Real-Time Topology Optimization Based on Convolutional Neural Network by Using Retrain Skill

Jun Yan[a*], Dongling Geng[b], Qi Xu[c], Haijiang Li[d*]

[a] Department of Engineering Mechanics, Dalian University of Technology, Liaoning, China,
Email: yanjun@dlut.edu.cn
[b] Department of Engineering Mechanics, Dalian University of Technology, Liaoning, China,
Email: 22003094@mail.dlut.edu.cn
[c] Department of Engineering Mechanics, Dalian University of Technology, Liaoning, China ,
Email: namexuqi@mail.dlut.edu.cn
[d] School of Engineering, Cardiff University, Cardiff, the United Kingdom,
Email: lih@cardiff.ac.uk

***Corresponding Author**:
Prof. Jun Yan
Department of Engineering Mechanics, Dalian University of Technology
Email: yanjun@dlut.edu.cn
No.2 Linggong Road, Ganjingzi District, Dalian, Liaoning Province, 116024

Prof. Haijiang Li
School of Engineering, Cardiff University,
Email: lih@cardiff.ac.uk
S2.10C, Queen's Buildings - South Building, 5 The Parade, Newport Road, Cardiff, CF24 3AA

Abstract: To realize a real-time structural Topology Optimization (TO), it is essential to use the information during the TO process. A step-to-step training method is proposed to improve the deep learning model prediction accuracy based on the Solid Isotropic Material with Penalization (SIMP) TO method. By increasing the use of optimization history information (such as the structure density matrix), the step-to-step method improves the model utilization efficiency for each sample data. This training method can effectively improve the deep learning model prediction accuracy without increasing the sample set size. The step-to-step training method combines several independent deep learning models (sub-models). The sub-models could have the same model layers and hyperparameters. It can be trained in parallel to speed up the training process. During the deep learning model training process, these features reduce the difficulties in adjusting sub-model parameters and the model training time cost. Meanwhile, this method is achieved by the local end-to-end training process. During the deep learning model predicting process, the increase in total prediction time cost can be ignored. The trained deep learning models can predict the optimized structures in real-time. Maximization of first eigenfrequency topology optimization problem with three constraint conditions is used to verify the effectiveness of the proposed training method. The method proposed in this study provides an implementation technology for the real-time TO of structures. The authors also provide the deep learning model code and the dataset in this manuscript (git-hub).

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

1. Introduction

With the rapid development of numerical analysis, structural topology optimization[1] has been widely applied in aerospace structures[2] and automotive[3] fields. The topology optimization method can determine the optimal distribution of materials to achieve an optimized structural performance[4]. Currently, many topology optimization methods have been developed, such as the solid isotropic material with penalization (SIMP) method[5], the moving morphable component method[6], and the evolution structure optimization method[7]. These methods have been successfully applied to static[8, 9], dynamic[10, 11], thermodynamic[12, 13], and other multidisciplinary optimization problems. However, the entire iterative process of classical topology optimization methods is accompanied by finite element calculation and sensitivity analysis. The calculation time cost increases rapidly with an increase in elements and problem dimensions. Therefore, a new algorithm is developed to realize real-time topology optimization.

Recently, deep learning has developed rapidly. It has been successfully applied to image classification[14], target detection[15], and language translation[16] problems. A neuron with simple operations is the basic unit of the deep learning model. The parameters in the model are updated using a gradient descent algorithm. Real-time topology optimization is the combination of topology optimization and deep learning. With the model training process, the model can gradually determine the approximate functional relationship between the input and output data. The rapid response of the trained deep learning model provides the possibility of calculating the topology optimization in real-time.

To accelerate the topology optimization process using a deep learning model, some scholars have replaced the topology optimization process with a deep learning model, which is usually realized by the offline model training method. Yu[17] used a convolutional neural network (CNN) to predict low-resolution results (32×32) using the load and constraint information in the X and Y directions. Subsequently, a conditional generational adversarial network was used to convert the low-resolution results to high-resolution results (128×128). Based on a study by Yu[17], Keigo[18] improved the prediction accuracy of a deep learning model by adding SPADE[19] layers to the model. Wang[20] trained a U-Net model based on displacement and strain information. The trained model could achieve multiple load conditions for real-time structural topology optimization. The above scholars accelerated the topology optimization process based on a deep learning model with an offline model training method. The offline training method means that deep learning model training is entirely independent of topology optimization. This method generally constructed the relationship between the initial physical information (like the load condition or volume fraction) and the optimized result. So this model training method is also known as the “end-to-end” method.

The method has the advantage of high prediction efficiency and is easily implemented.

In contrast to the offline training method, some scholars have integrated the deep learning models into the topology optimization calculation called the online training method. Aaditya[21] used a neural network to obtain the element density using the SIMP topology optimization method. As a result, optimal structures with different resolutions can be produced. To accelerate the topology optimization process, Chi[22] used a deep learning model to predict sensitivity information and replaced the classical analysis in some iterative steps. Guo[23] used a deep learning model to obtain microstructural stiffness and then accelerated the entire calculation process of multi-scale topology optimization. Hao[24] used a fully connected feedforward neural network to replace the implicit expression of the classical level set function, which can effectively generate diversified structures. The deep learning model in the online training method replaces part of the classical optimization process (such as accelerating sensitivity analysis[22] or obtaining multiple resolution results in one optimization calculation[21]). The optimization is still primarily controlled by classical optimization algorithms. In other words, the job complexity handled by the model in the online learning method is relatively small. This characteristic demonstrates that the model does not require high complexity and has relatively lax criteria for the sample set size. Meanwhile, the optimization results obtained by this method generally have higher accuracy and better adaptability for various situations because the optimization is still primarily based on classical optimization algorithms.

In addition to the influence of the model training methods discussed above, the ability of the deep learning model to solve nonlinear problems (learning ability) can also significantly influence the model's prediction accuracy[25]. Generally, properly adjusting the structure of the deep learning model can enhance its nonlinear preprocessing ability. The structural disconnection problem often occurred in SIMP real-time prediction results. By combining multiple deep learning models with different loss functions, Gorkem[26] alleviated this problem and improved the prediction accuracy of the deep learning model. Contrary to enhancing the learning ability of the deep learning model, Lin[27] divided the heat-dissipation topology optimization process into two stages and realized real-time topology optimization in the second stage using a CNN. The deep learning model can also be reduced by decreasing the complexity of the problem being solved. However, the model usually depends on the size of the sample set. The deep learning model may have an over-fitting problem owing to insufficient training with a limited sample-set size. To provide more valuable data for the deep learning model, Wang[20] increased the initial data type of the model input. The principal stress matrix contains more optimization information than the load and constraint information. Using the initial principal stress matrix as the initial input data for the model, Yan[28] solved the problem discussed above.

In this paper, a step-to-step (STS) training method is proposed. This method is an attempt to improve the popular online and offline training methods. Compared with the online training method, the STS method has better efficiency improvement. By increasing the use of optimization history information, the STS method improves the model utilization efficiency for each example. Without increasing the sample set size, the prediction accuracy of the deep learning model is improved. The low prediction accuracy is usually presented as structure fracture or gray element phenomenon (for SIMP method). Therefore, in addition to taking the objective function and loss function as the model evaluation criterion, a new function is proposed to evaluate the non-discreteness of structures. The training method is introduced in Section 3. In Section 4, several examples are used to verify the effectiveness of the proposed method. The final section summarizes the study.

2. The basic theory of fundamental frequency topology optimization based on the density method and convolutional neural network

2.1 Topology optimization of maximizing the fundamental eigenfrequency based on the density method

Structural vibration and noise control designs are significant in many industrial fields[29]. Maximizing the fundamental eigenfrequency of structural vibrations can effectively avoid structural resonance problems[10]. The SIMP topology optimization method uses explicit interpolation to describe the structural stiffness and mass matrices. The formulas for the topology optimization problem of maximizing the fundamental frequency based on SIMP with a structure volume constraint are as follows:

$$\text{find} \quad \mathbf{z} = \{\rho_i\} \tag{2.1}$$

$$\max \quad \lambda_1 = \omega_1^2 \tag{2.2}$$

$$\text{subject to} \quad \mathbf{K}\emptyset_j = \lambda_j \mathbf{M}\emptyset_j, j = 1,2,\dots,\mathrm{J} \tag{2.3}$$

$$\emptyset_j^T \mathbf{M}\emptyset_k = \delta_{jk}, k, j = 1,2,\dots,\mathrm{J} \tag{2.4}$$

$$\textstyle\sum_{e=1}^{N_E} \rho_e V_e - V^* \leq 0, V^* = \alpha V_0 \tag{2.5}$$

$$0 < \hat{\rho} \leq \rho_e \leq 1, e = 1,\dots,N_E \tag{2.6}$$

In Eq. (2.1), ρ_i is the element density, and $\mathbf{z}$ is the set of design variables. In Eqs. (2.2) and (2.3), λ_j is the j^{th} eigenvalue, and $\emptyset_j$ is the eigenvector corresponding to this eigenvalue. J is the total freedom degree of the structure. $\mathbf{K}$ and $\mathbf{M}$ are the stiffness and mass matrices of the structure, respectively. Eq. (2.4) is the normalization condition of the eigenvectors, where δ_{jk} is the Kronecker notation. Eq. (2.5) is the volume constraint condition for the structural optimization, where V^* is the upper limit of the allowable volume of the structure, α is the volume fraction, and V_0 is the total volume of the structure. N_E is the number of finite elements in the structure. To avoid numerical singularity difficulties during the topology optimization process, we set $\hat{\rho}$ as the lower bound of the element density ($\hat{\rho} = 0.001$ in this paper). $\mathbf{K}$ and $\mathbf{M}$ are assembled using the element stiffness matrix ($\mathbf{K}_e^*$) and mass matrix ($\mathbf{M}_e^*$), respectively, as shown in Eqs. (2.7) and (2.8).

$$\mathbf{K} = \sum_{e=1}^{N_E} \rho_e^p \mathbf{K}_e^* \tag{2.7}$$

$$\mathbf{M} = \sum_{e=1}^{N_E} \rho_e^q \mathbf{M}_e^* \tag{2.8}$$

p and q are the penalty coefficients of the stiffness and mass matrices, respectively ($p = 3$, $q = 2$). The method of moving asymptotes (MMA)[30] optimization algorithm is used for parameter update.

2.2 Theory of the convolutional neural network

CNN is widely used in feature extraction and image recognition[31]. Compared with the traditional back-propagation neural network (BP)[32], CNN has convolution and pooling layers. The convolution layer can extract features from samples, even if some feature-invariant operations, such as translation, rotation, and edge shearing, exist. Therefore, a CNN is also known as a 'translation-invariant neural network'[33].

The artificial-material-density topology optimization method uses finite element density values to describe structural geometry. The fundamental concept of this method is to discretize the density values of the finite element with 0 or 1 to represent the material distribution in the structure. When the density value is 1, the element is filled with the material. When the density value is 0, the element contains no material. However, in the actual optimization process, using an efficient gradient descent algorithm to update the discrete optimization variables is difficult. The discrete variables are frequently relaxed, and the element density ρ is defined as a continuous variable within [0, 1][34]. Owing to the characteristics discussed above, the configuration of the optimization results of the artificial-material-density topology optimization methods is frequently presented as a grey image.

The CNN has the local connection and weight-sharing characteristics[35], which can reduce the number of parameters of the CNN and enhance the feature extraction capability for large-scale matrix data represented by images. Therefore, CNN is adopted to solve the real-time topology optimization problem based on the SIMP method. Combining topology optimization with a deep learning model improves the efficiency of the structural vibration topology optimization calculation.

3. Proposed Step-to-Step Training Method

3.1 Structure of the submodels in the step-to-step training method

The structure of the CNN used in this study is shown in Figure 1. The model is composed of the encoder, decoder, and reducing parts. Activation functions can enhance the model nonlinear learning ability. The final layer adopts the sigmoid activation function[36], and the other uses the ReLU function[37]. The specific expressions of these two functions are shown in Eq. (3.1) and Eq. (3.2).

$$f(x)=\frac{1}{1+e^{-x}} \tag{3.1}$$

$$f(x)=\begin{cases} x & if\ x>0 \\ 0 & if\ x\leq 0 \end{cases} \tag{3.2}$$

The encoder part (including block-1, block-2, block-3, block-4 and block-5) encodes the input data and reduces the data dimension through a convolution operation; the decoder part (including block-6, block-7, block-8, block-9 and block-10) realizes data up-sampling through the transpose convolution operation[38]. Traditional CNN uses a fully connected layer to adjust output data size. And the total model parameters number will be enormous if the output matrix is large. However, structure topology optimization calculation is an element size-dependent problem. When the sample set size is limited, the over-fitting problem influences model prediction accuracy[39]. On the one hand, the model alleviated this phenomenon by adding the dropout layer[40], which is also a standard treatment. There is a hyperparameter called dropout rate in the dropout layer. In the current iteration, the dropout layer could arbitrarily cause some neurons to lose their learning capacity. It reduces the number of all parameters of the model to some extent. On the other hand, the reducing part imitates the characteristics of Google Net[41] and uses the convolution layer to replace the full connect layer in the model. This method can also reduce the number of parameters of the entire model. The table in the appendix shows the trainable parameters in two models. The layers without trainable parameters are omitted (such as the dropout layer and add layer). With a 221×31 matrix as the model input and sample set label, the number of model parameters with the reducing part is approximate $1/53$ of the model without it. The reducing part effectively reduces the parameter size of the deep learning model and enables the model to be trained with a limited sample set.

The mean square error (MSE) loss function is used in this model, as shown in Eq. (3.3), where $\hat{y}$ is the MES function result, y_i and y_i^p represent the original data and prediction result of the deep learning model, respectively, and n is the whole sample set. On the one hand, the MSE value can reflect the similarity degree of the two structures using the Euler distance. On the other hand, the convergence of the deep learning model can be evaluated by observing the change in the loss function value during the model training process.

$$\hat{y} = \frac{\sum_{i=1,\ldots,n}^{n}(y_i - y_i^p)^2}{n} \tag{3.3}$$

The CNN in this study used the adaptive moment estimation (Adam) optimizer[42]. This manuscript uses NVIDIA Corporation TU102 [Titan RTX] for 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.

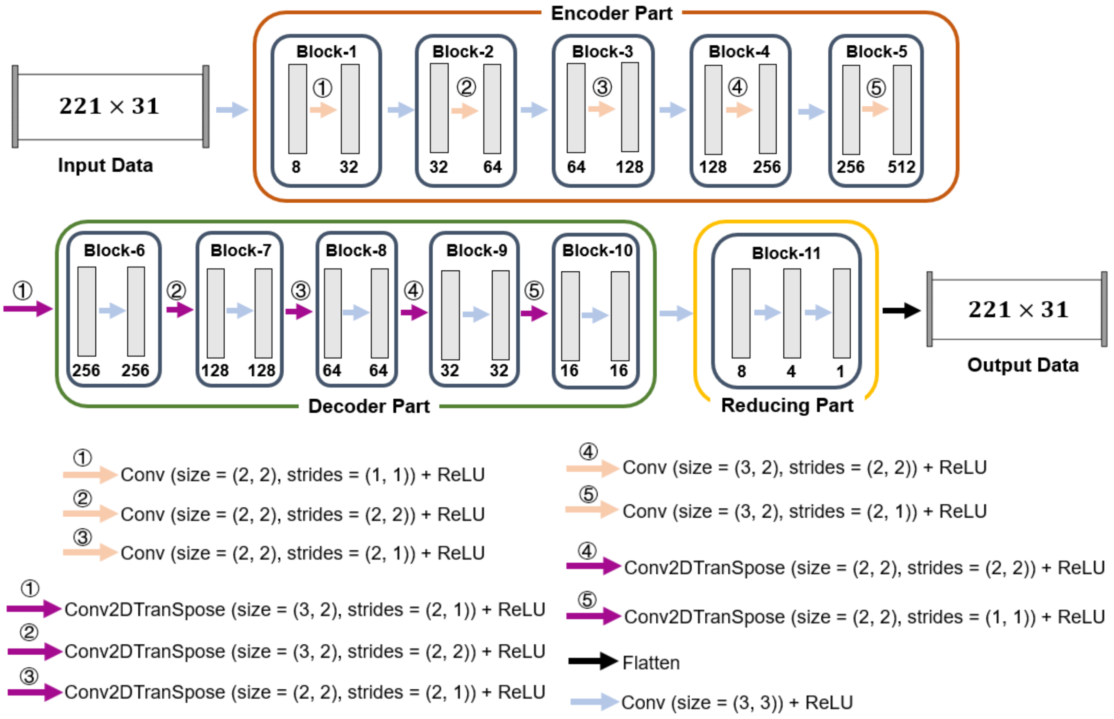

Encoder Part
221 × 31
Input Data
Block-1
①
8 32
Block-2
②
32 64
Block-3
③
64 128
Block-4
④
128 256
Block-5
⑤
256 512
①
Block-6
256 256
②
Block-7
128 128
③
Block-8
64 64
④
Block-9
32 32
⑤
Block-10
16 16
Decoder Part
Block-11
8 4 1
Reducing Part
221 × 31
Output Data
① Conv (size = (2, 2), strides = (1, 1)) + ReLU
② Conv (size = (2, 2), strides = (2, 2)) + ReLU
③ Conv (size = (2, 2), strides = (2, 1)) + ReLU
④ Conv (size = (3, 2), strides = (2, 2)) + ReLU
⑤ Conv (size = (3, 2), strides = (2, 1)) + ReLU
① Conv2DTranSpose (size = (3, 2), strides = (2, 1)) + ReLU
② Conv2DTranSpose (size = (3, 2), strides = (2, 2)) + ReLU
③ Conv2DTranSpose (size = (2, 2), strides = (2, 1)) + ReLU
④ Conv2DTranSpose (size = (2, 2), strides = (2, 2)) + ReLU
⑤ Conv2DTranSpose (size = (2, 2), strides = (1, 1)) + ReLU
Flatten
Conv (size = (3, 3)) + ReLU

Figure.1 Structure of CNN with encoder, decoder, and reducing Parts. The encoder part contains four blocks to downscale data size; the decoder part contains three blocks for data upsampling; the reducing part is used to resize the output data without dramatically increasing the total parameters of the CNN.

3.2 Stages of Topology Optimization Process

The topology optimization has two stages: Stage-1: a dramatic change occurs in the material distribution. A preliminary force transmission path is formed. Stage-2: local areas of the structure are optimized[27]. In the SIMP topology optimization method, the optimization process is frequently accompanied by the disappearance of intermediate density elements and the appearance of ‘black’ elements (whose element density is close to 1) and ‘white’ elements (whose element density is close to 0). The number of intermediate density elements can reflect both the structure clarity and the topology optimization iterative process. Therefore, based on the change in intermediate density elements, the topology optimization process can be divided into the preliminary optimization stage and the advanced optimization stage.

$$\psi = \frac{\eta}{N} \tag{3.4}$$

In Eq. (3.4), ψ is the grey element proportion coefficient. and η is the number of elements with element density values in the range $[\alpha - 0.1, \alpha + 0.1]$ (α is the volume fraction). For example, in Figure 2, both ends of the structure are fixed and the maximization structure fundamental eigenfrequency optimization problem is considered. The figure shows the change in ψ during the optimization process. The material distribution changes significantly during the preliminary optimization stage (the first 18 iterations for this load condition). An approximate structure is generated (the main load path is generated), and ψ decreases gradually. In the advanced optimization stage, the structure gradually generates local features and obtains the final results. ψ is usually a parameter determined by calculation experience. For the maximization fundamental eigenfrequency optimization problem, the structure is defined as the middle result when the value of ψ decreases below 0.1 for the first time.

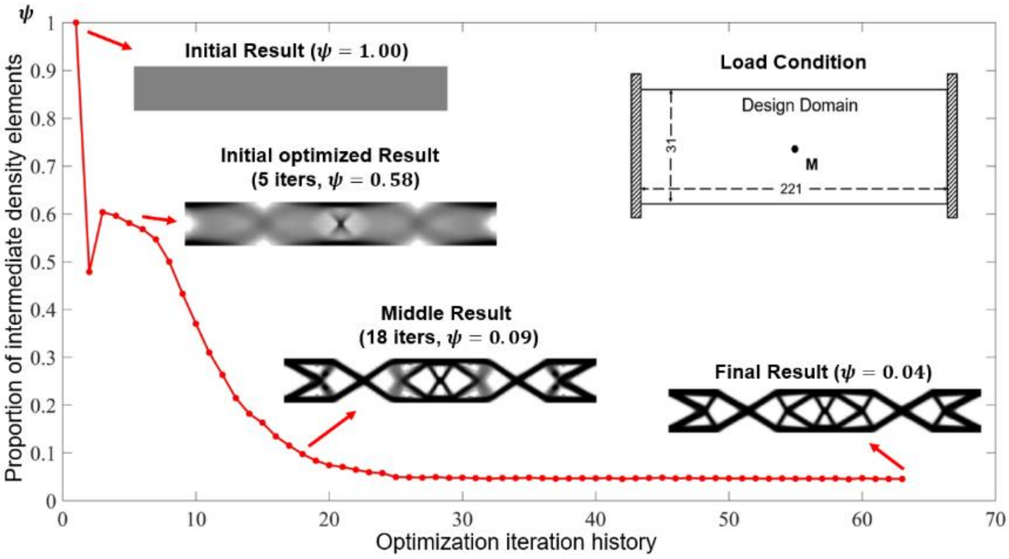


Figure.2 Value changes of ψ with the topology optimization process. The value of ψ for the initial result is 1.00. There are only middle-density elements in the design domain. The initial optimized result is the structure after five iterations. The structure whose ψ value decreases below 0.1 for the first time is called the middle result.

3.3 Step-to-Step Training Method

As discussed in Section 1, the traditional end-to-end (ETE) model training method[17, 18] (offline training method) frequently determines the functional relationship between the initial optimization information and final optimization results through the model training process. However, a dramatic change occurs in the material distribution during topology optimization. The traditional ETE training method cannot easily obtain accurate prediction results with a limited dataset[43]. However, the traditional online model training method[22, 23] frequently embeds a deep learning model in the calculation process of topology optimization. The model training process is coupled with topology optimization calculation, and it limits the optimization efficiency improvement.

The STS training method proposed in this paper is realized by combining two sub-models, and each sub-model is trained based on a local ETE training method. Figure 3 shows the steps of the proposed STS training method. The first step is to accumulate a dataset for model training. Information about the initial structure, load conditions, and optimized structure is saved. Additionally, the structure of the middle result between the preliminary optimization stage and advanced optimization stage is saved. In the second step, the deep learning model is trained based on the dataset obtained in the first step. The input of Model-1 is the initial structural information, and the label of the dataset is the optimized structure. The input of Model-2 is information about the middle result, and the optimized structure is also taken as the label.

Model-1 is trained using the traditional offline training method. However, an inevitable model prediction error occurs when the dataset is limited. Meanwhile, the model prediction error frequently occurs as a grey element problem and is primarily distributed in the local area of the structure[20, 26, 28]. In the early stages of the advanced optimization stage, many grey element distributions occur in the local area of the structure. Along with the optimization process, the local features of the structure gradually became clear. Considering the above characteristics during the topology optimization process, Model-2 is trained to solve the grey element problem and improve the accuracy of the prediction result of Model-1. Compared with Model-2, Model-1 requires a higher model learning ability. Thus, the structure of Model-1 is also suitable for Model-2. This characteristic can reduce the time cost during the model training process of the STS training method. Additionally, Model-1 and Model-2 do not have a coupling relationship during the training process. Both models can be trained in parallel. Compared with the traditional offline training method, the proposed STS training method has more deep learning models; however, it does not produce a significant time-cost increase in sample set accumulation, sub-model parameters adjustment, and sub-model training time-cost. The proposed STS training method is based on ensemble learning[44]. By providing different training data, each submodel has its functional characteristics, and the submodels are combined to achieve real-time topology optimization with high prediction accuracy. In this paper, the structures of both submodels are the same.

As shown in Step-3 in Figure 3, the Model-1 prediction results are processed using Model-2 to improve the prediction accuracy. However, both sub-models are trained using the local ETE training method. Compared with classical topology optimization, the total model prediction time cost can be ignored. The optimized structure can also be predicted in approximately real-time. In this paper, proposed STS training method's prediction efficiency and effect are verified using structural fundamental frequency optimization examples under three constrained conditions. This paper also compares the model training effect of the ETE and STS model training methods, proving the proposed method's effectiveness.

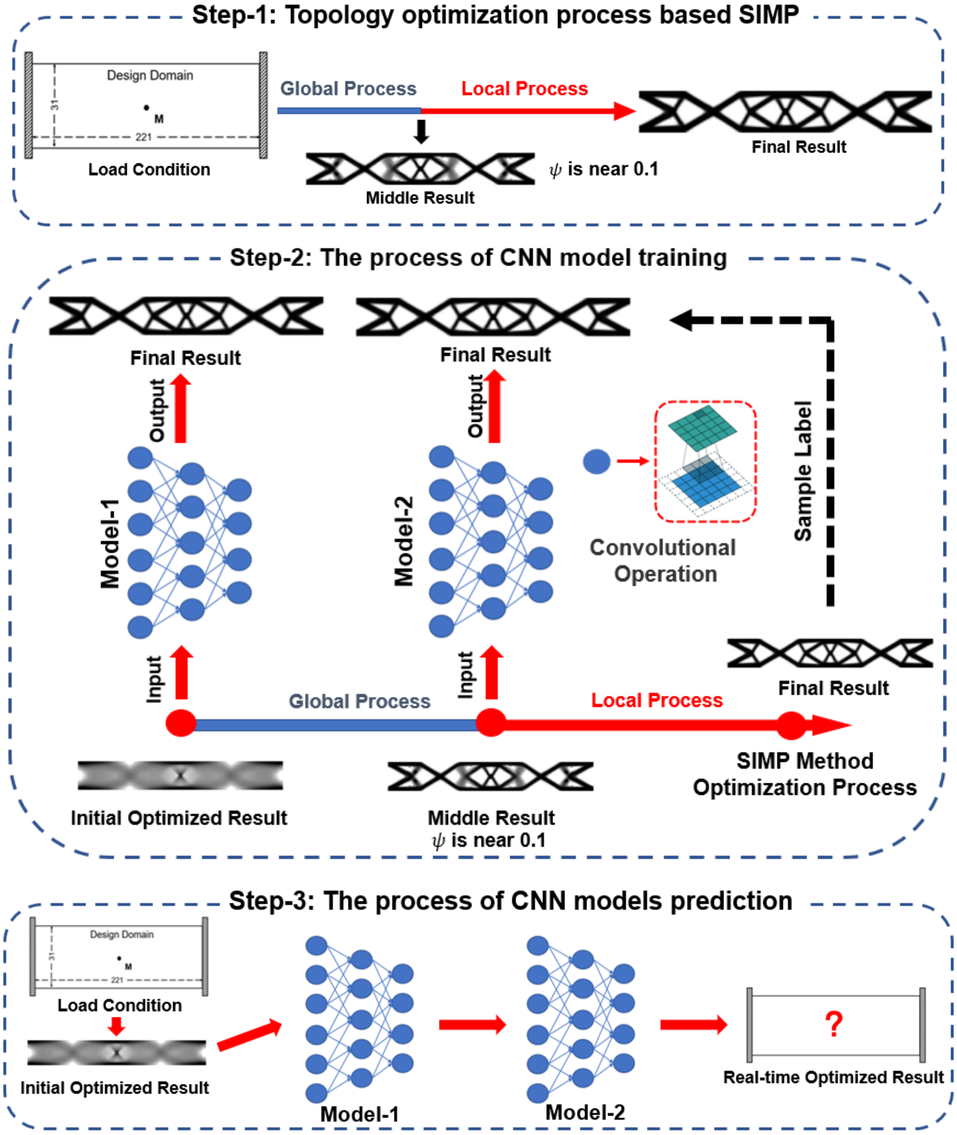


Figure.3 Description of the STS training method: Step-1 is the process of sample set accumulation; Step-2 is the sub-models (CNN) training process; Step-3 is the prediction process of CNN models.

3.4 Evaluation Function for Structure Clarity in SIMP

As mentioned in Section 3.3, compared with the traditional end-to-end training method, the step-to-step training method proposed has an additional "post-processing process". Model-2 is trained to learn the ability of classical optimization methods to deal with the unclear structural features of local regions. Therefore, an effective structure clarity evaluation mechanism should be determined. The MSE loss function compares the similarity of the two matrices using Euler distance. The model training situation can be obtained by the MSE value changes. However, the MSE function is more concerned with the difference between two matrices. It cannot reflect the changes in intermediate density element numbers. There are filtering techniques in classical topology optimization methods. M_{nd} proposed by Sigmund[45] is a popular method[46], $M_{nd} = \frac{\sum_{e=1}^{n} 4\rho(1-\rho)}{n} \times 100\%$. However, the weight of the M_{nd} function for each element density is fixed. This manuscript proposes a new evaluation function using the normal distribution function, as shown in Eq. (3.5).

$$\mathrm{TC} = \frac{1}{N_e} \times \sum_{i=1}^{\mathrm{N_e}} e^{\frac{-(\rho_i-\mu)^2}{2\sigma^2}}, \quad \rho_i \in [0,1] \tag{3.5}$$

In Eq. (3.5), N_e is the same as that in Eqs. (2.7) and (2.8), ρ_i represents the i^th element density value, and μ and σ^2 represent the mean and variance of the normal distribution function, respectively.

Figure 4 shows that 95.45% of the area of the normal distribution function is within $[\mu - 2\sigma, \mu + 2\sigma]$. This region is the central evaluation region of the target clarity (TC) function. The number of elements in the central evaluation area directly influences the TC value of the structure. The change in the TC value can be used to evaluate the clarity of the entire structure. For example, when all element density is 0.5, the TC value is 1. Only intermediate-density elements exist in the structure. If the TC value approaches 0, the element's density tends to be 0 or 1 (the structure tends to have a clear boundary distribution).

Figure 4 shows the weight distribution of the TC $(\mu = 0.5,\ \sigma^2 = 0.1^2)$ function, and the main evaluation region is $[0.3,\ 0.7]$. First, all the density elements in the structure are flattened. Second, the score of each element is calculated. Finally, the TC value is obtained using the average of all element scores.

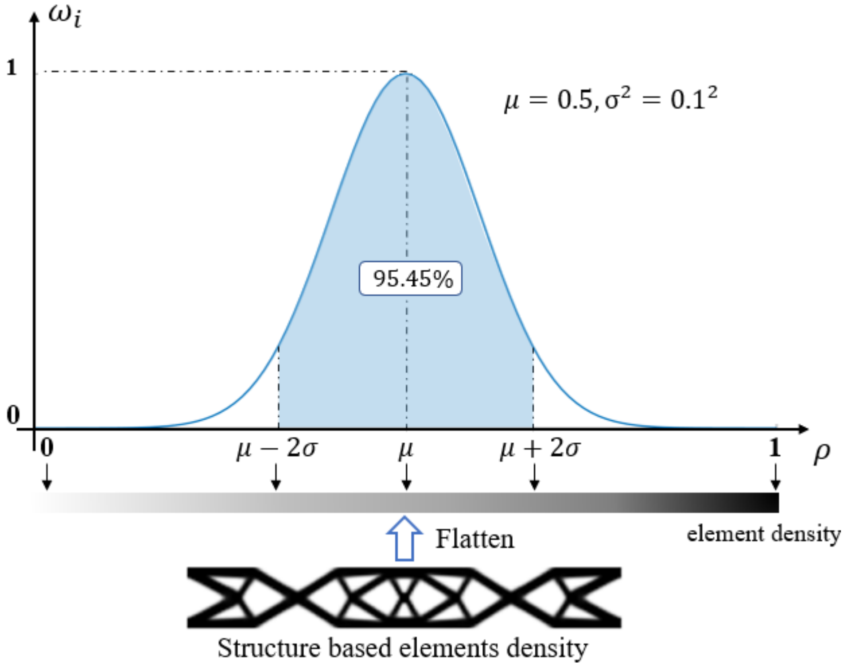


Figure.4 Weight distrubution and calculation process of TC ($\mu = 0.5,\ \sigma^2 = 0.1^2$) function.

The grey element problem in the SIMP method primarily raises from the existence of intermediate-density elements in the structure. However, defining the specific density range of the grey elements is difficult. Therefore, this paper evaluates the clarity of the structures based on three different σ^2 values ($\sigma_1^2 = 0.05^2$, $\sigma_2^2 = 0.1^2$, $\sigma_3^2 = 0.15^2$). As shown in Figure 5, the main evaluation region of the TC function increases with an increase in σ^2. The main evaluation regions of σ_1^2, σ_2^2, and σ_3^2 are $[0.4, 0.6]$, $[0.3, 0.7]$ and $[0.2, 0.8]$, respectively. Therefore, by changing the value of σ^2, the TC function can evaluate the clarity of the structure in the different evaluation regions.

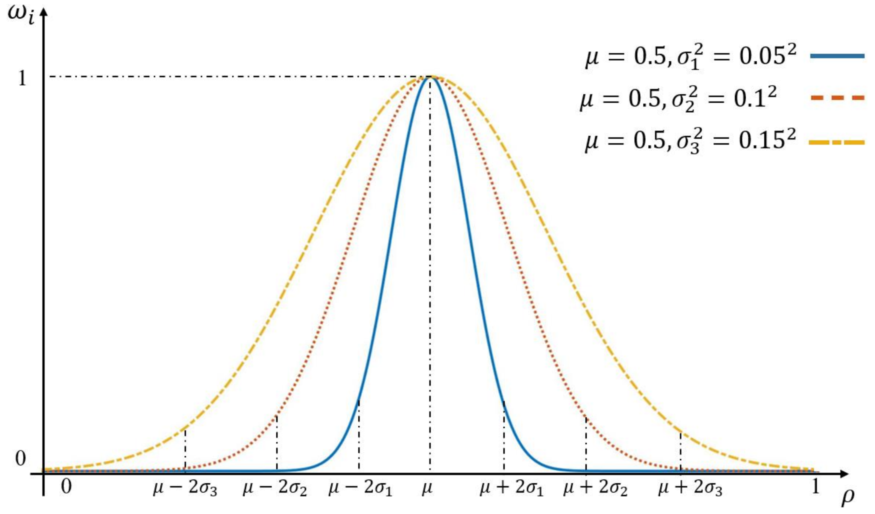


Figure.5 Main evaluation area of TC function with different σ^2 values. The blue, red, and yellow lines represent TC functions with $\sigma^2 = 0.05^2, \sigma^2 = 0.1^2$ and $\sigma^2 = 0.15^2$, respectively.

Table 1 shows the TC function scores with different σ^2 values when the density is 0 or 1. When the element density is 0 or 1, the scores of the TC function are always close to 0. Therefore, when the TC function value is close to 0, the structure tends to appear in the form of 'black' or 'white' elements and has a clear geometric configuration.

Table 1 Value of TC ($\mu = 0.5$, $\rho_i = 0$ or 1) with different σ^2. The value of TC at point $\rho_i = 0$ or 1 is close to 0 with different σ^2.

Different σ^2 values	$\sigma^2 = 0.05^2$	$\sigma^2 = 0.1^2$	$\sigma^2 = 0.15^2$
TC ($\rho_i = 0$ or 1)	1.93×10^{-22}	3.73×10^{-6}	3.87×10^{-3}

4. Verification of Step-to-Step Training Method

In this section, three examples under different constraint conditions are used to verify the effectiveness of the proposed STS model training method. The structure is discretized by 221×31 elements, and the volume fraction constraint α is set to 0.5. The conditions for the three examples are shown in Figure 6. In Condition-1, the middle nodes of the left and right boundaries of the structure are fixed. In Condition-2, both boundaries on the left and right sides of the structure are fixed. In Condition-3, the left boundary of the structure and the middle node of the right boundary are fixed. The concentrated mass is randomly distributed in the load domain for three conditions (blue region in Figure 4, Condition-1: 81×31, Condition-2: 41×31, Condition-3: 75×31). The initial structure, middle structure and optimized structure density matrix will be saved. And they will be disrupted to form the sample set. The total number of samples is 6107 (Condition-1: 2511, Condition-2: 1271, Condition-3: 2325), and 150 samples are extracted from each condition as the verification set.

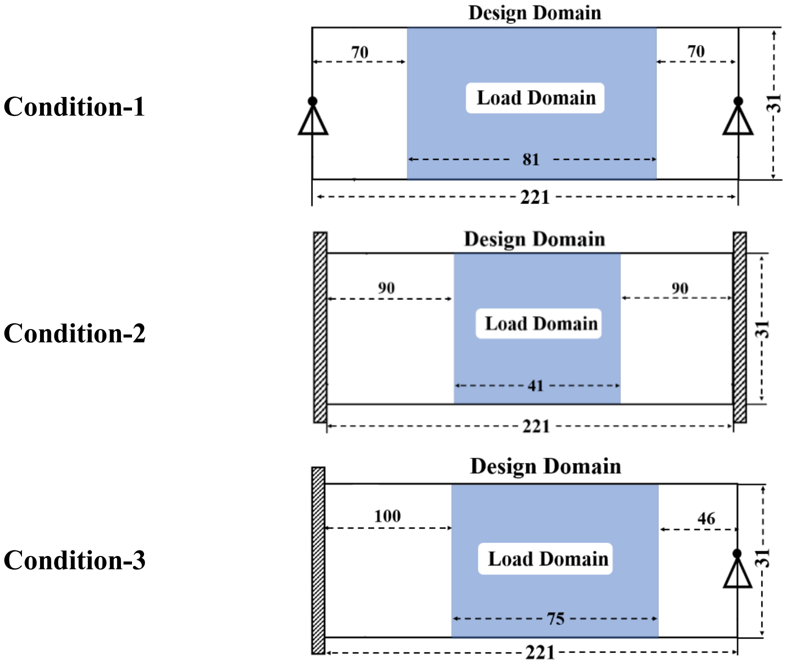


Figure.6 Different constraints for Condition-1, Condition-2, and Condition-3. The position of concentration mass is randomly placed in the load domain for each constraint condition.

The MSE function is used as the model loss function. Using Condition-1 as an example, Figure 7 shows the curve of MSE loss function values changing with iterations under Condition-1 constraints. The red and blue lines represent the loss function values of the training and verification sets, respectively, of Model-1. The black and green dotted lines represent the loss function values of the training and verification sets, respectively, of Model-2. For Condition-1 (Figure 7), the Model-1 loss function value of the verification set is approximately 0.0145, and that of the Model-2 verification set is approximately 0.0055. The Model-2 loss function value is lower than Model-1, and the training effect is also better. No noticeable over-fitting or under-fitting occurs in the training process of Model-1 and Model-2.

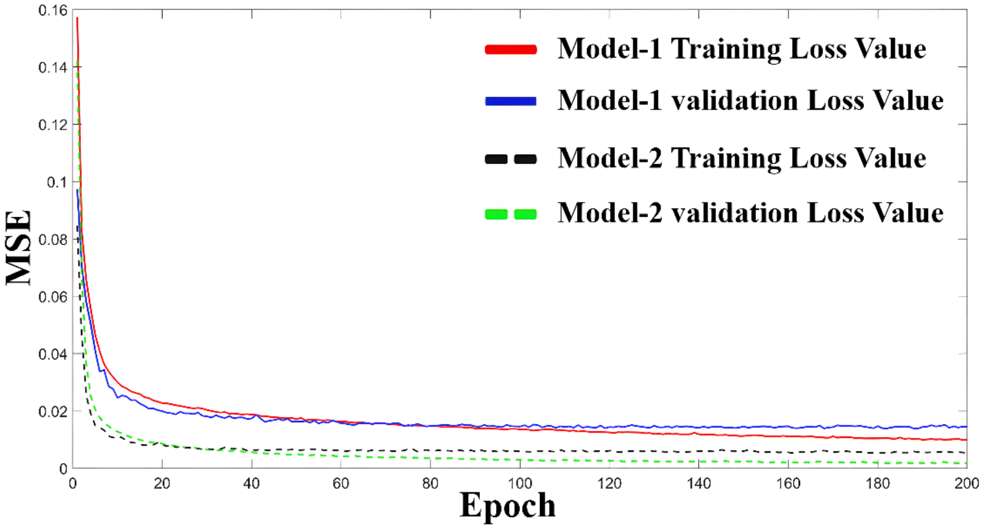


Figure.7 Loss function value (MSE) changes throughout the training process in Condition-1 for Model-1 and Model-2.

Table 2 shows the prediction results of models based on different training methods under Condition-1. In addition to the training method, the other factors, such as sample set size and model structure, are the same in the ETE and STS methods. In Table 2, the first two columns of “Predicted Structure” results are obtained based on ETE and STS training methods, respectively. Moreover, the last column’s results are optimized by the SIMP method. These are reference solutions to verify the model prediction accuracy.

The model prediction results in the first column of Table 2 show that the model trained based on the ETE model training method does not achieve a good prediction effect. A severe grey element problem occurs in the model prediction results. By comparing the STS method prediction results and SIMP optimization results in Table 2, the prediction error is primarily generated in the local region of the structure. The STS model prediction error distribution is similar to the results of other studies[17, 26, 47].

In classical topology optimization methods, the advanced optimization process is primarily used to generate local features of the structure. Therefore, the STS model training method is proposed to consider the topology optimization process information. Using Model-2 to learn the advanced optimization stage characteristics, the STS training method can effectively solve the grey element problem in the ETE method prediction results.

By comparing the model prediction results based on the ETE and STS training methods, the deep learning model trained based on the STS method can obtain more accurate predictions. Meanwhile, the fundamental eigenfrequency predicted by the STS method is closer to the reference solution optimized using the SIMP method. For example, in row 1 of Table 1, the STS method reduces the relative error of the objective function value compared with the reference result from 6.05% (ETE method) to 1.76% (STS method).

Table 2 Real-time topology optimization results for Condition-1 based on SIMP method: the 'end-to-end (ETE) predicts results' and 'step-to-step (STS) predicts results' are the deep learning model prediction results based on the ETE and STS training methods, respectively. 'optimized results based on SIMP' is the structure optimized by SIMP as the reference result. In addition to the training method, the other factors, such as sample set size and model structure, are the same in the ETE and STS methods.

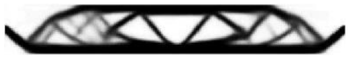
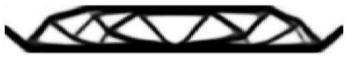
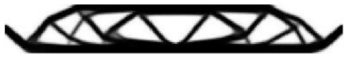
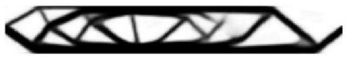
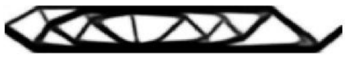
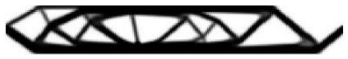
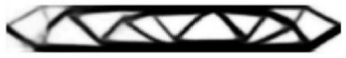
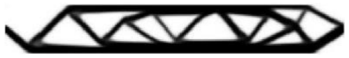
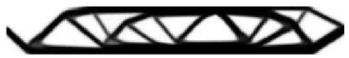
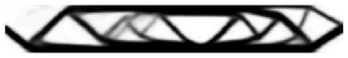
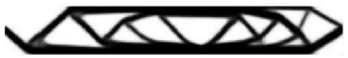
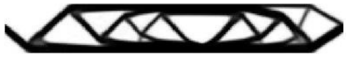
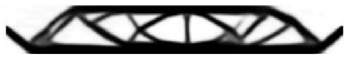
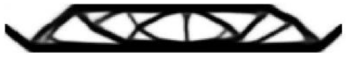
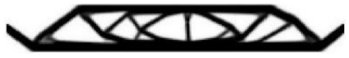
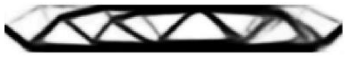
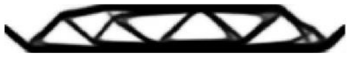
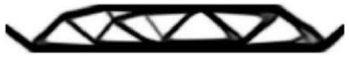
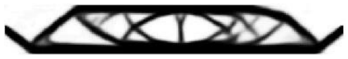
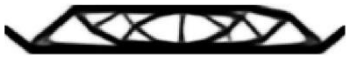
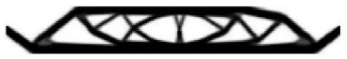
ETE Method Prediction Results		**STS Method Prediction Results**		**Optimized Results based on SIMP**	
Predicted Structures	Fundamental Eigenfrequency	Predicted Structures	Fundamental Eigenfrequency	Optimized Structures	Fundamental Eigenfrequency
	34.66		36.24		36.89
	35.42		36.69		36.50
	31.23		36.72		36.75
	36.63		37.10		36.33
	34.71		34.98		34.25
	35.17		36.09		35.07
	33.15		33.71		35.80

Figure 8 shows the number of artificial density elements in different density areas in the structure based on the ETE and STS methods under example 1, Condition-1 (row 1in Table 1). The blue and orange colors correspond to the distribution of elements in the structure obtained using the ETE and STS methods, respectively. Figure 8 shows that, compared with the results obtained using the ETE method, the number of low-density value elements (ρ∈[0,0.1)) and high-density value elements (ρ∈[0.9,1.0]) in the structure obtained using the STS method is relatively lower. The structure obtained using the ETE method has a severe grey element problem, whereas the structure predicted using the STS method significantly improves this phenomenon.

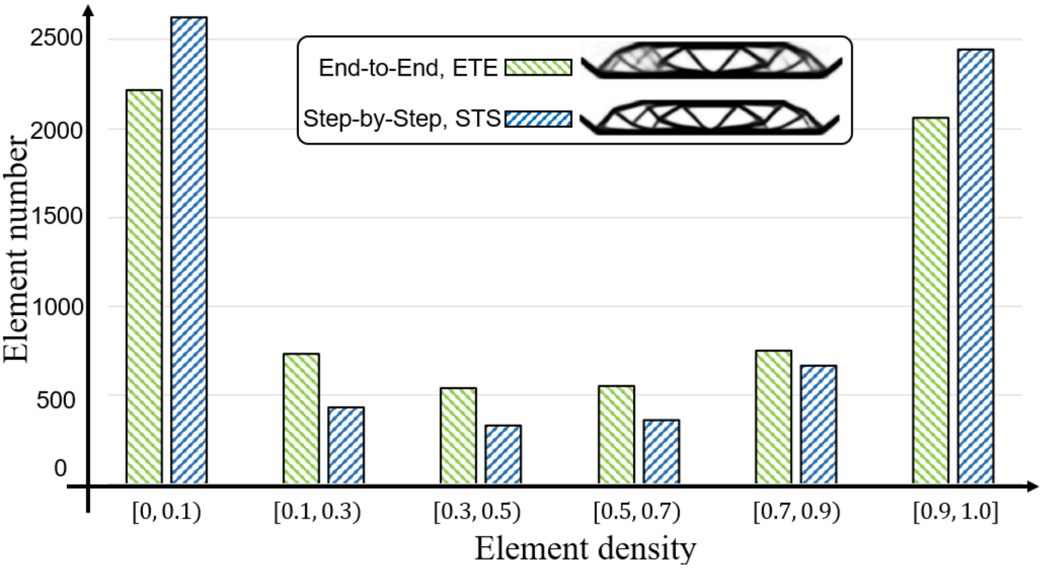


Figure.8 Element number distribution for different element density ranges such as example 1 in Condition-1 (Table 2 row 1). The green and blue colors correspond to the structures obtained using the ETE and STS methods.

The load conditions in Table 3 are the same as those in Figure 8 (Condition-1 row 1 in Table 2). Table 3 shows the TC values with different σ^2 values for different training methods (different σ^2 values can change the central area of the TC function for different element density values). If all element densities are equal to 0.5, TC is 1. And if a converged design is discrete, TC is near 0. The main evaluation area of the TC function can be adjusted by adjusting σ^2 values. It can be seen from Table 3 that the predicted results of the STS method are better than those of the ETE method with different σ^2 values. For example, when $\sigma^2 = 0.1^2$, the TC function concerned more about the element number whose density belongs to the $[0.3,\ 0.7]$. The relative errors obtained using the ETE and STS methods compared with the SIMP method are 74.14% and 8.62%, respectively. The STS method proposed in this paper significantly reduces the TC value errors compared with the ETE method. In other words, the STS method effectively improves the prediction accuracy of the model without increasing the sample set.

Table 3 TC function values of the structure with different σ^2 for Condition-1 row 1.

Training Method	$\sigma^2 = 0.05^2$	$\sigma^2 = 0.1^2$	$\sigma^2 = 0.15^2$
ETE Method	0.045	0.101	0.159
STS Method	0.031	0.063	0.104

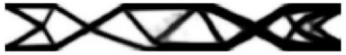
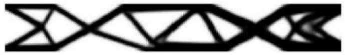
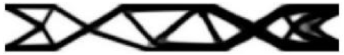
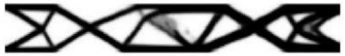
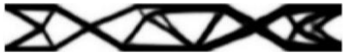
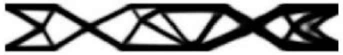
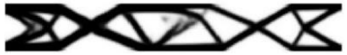
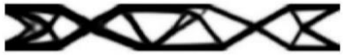
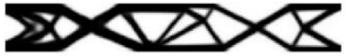
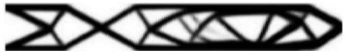
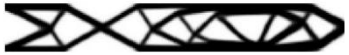
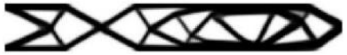
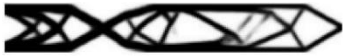
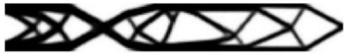
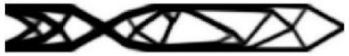
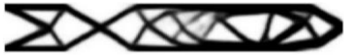
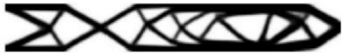
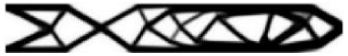
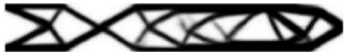
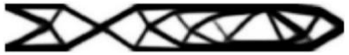
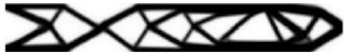
Based on the TC $(\mu = 0.5,\ \sigma^2 = 0.1^2)$ evaluation function, the model prediction results under Condition-2 and Condition-3 are shown in Table 4. The results of rows 1 to 3 belong to Condition-2, and the other results belong to Condition-3. The columns in Table 3 are the same as those in Table 2 except for the TC function column. The TC column shows the evaluation values of the TC function when $\mu = 0.5, \sigma^2 = 0.1^2$.

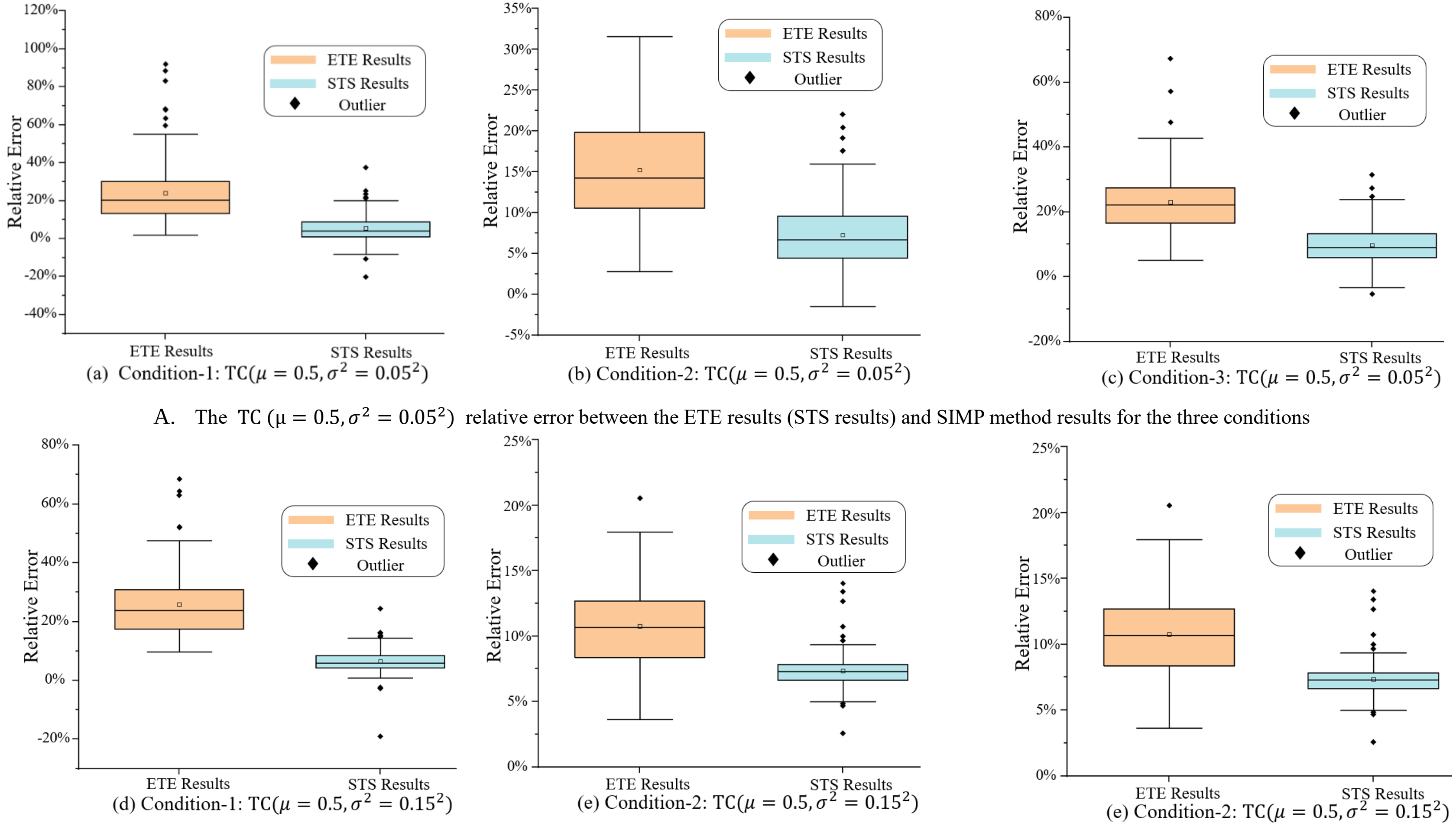
By comparing the results of the ETE and STS methods,. the TC function values of the STS predicted results are closer to the structures obtained using the SIMP optimization method. The proposed STS method effectively solves the grey element problem by fully utilizing the 'depth' for each sample. Thus, the accuracy of the model prediction improved. The iteration history information is effectively utilized by the proposed STS method without increasing the size of the sample set.

The analysis above is based on the TC $(\mathrm{\mu} = 0.5, \sigma^2 = 0.1^2)$ function, and the main evaluation region is $\rho \in [0.3, 0.7]$. Figure 9 shows the TC function values of all samples in verification sample sets from Condition-1, Condition-2, and Condition-3 with different σ^2. The orange and blue lines in Figure 9 correspond to the relative error distribution of the TC evaluation function values of the ETE and STS methods compared with the SIMP method, respectively. First, by analyzing subfigures (a)-(f), it can observe that the relative errors of the TC function values of the STS method are always lower than those of the ETE method with different load conditions and TC function attention areas. It demonstrates the effectiveness of the STS model training method. Second, by analyzing the box area for different training methods, the TC value variances of the STS method are always more minor than those of the ETE method, which indicates good stability of the STS method. Finally, using subfigures (a) and (d) as an example, if the main evaluation region of the TC function is adjusted from $[0.4, 0.6]$ to $[0.2, 0.8]$, the upper quartile (Q3) value of the ETE method decreases from 53.7% to 30.9%. The Q3 value of the STS method changes from 8.6% to 8.47%. The above results show that the number of element densities $\rho \in [0.4, 0.6]$ in the ETE method prediction results is larger than that in the SIMP results. A large number of intermediate density elements are present in the structure. When the TC function's main evaluation region is expanded to $[0.2, 0.8]$, the weight coefficient of the TC function at $[0.4,0.6]$ is correspondingly reduced. It enables the TC function value errors of the ETE method decrease from 53.7% to 30.9%. However, the Q3 value for the STS method results does not change significantly. It indicates that the model training method proposed in this paper affects the elements in the large density range, and the grey element problem can be solved effectively.

The STS training method considers the information during the topology optimization process. Model-2 has the role of the Heaviside function[45, 46] in the classical SIMP method in the STS method. The STS training method effectively solves the grey element problem in the ETE model training method and significantly improves the accuracy and clarity of the model prediction results. Inspired by the boosting ensemble learning method[48], the STS training method effectively combines Model-1 and Model-2, and an accurate real-time topology optimization calculation is realized without increasing the sample set size.

Table 4 Real-time optimization results for Condition-2 and Condition-3 based on the SIMP optimization method. The 'ETE predicts results', 'STS predicts results' and 'optimized results based on SIMP' are the same as those in Table 2. Here, the TC $(\mu = 0.5, \sigma^2 = 0.1^2)$ function is used to evaluate the clarity of the structures.

	ETE Method Prediction Results		**STS method Prediction Results**		**Optimized Results based on SIMP**	
	Predicted Structures	TC $\mu = 0.5, \sigma^2 = 0.1^2$	Predicted Structures	TC $\mu = 0.5, \sigma^2 = 0.1^2$	Optimized Structures	TC $\mu = 0.5, \sigma^2 = 0.1^2$
Condition-2		6.87×10^{-2}		6.13×10^{-2}		5.65×10^{-2}
		7.63×10^{-2}		6.64×10^{-2}		6.27×10^{-2}
		7.22×10^{-2}		6.72×10^{-2}		6.26×10^{-2}
Condition-3		8.08×10^{-2}		6.66×10^{-2}		6.57×10^{-2}
		7.68×10^{-2}		6.97×10^{-2}		6.32×10^{-2}
		8.89×10^{-2}		7.18×10^{-2}		6.64×10^{-2}
		8.27×10^{-2}		6.59×10^{-2}		6.03×10^{-2}



(a) Condition-1: TC($\mu = 0.5, \sigma^2 = 0.05^2$)

(b) Condition-2: TC($\mu = 0.5, \sigma^2 = 0.05^2$)

(c) Condition-3: TC($\mu = 0.5, \sigma^2 = 0.05^2$)

A. The TC ($\mu = 0.5, \sigma^2 = 0.05^2$) relative error between the ETE results (STS results) and SIMP method results for the three conditions

(d) Condition-1: TC($\mu = 0.5, \sigma^2 = 0.15^2$)

(e) Condition-2: TC($\mu = 0.5, \sigma^2 = 0.15^2$)

(e) Condition-2: TC($\mu = 0.5, \sigma^2 = 0.15^2$)

B. The TC ($\mu = 0.5, \sigma^2 = 0.15^2$) relative error between the ETE results (STS results) and SIMP method results for the three conditions

Figure.9 Relative error of TC values between ETE (STS) and SIMP method for three conditions. In A $\sigma^2 = 0.05^2$. (a), (b), and (c) in A correspond to Condition-1, Condition-2, and Condition-3, respectively. In B, $\sigma^2 = 0.15^2$. The corresponding relationship of sub-figures is the same as those in A.

Although two deep learning models are used in the STS training method, no coupling relationship occurs between the training of the two sub-models. Both sub-models can be trained in parallel. Therefore, the time cost of the model training process does not increase significantly. For the prediction process of the model, because the sub-models are trained based on the local ETE training method, the serial method of model prediction does not significantly increase the model prediction time. Table 5 shows the average model prediction calculation time for each example based on the different training methods under the three loading conditions. Although the deep learning model prediction time trained based on the STS method is increasing, it can still achieve real-time topology optimization calculation. Compared with the traditional ETE model training method, the accuracy of the model prediction results significantly improves.

Table 5 Prediction time cost for ETE method and STS method in three conditions. The computing environment is the same as described in Section 3.1

Prediction Time -Cost	Condition-1 (s)	Condition-2 (s)	Condition-3 (s)
ETE Method	0.00888	0.01971	0.01021
STS Method	0.01362	0.02797	0.01776

5. Conclusion

In this work, a STS deep learning model training method is proposed. By using the iterative history information of topology optimization, the STS method improves the model utilization efficiency for each example, which can improve the model prediction accuracy without increasing the sample set size. The deep learning model' unsatisfactory prediction accuracy is usually presented as the gray element phenomena or structure fracture phenomena. Therefore, in addition to taking objective function and model loss function as the evaluation, a new function is proposed to evaluate the model prediction accuracy from the structure non-discreteness perspective. The effectiveness of STS training method is verified using fundamental frequency optimization problem under three different constraint conditions. This work provides an implementation model training method for the combination of deep learning and topology optimization.

However, this training method still cannot avoide the model's smaple set dependence phenomenon. In future research, the authors believe that the deep learning model can alleviate this phenomenon by introducing the prior knowledge of topology optimization, such as the physical field information or building "structure gene pool".

Appendix: The total trainable parameters in the model

This table shows the feature map size and trainable parameters in each layer. The change of feature map size reflects the operation time of up-sampling and down-sampling in the two models. The trainable parameters refer to the weight and bias parameters of neurons in each layer (such as the convolution layer or down/up sampling layer). These parameters will change during the model training process. Therefore, the trainable parameter size will affect the model training efficiency. The layers without trainable parameters are omitted (such as the dropout layer or add layer).

The model with reducing part		The model without reducing part	
Feature map size	Parameters	Feature map size	Parameters
(221, 31, 1)	0	(221, 31, 1)	0
(221, 31, 8)	80	(221, 31, 8)	80
(220, 30, 32)	1,056	(220, 30, 32)	1,056
(220, 30, 32)	9,248	(220, 30, 32)	9,248
(110, 15, 64)	8,256	(110, 15, 64)	8,256
(110, 15, 64)	36,928	(110, 15, 64)	36,928
(55, 14, 128)	32,896	(55, 14, 128)	32,896
(55, 14, 128)	147,584	(55, 14, 128)	147,584
(27, 7, 256)	196,864	(27, 7, 256)	196,864
(27, 7, 256)	590,080	(27, 7, 256)	590,080
(13, 6, 512)	786,944	(13, 6, 512)	786,944
(27, 7, 256)	786,688	(27, 7, 256)	786,688
(27, 7, 256)	590,080	(27, 7, 256)	590,080
(55, 14, 128)	196,736	(55, 14, 128)	196,736
(55, 14, 128)	147,584	(55, 14, 128)	147,584
(110, 15, 64)	32,832	(110, 15, 64)	32,832
(110, 15, 64)	36,928	(110, 15, 64)	36,928
(220, 30, 32)	8,224	(220, 30, 32)	8,224
(220, 30, 32)	9,248	(220, 30, 32)	9,248
(221, 31, 16)	2,064	(221, 31, 16)	2,064
(221, 31, 16)	2,320	(221, 31, 16)	2,320
(221, 31, 8)	1,160	(221, 31, 8)	1,160
(221, 31, 4)	292	(221, 31, 4)	292
(221, 31, 1)	337	(6851, 1)	187,751,655
Total Parameters	3,624,129	Total Parameters	191,375,747

Declarations

Funding: This research was financially supported by the National Natural Science Foundation of China (No. U1906233,11732004), the Key R&D Program of Shandong Province (2019JZZY010801), the Fundamental Research Funds for the Central Universities (DUT20ZD213，DUT20LAB308).

Replication of results: The data set and deep learning model code are available from the "https://github.com/893801366/Real-time-topology-optimization-based-on-convolutional-neural-network-by-using-retrain-skill.git" (git-hub).

Reference

1. Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71:197–224. https://doi.org/10.1016/0045-7825(88)90086-2
2. Zhu JH, Zhang WH, Xia L (2016) Topology Optimization in Aircraft and Aerospace Structures Design. Arch Comput Methods Eng 23:595–622. https://doi.org/10.1007/s11831-015-9151-2
3. Wu P, Ma Q, Luo Y, Tao C (2016) Topology Optimization Design of Automotive Engine Bracket. Energy Power Eng 08:230–235. https://doi.org/10.4236/epe.2016.84021
4. Cheng KT (1981) On non-smoothness in optimal design of solid, elastic plates. Int J Solids Struct 17:795–810. https://doi.org/10.1016/0020-7683(81)90089-5
5. Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1:193–202. https://doi.org/10.1007/BF01650949
6. Guo X, Zhang W, Zhong W (2014) Doing topology optimization explicitly and geometrically-a new moving morphable components based framework. J Appl Mech Trans ASME 81:1–12. https://doi.org/10.1115/1.4027609
7. Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. Comput Struct 49:885–896. https://doi.org/https://doi.org/10.1016/0045-7949(93)90035-C
8. Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69:635–654. https://doi.org/10.1007/s004190050248
9. Huang X, Xie YM (2008) Optimal design of periodic structures using evolutionary topology optimization. Struct Multidiscip Optim 36:597–606. https://doi.org/10.1007/s00158-007-0196-1
10. Du J, Olhoff N (2007) Topological design of freely vibrating continuum structures for maximum values of simple and multiple eigenfrequencies and frequency gaps. Struct Multidiscip Optim 34:91–110. https://doi.org/10.1007/s00158-007-0101-y
11. Fan Z, Yan J, Wallin M, et al (2020) Multiscale eigenfrequency optimization of multimaterial lattice structures based on the asymptotic homogenization method. Struct Multidiscip Optim 61:983–998. https://doi.org/10.1007/s00158-019-02399-0
12. Gersborg-Hansen A, Bendsøe MP, Sigmund O (2006) Topology optimization of heat conduction problems using the finite volume method. Struct Multidiscip Optim 31:251–259. https://doi.org/10.1007/s00158-005-0584-3
13. Deng J, Yan J, Cheng G (2013) Multi-objective concurrent topology optimization of

thermoelastic structures composed of homogeneous porous material. Struct Multidiscip Optim 47:583–597. https://doi.org/10.1007/s00158-012-0849-6

14. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 770–778

15. Zou Z, Shi Z, Guo Y, Ye J (2019) Object Detection in 20 Years: A Survey. arXiv:1905.05055

16. Cho K, Van Merriënboer B, Gulcehre C, et al (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. EMNLP 2014 - 2014 Conf Empir Methods Nat Lang Process Proc Conf 1724–1734. https://doi.org/10.3115/v1/d14-1179

17. Yu Y, Hur T, Jung J, Jang IG (2019) Deep learning for determining a near-optimal topological design without any iteration. Struct Multidiscip Optim 59:787–799. https://doi.org/10.1007/s00158-018-2101-5

18. Nakamura K, Suzuki Y (2020) Deep learning-based topological optimization for representing a user-specified design area. arXiv:2004.05461

19. Tan Z, Chen D, Chu Q, et al (2021) Efficient Semantic Image Synthesis via Class-Adaptive Normalization. IEEE Trans Pattern Anal Mach Intell 1. https://doi.org/10.1109/TPAMI.2021.3076487

20. Wang D, Xiang C, Pan Y, et al (2022) A deep convolutional neural network for topology optimization with perceptible generalization ability. Eng Optim 54:973–988. https://doi.org/10.1080/0305215X.2021.1902998

21. Chandrasekhar A, Suresh K (2021) TOuNN: Topology Optimization using Neural Networks. Struct Multidiscip Optim 63:1135–1149. https://doi.org/10.1007/s00158-020-02748-4

22. Chi H, Zhang Y, Tang TLE, et al (2021) Universal machine learning for topology optimization. Comput Methods Appl Mech Eng 375:112739. https://doi.org/10.1016/j.cma.2019.112739

23. Yilin G, Fuh Ying Hsi J, Wen Feng L (2021) Multiscale topology optimisation with nonparametric microstructures using three-dimensional convolutional neural network (3D-CNN) models. Virtual Phys Prototyp 0:1–12. https://doi.org/10.1080/17452759.2021.1913783

24. Deng H, To AC (2021) A parametric level set method for topology optimization based on deep neural network. J Mech Des Trans ASME 143:. https://doi.org/10.1115/1.4050105

25. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Networks 2:359–366. https://doi.org/10.1016/0893-6080(89)90020-8

26. Ates GC, Gorguluarslan RM (2021) Two-stage convolutional encoder-decoder network to improve the performance and reliability of deep learning models for topology optimization. Struct Multidiscip Optim 63:1927–1950. https://doi.org/10.1007/s00158-020-02788-w

27. Lin Q, Hong J, Liu Z, et al (2018) Investigation into the topology optimization for conductive heat transfer based on deep learning approach. Int. Commun. Heat Mass Transf. 97:103–109. https://doi.org/10.1016/j.icheatmasstransfer.2018.07.001

28. Yan J, Zhang Q, Xu Q, et al (2022) Deep learning driven real time topology optimisation based on initial stress learning. Adv Eng Informatics 51:. https://doi.org/10.1016/j.aei.2021.101472

29. Du J, Olhoff N (2007) Minimization of sound radiation from vibrating bi-material structures using topology optimization. Struct Multidiscip Optim 33:305–321. https://doi.org/10.1007/s00158-006-0088-9

30. K. S (1987) The method of moving asymptotes - a new method for structural optimization. Int J Numer Methods Eng 24:359–373. https://doi.org/10.1002/nme.1620240207

31. Chen D, Liu F, Li Z (2020) Deep Learning Based Single Sample Per Person Face Recognition: A Survey. arXiv:2006.11395

32. Rumelhart DE, Hintont GE (1986) Learning Representations by Back-Propagating Errors. Nature 323:533–536. https://doi.org/10.7551/mitpress/1888.003.0013

33. Uang C-M, Yin S, Andres P, et al (1994) Shift-invariant interpattern association neural network. Appl Opt 33:2147. https://doi.org/10.1364/ao.33.002147

34. Pereira JT, Fancello EA, Barcellos CS (2004) Topology optimization of continuum structures with material failure constraints. Struct Multidiscip Optim 26:50–66. https://doi.org/10.1007/s00158-003-0301-z

35. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2323. https://doi.org/10.1109/5.726791

36. Han J, Moraga C (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. Lecture Notes in Computer Science 930:195-201. https://doi.org/10.1007/3-540-59497-3_175

37. A D (2000) Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature 442:947–951. https://doi.org/10.1038/35016072

38. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434

39. Huang J, Qu L, Jia R, Zhao B (2019) O2U-Net: A simple noisy label detection approach for deep neural networks. Proc IEEE Int Conf Comput Vis 2019-Octob:3325–3333. https://doi.org/10.1109/ICCV.2019.00342

40. Hinton GE, Srivastava N, Krizhevsky A, et al (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580

41. Szegedy C, Liu W, Jia Y, et al (2015) Going deeper with convolutions. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 07-12-June:1–9. https://doi.org/10.1109/CVPR.2015.7298594

42. Kingma DP, Ba J (2014) Adam: A Method for Stochastic Optimization. Comput Sci arXiv:1412.6980

43. Rade J, Balu A, Herron E, et al (2021) Algorithmically-consistent deep learning frameworks for structural topology optimization. Eng Appl Artif Intell 106:. https://doi.org/10.1016/j.engappai.2021.104483

44. Gowda SN, Yuan C (2019) ColorNet: Investigating the Importance of Color Spaces for Image Classification. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 11364 LNCS:581–596. https://doi.org/10.1007/978-3-030-20870-7_36

45. Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multidiscip Optim 33:401–424. https://doi.org/10.1007/s00158-006-0087-x

46. Xu S, Cai Y, Cheng G (2010) Volume preserving nonlinear density filter based on heaviside functions. Struct Multidiscip Optim 41:495–505. https://doi.org/10.1007/s00158-009-0452-7

47. Wang D, Xiang C, Pan Y, et al (2021) A deep convolutional neural network for topology optimization with perceptible generalization ability. Eng Optim 1–16. https://doi.org/10.1080/0305215X.2021.1902998

48. Friedman JH (2001) Greedy function approximation: A gradient boosting machine. Ann Stat 29:1189–1232. https://doi.org/10.1214/aos/1013203451