

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/159930/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Geng, Dongling, Yan, Jun, Xu, Qi, Zhang, Qi, Zhou, Mengfang, Fan, Zhirui and Li, Haijiang 2023. Real-time structure topology optimization using CNN driven moving morphable component method. Engineering Structures 290 , 116376. 10.1016/j.engstruct.2023.116376

Publishers page: <https://doi.org/10.1016/j.engstruct.2023.116376>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



# **Real-Time Structure Topology Optimization using CNN driven Moving Morphable Component Method**

Dongling Geng<sup>a</sup> Jun Yan<sup>a,\*</sup> Qi Xu<sup>a</sup> Qi Zhang<sup>a</sup> Mengfang Zhou<sup>a</sup>

Zhirui Fan<sup>a</sup> Haijiang Li<sup>b,\*</sup>

<sup>a</sup> Department of Engineering Mechanics, Dalian University of Technology, China

<sup>b</sup> School of Engineering, Cardiff University, Cardiff, the Unite Kingdom

\*Corresponding Author:

Prof. Jun Yan

Department of Engineering Mechanics, Dalian University of Technology, Dalian,

Liaoning Province, 116024, China

Email: [yanjun@dlut.edu.cn](mailto:yanjun@dlut.edu.cn)

Postal Address:

No.2 Linggong Road, Ganjingzi District, Dalian, Liaoning Province, P.R.C., 116024

\*Corresponding Author:

Prof. Haijiang Li

School of Engineering, Cardiff University, Cardiff, CF10 3AT, Unite Kingdom

Email: [lih@cardiff.ac.uk](mailto:lih@cardiff.ac.uk)

Postal Address:

S2.10C, Queen's Buildings, 5 The Parade, Newport Road, Cardiff, Cf24 3AA

## Abstract

Classical optimization methods require finite element analysis in iterations, which increase the computing time and decrease the algorithmic efficiency. The deep learning model can potentially realize real-time topology optimization design, but it normally requires large training set. This paper presents a real-time topology optimization algorithm based on the Moving Morphable Component (MMC) method using a Convolutional Neural Network (CNN). The optimization algorithm uses a new data pre-processing method, which can preserve the numerical characteristics and smoothness of the structure boundary, hence it can help CNN to capture data features with a limited sample set. The topology optimization boundary information of the optimized result is used as the sample set label to avoid the components dislocation phenomenon. The new algorithm effectiveness has been verified with several examples. The trained model can significantly improve the optimization efficiency of the MMC method and offer accurate results with a clear structure boundary.

**Keywords:** MMC; Topology Optimization; Real-Time Optimization; Convolutional Neural Network; Deep Learning

# 1. Introduction

Structural topology optimization is used to obtain the optimal distribution of materials to realize an optimized structural performance under certain conditions [1]. A variety of topology optimization methods have been developed, such as density-based optimization method [2–6] and boundary evolutionary optimization method [7–12]. Some methods have been successfully applied to fluid, phonon crystals, and multidisciplinary problems [13]. However, optimization methods usually need finite element analysis and repeat iterations. This increases the time cost of topology optimization computation. Moreover, when the number of elements and problem dimensions increase, the computation efficiency decreases rapidly.

Recently, deep learning has developed rapidly. Extensive research has been conducted on combining deep learning and topology optimization. The deep learning model uses neurons containing simple operations as the basic unit and establishes a functional relationship between the input and output. Therefore, a well-trained model can find the solution quickly.

Zhang [14] used structural displacement, strain, and volume fraction information as the input of a CNN. The model was trained with 80,000 samples, and achieved real-time topology optimization prediction of multiple load conditions. Yu [15] used a CNN and Generative Adversarial Network (GAN) trained with 100,000 samples to convert the low-resolution results to high-resolution results. Nakamura [16] added the spade layer [17] based on the work of Yu [15], which improved the performance of the model. Sosnovik [18] built a lightweight model to reduce the sample sets requirement and successfully accelerated the topology optimization calculation with 10,000 samples.

Artificial density topology optimization method describes the structure by the

1 element density, which is similar to gray image. The above studies are mainly based on  
2 the combination of Solid Isotropic Material with Penalization (SIMP) and deep learning.  
3  
4 However, owing to the probabilistic nature of the deep learning model, the inevitable  
5 prediction error may lead to the model prediction results without clear boundaries [16].  
6  
7 The MMC topology optimization method proposed by Guo [19] can generate clear  
8 structural boundaries. Xin [20] combined the KNN model with the MMC method to  
9 accelerate MMC topology optimization. Compared with Xin [20], Zheng [21]  
10 accelerated the MMC optimization process by using Attention U-Net [22]. However,  
11 due to the preprocessing method proposed by Zheng [21], the sample set label was  
12 composed of 0 or 1.  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23

24 As mentioned above, compared with the SIMP method, there are fewer studies on  
25 combining MMC with deep learning model. An important reason is the special data  
26 characteristics of the boundary evolution optimization method. Meanwhile, deep  
27 learning models usually have the requirements of the sample set size. Most of the  
28 models mentioned above were trained based on a large sample set (80,000 samples  
29 [14], 100,000 samples [15]), which hinders the deep learning application to topology  
30 optimization.  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

41 The combination of MMC and deep learning is studied in this paper. Based on the  
42 mathematical characteristics of the MMC method, the structure boundary information  
43 is chosen as sample set label. It can avoid the “component dislocation” phenomenon in  
44 the model prediction results. Meanwhile, a new data preprocessing method is proposed  
45 to ensure the sample label’s accuracy in mathematic. High-quality sample labels are  
46 helpful to improve the model prediction accuracy based on limited sample sets. Finally,  
47 in the MMC method, the initial optimization stage will be greatly influenced by the  
48 initial components’ distribution. A sampling method is proposed to avoid collecting the  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

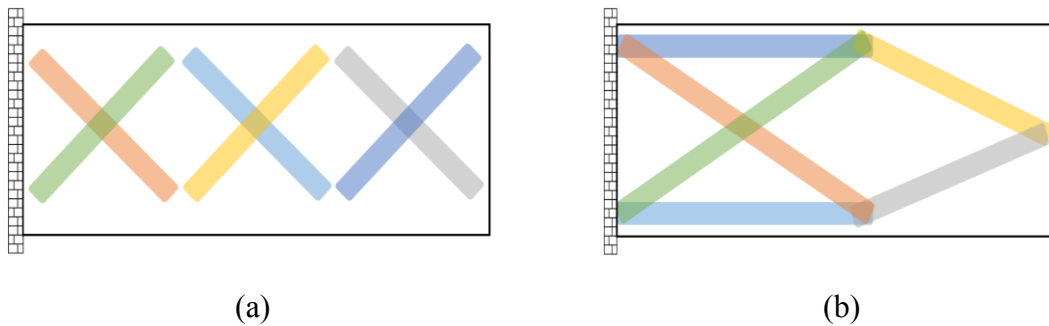
optimization information with severe fluctuations. Section 2 introduces the basic theory of the MMC method and CNN. In Section 3, a real-time topology optimization algorithm based on the MMC method is introduced. In Section 4, the algorithm's accuracy is verified by several numerical examples. Section 5 is the summary of this study.

## 2. The theoretical basis of a CNN-driven MMC real-time topology optimization

### 2.1. MMC topology optimization method

In the MMC topology optimization method, the structure is composed of components (bars). MMC can realize structure optimization by translation, rotation, and covering operations [19], as shown in Fig. 1.

The MMC method uses the topology description function (TDF)  $\phi_i(x, y)$  to describe the structural boundary:



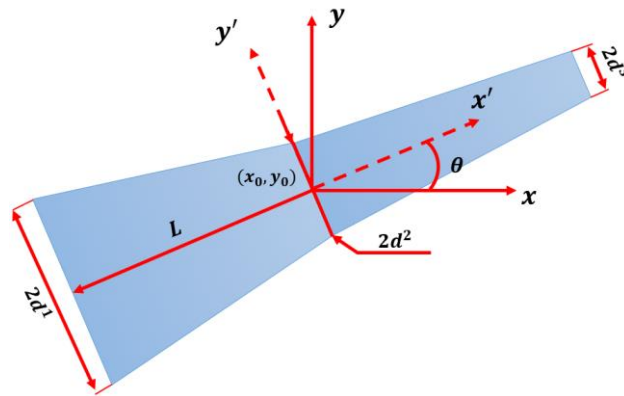
**Fig. 1** The basic concept of the MMC method: **(a)** distribution of components in the initial structure; **(b)** distribution of components in the optimized structure.

$$\phi_i(x, y) = \left( \frac{x'}{L_i} \right)^p + \left( \frac{y'}{f(x')} \right)^p - 1 \quad (1)$$

$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \begin{Bmatrix} x - x_{0i} \\ y - y_{0i} \end{Bmatrix} \quad (2)$$

In Eq. (1),  $p$  is the hyperelliptic parameter ( $p = 6$  in this study).  $x'$  and  $y'$  are the coordinates under the local system.  $x$  and  $y$  are the coordinates under the global system.  $L_i$  denotes the half-length of the  $i^{\text{th}}$  component. Eq. (2) shows the coordinate transformation relationship between the local and global systems.  $x_{0i}$  and  $y_{0i}$  represent the coordinates in the center of  $i^{\text{th}}$  component in the global system, and  $\theta_i$  represents the tilt angle of the component. Function  $f(x')$  describes the change of the component section. Guo [11] presented three types of expressions of  $f(x')$  including the uniform thickness component, linearly varying thickness component, and quadratically varying thickness component. The third type of  $f(x')$  is used in this study.

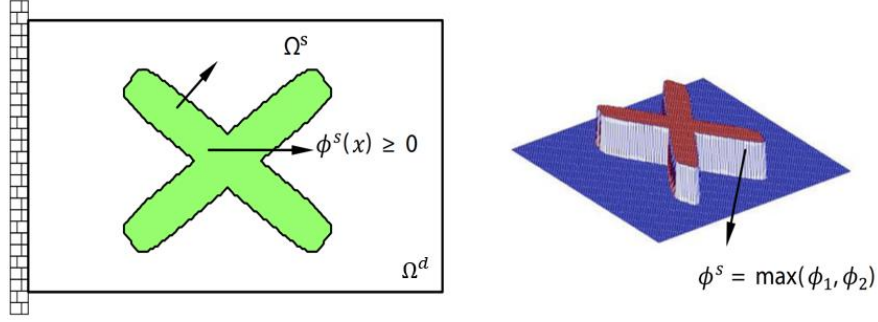
In Fig. 2,  $d^i$  is the half-width of the component at three different points. The meanings of the other parameters are the same as those in Eq. (1) and Eq. (2). In the MMC method, the geometric parameters of the components are stored in the component geometry description vector  $\mathbf{D}$ , and  $\mathbf{D} = (x_0, y_0, L, \sin\theta, d^i)^T$ .



**Fig. 2** Quadratically varying thicknesses component.

In the MMC method, the topology description boundary information (TDBI) matrix  $\phi^s(\mathbf{x})$ , which contains all the boundary information of the structure, is obtained according to  $\phi^s(\mathbf{x}) = \max(\phi_1, \dots, \phi_n)$ , and  $n$  is the number of components in the structure. According to Eq. (3), the TDBI matrix can describe the structure boundary clearly. The geometry meanings of Eq. (3) are shown in Fig. 3.

$$\begin{cases} \phi^s(\mathbf{x}) > 0, & \text{if } \mathbf{x} \in \Omega^s \\ \phi^s(\mathbf{x}) = 0, & \text{if } \mathbf{x} \in \partial\Omega^s \\ \phi^s(\mathbf{x}) < 0, & \text{if } \mathbf{x} \in \Omega^d/\Omega^s \end{cases} \quad (3)$$



**Fig. 3** The structural topology representation of each component [11].

The optimization formulation of the MMC method can be expressed as Eq. (4).

The objective function is compliance minimization, and the constraint is structure volume fraction.

$$\text{find } \mathbf{D} = ((\mathbf{D}^1)^T, \dots, (\mathbf{D}^i)^T, \dots, (\mathbf{D}^n)^T)^T, \mathbf{u}(\mathbf{x})$$

$$\text{min } C = \int_{\Omega^d} H(\phi^s(\mathbf{x}; \mathbf{D})) \mathbf{f} \cdot \mathbf{u} dV + \int_{\Gamma_t} \mathbf{t} \cdot \mathbf{u} dS$$

subject to

$$\int_{\Omega^s} (H(\phi^s(\mathbf{x}; \mathbf{D})))^q \mathbb{E} : \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) dV = \int_{\Omega^s} H(\phi^s(\mathbf{x}; \mathbf{D})) \mathbf{f} \cdot \mathbf{v} dV + \int_{\Gamma_t} \mathbf{t} \cdot \mathbf{v} dS, \forall \mathbf{v} \in U_{ad} \quad (4)$$

$$\int_{\Omega^s} H(\phi^s(\mathbf{x}; \mathbf{D})) dV \leq \bar{V}, \mathbf{D} \subset \mathbf{U}_D, \mathbf{u} = \bar{\mathbf{u}}, \text{ on } \Gamma_u$$

In Eq. (4),  $\Omega^s = \cup_i^n \Omega^i$ ,  $i = 1, \dots, n$  and  $\Omega^i$  represents the region occupied by  $i^{th}$  component.  $\mathbf{u}$  and  $\mathbf{v}$  are the displacement field and the corresponding test function defined on  $\Omega^s$  with  $U_{ad} = \{\mathbf{v} | \mathbf{v} \in H(\mathbf{D}), \mathbf{v} = 0 \text{ on } \Gamma_u\}$ ;  $H(\mathbf{x})$  is the Heaviside function.  $\mathbf{f}$  and  $\mathbf{t}$  are the body force density in  $\Omega^i$  and the surface traction on the Neumann boundary  $\Gamma_t$ .  $\bar{\mathbf{u}}$  denotes the displacement on the Dirichlet boundary  $\Gamma_u$ .  $\boldsymbol{\varepsilon}$  represents the second-order linear strain tensor, and  $\mathbb{E}^i$  represents the elastic tensor.  $\bar{V}$  represents the upper bound of the structure available volume, and  $\mathbf{U}_D$  is the constraint matrix for  $\mathbf{D}$ .



## 2.2. CNN model and evaluation method

Traditional neural networks, such as Back Propagation (BP) neural network [23], use dense layers to connect the entire model. Generally, such neural networks have unaffordable computational costs when dealing with large-scale data. Unlike the BP network, CNN has convolutional and pooling layers that enable the model to process large-scale data. The CNN is also called the "translational invariant neural network" [24]. It allows shift-invariant operations such as translations, rotations, and size scaling. In addition, the CNN has the characteristics of weight sharing and local connection, which can further reduce the number of model parameters.

The mean square error (MSE) loss function is chosen to deal with the regression problem, as shown in Eq. (5).  $y_i^p$  is the value predicted by the CNN, and  $y_i$  is MMC optimized result,  $n$  is the number of samples.

$$\text{MSE} = \frac{\sum_{i=1}^n (y_i - y_i^p)^2}{n} \quad (5)$$

The model performance can be obtained by observing the trend of the loss function. To alleviate the overfitting problem of the model, dropout layers [25] and batch normalization layers [26] are used in the model. Dropout operation is a standard treatment to deal with the model overfitting problem. There is a hyperparameter called dropout rate in the dropout layer. This layer will randomly cause some neurons to lose learning ability during the iteration process. The dropout rate determines the percentage of neurons without learning ability. Dropout operation can alleviate the overfitting problem by reducing model complexity. Batch normalization layers can process the data through normalization operations. It can improve the model training stability and overfitting phenomenon.

In this work, structural compliance (objective function), MSE value, and

multiscale structural similarity (MSSIM) [27] are chosen to evaluate the CNN prediction results. The MSE value reflects structural similarity based on the Euclidean distance. The MSSIM value comprehensively compares two specified pictures' brightness, contrast, and structural similarity. For the problem in this study, the brightness and contrast of the pictures are the same. Therefore, the MSSIM value is chosen as the evaluation function from the image perspective.

### **3. New data preprocessing method and the corresponding CNN training**

#### **3.1. Data preprocessing method for TDBI**

The MMC method's component geometry description matrix ( $\mathbf{D}$ ) contains the basic optimization parameters. And matrix  $\mathbf{D}$  is converted to TDBI by Eq. (1). Finally, TDBI describes the structure boundary with a smooth 0 contour line. Both TDBI and component geometry description matrix ( $\mathbf{D}$ ) contain geometric information. The intuitive idea is choosing matrix  $\mathbf{D}$  as the sample set label. However, choosing  $\mathbf{D}$  as the sample label will influence the model prediction accuracy. On the one hand, the deep learning model is probabilistic. There are inevitable prediction errors in results. Eq. (1) may greatly magnify the errors. On the other hand, each parameter in  $\mathbf{D}$  has the information enrichment characteristic, and minor prediction errors may lead to significant structural changes. Different from  $\mathbf{D}$ , choosing TDBI as the sample set label can avoid the processing of Eq. (1). The amplification of model prediction errors can be avoided. Although the scale of total parameters will increase significantly, CNN has a strong learning ability for such data. Meanwhile, as the total number of parameters increases, the information contained by each parameter in TDBI decreases. It provides

more tolerance for the model learning process. For the above reasons, the TDBI information in MMC is chosen as the sample set label in this work.

The initialization method of neurons in the deep learning model, such as the gaussian initialization and Xavier initialization [28], significantly influence the model training process. Generally, the sample also needs to be scaled down to a small numeric range through a normalization operation to accelerate the convergence of the CNN training process. TDBI describes the structure boundary with a continuous and smooth 0 contour line. The MMC optimization result boundary is clear in graphically and mathematically. In the process of data preprocessing, this characteristic should be preserved. However, the numerical range of TDBI is large. For example, the value range of TDBI is  $[-2786.3, 1.0]$  in Fig. 5(b), which cannot meet the requirement of CNN training. Meanwhile, for different load conditions, the range of TDBI is different. Therefore, finding a proper data preprocessing method for TDBI characteristics is difficult.

Eq. (6) is a standard data normalization method called the Max-Min normalization method.  $\mathbf{X}$  represents all the samples for different load conditions, and  $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^i, \dots, \mathbf{X}^n\}$ ,  $n$  is the sample set size. In Eq. (6) and Eq. (7), the  $\mathbf{X}_{\min} = \{\min(\mathbf{X}^1), \min(\mathbf{X}^2), \dots, \min(\mathbf{X}^n)\}$ ,  $\mathbf{X}_{\max} = \{\max(\mathbf{X}^1), \max(\mathbf{X}^2), \dots, \max(\mathbf{X}^n)\}$ .  $\bar{\mathbf{X}}_{\text{True}}$  is the normalized result of the TDBI. The data can be converted within  $[0, 1]$  by Eq. (6). Eq. (7) is the inverse operation of Eq. (6).  $\bar{\mathbf{X}}$  is the TDBI matrix predicted by the CNN. Because the CNN is probabilistic,  $\bar{\mathbf{X}} = \bar{\mathbf{X}}_{\text{True}} + \Delta\mathbf{X}$ .  $\Delta\mathbf{X}$  is the CNN prediction error, and  $\bar{\mathbf{X}}_{\text{True}}$  is the accurate result computed by Eq. (6).

$$\bar{\mathbf{X}}_{\text{True}} = \frac{\mathbf{X} - \mathbf{X}_{\min}}{\mathbf{X}_{\max} - \mathbf{X}_{\min}} \quad (6)$$

$$\mathbf{X} = \bar{\mathbf{X}}(\mathbf{X}_{\max} - \mathbf{X}_{\min}) + \mathbf{X}_{\min} \quad (7)$$

The Max-Min normalization method may seriously influence the training of CNN for real-time MMC topology optimization. This preprocessing method will blur the data information at the structural boundary. CNN cannot be trained based on this sample set. On the other hand, the TDBI preprocessed by Eq. (6) needs an inverse operation like Eq. (7). There are inevitable errors in the prediction results obtained by the deep learning model,  $\bar{\mathbf{X}} = \bar{\mathbf{X}}_{\text{True}} + \Delta\mathbf{X}$ . In Eq. (7), the  $\mathbf{X}_{\max} - \mathbf{X}_{\min}$  may greatly magnify the CNN prediction errors. For the above reasons, the Max-Min normalization method is unsuitable for TDBI preprocessing.

Considering the above problems, a boundary feature preserving normalization method is proposed. The data preprocessing method of TDBI is realized by Eq. (8) and Eq. (9). This method can effectively control the value range of TDBI.

$$\phi'_i(\mathbf{x}) = \exp(\phi_i(\mathbf{x})) - \lambda \quad (8)$$

$$\begin{cases} \phi'_i(\mathbf{x}) = -1 & \text{if } \phi'_i(\mathbf{x}) < -1 \\ \phi'_i(\mathbf{x}) = 1 & \text{if } \phi'_i(\mathbf{x}) > 1 \end{cases} \quad (9)$$

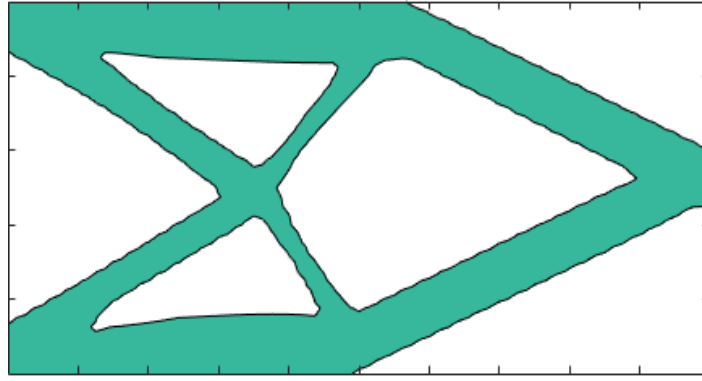
In Eq. (8),  $\lambda$  is the numerical boundary control parameter. If  $\lambda = 1$ , the TDBI still describes the structural boundary with 0 contour line. The definition of the numerical structure boundary does not change during data preprocessing ( $\lambda = 1$  in this study). This method preserves the accurate mathematical and geometric boundaries of TDBI.

The results of different TDBI preprocessing methods are shown in Fig. 4. Fig. 4(a) and Fig. 4(b) show the optimized structure and its level map, respectively. When Eq. (6) is used for TDBI data preprocessing, TDBI cannot use the 0 contour line to describe the structure boundary. The numerical boundary features of the TDBI matrix at the structure boundary are blurred (Fig. 4(c)). In contrast, Eq. (8) and Eq. (9) can adjust the value range while retaining the characteristics of TDBI (Fig. 4(d)). By comparing the

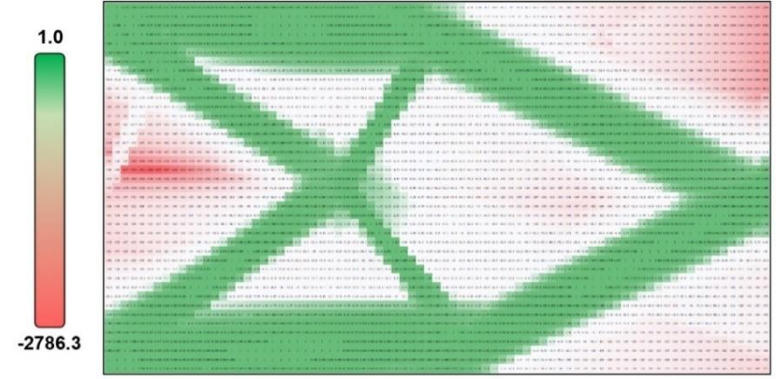
1 local features in Fig. 4(c) and Fig. 4(d), when using the preprocessing method proposed  
2 in this study (Eq. (8) and Eq. (9)), the TDBI numerical distribution is more uniform and  
3 reasonable.  
4  
5

6  
7 If  $\lambda = 1$ , the structural boundary is numerically unchanged. The proposed method  
8 is helpful to train the CNN based on accuracy structural boundaries. On the contrary,  
9 when using the Max-Min normalization method, the local feature of TDBI disappears  
10 (such Fig. 4(c)). CNN cannot effectively capture the features of TDBI based on this  
11 method. Finally, there is no inverse operation for this method. The CNN prediction error  
12 amplification problem is avoided. Meanwhile, the structure boundary numerical  
13 characteristics are retained when the value range of TDBI changes. Eq. (9) can remove  
14 the data features inside and outside the structure. It reduces the feature complexity of  
15 the input data and highlights the structure boundary features. This operation can help  
16 the model capture data characteristics with limited sample set. The preprocessing  
17 method has the effect of data enhancement.  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32

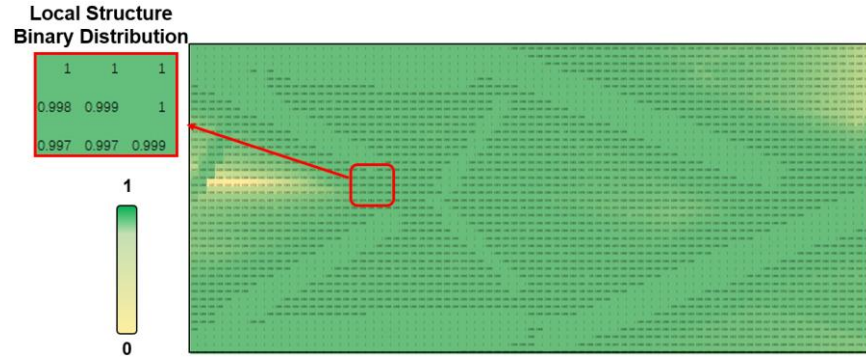
33  
34 The proposed preprocessing method is suitable for the data with Eq. (3)  
35 characteristics. Moreover, boundary evolution optimization methods generally have  
36 such features. The data preprocessing method proposed in this section is helpful for  
37 real-time boundary evolution topology optimization.  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65



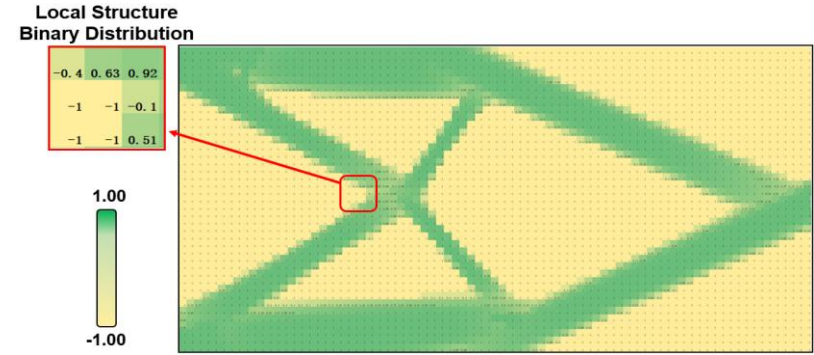
(a)



(b)



(c)

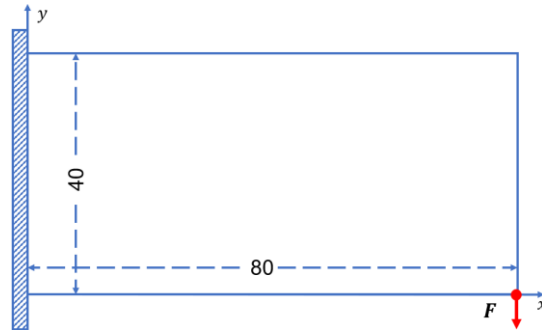


(d)

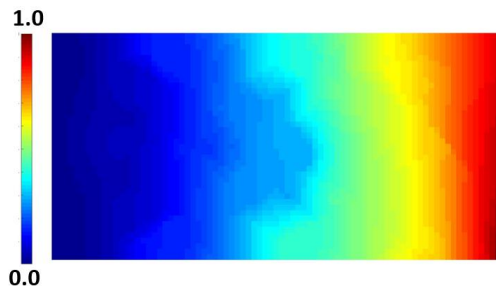
**Fig. 4** Numerical characteristics of TDBI matrices obtained with different data preprocessing methods: **(a)** Optimized result by the MMC and described the boundary with 0 contour line. **(b)** Optimized structure level map (unprocessed). **(c)** Optimized structure level map after data preprocessing by Eq. (6). **(d)** Optimized structure level map after preprocessing by Eq. (8) and Eq. (9) ( $\lambda = 1$ ).

### 3.2. The input data of the deep learning model

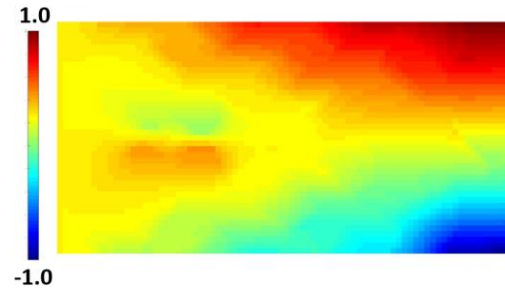
As mentioned above, TDBI was chosen as the sample set label. As shown in Fig. 6, there are four input data types: the node displacement in the x or y direction ( $U_x$ ,  $U_y$ ), node strain energy density matrix  $\mathbf{E}$  and the preliminary optimization structure matrix  $\mathbf{T}$ . The data type selection of sample set is mainly inspired by the research on real-time SIMP optimization methods. The input data of the model usually choose the element node displacement and initial optimization structure [15,29]. And there is no difference between SIMP method and MMC method. Yan [30] pointed out that using the data with various physical information can improve the model training effect. The node strain energy density matrix is also chosen as the input data for the model in this work.



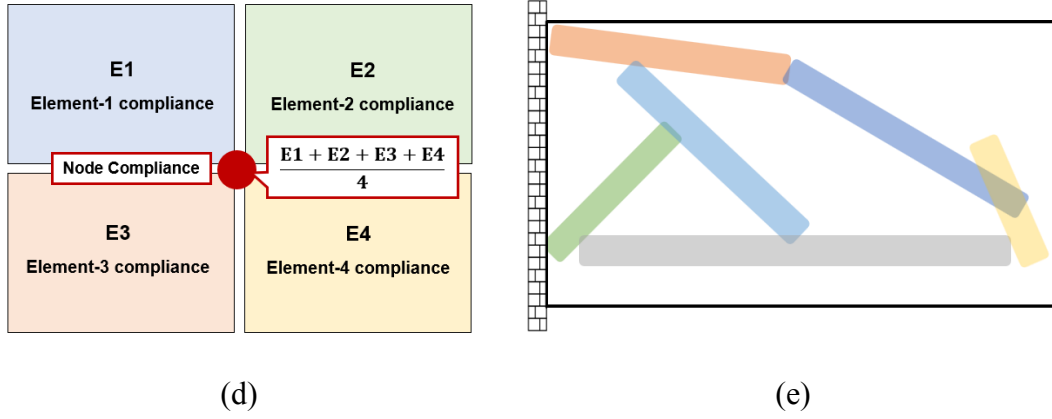
(a)



(b)

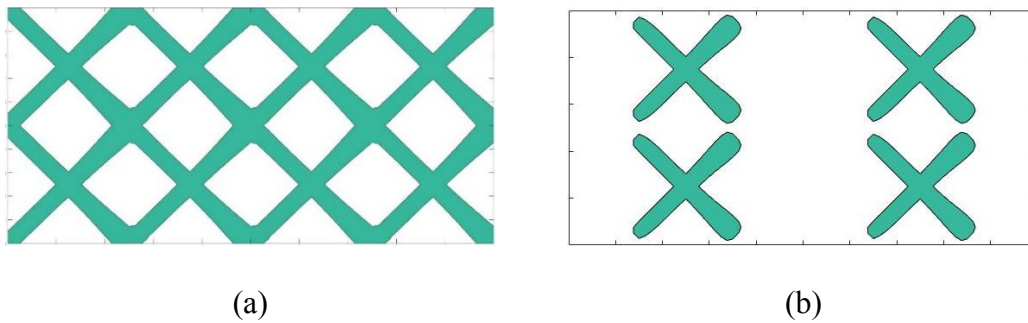


(c)



**Fig. 6** Four model input data types: **(a)** The specific load and constraint condition. **(b, c)** The node displacement in y and x direction. **(d)** The strain energy density at element nodes and its computing method. **(e)** The initial optimized structure.

The input data types for the model are discussed above, and the iterative steps to extract data will be described. The component is the basic unit of the MMC optimization method. Although the initial component layout has no significant influence on the MMC method's global optimization ability, the initial component layout will affect the initial optimization iteration process. In order to discuss this phenomenon, the cantilever beam is taken as an example. The structure is divided by  $40 \times 80$  finite element mesh. 16 and 8 components are used as the initial number of components, respectively. The component layout is shown in Fig. 7.

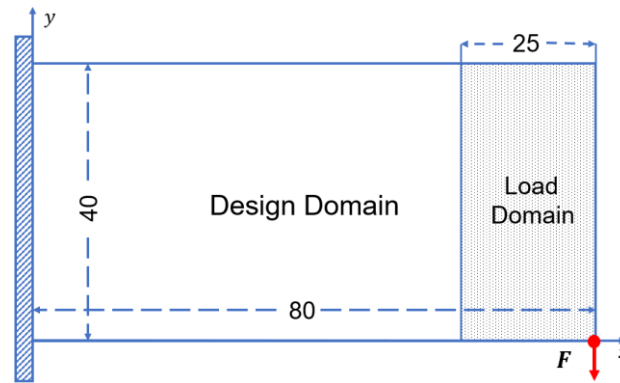


**Fig. 7** The components distribution with different initial component numbers: **(a)** and **(b)** show the initial distribution of 16 and 8 components layout.

The training set of the model is collected based on the load condition shown in Fig.

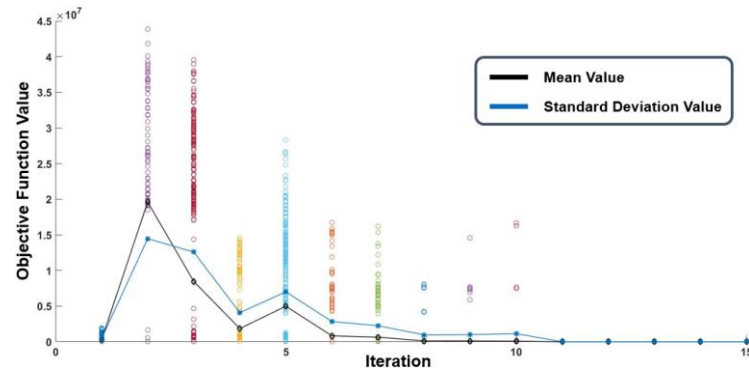


8. The boundary on the left is fixed, and the concentrated force along the  $-y$  direction is randomly placed in the load domain. The load domain is a rectangular area with 25 elements widths and 40 elements heights on the right side. Young's modulus is 1, and Poisson ratio is 0.3. Fig. 9 shows the changes of the objective function values in the first 15 optimization iteration steps with 16 initial components. For the 8 initial component distributions, Fig. 10 shows the objective function values changes in the first 25 optimization iteration steps. The specific statistics are shown in Tab. 1 and Tab. 2, respectively.

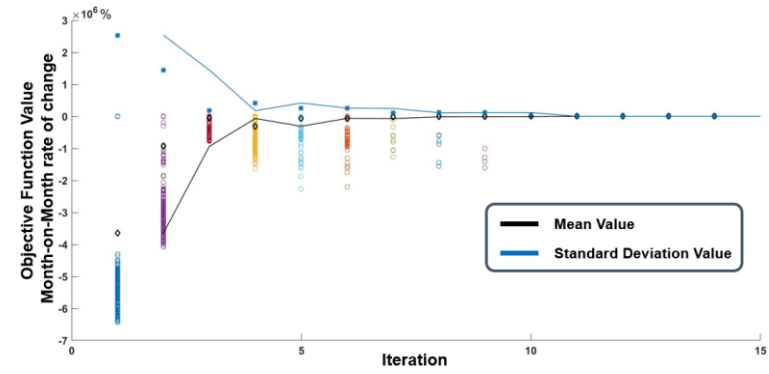


**Fig. 8** The load condition of cantilever beam example.

Because MMC optimization method regards components as the basic unit for structural optimization, the distribution of initial components will influence the MMC initial iteration process. For example, large objective function values will be generated when there is no component distribution at the loading point. Fig. 9 and Tab. 1 show the iterative process for the first 15 steps with 16 initial component distribution conditions. At the 15<sup>th</sup> iteration, the average change rate of objective function is less than 6% and close to 5%. It is called the stable calculation process. Moreover, for the 8 initial components distribution condition, the stable calculation process is entered in the 24<sup>th</sup> iteration step. Optimization parameters influenced by the component's distribution

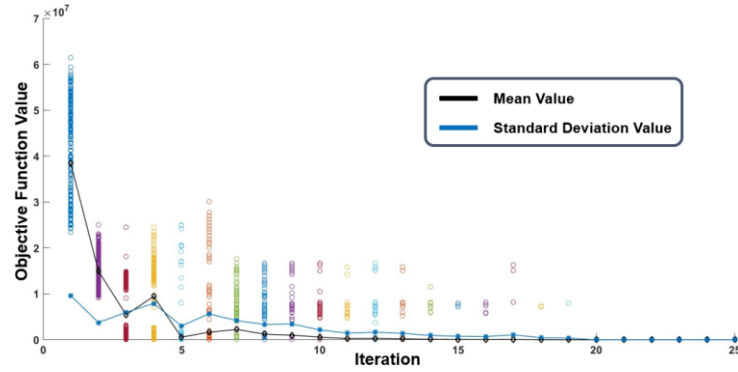


(a)

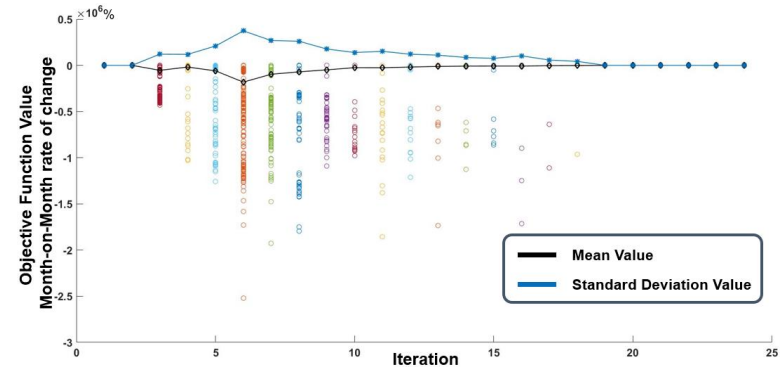


(b)

**Fig. 9** The first 15 steps of optimization process information (16 components distribution): **(a)** the objective function values variation. **(b)** the objective function value changes.



(a)



(b)

**Fig. 10** The first 25 steps of optimization process information (8 components distribution): **(a)** the objective function values variation. **(b)** the objective function value changes.

**Tab. 1** The first 15 steps optimization process numerical characteristics (16 initial components distribution condition).

Iteration	Mean values	Standard deviation values	Mean values of step-by-step ratio	Standard deviation values of step-by-step ratio
<b>1</b>	$2.89 \times 10^5$	$2.89 \times 10^5$	~	~
<b>2</b>	$1.96 \times 10^7$	$1.44 \times 10^7$	$-9.65 \times 10^5 \%$	$1.45 \times 10^6 \%$
<b>3</b>	$8.47 \times 10^6$	$1.26 \times 10^7$	$-3.08 \times 10^5 \%$	$1.77 \times 10^5 \%$
<b>:</b>	<b>:</b>	<b>:</b>	<b>:</b>	<b>:</b>
<b>13</b>	164.20	95.59	7.44%	5.72%
<b>14</b>	151.73	82.44	6.52%	5.10%
<b>15</b>	142.17	72.10	5.29%	4.82%

**Tab. 2** The first 25 steps optimization process numerical characteristics (8 initial components distribution condition).

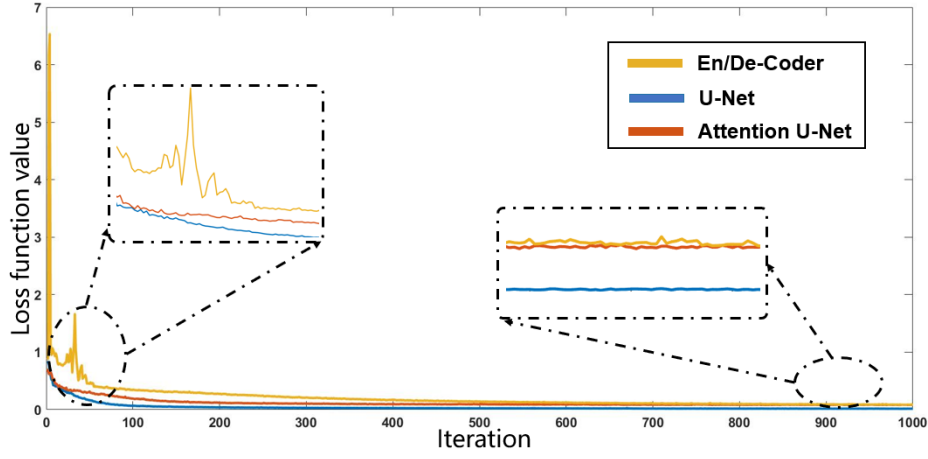
Iteration	Mean values	Standard deviation values	Mean values of step-by-step ratio	Standard deviation values of step-by-step ratio
<b>1</b>	$3.84 \times 10^7$	$9.63 \times 10^6$	~	~
<b>2</b>	$1.50 \times 10^7$	$3.74 \times 10^6$	53.79%	52.9%
<b>3</b>	$5.44 \times 10^6$	$5.97 \times 10^6$	$-5.39 \times 10^4 \%$	$1.21 \times 10^5 \%$
<b>:</b>	<b>:</b>	<b>:</b>	<b>:</b>	<b>:</b>
<b>23</b>	133.78	63.62	6.72%	4.09%
<b>24</b>	124.74	56.14	5.94%	3.65%
<b>25</b>	117.05	50.15	5.49%	3.60%

are minimal in this process. The optimization information in this process is problem-related and not significantly influenced by the initial component. The sample set is constructed during the stable calculation process. The sampling method can reduce the influence of the initial component distribution, and the deep learning model can be trained based on the data set related to the optimization problem.

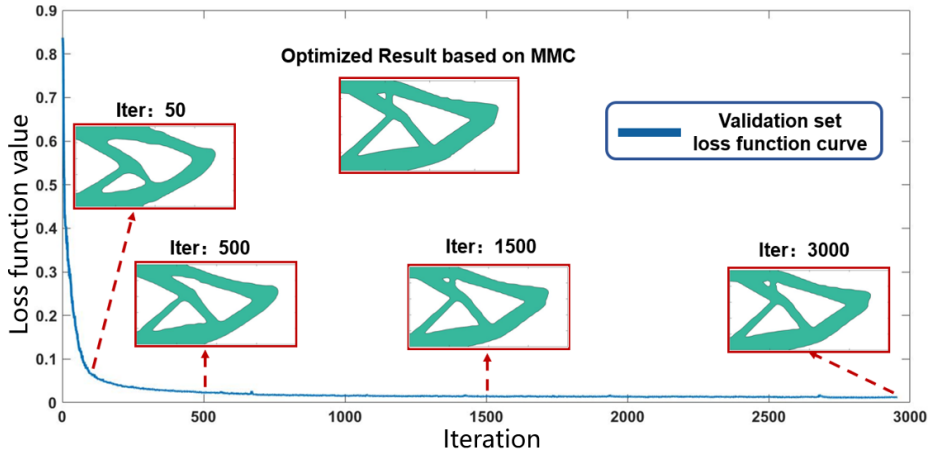
This manuscript uses NVIDIA Corporation TU102 [Titan RTX] for the deep learning model training. And the model is constructed by Python 3.6.12 and Keras 2.3.1. The sample set is obtained by MATLAB 2016b.

#### **4. Method validation and result analysis**

The accumulation of the sample set is shown in Fig. 8. The sample set is extended by data enhancement technology (data rotation). Finally, 2000 samples were obtained for model training. The ratio of the training set to validation set is 0.8:0.2. Fig. 11 shows the 1000 iterations process of classical CNN [24], U-Net [31], and attention U-Net [22]. The mean square error (MSE) loss function and Adam optimizer [32] are used in three models. Compared with the classical CNN, U-Net with a skip connection skill has better convergence stability. Due to the structure complexity of attention U-Net, the model requires longer iterations to achieve convergence. The primary purpose of this chapter is to verify the effectiveness of the data preprocessing method mentioned above,. The following discussion is based on the U-Net. In U-Net, the learning rate is 0.001. The batch size is 128, and the total iterations are 3000. The training curve of the U-Net model verification set is shown in Fig. 12.



**Fig. 11** The 1000 iteration history for different models.



**Fig. 12** The validation set loss function curve for U-Net model (3000 iterations).

#### 4.1. Model prediction results for 2D compliance optimization problem with different initial component numbers.

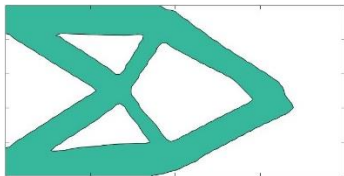
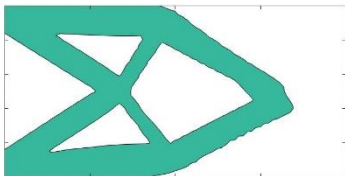
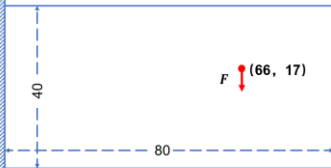
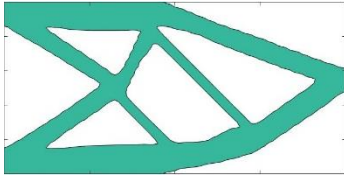
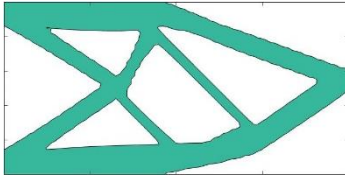
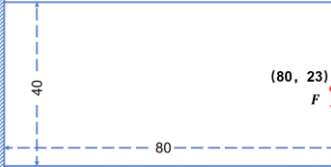
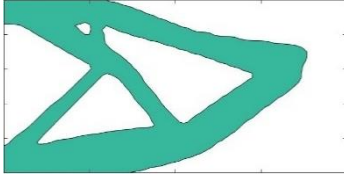
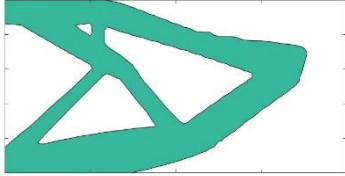
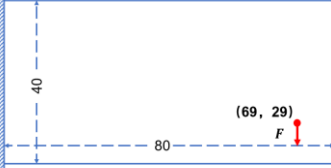
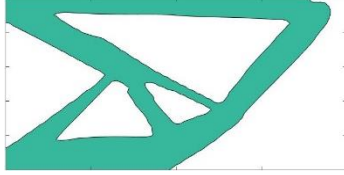
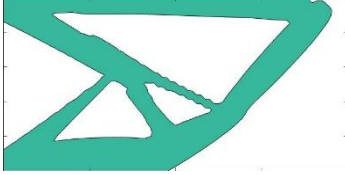
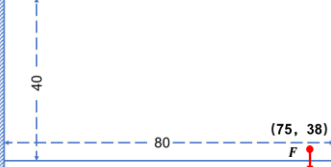
The model prediction results with 16 components' initial distribution are shown in Tab. 3. In comparison, the model prediction results with 8 components' initial distribution are shown in the Tab. 4. The "Obj" in Tab. 3 and Tab. 4 is the objective function value. MSE shows the structural similarity according to Euclidean distance. Multi-scale structural similarity (MSSIM) [27] comprehensively compares the brightness, contrast, and structural similarity of two pictures. When the MSSIM value is close to 1, it means that the similarity of the two images is high.

1 The deep learning model based on small sample set can obtain high prediction  
2 accuracy results by using the data preprocessing method proposed in this work. The  
3 relative errors of the objective function in Tab. 3 are -9.8%, 0.025%, 0.19%, and 5.5%,  
4 respectively, and those of the model prediction structures in Tab. 4 are 3.6%, 2.26%,  
5 0.68%, and 0.35%, respectively. The MES and MSSIM values also show the model's  
6 high prediction accuracy. Tab. 3 and 4 show that the model prediction accuracy is high  
7 for various initial component distributions. The sampling method can avoid the  
8 influence of initial component distribution and extract the information related to the  
9 optimization problem. It ensures the model's applicability to different initial component  
10 distributions.

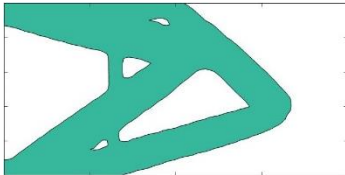
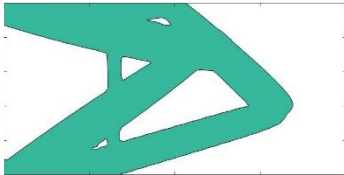
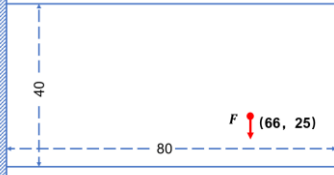
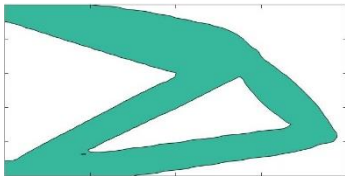
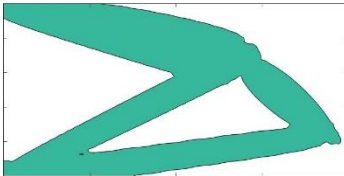
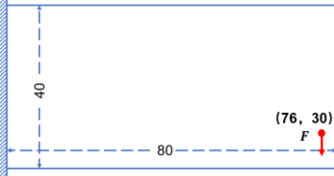
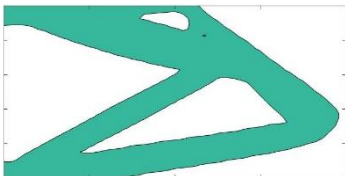
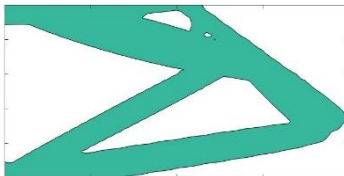
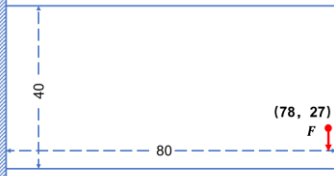
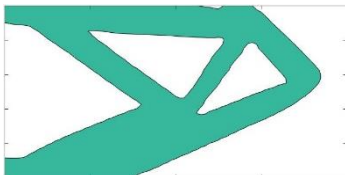
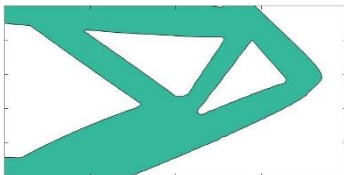
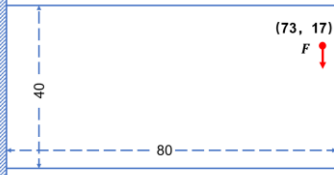
11 The preprocessing method proposed in this manuscript can adjust the range of  
12 TDBI matrix value and keep the structure boundary features accurate. It can enhance  
13 the model's ability to capture the data features. Enhancing the "information" contained  
14 in each sample can reduce the size of the sample set required for real-time topology  
15 optimization. This data preprocessing method is helpful for real-time boundary  
16 evolution optimization methods.

17 At the same time, the CNN model can handle large-scale matrix data, which makes  
18 it possible to use TDBI as the sample label. On the one hand, it avoids the TDF  
19 function's processing of prediction results, which may lead to the components  
20 dislocation phenomenon. For another, the model trained in such labels can directly  
21 predict the final optimized structure without any iteration. The above reasons ensure  
22 the model prediction results' integrity and continuity.

**Tab. 3** Real-time topology optimization results of the cantilever beam based on the CNN (16 initial components distribution).

The final results				Evaluation criteria		Load Condition	
Prediction results (CNN)	Obj	Optimized results (MMC)	Obj	MSSIM	MSE		
1		17.18		17.35	0.98	0.0049	
2		39.86		39.85	0.97	0.0067	
3		10.31		10.29	0.98	0.0047	
4		49.27		46.70	0.98	0.0060	

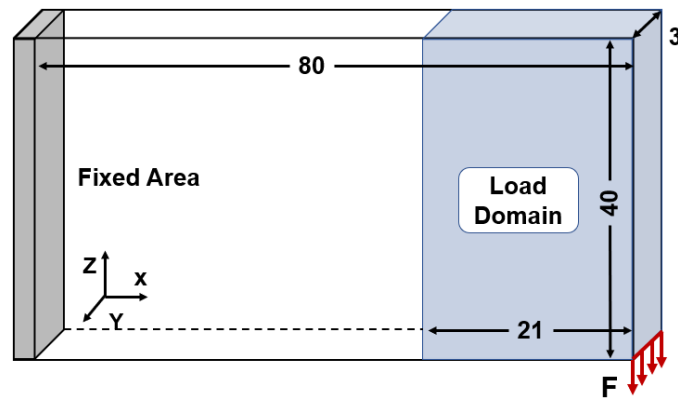
**Tab. 4** Real-time topology optimization results of the cantilever beam based on the CNN (8 initial components distribution).

The final results				Evaluation criteria		Load Condition	
Prediction results (CNN)	Obj	Optimized results (MMC)	Obj	MSSIM	MSE		
1		21.56		20.81	0.97	0.0073	
2		12.20		11.93	0.97	0.0070	
3		40.95		40.67	0.97	0.0066	
4		8.55		8.52	0.98	0.0073	

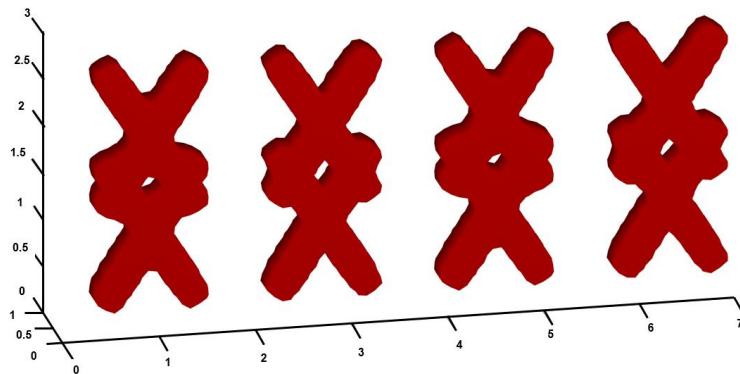


## 4.2. Model prediction results for 3D compliance optimization problem

In this section, a 3D compliance optimization problem is used to evaluate the validity of the proposed method. The structure is discretized by  $80 \times 3 \times 40$ . And the geometry size is  $7 \times 1 \times 3$ . The load condition is shown in Fig. 13 and the initial components distribution are shown in Fig. 14. The Young's model and Poisson ratio are the same as those in 2D optimization problem. The volume fraction constraint is 0.35. The external force location is randomly placed in the load domain with  $21 \times 3 \times 40$  on the right side. After the data enhancement operation, the sample set size is 1088. The training set to validation set ratio is approximately 0.8:0.2.



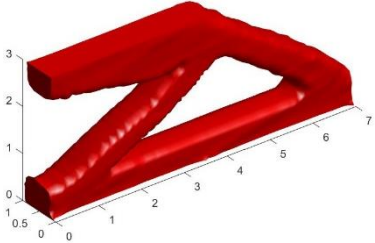
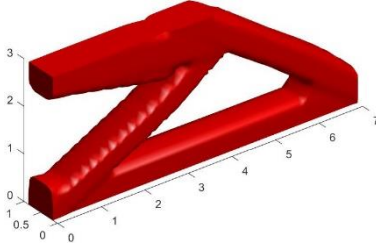
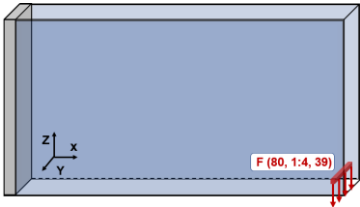
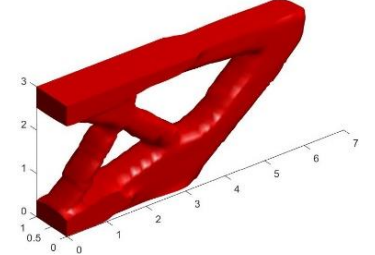
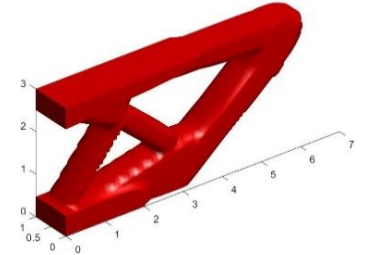
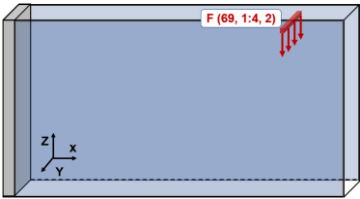
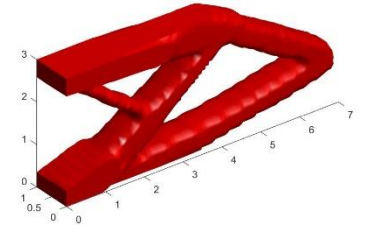
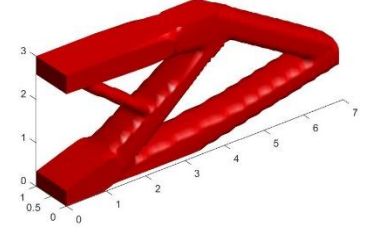

**Fig. 13** The load condition of cantilever beam example for 3D optimization problem



**Fig. 14** The initial components distribution for 3D optimization problem

16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

**Tab.5** Real-time topology optimization results of 3D cantilever beam example

Prediction results (CNN)	Optimized results (MMC)	Evaluation criteria
<div>1</div>  <p>Obj: 136.66</p>	 <p>Obj: 136.14</p>	 <p>Load Point: (80, 1:4, 39), MSE: <math>9.9 \times 10^{-3}</math> Relative error: <math>-0.38\%</math></p>
<div>2</div>  <p>Obj: 53.05</p>	 <p>Obj: 52.48</p>	 <p>Load Point: (69, 1:4, 2), MSE: <math>9.8 \times 10^{-3}</math> Relative error: <math>-1.08\%</math></p>
<div>3</div>  <p>Obj: 64.66</p>	 <p>Obj: 64.32</p>	 <p>Load Point5: (80, 1:4, 27), MSE: <math>1.5 \times 10^{-2}</math> Relative error: <math>-0.54\%</math></p>

For the loading condition shown in Fig. 13, the sampling method is the same as the 2D optimization condition. And the model training history is shown in Fig. 15. The MSE loss function is still used in this model. The relative errors of objective function shown in Tab. 5 are  $-0.38\%$ ,  $-1.08\%$  and  $-0.54\%$  and the MSE values are  $9.9 \times 10^{-3}$ ,  $9.8 \times 10^{-3}$  and  $1.5 \times 10^{-2}$ , respectively. For the validation set of the deep learning model, the model predictions' average absolute error and standard deviation of objective function is  $1.7\%$  and  $2.13 \times 10^{-2}$ , respectively. It can be seen from Tab. 5 that the proposed method is also applicable to 3D MMC optimization problems. The preprocessing method takes into account the mathematical characteristics of MMC method and can effectively improve the sample set accuracy.

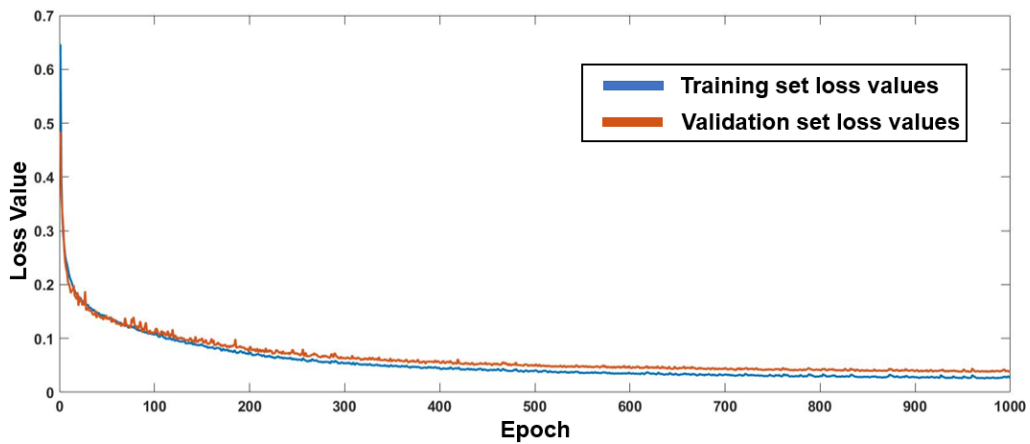


Fig. 15 The loss function value curve for 3D U-Net

Meanwhile, the average prediction time cost of the three conditions (2D structure with 16 components, 2D structure with 8 components and 3D structure with 16 components) is  $1.03 \times 10^{-2}$  s,  $9.29 \times 10^{-3}$  s,  $1.17 \times 10^{-1}$  s. Compared with the classical MMC optimization method, the model prediction time cost can be ignored. A well-trained model can offer the optimized structure in real-time.

### 4.3. Result analysis with different data preprocessing methods and models

As mentioned in the introduction, the previous MMC real-time topology optimization method was based on traditional machine learning methods. Deep learning has a stronger data feature extraction ability than machine learning. This part will compare the prediction effect of the traditional machine learning model and U-Net. The authors will also discuss the influence of different preprocessing methods on the model performance.

#### 4.3.1. Comparison of the U-Net and SVR model

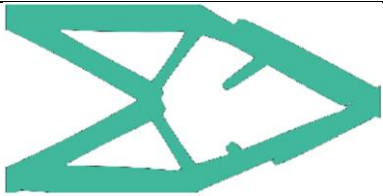
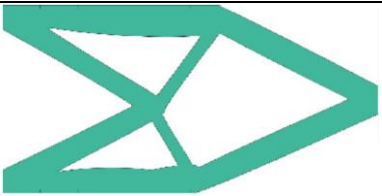
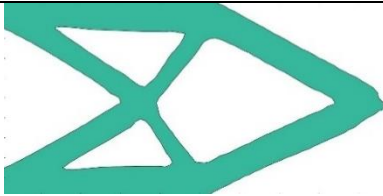
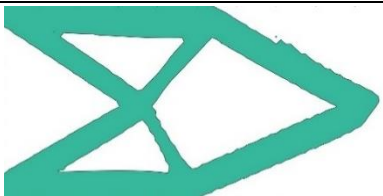

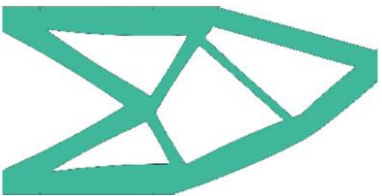
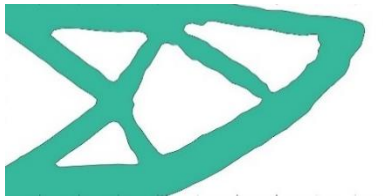
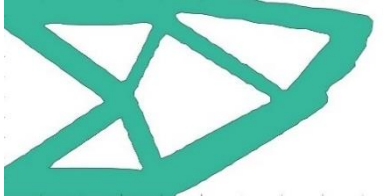

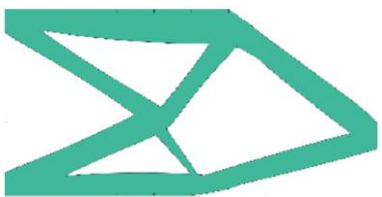
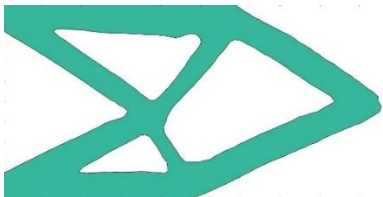
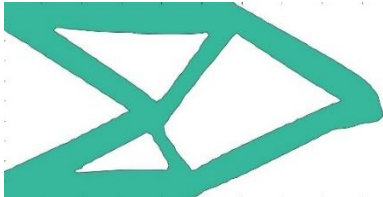
A comparison of the prediction results obtained by the SVR model [20] and the U-Net is shown in Tab. 6. The first and second columns are the prediction results based on the SVR model and reference results, respectively. The third column shows the MSSIM value of the structure in the first two columns. The fourth and fifth columns are the prediction results based on the U-Net and the reference results. The sixth column shows the MSSIM values for U-Net prediction results.

It can be seen from Tab. 6 that the prediction results of both models inherit the advantages of the MMC method, such as clear boundary and local features. However, by comparing MSSIM values and structures predicted by the two models, the accuracy of the structures predicted by U-Net is better than those of SVR. The MSSIM values of U-Net prediction results are significantly improved. There is an obvious component dislocation phenomenon in SVR prediction results, which leads to unfavorable structural continuity. On the contrary, the structures predicted by U-Net have good continuity. There are several reasons for this phenomenon. In addition to the influence

of data preprocessing of the TDBI matrix discussed in section 3, it can also be discussed from the following perspective. Limited by the parameter processing ability of the SVR model, the component geometry description vector  $\mathbf{D}$  is chosen as the sample label.

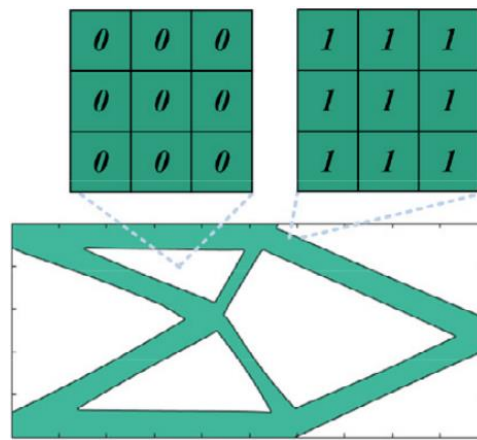
The parameters in matrix  $\mathbf{D}$  contain more information than those in TDBI. The model prediction error of parameters in  $\mathbf{D}$  has great influence on the prediction structures. Therefore, choosing  $\mathbf{D}$  as the sample set label requires higher prediction accuracy. Meanwhile, the parameters in  $\mathbf{D}$  have practical geometric meaning. For example, the prediction error of component tilt angle parameter may be amplified by the component length. The model needs to accurately predict every parameter in  $\mathbf{D}$ . Compared with matrix  $\mathbf{D}$ , the interaction between TDBI parameters is smaller. This means that the prediction error of a single parameter has little influence on the overall structure. Using TDBI as sample set label can reduce the requirement for the model prediction accuracy and effectively avoid the component dislocation phenomenon in prediction results.

**Tab. 6** A comparison of the results obtained with the SVR model and CNN. The SVR results are obtained from Lei [20]. The MSSIM value is calculated based on the prediction results and the reference results optimized by MMC.

SVR Model			CNN		
Predicted	Optimized structure	MSSIM	Predicted	Optimized structure	MSSIM
		0.8162			0.9042
		0.7473			0.8320
		0.7253			0.8936

### 4.3.2. Comparison of different data preprocess methods

Lei [20] adopted component parameters as the sample set labels for combination of MMC and deep learning. Similar to this manuscript, Zheng [33] also adopted the TDBI matrix as the sample set label. Therefore, they also faced the TDBI preprocessing problem. As shown in Fig. 13, a discretization data preprocess method is adopted. The area with the material is set to 1, and no material area is set to 0. The processing method is convenient. The sample set label is converted into a discrete data type. Therefore, based on this label, it is inclined to use the model classification ability to deal with real-time MMC optimization problem. Zheng also pointed out that the binary cross-entropy loss function had the best model training effect, which also supported the above conclusions.

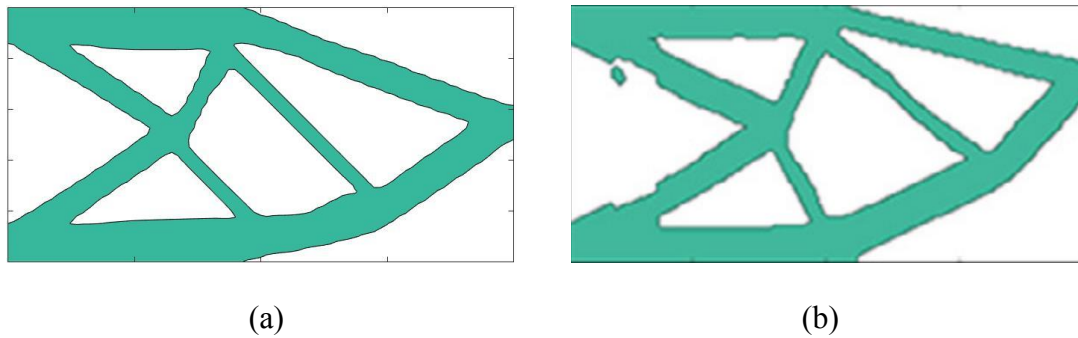


**Fig. 13** The TDBI matrix preprocess method proposed by Zheng [33]

The classical MMC optimization method uses a smooth level set function to describe the structure boundary. If the discretization preprocessing method based on finite element mesh is adopted, the boundary of the structure will be displaced. From the numerical perspective, describing the continuous boundary with discrete data will lead to the movement of structural boundaries. From the graphic perspective, the discrete data processing method will produce zigzag phenomenon at the structural

boundaries. Different from the above method, the data preprocessing method proposed in this manuscript takes the characteristics of the MMC optimization method into account, and proposes a new TDBI matrix preprocessing method. This preprocessing method effectively maintains the accuracy and continuity characteristics of structural boundaries. It also highlights the mathematical features of structure boundaries, which is more conducive to identification and extraction.

The accuracy of sample set label will affect model prediction accuracy. The data preprocessing method proposed in this manuscript will use the regression ability of deep learning model to deal with real-time MMC optimization. This preprocessing method can obtain sample set labels with continuous boundary characteristic. By providing more accurate sample set labels, the training effect of the model can be effectively improved.



**Fig. 14** The model prediction result based on different TDBI preprocess methods: **(a)** Model prediction result based on TDBI preprocess method proposed in this manuscript. **(b)** Model prediction result based on the discrete preprocess method.

## 5. Conclusion

This paper realizes real-time MMC topology optimization based on a trained U-Net with limited sample sets. By analyzing the mathematical characteristics of the MMC method, the TDBI is chosen as the sample label, which effectively avoiding the



component dislocation phenomenon in the model prediction results. CNN can predict the structure with a good continuous boundary. Based on the numerical characteristics of the TDBI, a new data preprocessing method is proposed. This preprocessing method can ensure the mathematical and geometric accuracy of the structure boundary. It can also enhance the data characteristics, and the model can be trained based on a small sample set. Several numerical examples demonstrate the method applicability. The computational efficiency of the MMC method is significantly improved. The algorithm implemented in this study can be applied to real-time topology optimization based on the MMC method, and the preprocessing method can also be employed in real-time topology optimization using other boundary evolution optimization methods.

## References

- [1] K.T. Cheng, On non-smoothness in optimal design of solid, elastic plates, *Int. J. Solids Struct.* 17 (1981) 795–810. [https://doi.org/10.1016/0020-7683\(81\)90089-5](https://doi.org/10.1016/0020-7683(81)90089-5).
- [2] M.P. Bendsøe, Optimal shape design as a material distribution problem, *Struct. Optim.* 1 (1989) 193–202. <https://doi.org/10.1007/BF01650949>.
- [3] Y.M. Xie, G.P. Steven, A simple evolutionary procedure for structural optimization, *Comput. Struct.* 49 (1993) 885–896. [https://doi.org/10.1016/0045-7949\(93\)90035-C](https://doi.org/10.1016/0045-7949(93)90035-C).
- [4] X.Y. Yang, Y.M. Xie, G.P. Steven, O.M. Querin, Bidirectional evolutionary method for stiffness optimization, *AIAA J.* 37 (1999) 1483–1488. <https://doi.org/10.2514/2.626>.
- [5] X. Huang, Smooth topological design of structures using the floating projection, *Eng. Struct.* 208 (2020) 110330. <https://doi.org/10.1016/j.engstruct.2020.110330>.
- [6] Z. Meng, Y. Pang, Y. Pu, X. Wang, New hybrid reliability-based topology optimization method combining fuzzy and probabilistic models for handling epistemic and aleatory uncertainties, *Comput. Methods Appl. Mech. Eng.* 363 (2020) 112886. <https://doi.org/10.1016/j.cma.2020.112886>.
- [7] G. Allaire, F. Jouve, A.M. Toader, A level-set method for shape optimization, *Comptes Rendus Math.* 334 (2002) 1125–1130. [https://doi.org/10.1016/S1631-073X\(02\)02412-3](https://doi.org/10.1016/S1631-073X(02)02412-3).
- [8] P. Wei, H. Ma, M.Y. Wang, The stiffness spreading method for layout optimization of truss structures, *Struct. Multidiscip. Optim.* 49 (2014) 667–682. <https://doi.org/10.1007/s00158-013-1005-7>.
- [9] X. Wang, P. Hu, Z. Kang, Layout optimization of continuum structures embedded with movable components and holes simultaneously, *Struct. Multidiscip. Optim.* 61 (2020) 555–573. <https://doi.org/10.1007/s00158-019-02378-5>.
- [10] J. Gao, L. Gao, Z. Luo, P. Li, Isogeometric topology optimization for continuum structures using density distribution function, *Int. J. Numer. Methods Eng.* 119 (2019) 991–1017. <https://doi.org/10.1002/nme.6081>.
- [11] W. Zhang, J. Yuan, J. Zhang, X. Guo, A new topology optimization approach based on Moving Morphable Components (MMC) and the ersatz material model, *Struct. Multidiscip. Optim.* 53 (2016) 1243–1260. <https://doi.org/10.1007/s00158-015-1372-3>.
- [12] J. Bai, W. Zuo, Hollow structural design in topology optimization via moving morphable component method, *Struct. Multidiscip. Optim.* 61 (2020) 187–205. <https://doi.org/10.1007/s00158-019-02353-0>.
- [13] M.P. Bendsøe, E. Lund, N. Olhoff, O. Sigmund, Topology optimization - Broadening the areas of application, *Control Cybern.* 34 (2005) 7–36. <https://doi.org/10.1007/s00422-004-0522-2>.

- [14] D. Wang, C. Xiang, Y. Pan, A. Chen, X. Zhou, Y. Zhang, A deep convolutional neural network for topology optimization with perceptible generalization ability, *Eng. Optim.* (2021) 1–16.  
<https://doi.org/10.1080/0305215X.2021.1902998>.
- [15] Y. Yu, T. Hur, J. Jung, I.G. Jang, Deep learning for determining a near-optimal topological design without any iteration, *Struct. Multidiscip. Optim.* 59 (2019) 787–799. <https://doi.org/10.1007/s00158-018-2101-5>.
- [16] K. Nakamura, Y. Suzuki, Deep learning-based topological optimization for representing a user-specified design area, *ArXiv E-Prints*. (2020) arXiv:2004.05461. <https://ui.adsabs.harvard.edu/abs/2020arXiv200405461N>.
- [17] Z. Tan, D. Chen, Q. Chu, M. Chai, J. Liao, M. He, L. Yuan, G. Hua, N. Yu, Efficient Semantic Image Synthesis via Class-Adaptive Normalization, *IEEE Trans. Pattern Anal. Mach. Intell.* (2021) 1.  
<https://doi.org/10.1109/TPAMI.2021.3076487>.
- [18] I. Sosnovik, I. Oseledets, Neural networks for topology optimization, *Russ. J. Numer. Anal. Math. Model.* 34 (2019) 215–223. <https://doi.org/10.1515/rnam-2019-0018>.
- [19] X. Guo, W. Zhang, W. Zhong, Doing topology optimization explicitly and geometrically-a new moving morphable components based framework, *J. Appl. Mech. Trans. ASME.* 81 (2014) 1–12. <https://doi.org/10.1115/1.4027609>.
- [20] X. Lei, C. Liu, Z. Du, W. Zhang, X. Guo, Machine learning-driven real-time topology optimization under moving morphable component-based framework, *J. Appl. Mech. Trans. ASME.* 86 (2019). <https://doi.org/10.1115/1.4041319>.
- [21] S. Zheng, H. Fan, Z. Zhang, Z. Tian, K. Jia, Accurate and real-time structural topology prediction driven by deep learning under moving morphable component-based framework, *Appl. Math. Model.* 97 (2021) 522–535.  
<https://doi.org/10.1016/j.apm.2021.04.009>.
- [22] O. Oktay, J. Schlemper, L. Le Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N.Y. Hammerla, B. Kainz, B. Glocker, D. Rueckert, Attention U-Net: Learning Where to Look for the Pancreas, (2018).  
<http://arxiv.org/abs/1804.03999>.
- [23] D.E. Rumelhart, G.E. Hinton, Learning Representations by Back-Propagating Errors, *Nature.* 323 (1986) 533–536.  
<https://doi.org/10.7551/mitpress/1888.003.0013>.
- [24] C.-M. Uang, S. Yin, P. Andres, W. Reeser, F.T.S. Yu, Shift-invariant interpattern association neural network, *Appl. Opt.* 33 (1994) 2147.  
<https://doi.org/10.1364/ao.33.002147>.
- [25] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, *Comput. Sci.* 3(4) (2012) 212–223. <http://arxiv.org/abs/1207.0580>.
- [26] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, *JMLR.Org.* 1 (2015) 448–456.
- [27] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Trans. Image Process.* 13

(2004) 600–612. <https://doi.org/10.1109/TIP.2003.819861>.

- [28] G. Xavier, B. Yoshua, Understanding the difficulty of training deep feedforward neural networks, *J. Mach. Learn. Res.* 9 (2010) 249–256. <https://doi.org/10.1109/LGRS.2016.2565705>.
- [29] Q. Lin, J. Hong, Z. Liu, B. Li, J. Wang, Investigation into the topology optimization for conductive heat transfer based on deep learning approach, *Int. Commun. Heat Mass Transf.* 97 (2018) 103–109. <https://doi.org/10.1016/j.icheatmasstransfer.2018.07.001>.
- [30] J. Yan, Q. Zhang, Q. Xu, Z. Fan, H. Li, W. Sun, G. Wang, Deep learning driven real time topology optimisation based on initial stress learning, *Adv. Eng. Informatics.* 51 (2022). <https://doi.org/10.1016/j.aei.2021.101472>.
- [31] W. Weng, X. Zhu, U-Net: Convolutional Networks for Biomedical Image Segmentation, *IEEE.* 9 (2021) 16591–16603. <https://doi.org/10.1109/ACCESS.2021.3053408>.
- [32] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, *Comput. Sci.* (2014) arXiv:1412.6980.
- [33] S. Zheng, H. Fan, Z. Zhang, Z. Tian, K. Jia, Accurate and real-time structural topology prediction driven by deep learning under moving morphable component-based framework, *Appl. Math. Model.* 97 (2021) 522–535. <https://doi.org/10.1016/j.apm.2021.04.009>.