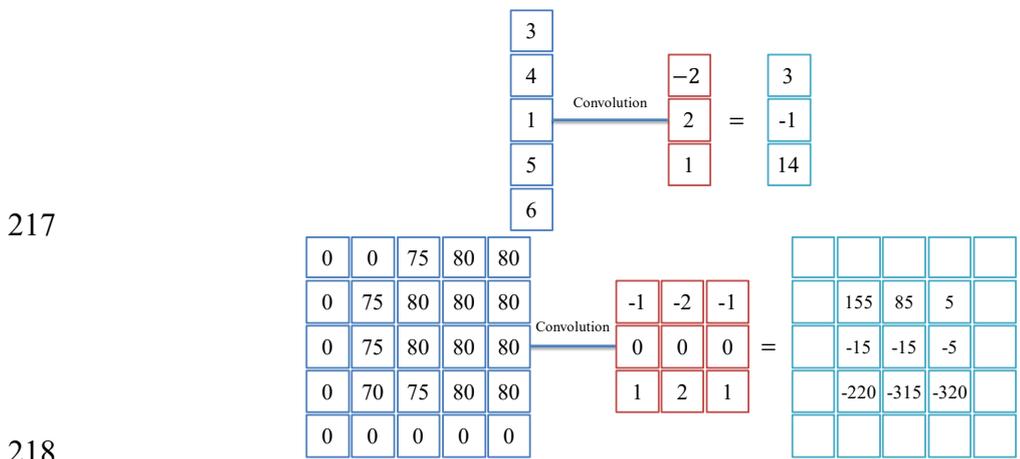
203
 204 **Figure 3 Modified CNN architecture for aerodynamic data modeling**

205 2.2.1. Convolutional Layer

206 The convolution layer processes the input data by a convolution filter and distill
 207 the local information and global information. If a two-dimensional input data is taken
 208 as the input of convolution layer (recorded as I and its coordinate is (m, n)), the con-
 209 volution filter is a two-dimensional matrix (recorded as K), and the obtained output
 210 data is a two-dimensional matrix identified as S and its coordinate is (i, j) . There-
 211 fore, the process of convolution can be expressed by the following formula:

$$212 \quad S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (5)$$

213 To be specific, as shown in Figure 4, A one-dimensional tensor with size of 5×1
 214 will output a one-dimensional tensor with size of 3×1 after convolution with a filter
 215 of size of 3×1 . A two-dimensional tensor with size of 5×5 will output a two-
 216 dimensional tensor with size of 3×3 after convolution with a filter of size of 3×3 .



218 **Figure 4 Instances of 1D convolution (top) and 2D convolution (bottom)**

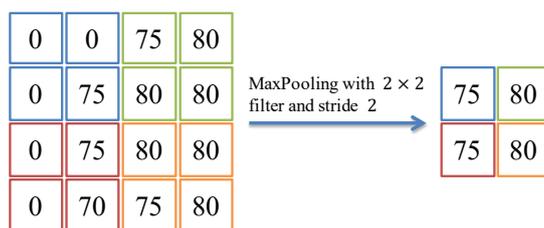
219 Different from the full connectivity of MLP, only the dynamic connection be-
 220 tween the convolutional filters and the neurons covered by the receptive field is estab-
 221 lished, which significantly reduces the number of parameters that need to be trained of
 222 CNN, and contributes to alleviate overfitting and the improvement of training effi-
 223 ciency.

225 2.2.2. Pooling Layer

226 It is common to periodically insert a pooling layer in-between successive convo-
 227 lutional layers in a CNN architecture. Its function is to progressively reduce the spa-
 228 tial size of the representation to reduce the amount of parameters and computation in
 229 the network, and accordingly to also control overfitting^{[41]-[42]}. The common pooling
 230 operations are maximum pooling and average pooling.

231 Maximum pooling keeps the maximum value in the pooling filter's receptive
 232 field and transfers it to the next layer, while discarding the rest of the values and mov-

233 ing the filter with a given stride to the next local region to do the same operation. For
234 a tensor with size of 4×4 , it will be processed into a tensor with size of 2×2 after the
235 maximum pooling with the 2×2 filter and stride 2. An instance is shown in Figure 5.
236 Note that the size of the tensor decreases by half through 2×2 pooling filter with the
237 stride of 2. Similarly, average pooling is an operation that selects the average value of
238 the region and pass it to the next layer.



239
240

Figure 5 An instance of maximum pooling

241 2.2.3. Fully Connected Layer

242 The pooling layer is usually followed by several fully connected layers, whose
243 structure is the same as that of MLP. The depth and width of the fully connected layer
244 are defined according to the complexity of the problem and the size of the data.

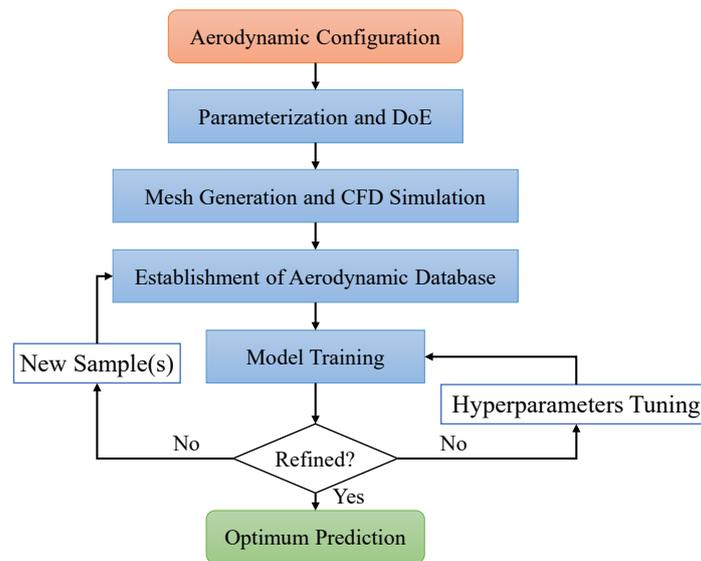
245 3. CNN-based Aerodynamic Data Modeling Method

246 The steps of the aerodynamic data modeling method using CNN, shown in Fig-
247 ure 6, can be described as follows:

248 a) Parameterization and design of experiments (DoE): Specify the parameter
249 space by defining the input variables and their range; Generate samples based on DoE
250 theory.

251 b) Mesh generation and CFD simulation: Set up flow field computational block
252 and meshing; obtain the aerodynamic data at the sample points by conducting CFD
253 simulation.

254 c) Establishment of aerodynamic database preparation: Organize the design
 255 space and its corresponding aerodynamic characteristics into an aerodynamic database.
 256 d) Model training: Update model parameters by optimization algorithm iterative-
 257 ly.
 258 e) Optimum prediction: Output the optimum prediction if achieving required ac-
 259 curacy, otherwise refine the model through sample augment and hyperparameter ad-
 260 justment.



261
 262

Figure 6 Flowchart of aerodynamic data modeling

263 A sufficient number of samples are required to build a data-driven model with
 264 reasonable accuracy. To obtain enough samples for training the model, we compute
 265 the aerodynamic force coefficients by CFD simulation, such as lift coefficient, drag
 266 coefficient, and pitching moment coefficient corresponding to different flight status
 267 (with a given range for free-stream Mach number and angle of attack) and different
 268 shapes described by the wing planar variables and profile variables.

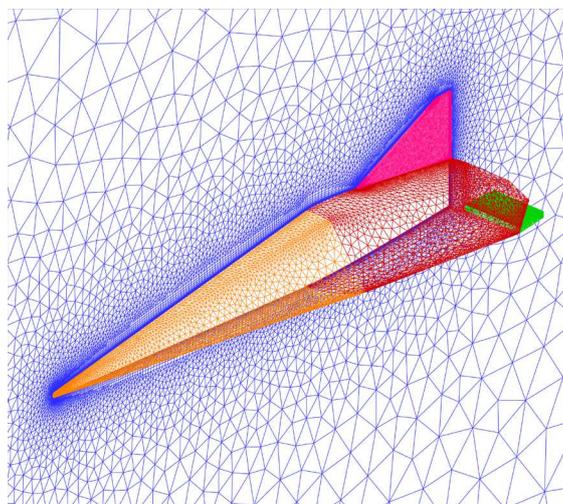
269 The wing of the Sanger aerospace plane carrier (the configuration of the aircraft
 270 is shown in Figure 9) is employed as a test case to validate the prediction capability of
 271 the CNN model. The following content details 5 procedures to get the aerodynamic

272 data and build the CNN.

273 3.1. Validation of RANS FlowSolver

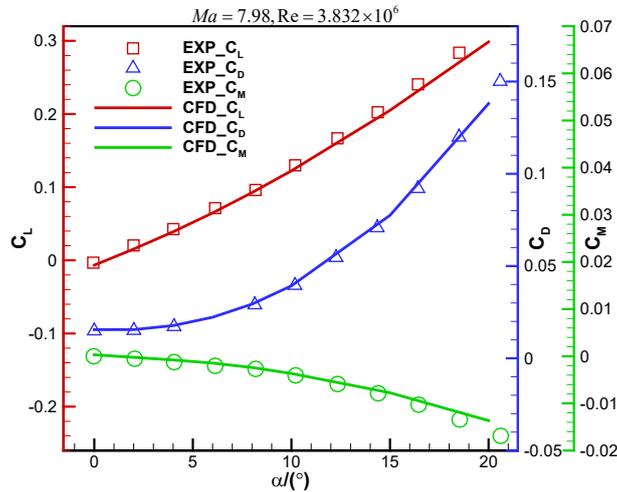
274 Before implementing CFD simulation to obtain aerodynamic data, the flow solv-
275 er used in this article needs to be validated. To validate the RANS solver, it is used to
276 simulate the hypersonic flow over the FDL-5A configuration. An AUSM (advection
277 upstream splitting method) scheme is used for spatial discretization and a $k - \omega SST$
278 turbulence model is adopted for turbulent closure. The sketch of unstructured compu-
279 tational mesh is shown in Figure 7 and the number of mesh cells is 0.48 million.

280 The hypersonic flow over FDL-5A is simulated at the status of
281 $Ma = 7.98, Re = 3.832 \times 10^6$. Figure 8 shows the comparison of computed force coeffi-
282 cient and experimental data. It can be seen that the lift, drag, and pitch moment coef-
283 ficients achieved a good match with the experimental data at different angles of attack.
284 Although the calculated Mach number for the RANS solver verified here is 7.98
285 above that of data for aerodynamic modeling, they are both in hypersonic regime.
286 Consequently, we think that it makes sense to validate the RANS solver for the CFD
287 simulation used to model.



288
289

Figure 7 Sketch of computational mesh for FDL-5A configuration

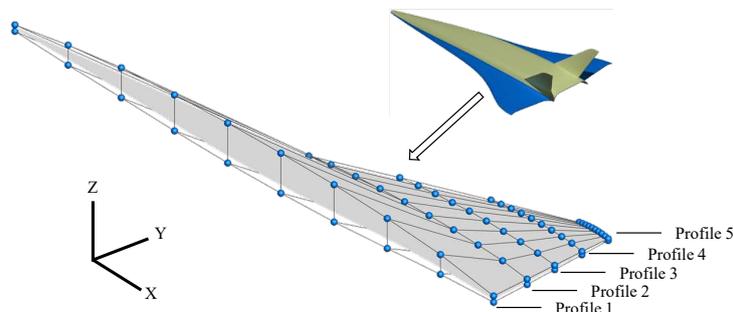


290
291
292

Figure 8 Comparison of computed force coefficients and experimental data for FDL-5A configuration

293 *3.2. Parameterization*

294 The wing is parameterized into profile variables with 100 control points by the
295 free-form deformation (FFD) method at 5 variable locations, and planar shape varia-
296 bles with 7 configuration variables of the wing. As shown in Figure 9, five profiles of
297 the wing are parameterized by the FFD method. There are 10 control points at the up-
298 per and lower sides respectively. Figure 10 shows 7 design variables used to parame-
299 terize the planar shape of the wing, namely, root chord, leading-edge sweep angle of
300 the inner wing segment, leading-edge sweep angle of the outer wing segment, trailing-
301 edge sweep angle of the inner wing segment, trailing-edge sweep angle of the outer
302 wing segment, span at kink, and wingspan.



303
304
305

Figure 9 Schematics of FFD control points in 5 wing profiles of Sanger aerospace plane carrier

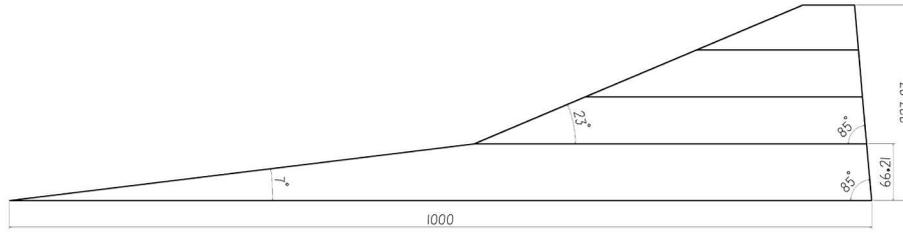


Figure 10 Schematic of parameterization for the planar shape of the wing of Sanger aerospace plane carrier

In total, there are 107 variables describing this configuration and 2 flight status variables (free-stream Mach number and angle of attack). The boundaries of these variables are presented in Table 2. In particular, the 100 wing profile variables (Index $\in [10,109]$) represent the Z-coordinates of the FFD control points which are varied from -10% to 10% on the benchmark of the wing.

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

Type	Variable	Index	Lower boundary	Upper boundary
Flight status	Mach number	1	5	6
	Angle of attack	2	0°	5°
Wing planar	Root chord	3	900	1100
	Span in kink	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°
Wing profile	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°
	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			

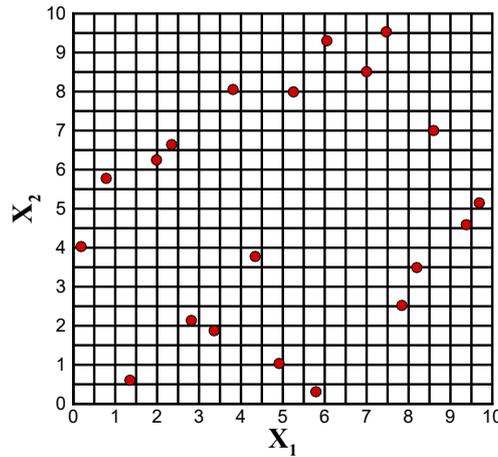
3.3. Design of Experiment

Latin hypercube sampling (LHS) is used as the design of experiments (DoE)

317 method to establish the distribution of input variable. LHS is a method of approximate
 318 random sampling from multivariate parameter distribution, which belongs to hierar-
 319 chical sampling technology and is often used in DoE. Samples x_j^i obtained by using
 320 the LHS method can be expressed as follows:

$$321 \quad x_j^i = \frac{\pi_j^i + U_j^i}{N}, 1 \leq j \leq d, 1 \leq i \leq N \quad (6)$$

322 Where i denotes the i -th samples, j denotes the j -th design variable, U denotes a ran-
 323 dom number in $[0,1]$, π_j denotes random permutation in $\{0,1,\dots,N-1\}$. As shown in
 324 Figure 11 is an instance of 20 sample points selected by the LHS method in a DoE
 325 problem of 2-dimensional input.



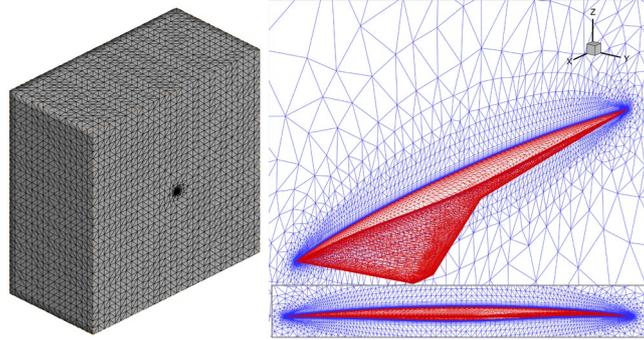
326
 327 **Figure 11 Schematic of 20 sample points selected by Latin hypercube sampling**

328 *3.4. Mesh Generation and CFD Simulation*

329 In this article, the mesh reconstruction method is used to generate meshes for dif-
 330 ferent shapes. A computational mesh is shown in Figure 12. The flow field is separ-
 331 ated by an unstructured mesh with the amount of about 0.48 million cells.

332 CFD simulation is conducted using RANS equations, and a two-equation
 333 $k-\omega SST$ turbulence model is adopted for turbulent closure. Finally, we build an

334 aerodynamic dataset consisting of 7431 sample points.



335

336 **Figure 12 Sketch of computational mesh for the wing of Sanger aerospace plane carrier**

337 3.5. CNN Training

338 Optimization for network weights and bias of the CNN is performed by using
339 backpropagation algorithm^[34]. As for the regression problem, the mean square error
340 (MSE) loss function of the model is expressed as followed:

$$341 \quad L = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (7)$$

342 Where N denotes the number of sample points in the training dataset. y_i is the nu-
343 merical simulation values calculated by CFD simulation, and \hat{y}_i denotes the predic-
344 tion values.

345 The dataset is divided into the training dataset and the test dataset in a ratio of
346 19:1. The reason for not using the validation dataset is that the CNN is trained in this
347 article without the training techniques requiring the validation dataset, such as early
348 stopping and dynamic adjustment of the learning rate, etc. Hence, there are 7059 wing
349 sample points fed to tune the parameters of the network and 372 sample points used to
350 test the predicted accuracy of CNN.

351 The Adam^[43] method is used to optimize the model to approximate the mapping
352 underlying the input data. The initial learning rate is set at 0.0001. The initial batch

353 size is set to 128. The training procedure is carried out based on a GPU (NVIDIA
 354 RTX 3080).

355 4. Results and Discussion

356 4.1. Influence of CNN Hyperparameters

357 The training and prediction performance of neural networks are directly affected
 358 by hyperparameters. Hence, in this section, we investigate several hyperparameters
 359 for CNN. Better hyperparameters can be found by observing the convergence perfor-
 360 mance of the loss function on the training dataset and test dataset.

361 4.1.1. Number of Convolutional Layers

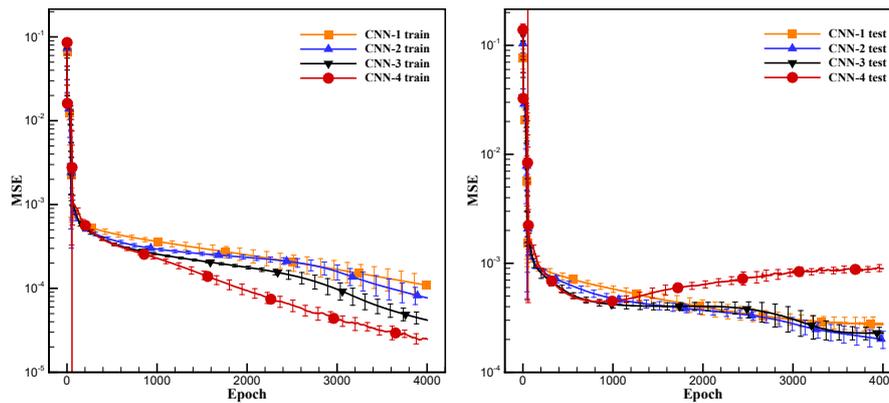
362 We set up four CNN networks, CNN-1, CNN-2, CNN-3, and CNN-4, whose
 363 numbers of convolutional layers are 1, 2, 3, and 4, respectively, and the whole net-
 364 work structure and its parameters are shown in Table 3. In the first convolutional layer
 365 of CNN-1, 16 indicates the number of filters, 3×1 indicates the convolutional filter
 366 size, and 2×1 indicates the pooling filter size. In the fully connected layer of CNN-1,
 367 3×128 indicates 3 layers with neurons of 128.

368 **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

369 Without loss of generality and in order to exclude the influence of the random

370 process on the training results, each CNN structure is trained ten times. The average
 371 values and standard deviation of MSE and is taken for comparison. Training is im-
 372 plemented for all four architectures, and the MSE convergence of the CNN is shown
 373 in Figure 13. The training histories of the four CNN structures with learning rate of
 374 0.00001 are given here. Although CNN-4 converges rapidly and performs better on
 375 the training dataset, a significant overfitting occurs, i.e., the MSE on the test dataset is
 376 larger, while the MSE on the test dataset increases as the MSE on the training set de-
 377 creases. Due to the minimum MSE, CNN-2 is selected as the benchmark architecture
 378 for further parameters studies.



379

380

Figure 13 MSE convergence history of CNN-1, CNN-2, CNN-3, and CNN-4

381

4.1.2. Number of Convolutional Filters

382

383

384

385

386

387

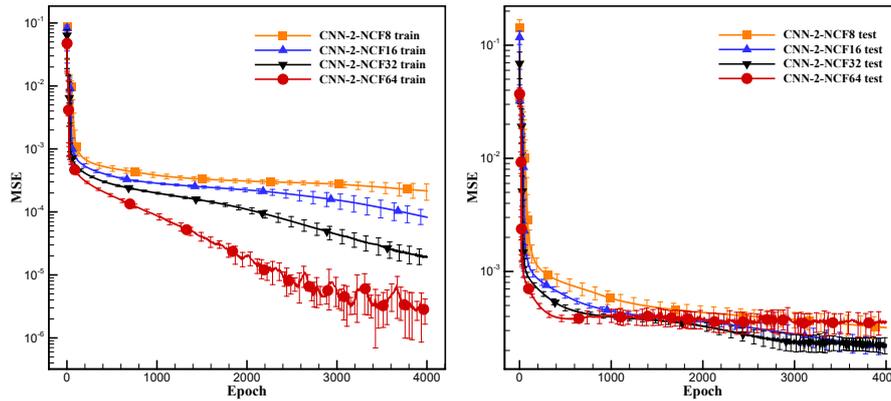
388

We investigate the influence of the number of filters by increasing the number of filters from 8 (NCF8) to 64 (NCF64) for the CNN-2's first convolutional layer and from 16 to 128 for the second convolutional layer, as detailed in Table 4. Figure 14 shows the effect of the number of filters on the MSE convergence of the model. It can be seen that the increase in the number of filters shows a significant reduction in the training MSE, while a larger number of filters accelerates the MSE convergence. However, on the test dataset, the MSEs of NCF8 and NCF64 are slightly higher than

389 NCF16 and NCF32. The MSE of the NCF16 is minimum and tends to continue to de-
 390 crease. Therefore, NCF16 is chosen as the next setting for the study.

391 **Table 4 Variation of CNN-2 with 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	$8, 3 \times 1, 2 \times 1$	$16, 5 \times 1, 2 \times 1$	$32, 6 \times 1, 2 \times 1$	$64, 8 \times 1, 2 \times 1$
Second convolution	$16, 3 \times 1, 2 \times 1$	$32, 5 \times 1, 2 \times 1$	$64, 6 \times 1, 2 \times 1$	$128, 8 \times 1, 2 \times 1$
Output	3×1	3×1	3×1	3×1



392
 393 **Figure 14 MSE convergence history with different number of convolutional filters.**

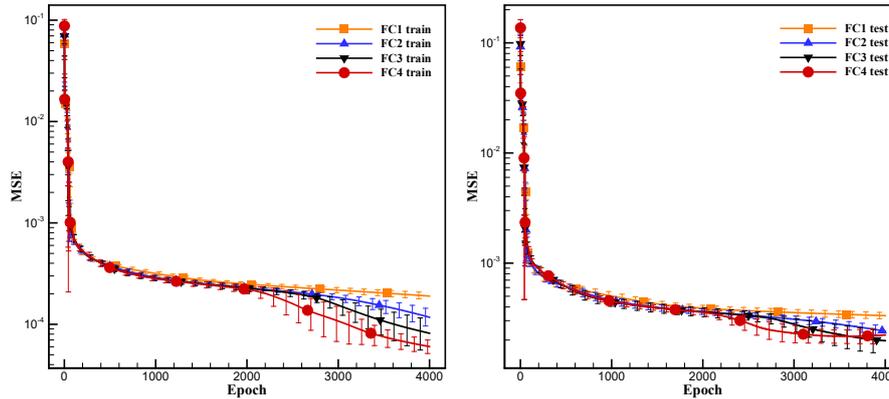
394 4.1.3. Number of Fully Connected Layers

395 To investigate the effect of the number of fully connected layers, CNN-2 is mod-
 396 ified to have one, two and four fully connected layers (CNN-2-FC1, CNN-2-FC3 and
 397 CNN-2-FC4), as shown in Table 5. The MSE convergence history is given in Figure
 398 15. More fully connected layers lead to improvement of MSE convergence, and CNN-
 399 2-FC4 has the best training MSE convergence, but it is CNN-2-FC3 that finally
 400 achieves the minimal MSE on the test dataset. Meanwhile CNN-2-FC3 has a tendency
 401 to decrease even further, which is better than CNN-2-FC4.

402
 403
 404

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

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



406

407

408

Figure 15 MSE convergence history of CNN-2 with different number of fully connected layers. (FC2 indicates two fully connected layers.)

409

4.1.4. Learning Rate

410

As a vital hyperparameter in the training of CNN, the learning rate (LR) largely

411

determines the model convergence efficiency and the final level of convergence. In

412

general, a larger and inappropriate learning rate will lead to the failure of the model to

413

converge, while a smaller and unsuited learning rate will bring a higher training time

414

cost and the risk of falling into a local optimum. The MSE convergence histories with

415

different learning rate (1×10^{-2} , 1×10^{-3} , 1×10^{-4} , and 1×10^{-5}) are shown in Figure 16.

416

It is observed from the results that LR0.01 and LR0.001 exhibit poor MSE conver-

417

gence and severe oscillations. This may be due to the fact that the optimization algo-

418

rithm skips the optimal path because of a large initial learning rate^[44]. Compared with

419

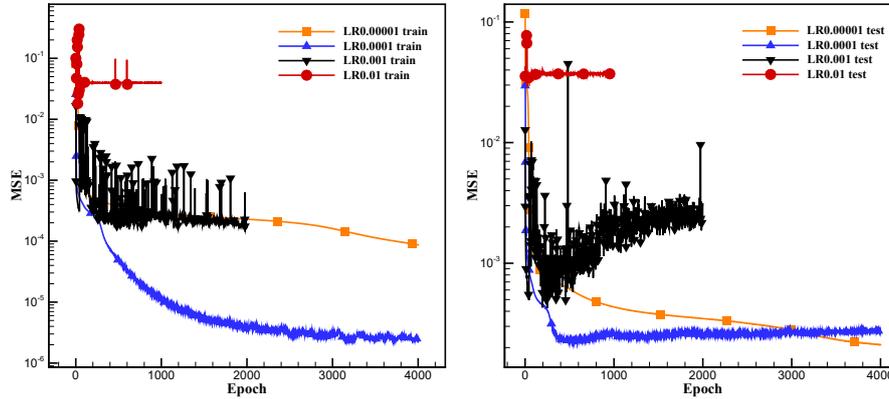
LR0.00001, LR0.0001 achieves faster convergence and then overfitting occurs, hence,

420

ultimately, we consider LR0.00001, which converges smoothly and still retains the

421

downward trend, to be the superior learning rate, despite its longest training time.

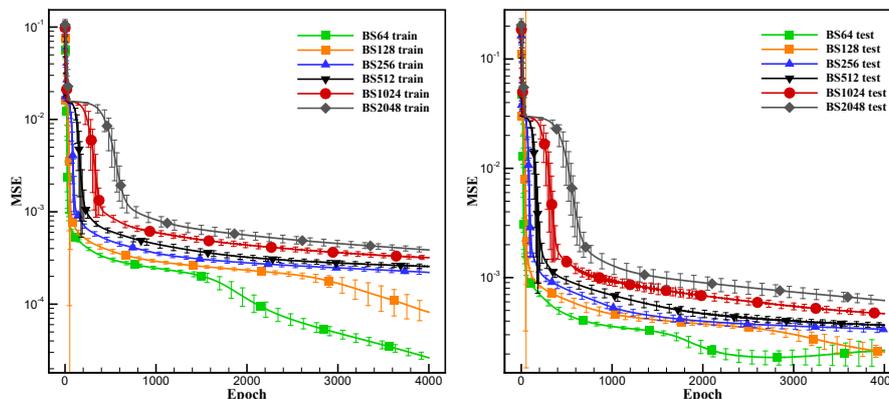


422
423

Figure 16 MSE convergence history of CNN-2 with different learning rate

424 4.1.5. Batch Size

425 Furthermore, study on the effect of batch size (BS) on MSE convergence is car-
 426 ried out in a series of batch size decreasing from 2048 (BS2048) to 64 (BS64) by a
 427 factor of 2. The impact of BS is mainly for the amount of computation, i.e., a larger
 428 BS produces faster computation but requires access to more examples to get the same
 429 error, since there are fewer updates per epoch^[45]. From the Figure 17, we can see that
 430 a smaller batch size leads to a smaller MSE in training dataset. Since BS128 has the
 431 minimum MSE in the test dataset and still retains the descent gradient, we use a mini-
 432 batch of 128 as the optimum for the training.



433
434

Figure 17 MSE convergence history of CNN-2 with different batch size

435 In addition, the effect of convolutional filter size on the CNN training is also in-
 436 vestigated and the results shows that its influence is very small, so the commonly used

437 filter size of 3×3 is used.

438 4.2. Validation of CNN prediction performance

439 Based on the aforementioned investigation, we finally adopt a CNN containing
440 two convolutional layers with a filter size of 3×1 , two pooling layers with a filter size
441 of 2×1 and three fully connected hidden layers with 128 neurons in each layer as the
442 model structure with 16 convolutional filters in first convolutional layer, 32 convolu-
443 tional filters in second convolutional layer. Learning rate and batch size are set to
444 0.00001 and 128 respectively. In order for the MSE to fully converge, an epoch of
445 6000 is used in the final training instead of the epoch of 4000 for less time cost during
446 the investigation of the influence of hyperparameters.

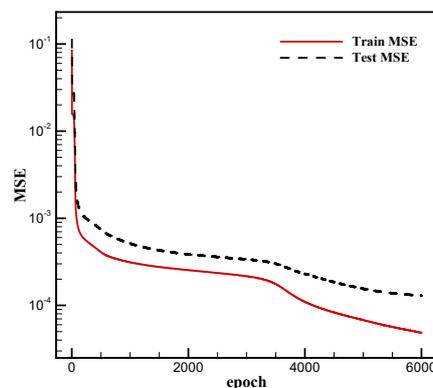
447 To validate the prediction performance of CNN. The relative error ε , the coeffi-
448 cient of determination R^2 , relative root-mean-square error (*RRMSE*), and relative
449 maximum absolute error (*RMAE*) are employed as the metrics and their formulas are
450 shown below.

$$\begin{aligned} \varepsilon &= \sum_{i=1}^N \frac{\|y_i - \hat{y}_i\|_2}{N \times \|y_i\|_2} \times 100\% \\ R^2 &= 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y}_i)^2}, \quad R^2 \in (-\infty, 1] \\ \text{451 } RRMSE &= \frac{1}{STD} \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (8) \\ RMAE &= \frac{1}{STD} \max |y_i - \hat{y}_i| \\ STD &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y}_i)^2}, \quad i = 1, 2, \dots, N \end{aligned}$$

452 where y_i is the simulation values of the i -th test sample calculated by CFD simulation.

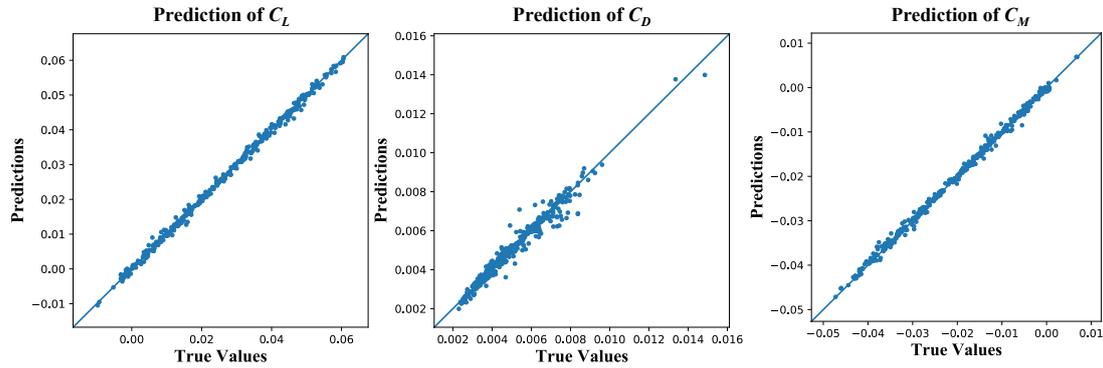
453 \hat{y}_i denotes the predicted value of the i -th test sample. N is the number of test sample.
454 The model is perfectly accurate when $R^2 = 1.0$, whereas $R^2 = 0.0$ indicates a bad ap-
455 proximation. The $RRMSE$ can reflect the global accuracy of the model and $RMAE$ is a
456 criterion that can indicate the local predicting performance.

457 Figure 18 shows the MSE convergence history of CNN training. It can be seen
458 that the MSE of both the training dataset and the test dataset decrease smoothly, and
459 no oscillation or overfitting. Figure 19 shows the comparison of prediction values ver-
460 sus CFD simulation value for the three aerodynamic coefficients in the test dataset.
461 Almost all sample points of the test dataset are clustered near the 45° line, which il-
462 lustrates the very reliable prediction error of the CNN. Among them, the performance
463 of prediction for C_L and C_M is better, while the C_D is relatively poor. The distribu-
464 tion plot of absolute error is shown in Figure 20. The y-axis indicates the number of
465 samples corresponding to the absolute error in x-axis. One can obviously see that the
466 error distribution is gaussian distribution. From Figure 20, it is clear that, for the pre-
467 diction of C_D in the test dataset, the errors are less than 10 counts (1 count indicates
468 the C_D of 0.0001) for the majority of the samples, and for the prediction of C_L and
469 C_M in the testing dataset, the errors are less than 0.002 for the majority of the samples.



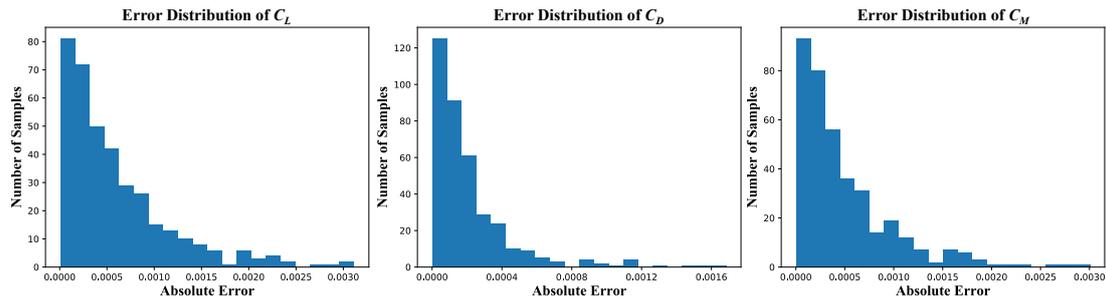
470
471

Figure 18 Convergence history of MSE for CNN



472
473

Figure 19 Prediction values versus CFD simulation values of C_L , C_D , and C_M

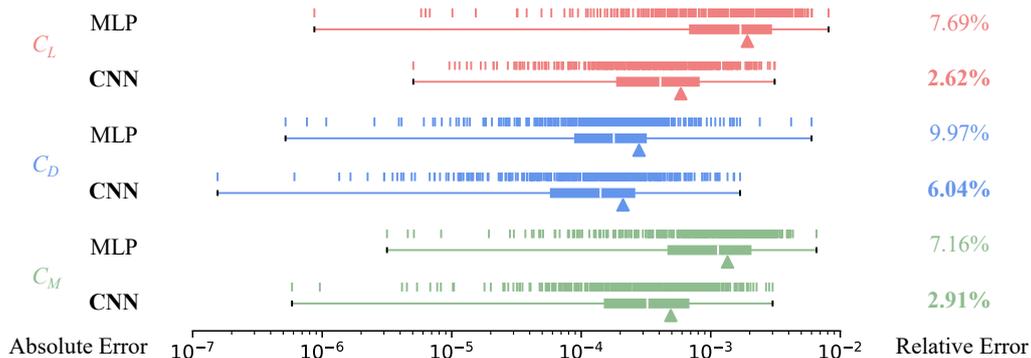


474
475

Figure 20 Absolute error distribution of CNN prediction for C_L , C_D , and C_M

476 As a comparison, we also train an MLP ten times and make predictions on the
 477 test dataset. Its structure consists of five fully connected hidden layers with neurons of
 478 128. The commonly used hyperparameters with a learning rate of 0.001, a batch size
 479 of 128, and an epoch of 6000 are adopted. Figure 21 shows the scatter plots of absolute
 480 error and the corresponding box plots. The scatter plot at the top provides the absolute
 481 error of predicted force coefficients at each configuration and flight status of the wing,
 482 and the box plot at the bottom provides the statistics for these absolute errors. The
 483 white line in the box plot indicates the median of these errors. The right and left edges
 484 of the box are the upper and lower quartile respectively. The location of the triangle
 485 below the box refers to the average of these errors. Specifically, for C_L , the largest
 486 error of CNN is about 0.0031. About 82.80% of the sample points are predicted by
 487 CNN with an error of less than 0.001. For C_D , the largest error of CNN is about 17

488 counts. The percentages of test sample points with an error less than 1 count, 5 counts,
 489 and 10 counts are 38.44%, 91.13%, and 97.58%, respectively. For C_M , the largest er-
 490 ror of CNN is about 0.0030. 86.83% of test sample points have an error of less than
 491 0.001. As a more detailed comparison (Table 6), we also compute three other accuracy
 492 metrics from which we can see that the CNN is clearly ahead of the MLP. In addition,
 493 compared to MLP with the same depth and width, the training process of CNN is less
 494 computationally intensive and takes less time to train due to the fewer parameters
 495 brought by the parameter sharing and local connectivity. The task is deployed on a
 496 GPU of NVIDIA RTX 3080 and the training time for the MLP is 1280.35s, while the
 497 CNN only takes 869.97s, yielding an improvement of about one-third in efficiency.



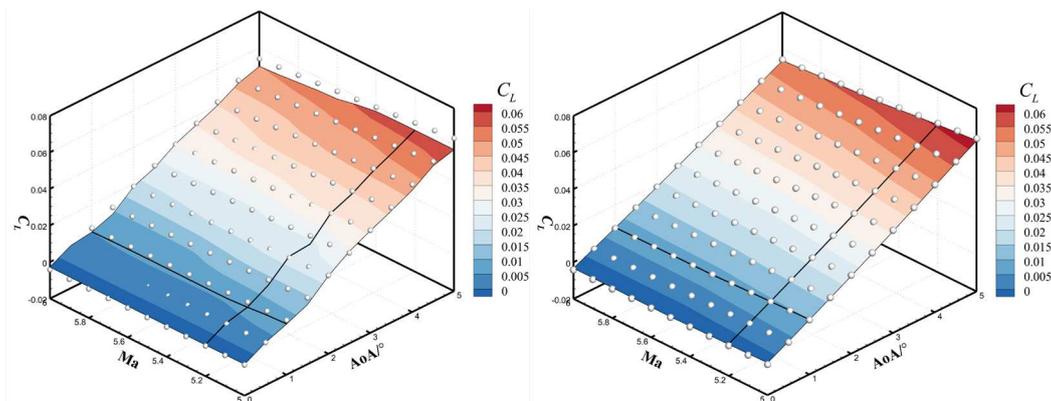
498
 499 **Figure 21 Comparison of the box plot and absolute error of prediction obtained by MLP and CNN**

500 **Table 6 Comparison of accuracy metrics for predicting aerodynamic performance by**
 501 **MLP and CNN in high-dimensional aerodynamic modeling with respect to flight status**
 502 **and wing shape variables**

Object	Model	Relative error ε	R^2	$RRMSE$	$RMAE$	Training time/s
C_L	MLP	7.69%	0.9813	0.1366	0.4638	1280.35
	CNN	2.62%	0.9978	0.0466	0.1779	869.97
C_D	MLP	9.97%	0.9076	0.3036	3.3977	1280.35
	CNN	6.04%	0.9661	0.1840	0.9539	869.97
C_M	MLP	7.16%	0.9815	0.1357	0.5212	1280.35
	CNN	2.91%	0.9970	0.0551	0.2391	869.97

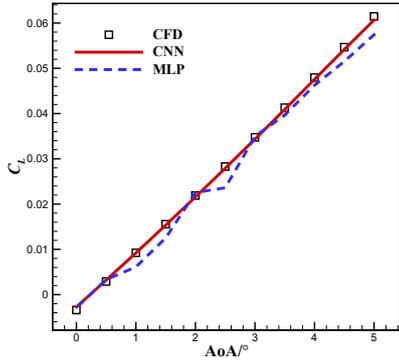
503 We also probe the ability of the trained MLP and CNN model to predict aerody-

504 namic coefficients of the baseline configuration at different flight status such as Mach
505 number and angle of attack. 121 additional sample points obtained by uniform sam-
506 pling method are labeled by conducting CFD simulation. These test sample points are
507 marked with white spheres, as plotted in Figure 22, with Mach number ranging from 5
508 to 6 and angle of attack in the range of 0 to 5 degrees. The response surfaces predicted
509 by MLP and CNN are shown in Figure 22 (a), (d), and (g), respectively, from which
510 several curves are sliced to obtain the variation of lift coefficient, drag coefficient and
511 pitch moment coefficient with respect to Mach number or angle of attack. One can see
512 that CNN produces smoother response surfaces that conform to the flow mechanism
513 and a better match to CFD simulation values than MLP, although the prediction error
514 for the drag coefficient at the α of 1° is relatively large. Figure 23 shows the absolute
515 error contour plots between the CFD simulation values and predicted values of MLP
516 and CNN for the three aerodynamic coefficients. The absolute error of CNN is signif-
517 icantly smaller than that of MLP. As a more detailed comparison (Table 7), we also
518 compute four accuracy metrics from which we can see that the CNN is clearly ahead
519 of the MLP.

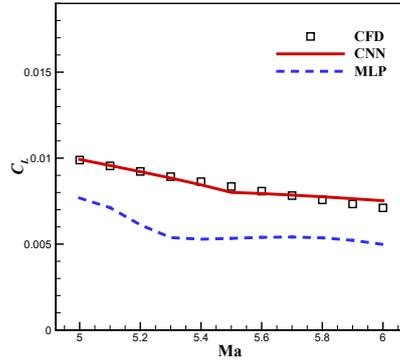


(a) Predicted C_L by MLP (left) and CNN (right)

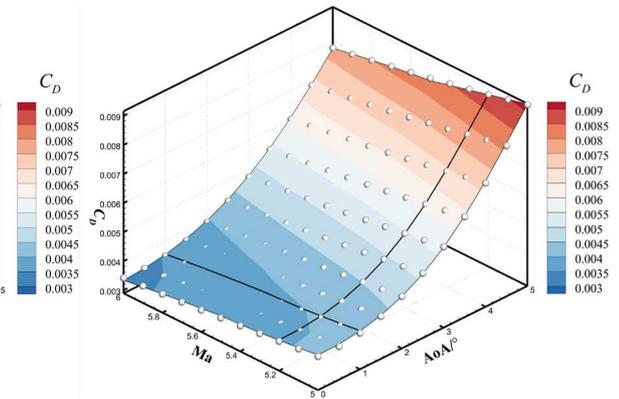
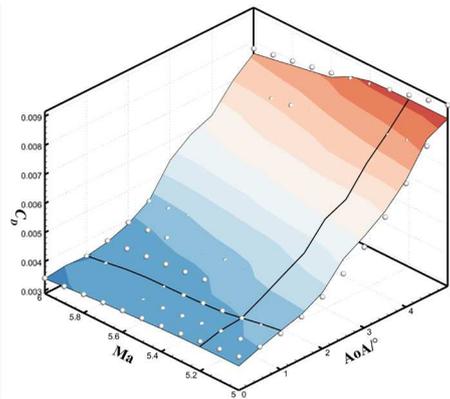
520
521



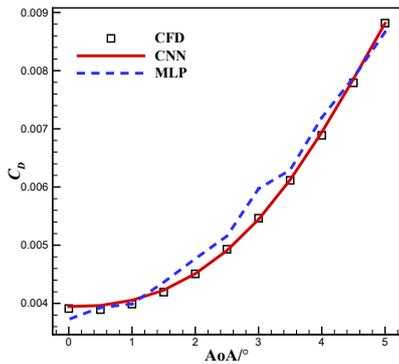
(b) Predicted C_L via MLP and CNN at $Ma = 5.2$



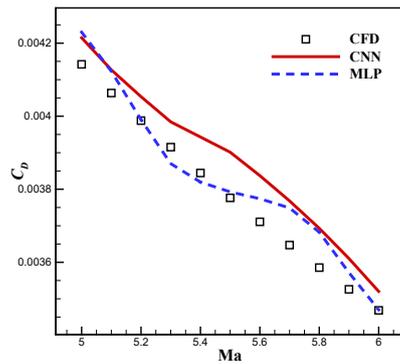
(c) Predicted C_L via MLP and CNN at $\alpha = 1^\circ$



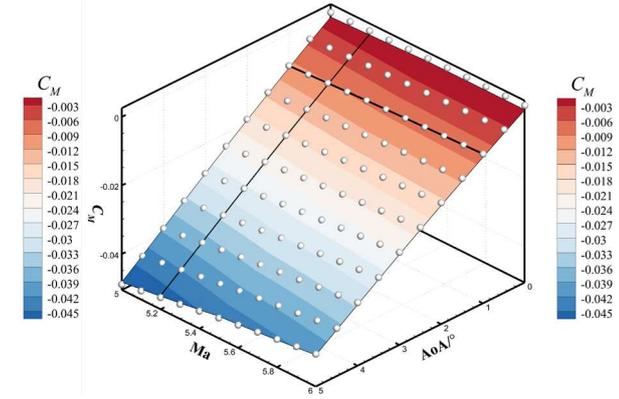
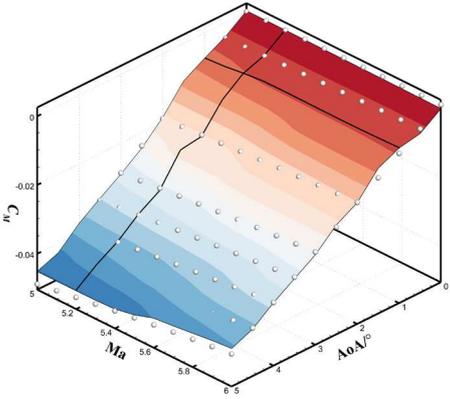
(d) Predicted C_D by MLP (left) and CNN (right)



(e) Predicted C_D via MLP and CNN at $Ma = 5.2$



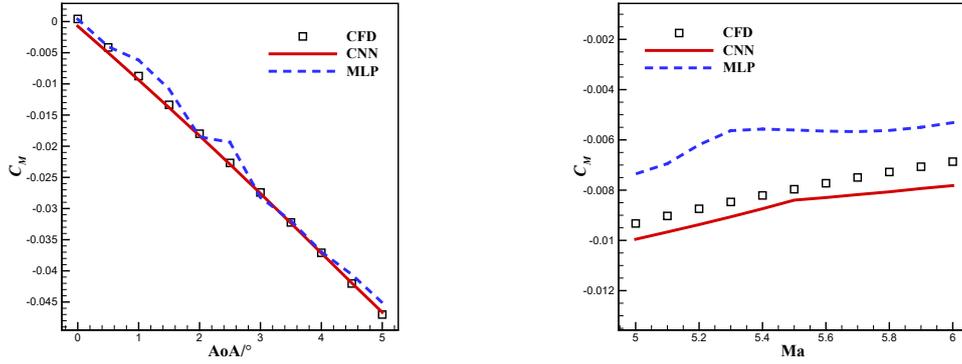
(f) Predicted C_D via MLP and CNN at $\alpha = 1^\circ$



(g) Predicted C_M by MLP (left) and CNN (right)

522
523

524
525



(h) Predicted C_M via MLP and CNN at $Ma = 5.2$ (i) Predicted C_M via MLP and CNN at $\alpha = 1^\circ$

526
527

Figure 22 Comparison of predicted aerodynamic coefficients using MLP and CNN with respect to Mach number and angle of attack

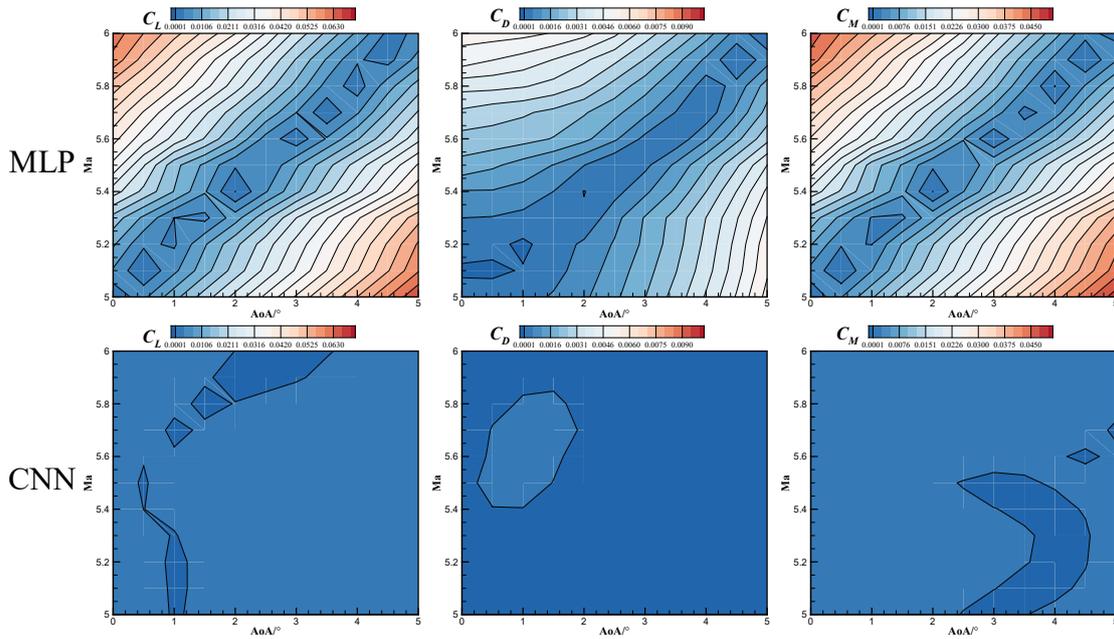


Figure 23 Comparison of absolute error contour of C_L (left), C_D (middle), and C_M (right) predicted by MLP versus CNN

528
529

530
531

Table 7 Comparison of accuracy metrics for predicting aerodynamic coefficients of the baseline wing by MLP and CNN at different Mach number and angle 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

532 4.3. Application to Fast Aerodynamic Shape Optimization

533 4.3.1 Optimization Problem Statement

534 To further demonstrate the modeling approach proposed in this paper, on the one
535 hand, genetic algorithm (GA) is implemented on the well-trained CNN to optimize
536 the five profiles of the wing to demonstrate its ability to support fast aerodynamic
537 shape optimization. On the other hand, three aerodynamic shape optimization cases
538 are preformed to verify the advantages of efficiency and convenience of the modeling
539 approach considering both flight status and shape variables in the case of multiple
540 flow status and optimization objectives. The details of the three optimization cases are
541 shown in Table 8, where t represents the maximum thickness of profiles. All the nota-
542 tions with subscript “0” mean baseline. The maximum thickness of the profile is also
543 constrained for the structural requirement.

544 Scaling of flow conditions from a 3D swept wing to a 2D profile is not consid-
545 ered here because the scaling rule for small or medium swept subsonic or transonic
546 wing may be not applicable to low aspect ratio and high swept hypersonic wing. Alt-
547 hough it is necessary to scale the flow conditions in a real-word design, it would still
548 make sense here to validate and demonstrate the applicability of CNN-based model-
549 ing method in efficient aerodynamic shape optimization.

550 **Table 8 Design cases for wing profile optimization**

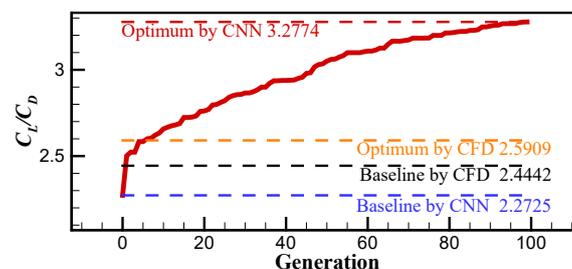
Optimization case	Optimization problem	Mach number	Altitude (km)	Reynolds number	Angle of attack (α)
1	$\max f_{obj}(x) = C_L / C_D$ $s.t. t - t_0 < 0.04t_0$	5.2	25	4.29×10^6	1°
2	$\max f_{obj}(x) = C_L / C_D$ $s.t. t - t_0 < 0.04t_0$	5.2	25	4.29×10^6	2°
3	$\min f_{obj}(x) = C_D$ $s.t. t - t_0 < 0.04t_0$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

551 Please be noticed that the five wing profiles selected as the baseline for optimiza-
552 tion already have been provided with good aerodynamic characteristics in hypersonic
553 regime, and thus we use adequate design variables incorporating 100 FFD control
554 points for a more refined optimization in anticipation of further improvement of aero-
555 dynamic performance.

556 4.3.2 Optimization Results

557 As can be seen from Figure 24 and Table 9, with the support of the CNN provid-
558 ing the correct optimization direction, the C_L / C_D of the baseline wing is boosted by
559 6.00% in optimization case 1, although the prediction of C_D is not very accurate re-
560 sulting in relatively poor predictions of C_L / C_D .

561 Figure 25 shows a comparison of geometric shape of baseline and optimized pro-
562 file. The maximum thickness of all profiles satisfies the constraints. The pressure co-
563 efficient contours and pressure distribution at three span-wise locations of the wing
564 using baseline and optimized wing profile are shown in Figure 26. A slightly larger
565 high-pressure regime is observed on the lower surface of the wing using optimized
566 profiles. It can be seen that the optimized wing profiles and pressure coefficient con-
567 tours are very similar to that of baseline, which is due to the fact that the baseline is
568 already a very good wing in hypersonic regime.



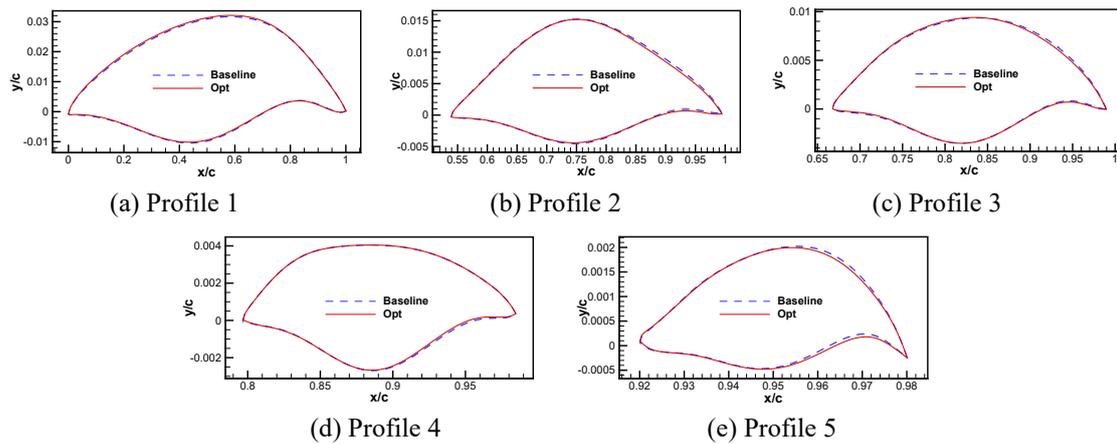
569
570 **Figure 24 Convergence history of aerodynamic shape optimization for the wing (optimi-**
571 **zation case 1)**

572
573

Table 9 Comparison of aerodynamic performance obtained from CNN prediction and CFD validation for baseline and optimized wing (optimization case 1)

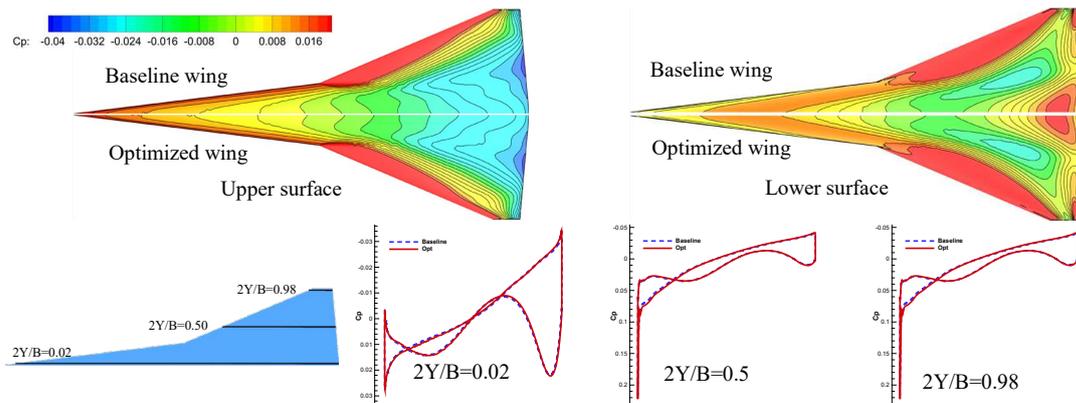
		C_L	C_D	C_L / C_D	C_M
CNN prediction	Baseline wing	0.00921	0.00405	2.2725	-0.00937
	Optimized wing	0.01095	0.00334	3.2774	-0.01016
	Δ	+18.89%	-17.53%	+44.22%	+8.43%
CFD validation	Baseline wing	0.00952	0.00389	2.4442	-0.00897
	Optimized wing	0.00984	0.00380	2.5909	-0.00914
	Δ	+3.38%	-2.48%	+6.00%	+1.92%

574



575

Figure 25 Comparison of five baseline and optimized wing profiles (optimization case 1)



576
577
578

Figure 26 Comparison of pressure coefficient contours and sectional pressure coefficient distribution at three span-wise locations of the wing using baseline and optimized profiles (optimization case 1)

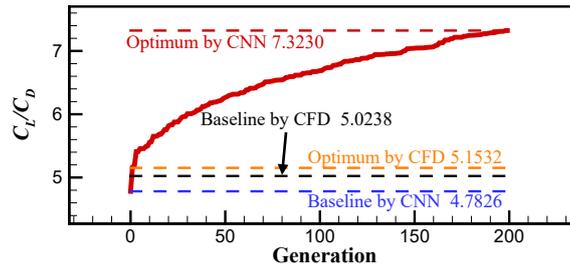
579

For the optimization case 2, as can be seen from Figure 27 and Table 10, the

580

C_L / C_D of the baseline wing is boosted by 2.57%. Figure 28 shows the comparison of

581 five baseline and optimized wing profiles. Pressure coefficient contours and the sec-
 582 tional pressure distribution are presented at Figure 29.

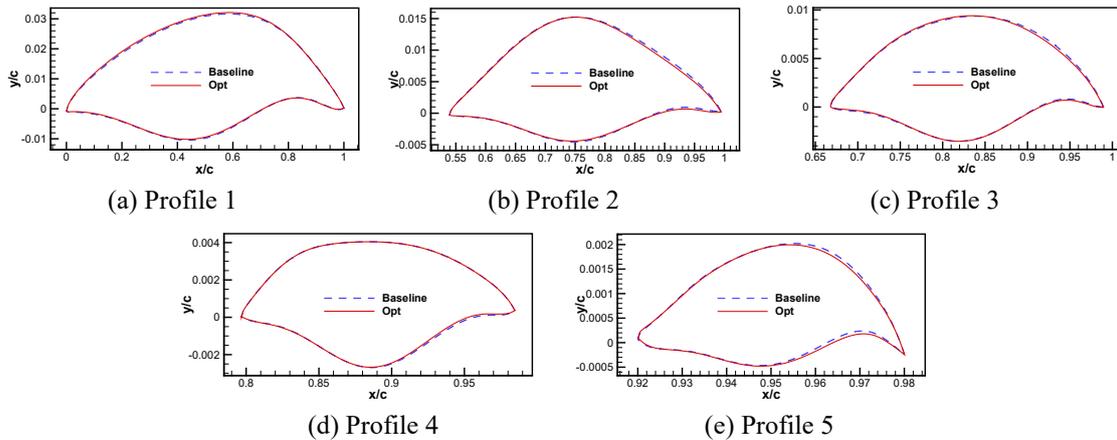


583
 584 **Figure 27 Convergence history of aerodynamic shape optimization for the wing (optimi-**
 585 **zation case 2)**

586 **Table 10 Comparison of aerodynamic performance obtained from CNN prediction and**
 587 **CFD validation for baseline and optimized wing (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	0.02385	0.00326	7.3230	-0.01934
	Δ	+10.56%	-27.79%	+53.12%	+5.61%
CFD validation	Baseline wing	0.02221	0.00442	5.0238	-0.01825
	Optimized wing	0.02256	0.00438	5.1532	-0.01842
	Δ	+1.57%	-0.98%	+2.57%	+0.99%

588



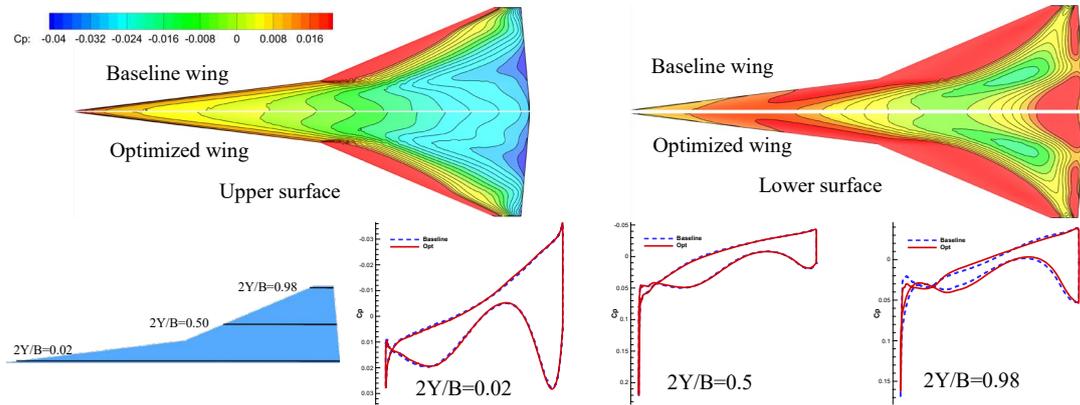
589 **Figure 28 Comparison of five baseline and optimized wing profiles (optimization case 2)**

590

591

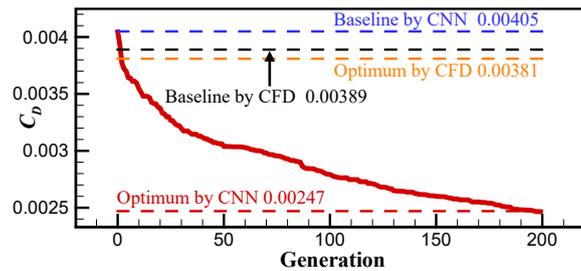
592

593



594 **Figure 29 Comparison of pressure coefficient contours and sectional pressure coefficient**
 595 **distribution at three span-wise locations of the wing using baseline and optimized pro-**
 596 **files (optimization case 2)**

597 For the optimization case 3, from Figure 30 and Table 11, one can see that the
 598 C_D of the optimized wing is reduced by 2.16%, while the C_L is also improved. A
 599 comparison of the five baseline and optimized profiles of the wing is given in Figure
 600 31. As we can see, the thicknesses of five wing profiles are changed with varying de-
 601 grees of decrease. Figure 32 shows the pressure coefficient contour and the sectional
 602 pressure coefficient distribution.



603
 604 **Figure 30 Convergence history of aerodynamic shape optimization for the wing (optimi-**
 605 **zation case 3)**

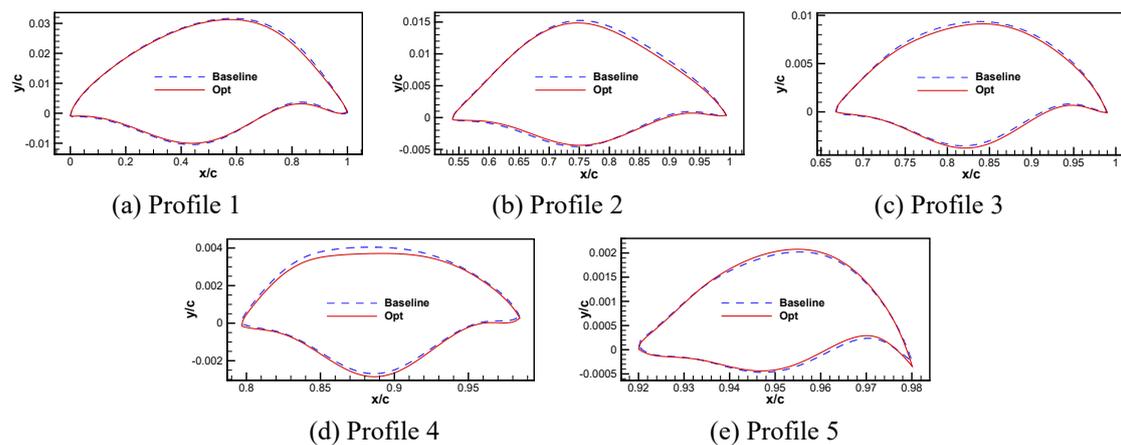
606
 607
 608
 609
 610
 611

612
613

Table 11 Comparison of aerodynamic performance obtained from CNN prediction and CFD validation for baseline and optimized wing (optimization case 3)

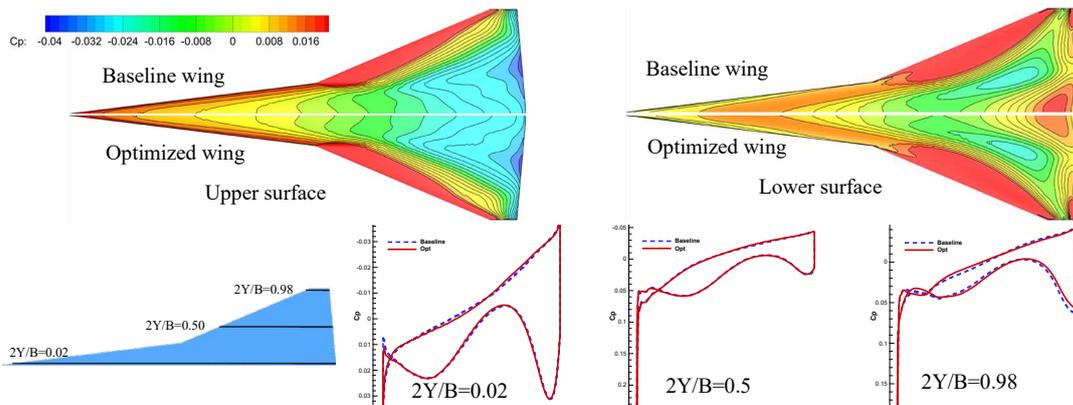
		C_L	C_D	C_L / C_D	C_M
CNN prediction	Baseline wing	0.00921	0.00405	2.2741	-0.00937
	Optimized wing	0.01001	0.00247	4.0582	-0.00977
	Δ	+8.65%	-39.12%	+78.46%	+4.32%
CFD validation	Baseline wing	0.00952	0.00389	2.4442	-0.00897
	Optimized wing	0.00975	0.00381	2.5589	-0.00908
	Δ	+2.44%	-2.16%	+4.69%	+1.23%

614



615

Figure 31 Comparison of five baseline and optimized wing profiles (optimization case 3)



616
617
618

Figure 32 Comparison of pressure coefficient contours and sectional pressure coefficient distribution at three span-wise locations of the wing using baseline and optimized profiles (optimization case 3)

619 4.3.3 Discussion

620

Table 12 shows the parameters of GA and the time cost in three optimization

621 cases for the wing, and it can be seen that the CNN-based high-dimensional aerody-
622 namic modeling method considering the flight status and shape variables can quickly
623 yield better aerodynamic shapes for all three optimization cases, i.e., different design
624 points and different optimization problems.

625 **Table 12 Parameter of GA and the time cost in three optimization cases**

Optimization case	Generation	Evaluation	Total time used (Intel core i9 11900K @3.50Hz)
1	100	5000	2 minutes 47 seconds
2	200	10000	5 minutes 51 seconds
3	200	10000	5 minutes 53 seconds

626 From the results of the three optimization cases, the aerodynamic coefficient
627 prediction error of the CNN for the shape that perturbs near the baseline is somewhat
628 larger than the global error, which reflects the fact that the amount of current database
629 used for modeling in the design space containing up to a hundred variables is not yet
630 sufficient and cannot give very accurate predictions, but the model can give the cor-
631 rect optimization orientation. Hence, It still makes sense to demonstrate the advantage
632 of the proposed modeling approach in high-dimensional aerodynamic modeling and
633 the ability to achieve fast aerodynamic shape optimization for multiple flight statuses
634 and optimization problems, although the improvement in aerodynamic characteristics
635 of our optimized shape is not very significant due to the fact that the chosen baseline
636 wing already has good aerodynamic characteristics in hypersonic regime resulting in
637 little room for improvement.

638 **5. Conclusions**

639 In this article, a CNN-based machine learning approach for high-dimensional
640 aerodynamic data modeling is proposed to provide fast and reliable aerodynamic per-

641 formance prediction. This modeling approach is demonstrated by a test aerodynamic
642 modeling case similar to the wing of Sanger aerospace plane carrier with 109-
643 dimensional input variables incorporating the flight status and aerodynamic shape var-
644 iables. Some conclusions can be drawn as follows.

645 a) MLP adapts to the augment of samples by expanding the network scale when
646 dealing with high-dimensional problems, which renders the reduction of training effi-
647 ciency and the trend of overfitting. The CNN with the advantages of sparse connec-
648 tion and weight sharing alleviates this problem.

649 b) How well the network converges depends largely on the hyperparameters of
650 the CNN. A learning rate (LR) of varying from 1×10^{-5} to 1×10^{-4} will be a better
651 choice, with higher LR leading to oscillation (even a failure to converge) and smaller
652 LR leading to higher training cost. Increasing the number of convolutional layers
653 brings about an enhancement of the ability to distill information, but too deep struc-
654 ture is prone to overfitting. Smaller batch size gives rise to faster MSE convergence
655 and greater computational intensity for the same epoch.

656 c) Compared with MLP, the CNN-based modeling method is dramatically more
657 accurate and efficient, not only for high-dimensional modeling problem with respect
658 to both aerodynamic shape variables and flight status variables, but also for response
659 surface prediction at different Mach number and angle of attack.

660 d) The surrogate model considering both flight status and shape enables fast aer-
661 odynamic shape optimization for multiple flight status without the need to conduct
662 expensive CFD simulations to build additional surrogate models.

663 The CNN-based aerodynamic modeling approach emerges as a gradient-free, fast
664 and accurate tool that complements other traditional methods in aerodynamic research

665 and provides a novel way of thinking for design optimization in practice. Furthermore,
666 targeted expansion for sample points and refinement in modeling method to improve
667 the prediction accuracy is conducive to enhance the quality of the optimization solu-
668 tion, which is the future research work.

669 **Declarations**

670 **Availability of Data and Materials**

671 The datasets generated during the current study are available from the corre-
672 sponding author on reasonable request.

673 **Competing Interests**

674 The author declare that they have no competing interests.

675 **Funding**

676 This work is supported by the National Numerical Wind Tunnel Project [grant
677 No. NNW2019ZT6-A12], Science Fund for Distinguished Young Scholars of Shaanxi
678 Province of China [grant No. 2020JC-31], and Natural Science Funding of Shaanxi
679 Province [grant No. 2020JM-127].

680 **Author Details**

681 *a. Institute of Aerodynamic and Multidisciplinary Design Optimization, National Key Laboratory of*
682 *Science and Technology on Aerodynamic Design and Research, School of Aeronautics, Northwestern*
683 *Polytechnical University, Xi'an 710072, China*

684 *b. Aircraft Swarm Intelligent Sensing and Cooperative Control Key Laboratory of Sichuan Province,*
685 *School of Aeronautics and Astronautics, University of Electronic Science and Technology of China,*
686 *Chengdu 611731, China*

687 **Authors' Contributions**

688 The research output comes from a joint effort. All Authors read and approved the
689 final manuscript.

690 **Acknowledgements**

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

693 **References**

- 694 [1] Brayton G H (1911) Stability in aviation. Macmillan and Company limited, Lon-
695 don.
- 696 [2] Rockwell International Space Division (1977) Aerodynamic design data book,
697 Vol. I, Orbiter Vehicle. Report No. SD 72-SH-0060-1L.
- 698 [3] Pamadi B N, Brauckmann G J, Ruth M J (2001) Aerodynamic characteristics, da-
699 tabase development, and flight simulation of the X-34 vehicle. *J Spacecraft*
700 *Rockets* 38(3): 334-344. doi: 10.2514/6.2000-900.
- 701 [4] Keshmiri S, Colgren R, Mirmirani M (2005) Development of an aerodynamic
702 database for a generic hypersonic air vehicle. In: AIAA Guidance, Navigation,
703 and Control Conference and Exhibit, San Francisco, California, 15-18 August
704 2005. doi: 10.2514/6.2005-6257. doi: 10.2514/6.2005-6257.
- 705 [5] Fu J M (2005) Three-dimensional aerodynamic mathematical model for tactical
706 missiles with jet steering. *Aerospace Shanghai* 22(4): 13-18. (Chinese)
- 707 [6] He K F, Wang W Z, Qian W Q (2004) Mathematic modeling for the missile aero-
708 dynamics with tail-wing according to wind-tunnel test results. *Experiments and*
709 *Measurements in Fluid Mechanics* 18(4): 62-66. (Chinese)
- 710 [7] Ghoreyshi M, Cummings R M, Da Ronch A (2013) Transonic aerodynamic load
711 modeling of X-31 aircraft pitching motions. *AIAA J* 51(10): 2447-2464. doi:
712 10.2514/1.J052309.
- 713 [8] Krige D G (1951) A statistical approach to some basic mine valuations problems
714 on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Engi-*

- 715 neering Society of South Africa 52(6): 119-139.
716 https://hdl.handle.net/10520/AJA0038223X_4792.
- 717 [9] Han Z H, Goertz S (2012) Hierarchical kriging model for variable-fidelity surro-
718 gate modeling. *AIAA J* 50(9): 1885-1896. doi: 10.2514/1.J051354.
- 719 [10] Han Z H, Görtz S, Zimmermann R (2013) Improving variable-fidelity surrogate
720 modeling via gradient-enhanced kriging and a generalized hybrid bridge function.
721 *Aerosp Sci Technol* 25(1): 177-189. doi: 10.1016/j.ast.2012.01.006.
- 722 [11] Liu J, Song W P, Han Z H (2017) Efficient aerodynamic shape optimization of
723 transonic wings using a parallel infilling strategy and surrogate models. *Struct*
724 *Multidiscip O* 55(3): 925-943. doi: 10.1007/s00158-016-1546-7.
- 725 [12] Shi Z W, Wang Z H, Li J C (2012) The research of RBFNN in modeling of non-
726 linear unsteady aerodynamics. *Acta Aerodynamica Sinica* 30(1): 108-112+119.
727 (Chinese)
- 728 [13] Wu C, Yao H, Peng X Z (2013) Application of support vector regression for aero-
729 dynamic modeling. *Computer Simulation* 30(10): 128-132. (Chinese)
- 730 [14] Santos M, Mattos B, Girardi R (2008) Aerodynamic coefficient prediction of air-
731 foils using neural networks. In: 46th AIAA Aerospace Sciences Meeting and Ex-
732 hibit. Reno, Nevada, 7-10 January 2008. doi: 10.2514/6.2008-887.
- 733 [15] Wang C, Wang G D, Bai P (2019) Machine Learning method for aerodynamic
734 modeling based on flight simulation data. *Acta Aerodynamica Sinica* 37(3): 488-
735 497. (Chinese) doi: 10.7638/kqdlxxb-2019.0024.
- 736 [16] Zhu L, Gao Z H (2007) Aerodynamic optimization design of airfoil based on neu-
737 ral networks. *Aeronautical Computing Technique* 37(3): 33-36.
- 738 [17] Fu J Q, Shi Z W, Chen K (2018) Applications of real-time recurrent neural net-
739 work based on extended Kalman filter in unsteady aerodynamics modeling. *Acta*
740 *Aerodynamica Sinica* 36(4): 658-663. (Chinese) doi: 10.7638/kqdlxxb-2016.0131.
- 741 [18] Shi Z W, Ming X (2005) The application of FNN in unsteady aerodynamics mod-
742 eling based on fuzzy clustering. *Acta Aerodynamica Sinica* 23(1): 21-24. (Chi-
743 nese)
- 744 [19] Zhang Y, Sung W J, Mavris D N (2018) Application of convolutional neural net-
745 work to predict airfoil lift coefficient. In: 2018 AIAA/ASCE/AHS/ASC Struc-
746 tures, Structural Dynamics, and Materials Conference. Kissimmee, Florida, 8-12

747 January 2018. doi: 10.2514/6.2018-1903.

748 [20] Du X, He P, Martins J R R A (2021) Rapid airfoil design optimization via neural
749 networks-based parameterization and surrogate modeling. *Aerosp Sci Technol*
750 113: 106701. doi: 10.1016/j.ast.2021.106701.

751 [21] Punjani A, Abbeel P (2015) Deep learning helicopter dynamics models. In: IEEE
752 International Conference on Robotics and Automation (ICRA). Seattle, Washing-
753 ton, May 26-30, 2015. doi: 10.1109/ICRA.2015.7139643.

754 [22] Wang Q, Qian W Q, Ding D (2016) A review of unsteady aerodynamic modeling
755 of aircrafts at high angles of attack. *Acta Aeronautica et Astronautica Sinica* 37(8):
756 2331-2347. (Chinese) doi: 10.7527/S1000-6893. 2016. 0072.

757 [23] Bouhlel M A, He S, Martins J R R A (2020) Scalable gradient-enhanced artificial
758 neural networks for airfoil shape design in the subsonic and transonic regimes.
759 *Struct Multidiscip O* 61(4): 1363-1376. doi: 10.1007/s00158-020-02488-5.

760 [24] Secco N R, de Mattos B S (2017) Artificial neural networks to predict aerody-
761 namic coefficients of transport airplanes. *Aircr Eng Aerosp Tec* 89(2): 211-230.
762 doi: 10.1108/AEAT-05-2014-0069.

763 [25] Barnhart S A, Narayanan B, Gunasekaran S (2021) Blown wing aerodynamic co-
764 efficient predictions using traditional machine learning and data science ap-
765 proaches. In: AIAA Scitech 2021 Forum. Virtual Event, 11-15 & 19-21 January
766 2021 doi: 10.2514/6.2021-0616.

767 [26] Karali H, Inalhan G, Umut Demirezen M (2021) A new nonlinear lifting line
768 method for aerodynamic analysis and deep learning modeling of small unmanned
769 aerial vehicles. *Int J Micro Air Veh* 13: 17568293211016817. doi:
770 10.1177/17568293211016817.

771 [27] Li J, Zhang M (2021) Data-based approach for wing shape design optimization.
772 *Aerosp Sci Technol* 112: 106639. doi: 10.1016/j.ast.2021.106639.

773 [28] LeCun Y, Bottou L, Bengio Y (1998) Gradient-based learning applied to docu-
774 ment recognition. *P IEEE* 86(11): 2278-2324. doi: 10.1109/5.726791.

775 [29] LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553): 436-444.
776 doi: 10.1038/nature14539.

777 [30] Thuerey N, Weißenow K, Prantl L (2020) Deep learning methods for Reynolds-
778 averaged Navier–Stokes simulations of airfoil flows. *AIAA J* 58(1): 25-36. doi:

- 779 10.2514/1.J058291.
- 780 [31] Yu B, Xie L, Wang F (2020) An improved deep convolutional neural network to
781 predict airfoil lift coefficient. In: Jing, Z. (eds) Proceedings of the International
782 Conference on Aerospace System Science and Engineering 2019. ICASSE 2019.
783 Lecture Notes in Electrical Engineering, vol 622. Springer, Singapore. doi:
784 10.1007/978-981-15-1773-0_21.
- 785 [32] Chen H, He L, Qian W (2020) Multiple aerodynamic coefficient prediction of
786 airfoils using a convolutional neural network. *Symmetry* 12(4): 544. doi:
787 10.3390/sym12040544.
- 788 [33] He L, Qian W Q, Dong K S (2022) Aerodynamic characteristics modeling of iced
789 airfoil based on convolution neural networks. *Acta Aeronautica et Astronautica*
790 *Sinica* 43(10): 126434. (Chinese) doi: 10.7527/S1000-6893.2021.26434.
- 791 [34] Rumelhart D E, Hinton G E, Williams R J (1986) Learning representations by
792 back-propagating errors. *Nature* 323(6088): 533-536. doi: 10.1038/323533a0.
- 793 [35] Shan S, Wang G G (2010) Survey of modeling and optimization strategies to
794 solve high-dimensional design problems with computationally-expensive black-
795 box functions. *Struct Multidiscip O* 41(2): 219-241. doi: 10.1007/s00158-009-
796 0420-2.
- 797 [36] Han Z H, Xu C Z, Zhang L (2020) Efficient aerodynamic shape optimization us-
798 ing variable-fidelity surrogate models and multilevel computational grids. *Chi-
799 nese J Aeronaut* 33(1): 31-47. doi: 10.1016/j.cja.2019.05.001.
- 800 [37] Han Z H, Zhang Y, Song C X (2017) Weighted gradient-enhanced kriging for
801 high-dimensional surrogate modeling and design optimization. *AIAA J* 55(12):
802 4330-4346. doi: 10.2514/1.J055842.
- 803 [38] Liu F, Han Z H, Zhang Y (2019) Surrogate-based aerodynamic shape optimiza-
804 tion of hypersonic flows considering transonic performance. *Aerosp Sci Technol*
805 93, 105345. doi: 10.1016/j.ast.2019.105345.
- 806 [39] Krizhevsky A, Sutskever I, Hinton G E (2012) Imagenet classification with deep
807 convolutional neural networks. *Advances in Neural Information Processing Sys-
808 tems*, 25.
- 809 [40] Albawi S, Mohammed T A, Al-Zawi S (2017) Understanding of a convolutional
810 neural network. In: 2017 International conference on engineering and technology

811 (ICET). Antalya, Turkey, 21-23 August 2017. doi:
812 10.1109/ICEngTechnol.2017.8308186.

813 [41]Narasinga Rao M R, Venkatesh Prasad V, Sai Teja P (2018) A survey on preven-
814 tion of overfitting in convolution neural networks using machine learning tech-
815 niques. *Int J Eng Technol In* 7(2.32): 177-180. doi: 10.14419/ijet.v7i2.32.15399.

816 [42]Xiao M, Wu Y, Zuo G (2021) Addressing overfitting problem in deep learning-
817 based solutions for next generation data-driven networks. *Wirel Commun Mob*
818 *Com.* <https://doi.org/10.1155/2021/8493795>.

819 [43]Kingma D P, Ba J (2014) Adam: A method for stochastic optimization. arXiv pre-
820 print arXiv: 1412. 6980. doi: 10.48550/arXiv.1412.6980.

821 [44]Sekar V, Zhang M, Shu C (2019) Inverse design of airfoil using a deep convolu-
822 tional neural network. *AIAA J* 57(3): 993-1003. doi: 10.2514/1.J057894.

823 [45]Bengio Y (2012) Practical recommendations for gradient-based training of deep
824 architectures. In: *Neural Networks: Tricks of the Trade* (pp. 437-478). Springer,
825 Berlin, Heidelberg. doi: 10.1007/978-3-642-35289-8_26.

826