Topology Optimization Using an Improved DoubleU-Net Model on Four Structural Datasets

Authors: Li Shun, white-period wind, Lin Nankai, Zeng Peijian, Yang Aimin, Lin Jianghao, Yang Aimin, Lin Jianghao

Date: 2023-06-15T00:00:00+00:00

Abstract

Topology optimization is widely employed in the engineering design phase to maximize product performance through mathematical modeling and optimization of material distribution within the design space. However, deep learning approaches for solving topology optimization problems suffer from data insufficiency and weak adaptability of trained models to varying boundary conditions. Therefore, a data sample generation method based on the Topy library was adopted to generate 400,000 two-dimensional samples across four boundary conditions: random structures, cantilever beams, continuous beams, and simply supported beams, with each sample containing two classes of resolution data, thereby establishing this dataset. An improved DoubleU-Net network is proposed for real-time, high-accuracy prediction of topology optimization. In the generated dataset, the average IoU accuracies for the four structural models —random beams, cantilever beams, continuous beams, and simply supported beams—are 93.26%, 96.71%, 96.35%, and 97.38%, respectively. Experimental results demonstrate that DoubleU-Net exhibits superior adaptability to data of different resolutions. Models trained using the random structure dataset demonstrate strong generalization capability and hold tremendous potential for real-time structural optimization in large-scale projects.

Full Text

Topology Optimization Based on Improved DoubleU-Net Using Four Structural Datasets

Li Shun[1], Bai Qifeng[1], Lin Nankai[1], Zeng Peijian[1], Yang Aimin[12]*, Lin Jianghao*[3]

[1]School of Computer Science, Guangdong University of Technology, Guangzhou 510000, China

[2]School of Computer Science and Intelligence Education, Lingnan Normal University, Zhanjiang 524000, China
[3]School of Automation, Guangdong University of Technology, Guangzhou 510000, China

This work is supported by the Guangdong Provincial Key R&D Program Project "Domestic Structural Dynamics CAE Software" (Project No.: 2021B0101190004).

Keywords: Topology optimization, Deep learning, Data generation, DoubleU-Net

Abstract

Topology optimization is widely employed in the engineering design phase to maximize product performance by mathematically modeling and optimizing material distribution within the design space. However, deep learning approaches for solving topology optimization problems suffer from insufficient training data and weak adaptability to varying boundary conditions. To address these challenges, we propose a Topy library-based data sample generation method to produce 400,000 2D samples across four boundary conditions: random structures, cantilever beams, continuous beams, and simply supported beams. Each category includes two resolution types. We introduce an improved DoubleU-Net network for real-time, high-accuracy topology optimization prediction. On our generated dataset, the improved DoubleU-Net achieves average IoU accuracies of 93.26%, 96.71%, 96.35%, and 97.38% for random beams, cantilever beams, continuous beams, and simply supported beams, respectively. Experimental results demonstrate that DoubleU-Net can effectively adapt to different resolution data. Notably, models trained on the random structure dataset exhibit strong generalization capability, showing great potential for real-time structural optimization in large-scale engineering projects.

1. Introduction

Topology optimization is a structural optimization method that aims to find the optimal structural shape and material distribution within a given design space by adjusting material layout and distribution [1, 2]. This process enables weight reduction, strength improvement, and maximization of material utilization efficiency while meeting performance requirements [3]. Topology optimization has found widespread applications in engineering design [4], materials science [5], aerospace engineering [6], and other fields.

With the advancement of artificial intelligence and finite element analysis methods, topology optimization has become a mainstream optimization approach.

Existing primary topology optimization methods include SIMP (Solid Isotropic Material with Penalization) [7, 8], homogenization (HDM) [9], level set (LST) [10, 11], evolutionary algorithms [12], MMTO (Multi-Material Topology Optimization) [13], MMC (Moving Morphable Components) [14], multi-component methods [15], and the Jaya algorithm [16]. These methods can produce structures that meet practical engineering requirements based on constraints and optimization objectives. However, as engineering structures become increasingly complex, the number of elements in these structures surges, causing the computational cost of topology optimization algorithms to multiply and severely hindering the development of refined, real-time topology optimization design and applications. To address this, Liao et al. [17] proposed a triple acceleration method for topology optimization. Li et al. [18] introduced a scalable approach to significantly accelerate convergence in topology optimization simulations. Jang et al. [19] proposed optimizing the design space based on fixed-grid topology, which improves computational accuracy while effectively reducing computational costs. Nevertheless, these methods share common limitations: they typically perform iterative calculations based on an initial design to gradually improve the structural shape, with each improvement relying on local information from the current design state, making them prone to falling into local optima. Moreover, each iteration requires finite element analysis to evaluate design performance, and this iterative process demands substantial computational resources and time, particularly for complex structures and large-scale problems where computational costs become prohibitively high.

Deep learning technologies have provided new approaches for topology optimization research. Sosnovik et al. [20] first introduced deep learning models into topology optimization, using CNNs to directly predict topology-optimized structures, eliminating intermediate iterative calculation steps and reducing total time consumption. Zheng et al. [21] constructed a U-Net [22] network to train datasets, enabling optimal configuration acquisition without optimization iterations or finite element analysis. Wang et al. [23] proposed a deep convolutional neural network with perceptible generalization ability to reduce the computational cost of the SIMP method. Zhang et al. [24] used hierarchical deep-learning neural networks to compute finite element analysis in topology optimization, significantly reducing computational costs through node coordinate optimization. Hao et al. [25] proposed a parametric level set topology optimization algorithm based on deep neural networks to achieve topological configuration diversity. Ye et al. [26] introduced conditional generative adversarial neural networks to solve cross-resolution topology optimization problems and used neural networks to accelerate the iterative process. Yu et al. [27] employed CNNs to accelerate the SIMP topology optimization method and used generative adversarial networks (GANs) to establish mappings between low-resolution and high-resolution topology optimization results. Wang et al. [28] applied deep learning methods to optimize the design of modern composite material structures for rapid production of lightweight materials with efficient structural configurations. Yan et al. [29] proposed predicting topology optimization from initial stress (LIS), reducing dataset requirements and substantially improving topology optimization speed. Kapania et al. [30] used U-Net [22] and U-Net++ [31] models to predict optimal topological structures, generating input and output data using commercial finite element analysis software.

Concurrently, researchers have improved deep learning models to enhance optimization performance. Ates et al. [32] proposed a two-stage network model to improve the prediction performance of deep neural networks for topology optimization. Nie [33] introduced a novel data-driven TopologyGAN model that utilizes various physical fields computed on the original unoptimized material domain as input, further improving overall accuracy. Wang et al. [34] proposed an improved U-Net model (Cba-U-Net) for topology optimization configuration prediction to obtain topological configurations in real time. Jeong et al. [35] presented a physics-informed neural network-based topology optimization framework that employs energy-based PINNs to replace traditional finite element analysis (FEA) in structural topology optimization.

While existing deep learning methods have achieved certain improvements in optimization efficiency, deep neural networks require large amounts of high-quality training data to learn deep semantic information from massive datasets and satisfy large-scale parameter learning while avoiding model overfitting. However, the topology optimization field currently suffers from scarce data and single boundary conditions, with no high-precision structural datasets available for model training. Furthermore, deep learning approaches for topology optimization exhibit weak boundary condition adaptability—once boundary conditions change, model prediction performance degrades significantly. Retraining models for new boundary conditions demands high data requirements and incurs substantial time costs, posing a critical challenge for research in this domain. To address this, we generate datasets with four different boundary conditions and two different resolutions based on the SIMP method for deep learning training, solving the dataset insufficiency problem. We employ an improved DoubleU-Net model for topology optimization, leveraging its strong generalization capability to address weak boundary adaptability. Our main contributions are summarized as follows:

1. We generated 400,000 2D samples using the Topy library based on four boundary conditions (random structures, cantilever beams, continuous beams, and simply supported beams) at different resolutions. Each structure comprises 100,000 samples, with 50,000 low-resolution (32$\times 64)and50,000high - resolution(64\times$128) samples each, totaling 70GB of data. This constitutes the most structurally diverse and largest-scale topology optimization research dataset currently available, which we have open-sourced on GitHub: https://github.com/BigDLishun/Topology-Optimization-Dataset.

2. This paper proposes an improved DoubleU-Net convolutional neural network model that achieves accurate and real-time topology optimization structure prediction, delivering the best performance across different struc-

tural datasets without being affected by resolution changes.

3. We explore the impact of different structures and dataset accuracies on experimental results and demonstrate the generalization capability of models trained on four different datasets across four boundary condition types. Experiments prove that training with the random structure dataset yields excellent generalization ability, showing tremendous potential for real-time structural optimization in large-scale engineering applications.

2. Methodology

2.1 SIMP-Based Topology Optimization

The most commonly used method in topology optimization is the Solid Isotropic Material with Penalization (SIMP) [7, 8], a continuous density topology optimization approach. Research on continuous density topology optimization is relatively mature, and SIMP has been applied in many commercial optimization software packages.

The SIMP method defines a finite element mesh within the design domain to be optimized, where each element has an associated density variable. SIMP optimization is based on continuous density variable r, using a penalty function to drive all intermediate density elements toward 0 or 1. To ensure numerical stability in finite element analysis, m_{inr} represents the minimum allowable relative density value.

The most common objective function is to maximize overall structural stiffness or minimize global compliance under specified mass reduction constraints. The optimization algorithm adjusts elemental densities through an iterative process to minimize global structural compliance.

2.2 SIMP-Based Dataset Generation

For dataset generation, we used CPU-based computation, while model training was performed on GPU using the PyTorch framework. The experimental configuration includes: CPU: Intel(R) Xeon(R) Gold-6230R @ 2.10GHz, GPU: NVIDIA A100 PCIE-40GB, Framework: torch-1.9.0+cu111, Python version: 3.8.8.

Currently, three common tools are used for dataset generation in topology optimization, as shown in : (1) Real simulation experiments using CAE software (e.g., Abaqus), which incur high time and software operation costs; (2) Simulation experiments using MATLAB, which reduce time and operation costs compared to real simulations but are inconvenient for batch dataset generation; (3) The Python Topy package, which integrates CAE simulation analysis with the SIMP algorithm for topology optimization problems including minimum compliance, heat conduction, and mechanism design, saving time on prior knowledge acquisition and offering simpler operation compared to the previous two methods, making it suitable for large-scale dataset generation.

Previous literature has used limited data with single boundary conditions and has not considered the impact of different resolution data on models, preventing them from achieving optimal results. Currently available public datasets are also limited. For example, Sosnovik et al. [20] used 10,000 random structure datasets at 40$×40resolutiontotrainaCNNmodel, achievinglimitedperformance.Nieetal.[33]used49,078cantileverbea
resolution to train a TopologyGAN model, achieving good results on cantilever beams but with relatively single boundary conditions. Some scholars have employed data augmentation methods to expand datasets during training; however, augmented data is not real data and its effects are not always positive. Inappropriate data augmentation may produce negative impacts.

Therefore, we used ToPy [37], a SIMP-based topology optimization framework, to generate our dataset. We generated 400,000 2D samples across four boundary conditions: random structures, cantilever beams, continuous beams, and simply supported beams, with 100,000 samples for each structure type. Low-resolution and high-resolution data each account for half (50,000 samples at 32$×64and50, 000at64×$128). Each dataset is obtained based on the SIMP method under random constraint conditions. In summary, our generated dataset includes not only four boundary conditions but also two different resolution types for each condition, with 50,000 samples per data type, enabling analysis of topology optimization effects under different boundary conditions and resolutions while satisfying model data requirements.

The fixed-point boundary condition details for the three industrial beam structure datasets are shown in [Figure 1: see original paper]. Random structures have randomly selected nodes fixed without specific boundary conditions.

Dataset parameter settings: - Resolution: 32$×64and64×$128 - Boundary conditions: See [Figure 1: see original paper] - Volume fraction: Normal distribution (0.8, 0.1) - SIMP penalty: 1 - SIMP filter radius: 2 - Number of forces: Uniform distribution (1, 10) - Load position: Any position within boundary conditions - Load direction: X+, X−, Y+, Y− - Iterations: 40

Dataset details based on SIMP generation are shown in . Two resolutions are available: (32$×64)and(64×$128). Volume fraction is randomly selected from a normal distribution (0.8, 0.1). The initial SIMP penalty coefficient is set to 1. The filter radius is 2. The number of forces is randomly selected from (1-10) using a random function. Load positions are randomly chosen from any node in the initial domain, with edge positions having 500 times the selection probability of interior positions. Force directions include four types: X+, X−, Y+, Y−. The final topology is obtained after 40 iterations of the SIMP method.

[Figure 2: see original paper] shows visualizations of the generated dataset, including four structure types at different resolutions. High-resolution structures exhibit more detailed boundaries and higher precision than low-resolution structures.

3. Model Architecture and Training

3.1 Data Generation and Model Training Process

The data generation and model training process using DoubleU-Net for topology optimization is illustrated in [Figure 3: see original paper].

(1) **Data Generation Stage**: Data is generated using Python and the Topy library based on the SIMP method. First, the 2D dimensions representing the design domain are defined, along with the domain' s boundary conditions. ToPy then creates a meshed domain and determines boundary condition and load positions on the mesh. The SIMP method performs topology optimization on the structure based on defined loads and boundary conditions through an iterative optimization process. The first iteration result is used as input data, while the final iteration result serves as label data.

(2) **Model Training Stage**: The first iteration result is used as input. During preprocessing, images are converted to tensor matrix form containing image information. Input data is then normalized to facilitate subsequent data processing and accelerate convergence during program execution. A batch of tensor matrices is fed into the model, which learns data features through multiple iterations and updates weights via backpropagation using the loss function.

3.2 Improved DoubleU-Net Network Architecture

Convolutional neural networks, particularly encoder-decoder-based architectures, have achieved remarkable results in topology optimization. Among various encoder-decoder networks, the U-Net [22] has gained widespread attention for its outstanding performance, providing an efficient and accurate new method for solving topology optimization problems, though its performance on high-precision structures is average.

Building upon the original DoubleU-Net [38] model for medical segmentation, we propose an improved DoubleU-Net model specifically adapted for topology optimization tasks to achieve precise topology configuration prediction and acceleration, enabling real-time topology optimization with high accuracy. The overall architecture of the improved DoubleU-Net network is shown in [Figure 4: see original paper].

(1) Pre-training Module: Instead of using the conventional pre-trained VGG16 encoder, we employ a U-Net architecture encoder block. The U-Net encoder progressively extracts image features through multiple convolutional layers, while the decoder restores image details by fusing information from the encoder and skip connections. Skip connections serve as bridges connecting corresponding layers between the encoder and decoder. Pre-training uses topology optimization training data (excluding test data) to obtain more useful features from topology optimization data. Using pre-trained models for feature extraction avoids redundant model training and improves training efficiency. This approach enhances DoubleU-Net' s prediction capability without significantly increasing computational cost or parameter count.

Input data passes through the pre-trained U-Net1 encoder and decoder to obtain a preliminary prediction result (output1). The original input is then multiplied with Sigmoid(output1). The Sigmoid function maps outputs to (0,1), better aligning with pixel value characteristics. The multiplication result serves as input to U-Net2' s encoder. The motivation for using two U-Nets stems from the observation that feature maps from U-Net1 can be further improved by re-accessing the original input image and corresponding mask.

(2) Encoder Module: Encoder2 contains four encoder blocks, each performing two Conv2d convolutions for upsampling operations. The input-output data format is as follows:

$$\text{Output} = \text{Input}, (\text{Conv2d} : k[0] = 1, s[2] = 2, p[0] = d[0] = 2, p[1] = d[1] = s[1] = (k[1]{-}1) = 1)$$

where N is the batch size of input data, C_{in} and C_{out} are input and output channel sizes, p is zero-padding added to both sides of input, d is the spacing between kernel elements, k is the convolution kernel size, s is the convolution stride, and H, W and H_{out}, W_{out} are the height and width of input and output planes in pixels.

Two BatchNorm2d normalization operations are then performed. Normalization helps reduce data dispersion, making it easier for the model to learn potential relationships between features, preventing network performance instability due to excessively large data before ReLU activation, and thereby improving model generalization performance.

Finally, the ReLU activation function introduces non-linearity into the model: $\text{ReLU}(x) = \max(0, x)$, where x is the normalized input data.

(3) Atrous Spatial Pyramid Pooling (ASPP): After the Encoder2 blocks, information is fed into ASPP [39]. ASPP obtains multi-scale contextual information by applying atrous convolution (dilated convolution) at different sampling rates. Atrous convolution introduces a dilation parameter in standard convolution, making the convolution kernel convolve with pixels at certain intervals, thereby expanding the receptive field and obtaining broader contextual information. ASPP performs pooling operations on atrous convolution results from multiple sampling rates to obtain multi-scale feature representations, which are then concatenated or weighted fused to produce more comprehensive and richer feature representations.

[Figure 5: see original paper] compares standard convolution and atrous convolution, showing how blue inputs produce green outputs with moving shadow representing the convolution kernel.

(4) Decoder Module: In Decoder2, feature information from skip connections of the two preceding encoders and output from ASPP processing are utilized. Other operations are similar to the encoder, except using Conv2d for downsampling. Two Conv2d operations are performed again, each followed by batch normalization and a ReLU activation function. Through this series of convolution, pooling, and feature fusion operations, the input data yields high-precision topology optimization prediction results with the same resolution as the input image.

3.3 Loss Functions

In the DoubleU-Net model and U-Net, we compared the effects of three different loss functions and evaluated their performance: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Structural Similarity Index Measure (SSIM) for assessing differences between model predictions and actual labels. When calculating loss between image input and labels, images are represented as pixel value matrices.

MSE is the average of squared differences between predicted and actual values. For two images of the same size, MSE is calculated as:

$$\mathrm{MSE}(I_1, I_2) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [I_1(i,j) - I_2(i,j)]^2$$

MAE is the average of absolute differences between predicted and actual values. For two images of the same size, MAE is calculated as:

$$\mathrm{MAE}(I_1, I_2) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} |I_1(i,j) - I_2(i,j)|$$

where $I_1(i,j)$ is the pixel value at row i, column j in the input image; $I_2(i,j)$ is the pixel value at row i, column j in the label image; and M and N are the number of rows and columns in the images.

SSIM measures structural similarity between two images. For two images of the same size, SSIM is calculated as:

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

In practical engineering calculations, to reduce computational complexity, SSIM is generally simplified as $\mathrm{SSIM}(x, y) = [l(x,y)]^\alpha \cdot [c(x,y)]^\beta \cdot [s(x,y)]^\gamma$ with $\alpha =$

$\beta = \gamma = 1$ and $C_3 = C_2/2$. Here, x and y represent input and label images, $l(x, y)$ is luminance comparison, $c(x, y)$ is contrast comparison, $s(x, y)$ is structure comparison, μ_x and μ_y are means of x and y, σ_x and σ_y are standard deviations of x and y, σ_{xy} is covariance of x and y, and C_1, C_2, C_3 are constants to avoid division-by-zero errors.

3.4 Evaluation Metrics

IoU (Intersection over Union) is a commonly used performance metric for evaluating segmentation tasks. In computer vision, IoU typically measures the overlap between predicted bounding boxes and ground truth. Since this task involves predicting real topology-optimized images, IoU is highly appropriate as an evaluation criterion. IoU is calculated as:

$$\text{IoU} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} [I_1(i,j) \cdot I_2(i,j)]}{\sum_{i=1}^{M} \sum_{j=1}^{N} [I_1(i,j) + I_2(i,j) - I_1(i,j) \cdot I_2(i,j)]}$$

where $I_1(i, j)$ is the pixel value at row i, column j in the input image, and $I_2(i, j)$ is the pixel value at row i, column j in the label image.

---

4. Experiments

4.1 Experimental Setup

Each structural dataset is divided into training, validation, and test sets in an 80%, 10%, 10% ratio, respectively, using random sampling. The test and validation sets are not used for model training. To evaluate the optimization performance of the improved DoubleU-Net model, we conducted experimental prediction assessments on four datasets. Model training uses 100 epochs, batch size of 128, and dynamically adjusted learning rate with an initial rate of 0.01, decreasing by 50% every 10 epochs.

shows the quantities of training, validation, and test sets for each structure.

4.2 Random Structure Dataset Evaluation

On the Random dataset, we compared the topology optimization performance of DoubleU-Net and U-Net models at two resolutions (32$\times 64 and 64 \times$128) and analyzed the impact of different loss functions (MAE, MSE, SSIM) on topology optimization effectiveness.

Experimental results are shown in . For both 32$\times 64and64\times 128resolutions, ourproposedDoubleU-
Netmodelachievesbetterpredictionperformancethan U-Netontraining, validation, andtestsets, withanavera
2\times 64resolution, DoubleU-NetandU-NetmodelsusingSSIMlossfunctionachievethebestvalidationandtestI
resolution, the U-Net model achieves the best results on training, validation,

and test sets with MSE loss function, which is surprising. Additionally, MSE performs best on most training sets but not on test sets, indicating some overfitting.

Analyzing the different results from different loss functions, we observe that using MAE as the loss function yields the lowest prediction accuracy because its gradient updates remain constant, which is unfavorable for model training. Using MSE as the loss function performs well on training sets but poorly on test sets due to overfitting. In contrast, the SSIM loss function achieves the best results because, unlike MAE and MSE, SSIM does not measure absolute error but evaluates loss from three perspectives—luminance, contrast, and structure—better aligning with human visual perception. Overall, the best-performing combination on the Random dataset is DoubleU-Net + SSIM, achieving the highest test accuracies of 92.38% and 93.14% at 32\$$\times 64 and 64\times$\$128 resolutions, respectively. Experimental data demonstrates that the SSIM loss function is more suitable for the DoubleU-Net model and yields optimal results. Therefore, for experimental fairness, we select SSIM as the model’ s loss function in subsequent experiments.

compares training loss convergence values for the improved DoubleU-Net. The overall training loss iteration processes for our proposed DoubleU-Net and U-Net models under different loss functions are shown in [Figure 6: see original paper]. As iteration count increases, the DoubleU-Net model exhibits more stable loss function convergence compared to U-Net. After 100 epochs, DoubleU-Net achieves lower loss convergence values than U-Net on both training and validation sets, indicating better fitting on the training set and improved data feature extraction.

4.3 Three Industrial Beam Structure Dataset Evaluation

The prediction performance comparison between DoubleU-Net and U-Net at two resolutions on three industrial structural beams is shown in . Having proven SSIM's excellent performance on the random dataset, we directly adopt SSIM for experimental fairness. DoubleU-Net outperforms U-Net on training, validation, and test sets across all three beam structures at both resolutions. On Cantilever data, test accuracies reach 96.57% and 96.85% at 32\$$\times 64 and 64\times$\$128 resolutions, respectively. On Continuous data, test accuracies are 96.30% and 96.40%. On Simply supported data, test accuracies are 97.47% and 97.30%. Analysis across different resolutions shows that DoubleU-Net prediction accuracy is nearly unaffected, with prediction errors within 1% across resolutions, demonstrating good robustness.

4.4 Prediction Performance Across Four Structures and Resolutions

We present partial prediction effect diagrams of the DoubleU-Net model across four structures at two resolutions, comparing predictions with corresponding real topology-optimized structures in [Figure 8: see original paper]. By comparing each image group, we observe that DoubleU-Net performs excellently on the test set and can effectively predict datasets with four different structures at two resolutions, producing results very close to real topology-optimized structures. This further highlights the effectiveness of the proposed DoubleU-Net model in the topology optimization domain. Experimental results demonstrate outstanding performance across all conditions.

4.5 Generalization Capability Test

A notable drawback of deep learning methods for topology optimization is that trained neural network models are only applicable to specific boundary conditions—prediction performance drops significantly when boundary conditions change. Therefore, we analyze the generalization capability of our proposed model. To our knowledge, this is the first study to examine generalization capability in neural network applications for topology optimization by analyzing models trained on four different datasets across all four dataset types. Results are shown in .

High-resolution datasets better reflect model performance, so testing is conducted at 64$×$128 resolution. We use models trained on four structural datasets to test prediction accuracy on all four structural datasets, focusing particularly on trained model performance on other datasets. Experimental results show that the model trained on the Random structure dataset exhibits the best generalization capability, performing excellently on the other three beam structure datasets with an average accuracy of 92.7%. In contrast, models trained on the other three beam structure datasets only perform excellently on their own structural dataset but poorly on the other three, achieving average accuracies of 87.81%, 89.91%, and 88.49% on different structural datasets for cantilever, continuous, and simply supported beams, respectively—approximately 4% lower than the random dataset-trained model. This proves that training with the random structure dataset yields excellent generalization capability with great potential for real-time structural optimization in large-scale engineering projects.

4.6 High-Quality Prediction Structure Evaluation and Quantity

In topology optimization, high-quality predicted structures are essential for practical application. Therefore, we investigate the quality distribution across test sets at 64$×$128 high resolution under different IoU metric ranges, with results shown in [Figure 9: see original paper]. Each structural test set contains 5,000 samples. Typically, IoU values below 0.7 indicate poor prediction quality where most predicted structures do not match real structures. Under our DoubleU-Net model, low-quality predictions with IoU $<$ 0.7 constitute only 0.1% of all structures. In the IoU range of 0.7-0.9, which we consider moderate quality where most predictions match real structures but details are poorly predicted, the proportion is 7.53%. IoU $>$ 0.9 is considered high-quality prediction where predicted structures are similar to real topology-optimized structures with successfully predicted details, accounting for a large proportion of 92.34%. This quality distribution further validates DoubleU-Net’ s high-quality prediction performance.

---

5. Conclusion

We generated a total of 400,000 2D samples across four boundary conditions: random structures, cantilever beams, continuous beams, and simply supported beams. Each structure includes 100,000 samples, with low-resolution and high-resolution data each accounting for half. This constitutes the most diverse and largest-scale open topology optimization dataset currently available. We proposed an improved DoubleU-Net convolutional neural network model that replaces the first encoder and decoder with pre-trained U-Net encoding and decoding modules to better extract data features, achieving accurate and real-time topology optimization design. The model delivers excellent results across all four boundary condition datasets, achieving 1-2% performance improvement over U-Net. We explored model generalization capability and proved that training with the random structure dataset yields excellent generalization potential for real-time structural optimization in large-scale engineering projects. We are the first to apply SSIM loss to topology optimization, achieving the best results. Since topology optimization structures require high-precision results, we also proposed a quality evaluation method for predicted structures and examined the quantity distribution across different quality levels.

Currently, our research focuses on 2D topology optimization. Future work will emphasize practical applications in topology optimization, including generating 3D topology optimization training data and collecting sufficient high-quality data for datasets. We also face challenges regarding substantial computational resource requirements. Furthermore, to improve deep learning generalization capability in real-time optimization, we need to obtain 3D topology optimization data for different structures to achieve real-time 3D topology optimization. Our experience in the 2D domain provides innovative ideas and valuable references for achieving intelligent, refined, and realistic 3D topology optimization.

References

[1] MICHELL A G M J T L, EDINBURGH,, MAGAZINE D P, SCIENCE J O. LVIII. The limits of economy of material in frame-structures [J]. 1904, 8(47): 589-97.
[2] BENDSøE M P, SIGMUND O. Topology optimization [M]. Optimization of Structural and Mechanical Systems. World Scientific. 2007: 161-94.
[3] SIGMUND O, MAUTE K. Topology optimization approaches [J]. Structural and Multidisciplinary Optimization, 2013, 48(6): 1031-55.

[4] DBOUK T J A T E. A review about the engineering design of optimal heat transfer systems using topology optimization [J]. 2017, 112: 841-54.
[5] HUANG X, ZHOU S, XIE Y, et al. Topology optimization of microstructures of cellular materials and composites for macrostructures [J]. 2013, 67: 397-407.
[6] ZHU J-H, ZHANG W-H, XIA L J A O C M I E. Topology optimization in aircraft and aerospace structures design [J]. 2016, 23: 595-622.
[7] BENDSøE M P. Optimal shape design as a material distribution problem [J]. Structural optimization, 1989, 1(4): 193-202.
[8] ZHOU M, ROZVANY G I N. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization [J]. Computer Methods in Applied Mechanics and Engineering, 1991, 89(1):
[9] BENDSøE M, KIKUCHI N J C M I A M, ENGINEERING. Generating optimal topologies in structural design using a homogenization method [J]. 1988, 71(2): 197-224.
[10] ALLAIRE G, JOUVE F, TOADER A-M. A level-set method for shape optimization [J]. Comptes Rendus Mathematique, 2002, 334(12): 1125-30.
[11] WANG M Y, WANG X, GUO D J C M I A M, et al. A level set method for structural topology optimization [J]. 2003, 192(1-2): 227-46.
[12] XIE Y M, STEVEN G P J C, STRUCTURES. A simple evolutionary procedure for structural optimization [J]. 1993, 49(5): 885-96.
[13] DOAN Q H, LEE D, LEE J, et al. Multi-material structural topology optimization with decision making of stiffness design criteria [J]. 2020, 45: 101098.
[14] GUO X, ZHANG W, ZHONG W J J O A M. Doing topology optimization explicitly and geometrically—a new moving morphable components based framework [J]. 2014, 81(8).
[15] ZHU J, ZHANG W, BECKERS P. Integrated layout design of multi-component system [J]. 2009, 78(6): 631-51.
[16] DEGERTEKIN S O, LAMBERTI L, UGUR I B J A S C. Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm [J]. 2019, 79: 363-90.
[17] LIAO Z, ZHANG Y, WANG Y, et al. A triple acceleration method for topology optimization [J]. Structural and Multidisciplinary Optimization, 2019, 60(2): 727-44.
[18] LI W, SURYANARAYANA P, PAULINO G H J M R C. Accelerated fixed-point formulation of topology optimization: Application to compliance minimization problems [J]. 2020, 103: 103469.
[19] JANG I G, KWAK B M. Evolutionary topology optimization using design space adjustment based on fixed grid [J]. International Journal for Numerical Methods in Engineering, 2006, 66(11):
[20] SOSNOVIK I, OSELEDETS I J R J O N A, MODELLING M. Neural networks for topology optimization [J]. 2019, 34(4): 215-23.
[21] ZHENG S, HE Z, LIU H. Generating three-dimensional structural topologies via a U-Net convolutional neural network [J]. Thin-Walled Structures, 2021, 159: 107263.
[22] RONNEBERGER O, FISCHER P, BROX T. U-Net: Convolutional

Networks for Biomedical Image Segmentation; proceedings of the Medical Image Computing and Computer-Assisted Intervention -MICCAI 2015, Cham, F 2015//, 2015 [C]. Springer International Publishing.
[23] WANG D, XIANG C, PAN Y, et al. A deep convolutional neural network for topology optimization with perceptible generalization ability [J]. Engineering Optimization, 2022, 54(6): 973-88.
[24] ZHANG L, CHENG L, LI H, et al. Hierarchical deep-learning neural networks: finite elements and beyond [J]. 2021, 67: 207-30.
[25] DENG H, TO A C. A Parametric Level Set Method for Topology Optimization Based on Deep Neural Network [J]. Journal of Mechanical Design, 2021, 143(9).
[26] LI J, YE H, YUAN B, et al. Cross-resolution topology optimization for geometrical non-linearity by using deep learning [J]. 2022, 65(4): -.
[27] YU Y, HUR T, JUNG J, et al. Deep learning for determining a near-optimal topological design without any iteration [J]. 2019, 59(3): 787-99.
[28] WANG Y, SOUTIS C, ANDO D, et al. Application of deep neural network learning in composites design [J]. European Journal of Materials, 2022, 2(1): 117-70.
[29] YAN J, ZHANG Q, XU Q, et al. Deep learning driven real time topology optimisation based on initial stress learning [J]. Advanced Engineering Informatics, 2022, 51: 101472.
[30] SEO J, KAPANIA R K J S, OPTIMIZATION M. Topology optimization with advanced CNN using mapped physics-based data [J]. 2023, 66(1): 1-20.
[31] ZHOU Z, RAHMAN SIDDIQUEE M M, TAJBAKHSH N, et al. UNet++: A Nested U-Net Architecture for Medical Image Segmentation; proceedings of the Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Cham, F 2018//, 2018 [C]. Springer International Publishing.
[32] ATES G C, GORGULUARSLAN R M J S, OPTIMIZATION M. Two-stage convolutional encoder-decoder network to improve the performance and reliability of deep learning models for topology optimization [J]. 2021, 63(4): 1927-50.
[33] NIE Z, LIN T, JIANG H, et al. TopologyGAN: Topology Optimization Using Generative Adversarial Networks Based on Physical Fields Over the Initial Domain [J]. Journal of Mechanical Design, 2021, 143(3).
[34] WANG L, SHI D, ZHANG B, et al. Deep learning driven real time topology optimization based on improved convolutional block attention (Cba-U-Net) model [J]. Engineering Analysis with Boundary Elements, 2023, 147: 112-24.
[35] JEONG H, BAI J, BATUWATTA-GAMAGE C P, et al. A Physics-Informed Neural Network-based Topology Optimization (PINNTO) framework for structural optimization [J]. Engineering Structures, 2023, 278: 115484.
[36] BEHZADI M M, ILIEŞ H T. GANTL: Toward Practical and Real-Time Topology Optimization With Conditional Generative Adversarial Networks and Transfer Learning [J]. Journal of Mechanical Design, 2021, 144(2).
[37] HUNTER W, OTHERS. ToPy - Topology optimization with Python [J]. GitHub repository, 2017.

[38] JHA D, RIEGLER M, JOHANSEN D, et al. DoubleU-Net: A Deep Convolutional Neural Network for Medical Image Segmentation [M]. 2020.
[39] CHEN L-C, PAPANDREOU G, SCHROFF F, et al. Rethinking atrous convolution for semantic image segmentation [J]. 2017.

Corresponding author: Yang Aimin (E-mail: amyang@gdut.edu.cn)
Corresponding author: Lin Jianghao (E-mail: lin_{hao}@foxmail.com)

Author Contributions:
Li Shun: Conceptualization, methodology, data acquisition, experimentation, writing-original draft
Bai Qifeng: Conceptualization, data acquisition
Lin Nankai: Conceptualization, data acquisition
Zeng Peijian: Conceptualization, data acquisition
Yang Aimin: Conceptualization, writing-original draft
Lin Jianghao: Writing-review & editing

Note: Figure translations are in progress. See original paper for figures.