

# Problem-independent machine learning (PIML)-based topology optimization - A universal approach

Mengcheng Huang<sup>1</sup>, Zongliang Du<sup>1,2\*</sup>, Chang Liu<sup>1,2\*</sup>, Yonggang Zheng<sup>1</sup>, Tianchen Cui<sup>1</sup>,  
Yue Mei<sup>1,2</sup>, Xiao Li<sup>3</sup>, Xiaoyu Zhang<sup>3</sup>, Xu Guo<sup>1,2\*</sup>

<sup>1</sup>*State Key Laboratory of Structural Analysis for Industrial Equipment,*

*Department of Engineering Mechanics,*

*Dalian University of Technology, Dalian, 116023, P.R. China*

<sup>2</sup>*Ningbo Institute of Dalian University of Technology, Ningbo, 315016, P.R. China*

<sup>3</sup>*Beijing Institute of Spacecraft System Engineering, Beijing 100094, P.R. China*

## Abstract

Solving topology optimization problem is very computationally demanding especially when high-resolution results are sought for. In the present work, a problem-independent machine learning (PIML) technique is proposed to reduce the computational time associated with finite element analysis (FEA) which constitutes the main bottleneck of the solution process. The key idea is to construct the structural analysis procedure under the extended multi-scale finite element method (EMsFEM) framework, and establish an implicit mapping between the shape functions of EMsFEM and element-wise material densities of a coarse-resolution element through machine learning (ML). Compared with existing works, the proposed mechanistic-based ML technique is truly problem-independent and can be used to solve any kind of topology optimization problems without any modification once the easy-to-implement off-line training is completed. It is demonstrated that the proposed approach can reduce the FEA time significantly. In particular, with the use of the proposed approach, a topology optimization problem with 200 million of design variables can be solved on a personal workstation with an average of only two minutes for FEA per iteration step.

**Keywords:** Topology optimization; Extended multi-scale finite element method (EMsFEM); Shape function; Problem independent machine learning (PIML).

---

\*Corresponding authors. E-mail: [zldu@dlut.edu.cn](mailto:zldu@dlut.edu.cn) (Zongliang Du), [c.liu@dlut.edu.cn](mailto:c.liu@dlut.edu.cn) (Chang Liu), [guoxu@dlut.edu.cn](mailto:guoxu@dlut.edu.cn) (Xu Guo)

## 1. Introduction

Topology optimization is a powerful tool for helping engineers design innovative structures and products and has already found successful applications in a variety of industrial fields [1–4]. It is, however, well-known that the solution of topology optimization problems is highly computationally demanding and often involves a large amount of computational effort. In order to enhance the solution efficiency of topology optimization problems, many approaches such as the parallel computing-based methods [5–8], the multi-scale methods [9–11], the multi-resolution methods [12–16], the design variable reduction methods [17, 18] and the degrees of freedom elimination strategy [19, 20], just to name a few, have been proposed in the literature.

With the fast development of artificial intelligence (AI) and machine learning (ML), recent years have witnessed a great interest on utilizing AI/ML techniques to address the issue of high computational cost associated with topology optimization. For example, numerous researches have focused on establishing an end-to-end relationship between the prescribed optimization parameters such as design domain, boundary conditions as well as positions/magnitudes of the external loads, and the final optimized structures to achieve a so-called real time topology optimization [21–26]. Although encouraging results have been obtained, there still lacks systematic investigations on their performances when these approaches are applied to solve problems similar to the ones included in the training sets. Besides, since these methods intend to construct a direct mapping between the optimization parameters and the optimized structural layout, the involved computational cost is prohibitive since a large number of expansive topology optimization problems must be solved for generating the samples required for training the ML models.

Following a quite different path, rather than pursuing a more attractive but less generalizable real-time topological design, some researchers have devoted to reducing the computation time associated with finite element analysis (FEA), which constitutes the main bottleneck for an efficient topology optimization algorithm, by utilizing the state-of-the-art ML techniques. In particular, Chi et al. proposed a so-called universal on-line ML approach for topology optimization under a multi-resolution framework by collecting train data from earlier iterations of the topology optimization process [27]. Later on, this approach has been further refined by adopting an off-line training strategy that separates the model training process from its application in actual topology

optimization [28]. In this refined approach, a ML model is established to predict the fine-resolution sensitivities required for updating the design variables efficiently using the individual displacement field and the local material densities on a coarse-resolution supercell as input. It has been demonstrated that with the use of this approach, large scale design problems up to 38 million design variables and a speedup up to 30 times can be achieved. Besides, a so-called finite element convolutional neural networks (FE-CNNs) was also proposed to accelerate structural topology optimization algorithm [29]. The constructed FE-CNN is used as a mapping function between high-resolution and low-resolution finite element systems, to cut down the scale of FEA in the optimization process. Connection layers and activation function based on prior knowledge of mechanics and topology optimization are designed and used, which makes the proposed neural networks accurate and adaptable. It has been shown that this approach can accelerate optimization by up to an order of magnitude in computational time.

Although remarkable results have been made by the above approaches, there still exists some challenging issues deserved to be further addressed. Firstly, the ML model constructed in these approaches is not fully problem-independent since a set of topology optimization problems with specific boundary/loading conditions (e.g., bending/torsion dominated problems, etc.) must be solved for generating the samples for training the ML models. It is unclear whether (at least theoretically) the adopted problems for training are still representative for the solution of more general topology optimization problems. Secondly, the validity of the assumption of local dependency of the ML model in [28] remains elusive since theoretically speaking the displacement of a coarse-resolution supercell is actually determined by the values of all design variables in the design domain rather than only the values of the material densities in the supercell. Although this limitation has been addressed by incorporating all material densities into the ML model in [29], this treatment may lead to prohibitive computational resources for training a heavy ML model, and actually computationally intractable for large-scale topology optimization problems. Finally, in order to keep a balance between the computational efficiency and prediction accuracy of the developed ML model, the size of the coarse-resolution supercell in these approaches cannot take a too large value (i.e., 2-3 times of the size of the fine-resolution element in [28]). This, however, will inevitably reduce the speedup ratio of FEA when ML-enhanced techniques are applied for solving

topology optimization problems.

In the present work, we intend to solve the aforementioned issues in a unified way. The central idea is to resort to a multi-resolution scheme and use a novel problem-independent machine learning (PIML) technique to save the computational effort substantially. Actually, unlike in existing works where the displacement associated with fine-resolution mesh is learned to predict the sensitivity required for fine-scale update of design variables, we suggest to establish an implicit mapping between the shape functions in the context of extended multi-scale finite element method (EMsFEM) framework and element-wise material densities of a coarse-resolution element through a separate off-line ML process. Once the shape functions are obtained, they can be employed to construct the corresponding local stiffness matrix associated with the coarse-resolution mesh for EMsFEM analysis. Compared with the existing treatment in other works, the proposed PIML technique is truly problem-independent in the sense that it does not rely on any specific topology optimization problems for the collection of training samples and only local information for material distribution in the coarse-resolution element is required as the input. Moreover, the developed ML model can be used to solve any kind of topology optimization problems governed by the same type of partial differential equations without any modification once the training is completed. A large-scale topology optimization problem involving 200 million of design variables fully demonstrates the effectiveness of the proposed approach for reducing the computational effort associated with topology optimization.

The reminder of the article is organized as follows. After a brief presentation of the adopted setting of the topology optimization problem in Section 2, Section 3 introduces the extended multi-scale finite element method (EMsFEM) used in the proposed approach for finite element analysis. Section 4 discusses how to construct the design-dependent shape functions of EMsFEM efficiently through a PIML technique and integrate them into a topology optimization framework seamlessly. Based on the above preparations, the overall solution procedure is described in Section 5. Section 6 provides some numerical examples to demonstrate the effectiveness of the proposed approach. The article is concluded by a summary of the main contributions, distinctive advantages and possible extension directions of the present work in Section 7.

## 2. Setting of the topology optimization problem

Since the main purpose of the present work is just to illustrate the basic idea of the proposed PIML-enhanced technique, only compliance minimization problems with a volume constraint under the linear elasticity assumption is considered in the following. Although the proposed technique can be combined with any topology methods such as the level set method [30, 31], the BESO method [32] as well as the MMC/MMV method [17, 33] for achieving an efficient solution process, the most popular Solid Isotropic Material with Penalization (SIMP) method [34] is adopted here, and a density filter [35, 36] is introduced to regularize the problem and avoid the checkerboard patterns.

Under the SIMP framework, the considered topology optimization problem can be formulated as follows:

$$\begin{aligned}
& \text{Find } \boldsymbol{\rho} = (\rho_1, \dots, \rho_n)^\top, \mathbf{U} \\
& \text{Min. } C(\boldsymbol{\rho}, \mathbf{U}) = \mathbf{U}^\top \mathbf{K} \mathbf{U} = \sum_{e=1}^n E_e(\rho_e) \mathbf{u}_e^\top \mathbf{k}_0 \mathbf{u}_e \\
& \text{S. t.} \\
& \quad \mathbf{K} \mathbf{U} = \mathbf{F}, \\
& \quad V(\boldsymbol{\rho})/V_0 - f \leq 0, \\
& \quad 0 \leq \rho_e \leq 1, e = 1, \dots, n.
\end{aligned} \tag{1}$$

In Eq. (1),  $\boldsymbol{\rho}$  is the vector of the design variables with  $\rho_e$  denoting the density of the  $e$ -th element. The symbol  $n$  denotes the total number of finite elements used for discretizing the prescribed design domain. The symbol  $\mathbf{K}$  is the global stiffness matrix with  $\mathbf{k}_0$  representing the stiffness matrix corresponding to a solid element. The quantities  $V(\boldsymbol{\rho})$  and  $V_0$  are the volumes of the structure and the design domain, respectively,  $f$  is the prescribed upper bound of the available solid material, and  $\mathbf{F}$  as well as  $\mathbf{U}$  are the vectors of external load and the displacement, respectively. In Eq. (1),  $E_e$  represents the Young's modulus of the  $e$ -th element, which can be interpolated as

$$E_e(\rho_e) = E_{\min} + \rho_e^p (E_0 - E_{\min}), \tag{2}$$

where  $E_0$  is the Young's modulus of the solid material,  $E_{\min}$  is a small positive real number (e.g.,  $E_{\min} = 10^{-3} \sim 10^{-6}$ ) used for avoiding the singularity of the global stiffness matrix. And  $p$  is a penalization factor ( $p = 3$  in this paper) introduced to ensure black-and-white designs.

For the SIMP approach, in order to eliminate mesh-dependency and numerical instabilities

such as the checkerboard patterns, a density filter is introduced to transform the original densities  $\rho_e$  as

$$\tilde{\rho}_e = \frac{\sum_{i \in N_e} H_{ei} \rho_i}{\sum_{i \in N_e} H_{ei}}, \quad (3)$$

where  $N_e$  is the influence domain of elements  $e$ , i.e.,  $N_e = \{i | d(i, e) \leq r_{\min}\}$  with  $d(i, e)$  denoting the center-to-center distance between the  $i$ -th element and the  $e$ -th element, and  $r_{\min}$  is the radius of the density filter. In Eq. (3),  $H_{ei}$  is the weight factor defined as

$$H_{ei} = \max(0, r_{\min} - d(i, e)). \quad (4)$$

By taking the filter operation into consideration, the sensitivities of the structural compliance  $C$  and the material volume  $V$  with respect to the  $j$ -th design variable  $\rho_j$  can be obtained by the chain rule as

$$\frac{\partial C}{\partial \rho_j} = \sum_{e \in N_j} \frac{\partial C}{\partial \tilde{\rho}_e} \frac{\partial \tilde{\rho}_e}{\partial \rho_j} = \sum_{e \in N_j} \frac{H_{je}}{\sum_{i \in N_e} H_{ei}} \frac{\partial C}{\partial \tilde{\rho}_e} = - \sum_{e \in N_j} \frac{H_{je}}{\sum_{i \in N_e} H_{ei}} p \tilde{\rho}_e^{p-1} (E_0 - E_{\min}) \mathbf{u}_e^\top \mathbf{k}_0 \mathbf{u}_e \quad (5)$$

and

$$\frac{\partial V}{\partial \rho_j} = \sum_{e \in N_j} \frac{\partial V}{\partial \tilde{\rho}_e} \frac{\partial \tilde{\rho}_e}{\partial \rho_j} = \sum_{e \in N_j} \frac{H_{je}}{\sum_{i \in N_e} H_{ei}} \frac{\partial V}{\partial \tilde{\rho}_e} = \sum_{e \in N_j} \frac{H_{je}}{\sum_{i \in N_e} H_{ei}}, \quad (6)$$

respectively.

### 3. Extended multi-scale finite element method (EMsFEM)

In order to achieve an ML-enhanced acceleration for solving topology optimization problems, in the present work, the so-called extended multi-scale finite element method (EMsFEM) developed in [37] for mechanical analysis of heterogeneous materials is adopted for FEA. Actually, EMsFEM is an extended vector field-oriented version of the multi-scale finite element method developed in [38] for scalar field problems. In EMsFEM, two kinds of elements are constructed to discretize the design domain, i.e., the whole design domain is first discretized with a set of coarse-resolution elements and each coarse-scale element is further discretized by a set of fine-resolution elements. Therefore, the detailed micro-scale material heterogeneity can be reasonably described by the fine-resolution elements, and the overall boundary value problem can be efficiently solved based on the coarse-resolution elements with much fewer degrees of freedom. The bridging between coarse and fine-resolution elements is achieved by constructing a set of *numerical shape functions*. Previous

works showed that EMsFEM can be applied successfully to solve various types of multi-scale mechanical analysis problems [39, 40].

In EMsFEM, the multi-scale numerical shape functions are constructed by assuming (for the sake of simplicity, only two-dimensional problem is discussed here. Extension to three-dimensional problem is straightforward)

$$\begin{aligned} u_E^l &= \sum_{i=1}^M N_{ixx}^l \hat{u}_E^i + \sum_{i=1}^M N_{ixy}^l \hat{v}_E^i, \\ v_E^l &= \sum_{i=1}^M N_{iyx}^l \hat{u}_E^i + \sum_{i=1}^M N_{iyy}^l \hat{v}_E^i, \end{aligned} \quad (7)$$

where  $\hat{u}_E^i$  ( $\hat{v}_E^i$ ) is the displacement along  $x$  ( $y$ ) direction of the  $i$ -th node of the  $E$ -th coarse-resolution element  $\Omega^E$  and  $u_E^l$  ( $v_E^l$ ) is the displacement along  $x$  ( $y$ ) direction of the  $l$ -th node of the fine-resolution mesh inside this coarse-resolution element (see Fig. 1 for reference). The central idea of EMsFEM is to calculate the values of  $N_{ixx}^l$ ,  $N_{ixy}^l$ ,  $N_{iyx}^l$  and  $N_{iyy}^l$ , where  $i = 1, \dots, M$ , on every node of the fine-resolution mesh inside the coarse-resolution element *numerically*. This is achieved by setting  $\hat{u}_E^i = 1$  and  $\hat{v}_E^i = 1$  for all  $i = 1, \dots, M$  separately and imposing appropriate boundary conditions along  $\partial\Omega^E$ . Furthermore, in order to guarantee the consistency between the displacement fields associated with the fine and coarse-resolution mesh, the following constraints on  $N_{ixx}^l$ ,  $N_{ixy}^l$ ,  $N_{iyx}^l$  and  $N_{iyy}^l$  should also be satisfied

$$\begin{aligned} \sum_{i=1}^M N_{ixx}^l &= 1, \quad \sum_{i=1}^M N_{iyy}^l = 1, \quad l = 1, \dots, N, \\ \sum_{i=1}^M N_{iyx}^l &= 0, \quad \sum_{i=1}^M N_{ixy}^l = 0, \quad l = 1, \dots, N, \end{aligned} \quad (8)$$

where  $N$  is the total number of nodes in the fine-resolution mesh.

In the present work, as shown in Fig. 2, a four-node coarse-resolution element is adopted and a linear boundary condition is imposed on  $\partial\Omega^E$  for constructing the numerical shape functions. In addition, the coarse-resolution element is also discretized by a set of four-node fine-resolution elements. Once the numerical shape functions are determined, the  $8 \times 8$  stiffness matrix  $\mathbf{K}^E$  associated with the coarse-resolution element can be obtained by variational principle as

$$\mathbf{K}^E = \sum_{f=1}^m (\mathbf{N}_f)^\top \mathbf{k}^f \mathbf{N}_f, \quad \mathbf{N}_f = [\mathbf{N}_{f1}^\top, \mathbf{N}_{f2}^\top, \mathbf{N}_{f3}^\top, \mathbf{N}_{f4}^\top]^\top, \quad (9)$$

where  $\mathbf{k}^f$  donates the  $8 \times 8$  stiffness matrix of the  $f$ -th fine-resolution element,  $m$  is the total number of fine-resolution elements in a coarse-resolution element. In Eq. (9),  $\mathbf{N}_{fj_e} = [N_{1xx}^{l[j_e]}, N_{1xy}^{l[j_e]}, N_{2xx}^{l[j_e]}, N_{2xy}^{l[j_e]}, N_{3xx}^{l[j_e]}, N_{3xy}^{l[j_e]}, N_{4xx}^{l[j_e]}, N_{4xy}^{l[j_e]}, N_{1yx}^{l[j_e]}, N_{1yy}^{l[j_e]}, N_{2yx}^{l[j_e]}, N_{2yy}^{l[j_e]}, N_{3yx}^{l[j_e]}, N_{3yy}^{l[j_e]}, N_{4yx}^{l[j_e]}, N_{4yy}^{l[j_e]}]$ ,  $j_e = 1, \dots, 4$  and  $l[j_e]$  denotes the global index number of the  $j$ -th local node of the  $f$ -th fine-resolution element sorting in the corresponding coarse-resolution element.

The coarse-resolution element stiffness matrix obtained in this way can then be assembled to find the coarse-resolution displacement vectors  $\hat{\mathbf{u}}^E = (\hat{u}_e^1, \hat{v}_e^1, \dots, \hat{u}_e^4, \hat{v}_e^4)^\top, E = 1, \dots, NE$  with  $NE$  denoting the total number of coarse-resolution elements in the design domain. Once  $\hat{\mathbf{u}}^E$  is obtained, the fine-resolution displacement  $\mathbf{u}^E = (u_E^1, v_E^1, \dots, u_E^N, v_E^N)^\top$  in  $\Omega^E$  can be calculated from the nodal values of  $N_{ixx}^l, N_{ixy}^l, N_{iyx}^l$  and  $N_{iyy}^l, i = 1, \dots, 4; l = 1, \dots, N$  as

$$\begin{bmatrix} u_E^1 \\ v_E^1 \\ \vdots \\ u_E^N \\ v_E^N \end{bmatrix} = \begin{bmatrix} N_{1xx}^1 & N_{1xy}^1 & \cdots & N_{4xx}^1 & N_{4xy}^1 \\ N_{1yx}^1 & N_{1yy}^1 & \cdots & N_{4yx}^1 & N_{4yy}^1 \\ & & \vdots & & \\ N_{1xx}^N & N_{1xy}^N & \cdots & N_{4xx}^N & N_{4xy}^N \\ N_{1yx}^N & N_{1yy}^N & \cdots & N_{4yx}^N & N_{4yy}^N \end{bmatrix} \begin{bmatrix} \hat{u}_E^1 \\ \hat{v}_E^1 \\ \vdots \\ \hat{u}_E^4 \\ \hat{v}_E^4 \end{bmatrix}. \quad (10)$$

It is worth noting that although only four-node coarse-resolution element and linear boundary conditions are used in the present work for EMsFEM analysis, actually more nodes and more complex boundary conditions can be introduced to enhance the accuracy of EMsFEM. We refer the readers to [37] for more details on this aspect.

#### 4. Machine learning-based construction of the EMsFEM shape functions

The calculation of the EMsFEM shape functions is described in previous subsection. It is, however, worth noting that when EMsFEM is applied in topology optimization, the shape functions must be constructed at every step of iteration since the material distribution is constantly changed in every coarse-resolution element. Under this circumstance, it cannot be expected to reduce the FEA time substantially even when EMsFEM is used for structural analysis.

In order to circumvent the above difficulty, we propose a novel procedure to *replace the time-consuming online shape function construction process by an off-line trained ML model*. Under this treatment, the input parameters of the ML model are just the fine-resolution element densities in a



specific coarse-resolution element, and the outputs are the nodal values of  $N_{ixx}^l$ ,  $N_{ixy}^l$ ,  $N_{iyx}^l$  and  $N_{iyy}^l$ ,  $i = 1, \dots, 4; l = 1, \dots, N$  in this element. From the predicted nodal values of these shape functions, the stiffness matrix of the coarse-resolution element can be generated very quickly and used directly for EMsFEM analysis. In the following, the construction of the aforementioned ML model will be described in detail.

#### 4.1 The construction of the atrial neural network (ANN)

The artificial neural network used for training the shape functions is illustrated schematically in Fig. 3. Actually, it is just an ordinary feedforward neural network, where the activation functions in each layer are set as the elu function or tanh function, respectively. It is interesting to note that even with this ordinary ANN without any special architectures, fairly good learning results are achieved which can be reflected clearly in the examples provided in Section 6.

At this position, it is necessary to point out that since the existence of the relations in Eq. (8), only 12 out of 16  $N_i^l$ s are actually independent (for 2D case). In other words, once the values of  $N_{1xx}^l, N_{1xy}^l, \dots, N_{3xx}^l, N_{3xy}^l, N_{3yx}^l, N_{3yy}^l$  are determined,  $N_{4xx}^l, N_{4xy}^l, N_{4yx}^l$  and  $N_{4yy}^l$  can be calculated from Eq. (8) directly. Taking this fact into consideration, only the values of  $N_{1xx}^l, N_{1xy}^l, \dots, N_{3xx}^l, N_{3xy}^l, N_{3yx}^l, N_{3yy}^l$  inside the coarse-resolution element are set as the output of the neural network.

#### 4.2 The training of the ANN

The generation of samples, the loss function and the training method are described briefly in this subsection. First, the local densities in each coarse-resolution element corresponding to an individual sample for training are generated in the range of  $[0, 1]$  through a random process and there is no need to collect the samples during a real optimization process. Second, the loss function contains two parts: one is the mean square error between the prediction and true outputs; the other one is the mean square error between the stiffness matrix calculated by the predicted shape functions and the exact stiffness matrix obtained by the EMsFEM of the coarse-resolution elements. Similar to the idea of physics-informed neural network [41], the second part is actually a physical constraint imposed on output to guarantee the accuracy of the stiffness matrix of the coarse-resolution element for FEA, which directly determines the accuracy of the displacement field obtained by EMsFEM. This part is crucial for the success of the constructed deep neural network. During the training

process, the automatic-differentiation algorithm in TensorFlow is adopted, and the derivative of the loss function to the weight coefficient in the neural network is obtained by using the random gradient algorithm. Finally, the weight coefficients are updated by the Adam optimizer. In fact, during the process of training, we only set the learning rate of the Adam optimizer as 0.001, and the rest of the parameters are set by default in TensorFlow 2 without any parameter adjustment. In the present work, two deep neural networks are constructed to predict the shape functions for  $m = 5 \times 5 = 25$  and  $m = 10 \times 10 = 100$ , respectively. For  $m = 100$ , there are totally 11 layers in the intermediate layer, where the activation functions are set as [tanh, elu, tanh, elu, tanh, elu, elu, tanh, elu, tanh, elu] and the number of activation functions in each layer is set as [100, 120, 140, 160, 180, 200, 180, 160, 140, 120, 100], respectively. For  $m = 25$ , the activation functions are the same as in the case of  $m = 100$ , but the number of activation functions in each layer is set as [50, 60, 70, 80, 90, 100, 90, 80, 70, 60, 50].

The merit of the proposed ML scheme can be summarized as follows: (1) It is *totally problem-independent* in the sense that the learned shape functions are actually the discretized versions of Green's functions of the underlying governing partial differential equations (PDEs), which are obviously totally independent on the settings of the topology optimization problems (e.g., boundary conditions, design domain and external loads, etc) to be solved. (2) It is also *theoretically* locally-determined since learned shape functions are solely determined by the material distribution in the considered coarse-resolution element and irrespective of the material distributions in any other coarse-resolution elements. To the best of the authors' knowledge, this property cannot be shared by existing schemes, at least theoretically. (3) Unlike in most existing works, the proposed ML model's training process does not rely on any specific topology optimization problem and can be accomplished only by randomly generating the local material densities. Therefore, the trained shape functions can be applied to solve any type of topology optimization problem governed by the same PDEs.

## 5. Topology optimization solution procedure

Compared with the traditional solution procedure of SIMP method, the only difference in the present method is that the ML-enhanced EMsFEM is used to calculate the nodal displacement of the fine-resolution mesh. The flow chart of the proposed solution procedure is shown in Fig. 4.

Actually, high solution efficiency of the proposed approach mainly comes from the two facts: (1) The dimension of the global stiffness matrix is substantially decreased by the EMsFEM where only a coarse-resolution problem is solved. Theoretically, the computational complexity of FEA can be reduced from  $O(n^3)$  to  $O((n/L)^3)$  where  $n$  and  $L$  denote the numbers of degrees of freedom associated with the fine-resolution mesh on the entire design domain and a single coarse-resolution element, respectively. (2) The computation time of EMsFEM shape functions is greatly reduced by the developed ML model. In addition, in order to further improve the efficiency of generating the element stiffness matrix of coarse-resolution elements, two thresholds  $\bar{\rho}$  and  $\underline{\rho}$  are introduced. For coarse-resolution elements with average densities greater than the  $\bar{\rho}$  or lower than  $\underline{\rho}$ , they are treated as pure solid or weak material, respectively. The EMsFEM shape functions and the stiffness matrix of these two kinds of coarse-resolution elements can be stored in advance and called directly during the optimization process. With the use of this treatment, the computational time for generating the element stiffness matrix of coarse-resolution elements can be reduced significantly, especially for large-scale examples or design problems with a low available volume fraction. It is also worthwhile to note that the proposed approach can be applied under the MMC-based topology optimization framework to further reduce the computational time by resorting to the degrees of freedom elimination scheme [19, 20].

Since structural compliance is taken as the objective function in the present work, once the displacement field associated with the fine-resolution mesh is obtained, it can be directly substituted into Eq. (5) to calculate the sensitivity, and then the optimality criteria (OC) method [36] is used to update the densities of the fine-resolution mesh.

## 6. Numerical examples

In this section, a short cantilever beam example, an L-shaped beam example and a MBB beam example are solved to validate the effectiveness and accuracy of the proposed approach. All the parameters involved in these numerical examples are assumed to be dimensionless. The Young's moduli of solid and weak material are  $E_0 = 1$  and  $E_{\min} = 10^{-3}$ , respectively, and the Poisson's ratios for both solid and weak materials are 0.3. The radius of the filter  $r_{\min}$  is taken as three times of the fine-resolution element size and the upper bound of the volume fraction of the available solid material is 0.5 in all tested examples. In addition, we assumed the threshold values for generating

the coarse-resolution element stiffness matrix as  $\bar{\rho} = 0.95$  and  $\underline{\rho} = 0.002$ , respectively. In addition, optimality criteria (OC) method is used to update design variables and the optimization process is terminated when the relative change of the objective function value is less than 0.0002 in the last five consecutive iterations. All examples are solved on a laptop equipped with an Intel(R) Xeon (R) Gold 6256 3.60GHz CPU and 512.0GB of RAM.

### 6.1 The short cantilever beam example

The accuracy and efficiency of the ANN based EMsFEM are verified in this example as shown in Fig. 5. The design domain is discretized by a set of fixed fine-resolution  $3200 \times 1600$  uniform quadrilateral plane stress elements. Two sets of coarse-resolution mesh (i.e.,  $640 \times 320$  and  $320 \times 160$ ), where  $5 \times 5$  and  $10 \times 10$  fine-resolution elements are contained in each coarse-resolution element, respectively, are examined. As illustrated in Table 1, the optimized designs obtained by the proposed approach are similar to that resulting from the classical SIMP method. In particular, for the case  $m = 5 \times 5$ , the optimized value of the structural compliance is almost the same as that obtained from direct fine-scale calculation (the relative error is only  $8.80 \times 10^{-5}$ ). Meanwhile, very small relative errors (i.e.,  $3.00 \times 10^{-4}$  ( $m = 5 \times 5$ ) and  $8.72 \times 10^{-3}$  ( $m = 10 \times 10$ ), respectively) of the optimized values of the objective function between the EMsFEM ( $C_{\text{EMs}}$ ) and ANN-EMsFEM ( $C_{\text{ANN-EMs}}$ ) methods verify the accuracy of the proposed ML model. From another point of view, the EMsFEM used in this paper is based on the assumption of linear boundary conditions, i.e., the displacements are linearly distributed on the boundaries of the coarse-resolution elements, and this assumption leads to the difference between the results of the EMsFEM analysis and the fine-scale analysis of the optimized structures.

In addition, the filter radius of this example is fixed as 3 times the size of the fine-resolution mesh and smaller than the size of the coarse-resolution meshes, there is no checkerboard phenomenon or QR modes in the optimized structures. Actually, this performance is difficult to achieve in traditional multi-resolution topology optimization approaches [12]. This result clearly demonstrates the developed PIML model's accuracy for predicting the coarse-resolution elements' stiffness matrix.

The average time for one iteration step  $t_{\text{it}}$  is also compared in Table 1. Compared to the value associated with the classical SIMP method,  $t_{\text{it}}$  is reduced by 75.77% (for the  $10 \times 10$  case) and

74.52% (for the case  $m = 5 \times 5$ ), respectively. Compared with the EMsFEM of coarse-resolution elements containing  $5 \times 5$  fine meshes, more numerical shape functions and more complicated stiffness matrix are required for the coarse-resolution elements containing  $10 \times 10$  fine meshes. Considering the fact that, the number of coarse elements of the latter case is 4 times that of the former, the total solution time of those two cases are almost the same.

It is also noticed that, in the proposed PIML-enhanced approach, the time cost for OC update takes the majority in each iteration as illustrated in Fig. 6. In this figure, the FEM time pertaining to the proposed approach includes the time consumed in generating the EMsFEM shape functions of each coarse-resolution element and calculating the fine-resolution displacement. Interestingly, in the classical SIMP method, 80.82% of the total computational time is spent on FEA, while in the proposed approach, the time cost of OC update consumes more than 77% of computational time. Actually, almost 10 times of reduction in FEA has been achieved by the proposed approach for this problem. Since the computational complexity of solving a set of linear algebraic equations is proportional to the cube of the number of unknowns, it can be expected that the larger the topology optimization problem to be solved, the more pronounced acceleration can be achieved for FEA. Furthermore, since the proposed PIML-enhanced approach is fully scalable, parallel computing technique can be adopted naturally to further improving the efficiency of the solution process.

## 6.2 The L-shaped beam example

To further test the prediction accuracy of the neural networks and to show the problem independent advantage of the proposed ML models, an L-shaped beam example with 4 million fine-resolution meshes is provided as shown in Fig. 7. The corresponding optimized results are shown in Table 2. For the case  $m = 5 \times 5$ , it can be found that even for this model with complex stress states, the relative error of the  $C_{ANN-EMs}$  and  $C_{EMs}$  is only  $1.18 \times 10^{-3}$ , and more interestingly, the optimized value of the structural compliance ( $C_f$ ) is also almost the same as that obtained from direct fine-scale calculation (the relative error is only  $3.92 \times 10^{-4}$ ). All these facts again demonstrate the effectiveness of the proposed ML model. For the case  $m = 10 \times 10$ , the relative error of the  $C_{ANN-EMs}$  and  $C_{EMs}$  is  $3.03 \times 10^{-2}$  and the optimized value of the structural compliance ( $C_f$ ) is 1.12% higher than the value obtained from direct fine-scale calculation, which are all much larger than the corresponding errors of the  $5 \times 5$  case. This can be understood that, the total number of

output values of the neural network for the node-shaped function values of the  $10 \times 10$  case is 972, which is as many as 5 times of the number of the output values of the  $5 \times 5$  case (i.e., 192). Even the loss function of the two neural networks is decreased to the same level, the neural network of  $10 \times 10$  case probably has a larger error in the obtained structural response due to the error accumulation. This will be one of the directions of our subsequent work to improve the prediction accuracy as well as to explore the impact of different types of neural networks for training.

### 6.3 The MBB beam example

In order to demonstrate that the neural network in this paper can optimize any boundary problem with the same type of unit, an MBB beam example (see Fig. 8 for the setting of the problem) is provided. The optimization results, the corresponding solution time associated with classical SIMP method and the proposed ANN-EMsFEM (using the developed ML model for FEA) are compared and shown in Table 3. It also can be observed that the optimized structures obtained by the ANN-EMsFEM method with different numbers of coarse-resolution elements are very similar to the structure obtained by using SIMP method directly. Actually, the very small relative errors (i.e.,  $5.21 \times 10^{-4}$  ( $5 \times 5$ ) and  $8.60 \times 10^{-3}$  ( $10 \times 10$ ), respectively) of the optimized value of the objective function between the SIMP and ANN-EMsFEM method verify the universality of the proposed ML model. Meanwhile, the average time of the proposed ANN-EMsFEM method for one iteration step is only about 30% of that of SIMP method, which indicates the proposed approach does enhance the efficiency of topology optimization effectively.

Finally, in order to demonstrate the ability of the proposed method for dealing with extremely large-scale topology optimization problems, the same problem is also solved by a  $2000 \times 1000$  coarse-resolution mesh with  $10 \times 10 = 100$  fine-resolution elements in each coarse-resolution element by taking the symmetry of the problem into consideration (i.e., 200 million fine-resolution elements in *half* of the design domain) and the optimized structure is shown in Fig. 8. It can be observed that more structural details have been revealed by increasing the design resolution. Table 4 provides the breakdown of the computational times associated with different parts of the solution process for a typical iteration step. In Table. 4, The ANN time is the computational time used for generating the shape functions by ANN to obtain the stiffness matrix of each coarse-resolution element, while the EMsFEM time is the computational time to calculate the fine-resolution

displacement. As the iteration proceeds, the structure becomes clearer. In other words, there are more and more weak and solid coarse-resolution elements in the structure, which also means that there are fewer coarse-resolution elements to be predicted, leading to the reduction of ANN time as shown in Table 4. It can be found that for this extremely large-scale topology optimization problem, the FEA time (the sum of the ANN time and the EMsFEM time) is only about two minutes in a typical iteration step with the use of the proposed PIML-enhanced technique. Actually, for this problem most of the computational time (more than 85%!) is spent on updating the design variables. Furthermore, it is worth noting that this problem cannot be solved with the SIMP method directly using the available hardware for the limitation of computation power.

## 7. Concluding remarks

In the present work, we propose a universal PIML-enhanced topology optimization approach which can solve topological design problems in a significantly effective way. This is achieved by resorting to the extended multi-scale finite element method (EMsFEM) and constructing the corresponding EMsFEM shape functions through an off-line ML scheme, which can establish an implicit mapping between the shape functions and element-wise material densities of a coarse-resolution element. The proposed ML scheme is mechanistic-based and totally problem-independent in the sense that the objects under ML are actually the discretized versions of the Green's functions of the underlying governing partial differential equations (PDEs), which are obviously physics relevant and totally independent on the settings of the topology optimization problems (e.g., boundary conditions, design domain and external loads, etc) to be solved. The proposed ML scheme is also *theoretically* locally-determined which can be understood from the perspective that the functions to be learned are solely determined by the material distribution in the considered coarse-resolution element and irrespective of the material distributions in any other coarse-resolution elements. This is actually a very important advantage for guaranteeing the generalization ability of the proposed scheme, which cannot be shared by existing ML-based topology optimization schemes at least theoretically. Furthermore, unlike in most existing works where the samples used for ML are often collected/generated from a collection of specific topology optimization problems in order to cover a considerably wide range of stress/strain states, it is only necessary to increase the diversity of the samples in the proposed ML scheme by enhancing the

variety of material heterogeneity in the coarse-resolution element, which is obviously stress/strain state-independent. Therefore, our ML scheme has great potential for generalization since once the shape functions are trained successfully, it can be applied to solve any types of topology optimization problems governed by the same PDEs.

Several numerical examples are provided to illustrate the capabilities of the proposed approach. It is found that the PIML-enhanced approach behaves very well under various problem settings. It is worthy to note that the proposed approach can reduce the computational time for FEA by about two orders of magnitude and solve an ultra-large-scale topology optimization problem with 200 million design variables, which cannot be tackled by direct computation through available hardware on a personal workstation.

The present work can be extended along various directions. Firstly, since only the preliminary fully-connected ANN is employed for ML in the present work, it is interesting to explore how to enhance the performance of the proposed approach by introducing state-of-the-art ML architectures (e.g., convolutional neural networks). And this framework can also be directly extended to three-dimensional design problems with the help of 3D EMsFEM. Some attractive results for large-scale three-dimensional topology optimization problems have been obtained and the related work will appear soon. Secondly, although the proposed approach is only examined within the most popular SIMP framework in the present work, it can also be combined with other topology optimization approaches to increase the computational efficiency. For example, as shown in the ultra-large-scale example with 200 million design variables, most of the computational time (actually more than 85%!) is consumed on updating the design variables. Therefore, it is highly expected that, besides the cost of FEA, substantial saving of computation effort on numerical optimization can also be achieved if the proposed approach is carried out under the moving morphable component (MMC)-based topology optimization framework where the number of design variables can be reduced by 1-2 orders of magnitude. Last but not least, since the proposed approach is mechanistic-based and totally problem-independent, in principle, it can be generalized to solve topology optimizations involving various types of objective/constraint functions and multidisciplinary physics since EMsFEM has already shown its great potential for dealing with various types of boundary value problems. Corresponding research results will be reported elsewhere.





## **Acknowledgement**

This work is supported by the National Key Research and Development Plan (2020YFB1709401), the National Natural Science Foundation (11821202, 11732004, 12002077, 12002073), the Fundamental Research Funds for Central Universities (DUT21RC(3)076, DUT20RC(3)020), Doctoral Scientific Research Foundation of Liaoning Province (2021-BS-063) and 111 Project (B14013).

## References

- [1] G.I.N. Rozvany, A critical review of established methods of structural topology optimization, *Struct. Multidiscip. Optim.* 37 (2009) 217–237. <https://doi.org/10.1007/s00158-007-0217-0>.
- [2] J.D. Deaton, R.V. Grandhi, A survey of structural and multidisciplinary continuum topology optimization: post 2000, *Struct. Multidiscip. Optim.* 49 (2014) 1–38. <https://doi.org/10.1007/s00158-013-0956-z>.
- [3] X. Guo, G.-D. Cheng, Recent development in structural design and optimization, *Acta Mech Sin.* 26 (2010) 807–823. <https://doi.org/10.1007/s10409-010-0395-7>.
- [4] O. Sigmund, K. Maute, Topology optimization approaches, *Struct. Multidiscip. Optim.* 48 (2013) 1031–1055. <https://doi.org/10.1007/s00158-013-0978-6>.
- [5] T. Borrvall, J. Petersson, Large-scale topology optimization in 3D using parallel computing, *Comput. Methods Appl. Mech. Engrg.* 190 (2001) 6201–6229. [https://doi.org/10.1016/S0045-7825\(01\)00216-X](https://doi.org/10.1016/S0045-7825(01)00216-X).
- [6] N. Aage, B.S. Lazarov, Parallel framework for topology optimization using the method of moving asymptotes, *Struct. Multidiscip. Optim.* 47 (2013) 493–505. <https://doi.org/10.1007/s00158-012-0869-2>.
- [7] A. Evgrafov, C.J. Rupp, K. Maute, M.L. Dunn, Large-scale parallel topology optimization using a dual-primal substructuring solver, *Struct. Multidiscip. Optim.* 36 (2008) 329–345. <https://doi.org/10.1007/s00158-007-0190-7>.
- [8] N. Aage, E. Andreassen, B.S. Lazarov, O. Sigmund, Giga-voxel computational morphogenesis for structural design, *Nature* 550 (2017) 84–86. <https://doi.org/10.1038/nature23911>.
- [9] H. Rodrigues, J.M. Guedes, M.P. Bendsoe, Hierarchical optimization of material and structure, *Struct. Multidiscip. Optim.* 24 (2002) 1–10. <https://doi.org/10.1007/s00158-002-0209-z>.
- [10] P.G. Coelho, P.R. Fernandes, J.M. Guedes, H.C. Rodrigues, A hierarchical model for concurrent material and topology optimisation of three-dimensional structures, *Struct. Multidiscip. Optim.* 35 (2008) 107–115. <https://doi.org/10.1007/s00158-007-0141-3>.
- [11] L. Liu, J. Yan, G. Cheng, Optimum structure with homogeneous optimum truss-like material, *Comput. Struct.* 86 (2008) 1417–1425. <https://doi.org/10.1016/j.compstruc.2007.04.030>.
- [12] T.H. Nguyen, G.H. Paulino, J. Song, C.H. Le, A computational paradigm for multiresolution

topology optimization (MTOP), *Struct. Multidiscip. Optim.* 41 (2010) 525–539. <https://doi.org/10.1007/s00158-009-0443-8>.

[13] T.H. Nguyen, C.H. Le, J.F. Hajjar, Topology optimization using the p-version of the finite element method, *Struct. Multidiscip. Optim.* 56 (2017) 571–586. <https://doi.org/10.1007/s00158-017-1675-7>.

[14] J.P. Groen, M. Langelaar, O. Sigmund, M. Ruess, Higher-order multi-resolution topology optimization using the finite cell method: Higher-order multi-resolution topology optimization using the finite cell method, *Int. J. Numer. Methods Eng.* 110 (2017) 903–920. <https://doi.org/10.1002/nme.5432>.

[15] C. Liu, Y. Zhu, Z. Sun, D. Li, Z. Du, W. Zhang, X. Guo, An efficient moving morphable component (MMC)-based approach for multi-resolution topology optimization, *Struct. Multidiscip. Optim.* 58 (2018) 2455–2479. <https://doi.org/10.1007/s00158-018-2114-0>.

[16] H. Liu, Y. Wang, H. Zong, M.Y. Wang, Efficient structure topology optimization by using the multiscale finite element method, *Struct. Multidiscip. Optim.* 58 (2018) 1411–1430. <https://doi.org/10.1007/s00158-018-1972-9>.

[17] X. Guo, W. Zhang, W. Zhong, Doing topology optimization explicitly and geometrically—a new Moving Morphable Components based framework, *ASME Trans. J. Appl. Mech.* 81 (2014) 081009. <https://doi.org/10.1115/1.4027609>.

[18] W. Zhang, J. Yuan, J. Zhang, X. Guo, A new topology optimization approach based on Moving Morphable Components (MMC) and the ersatz material model, *Struct. Multidiscip. Optim.* 53 (2016) 1243–1260. <https://doi.org/10.1007/s00158-015-1372-3>.

[19] W. Zhang, J. Chen, X. Zhu, J. Zhou, D. Xue, X. Lei, X. Guo, Explicit three dimensional topology optimization via Moving Morphable Void (MMV) approach, *Comput. Methods Appl. Mech. Engrg.* 322 (2017) 590–614. <https://doi.org/10.1016/j.cma.2017.05.002>.

[20] Z. Du, T. Cui, C. Liu, W. Zhang, Y. Guo, X. Guo, An efficient and easy-to-extend Matlab code of the Moving Morphable Component (MMC) method for three-dimensional topology optimization, *Struct. Multidiscip. Optim.* 65 (2022) 158. <https://doi.org/10.1007/s00158-022-03239-4>.

[21] E. Ulu, R. Zhang, L.B. Kara, A data-driven investigation and estimation of optimal topologies under variable loading configurations, *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* 4

- (2016) 61–72. <https://doi.org/10.1080/21681163.2015.1030775>.
- [22] G.X. Gu, C.-T. Chen, M.J. Buehler, De novo composite design based on machine learning algorithm, *Extreme Mech. Lett.* 18 (2018) 19–28. <https://doi.org/10.1016/j.eml.2017.10.001>.
- [23] X. Lei, C. Liu, Z. Du, W. Zhang, X. Guo, Machine learning-driven real-time topology optimization under Moving Morphable Component-based framework, *ASME Trans. J. Appl. Mech.* 86 (2019) 011004. <https://doi.org/10.1115/1.4041319>.
- [24] Y. Yu, T. Hur, J. Jung, I.G. Jang, Deep learning for determining a near-optimal topological design without any iteration, *Struct. Multidiscip. Optim.* 59 (2019) 787–799. <https://doi.org/10.1007/s00158-018-2101-5>.
- [25] Y. Zhang, B. Peng, X. Zhou, C. Xiang, D. Wang, A deep Convolutional Neural Network for topology optimization with strong generalization ability, *Eng. Optim.* 54 (2021) 973–988. <https://doi.org/10.1080/0305215X.2021.1902998>.
- [26] Z. Nie, T. Lin, H. Jiang, L.B. Kara, TopologyGAN: Topology optimization using generative adversarial networks based on physical fields over the initial domain, *J. Mech. Des.* 143 (2021). 031715. <https://doi.org/10.1115/1.4049533>.
- [27] H. Chi, Y. Zhang, T.L.E. Tang, L. Mirabella, L. Dalloro, L. Song, G.H. Paulino, Universal machine learning for topology optimization, *Comput. Methods Appl. Mech. Engrg.* 375 (2021) 112739. <https://doi.org/10.1016/j.cma.2019.112739>.
- [28] F.V. Senhora, H. Chi, Y. Zhang, L. Mirabella, T.L.E. Tang, G.H. Paulino, Machine learning for topology optimization: Physics-based learning through an independent training strategy, *Comput. Methods Appl. Mech. Engrg.* 398 (2022) 115116. <https://doi.org/10.1016/j.cma.2022.115116>.
- [29] T. Yue, H. Yang, Z. Du, C. Liu, K.I. Elkhodary, S. Tang, X. Guo, A mechanistic-based data-driven approach to accelerate structural topology optimization through finite element convolutional neural network (FE-CNN), (2021). <https://doi.org/10.48550/arXiv.2106.13652>.
- [30] M.Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, *Comput. Methods Appl. Mech. Engrg.* 192 (2003) 227–246. [https://doi.org/10.1016/S0045-7825\(02\)00559-5](https://doi.org/10.1016/S0045-7825(02)00559-5).
- [31] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, *J. Comput. Phys.* 194 (2004) 363–393. <https://doi.org/10.1016/j.jcp.2003.09.032>.

- [32] O.M. Querin, G.P. Steven, Y.M. Xie, Evolutionary structural optimisation (ESO) using a bidirectional algorithm, *Eng. Comput.* 15 (1998) 1031–1048. <https://doi.org/10.1108/02644409810244129>.
- [33] W. Zhang, W. Yang, J. Zhou, D. Li, X. Guo, Structural topology optimization through explicit boundary evolution, *ASME Trans. J. Appl. Mech.* 84 (2017) 011011. <https://doi.org/10.1115/1.4034972>.
- [34] M. Zhou, G.I.N. Rozvany, The COC algorithm, Part II: Topological, geometrical and generalized shape optimization, *Comput. Methods Appl. Mech. Engrg.* 89 (1991) 309–336. [https://doi.org/10.1016/0045-7825\(91\)90046-9](https://doi.org/10.1016/0045-7825(91)90046-9).
- [35] B. Bourdin, Filters in topology optimization, *Int. J. Numer. Methods Eng.* 50 (2001) 2143–2158. <https://doi.org/10.1002/nme.116>.
- [36] E. Andreassen, A. Clausen, M. Schevenels, B.S. Lazarov, O. Sigmund, Efficient topology optimization in MATLAB using 88 lines of code, *Struct. Multidiscip. Optim.* 43 (2011) 1–16. <https://doi.org/10.1007/s00158-010-0594-7>.
- [37] H.W. Zhang, Y. Liu, S. Zhang, J. Tao, J.K. Wu, B.S. Chen, Extended multiscale finite element method: its basis and applications for mechanical analysis of heterogeneous materials, *Comput. Mech.* 53 (2014) 659–685. <https://doi.org/10.1007/s00466-013-0924-x>.
- [38] Y. Efendiev, T.Y. Hou, *Multiscale Finite Element Methods*, Springer New York, New York, NY, 2009. <https://doi.org/10.1007/978-0-387-09496-0>.
- [39] J. Lv, H. Liu, H.W. Zhang, A multiscale co-rotational method for geometrically nonlinear shape morphing of 2D fluid actuated cellular structures, *Mech. Mater.* 79 (2014) 1–14. <https://doi.org/10.1016/j.mechmat.2014.08.004>.
- [40] Y. Zheng, H. Zhang, J. Lv, H. Zhang, An arbitrary multi-node extended multiscale finite element method for thermoelastic problems with polygonal microstructures, *Int. J. Mech. Mater. Des.* 16 (2020) 35–56. <https://doi.org/10.1007/s10999-019-09458-w>.
- [41] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, 378 (2019) 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>.

## Figures

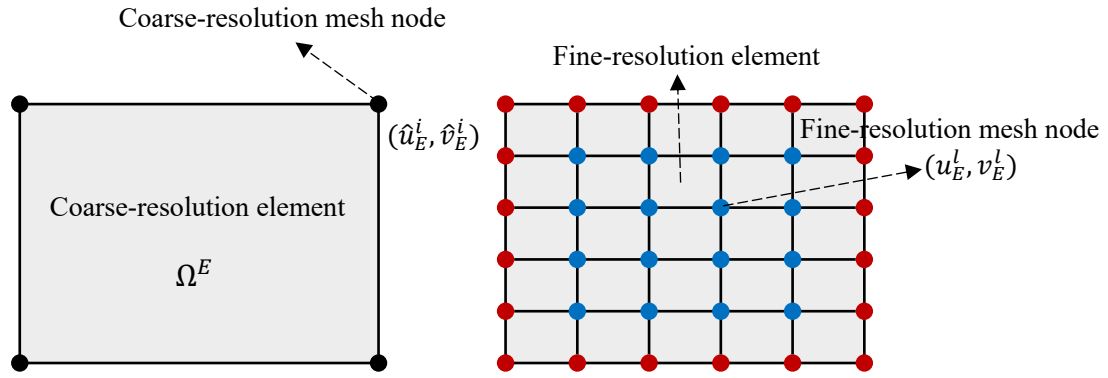


Fig. 1. Construction of the EMsFEM by setting coarse/fine-resolution meshes.

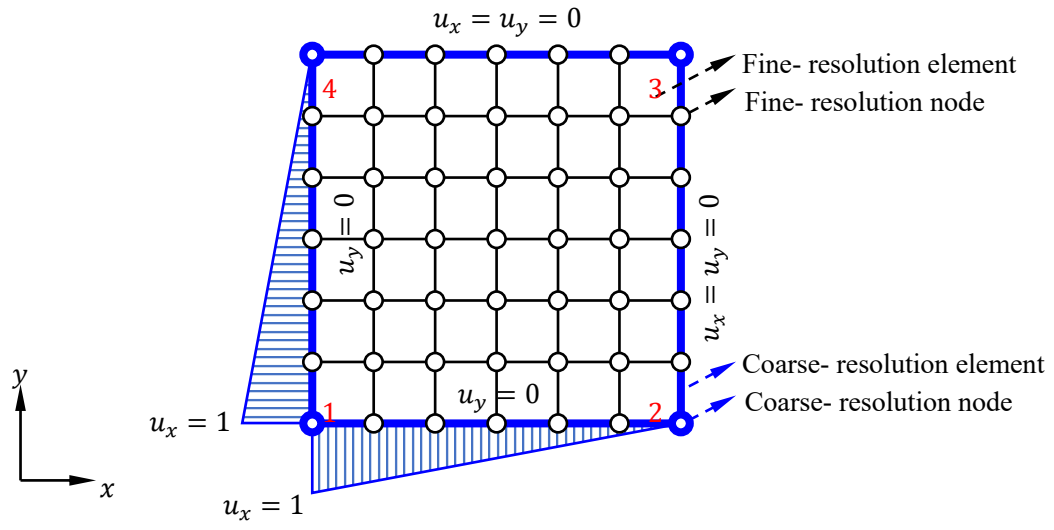


Fig. 2. Constructing EMsFEM shape functions by applying linear boundary condition.



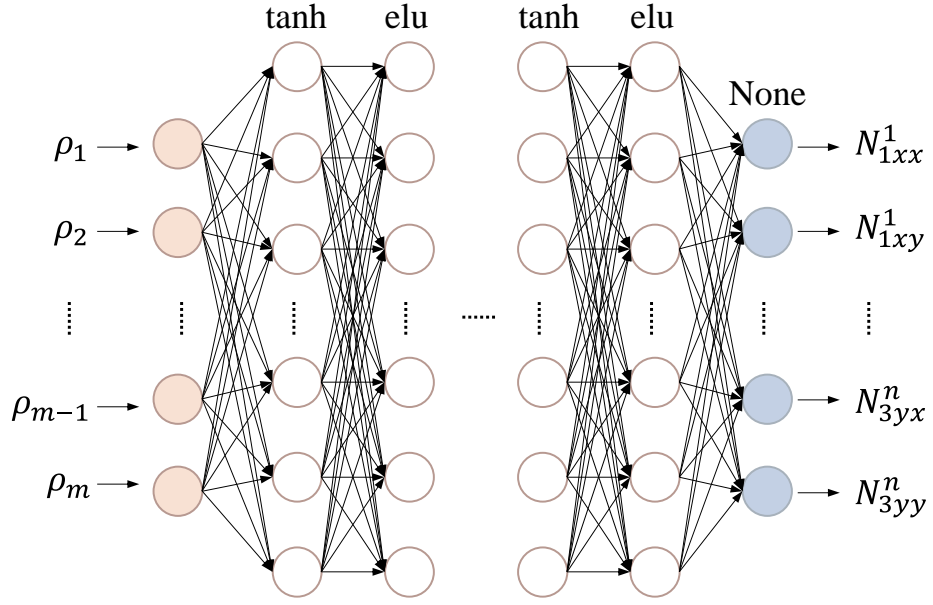


Fig. 3. The ANN used for ML. Input: density values of  $m$  fine-solution elements in a coarse-resolution element. Output: nodal values of independent EMsFEM shape functions.

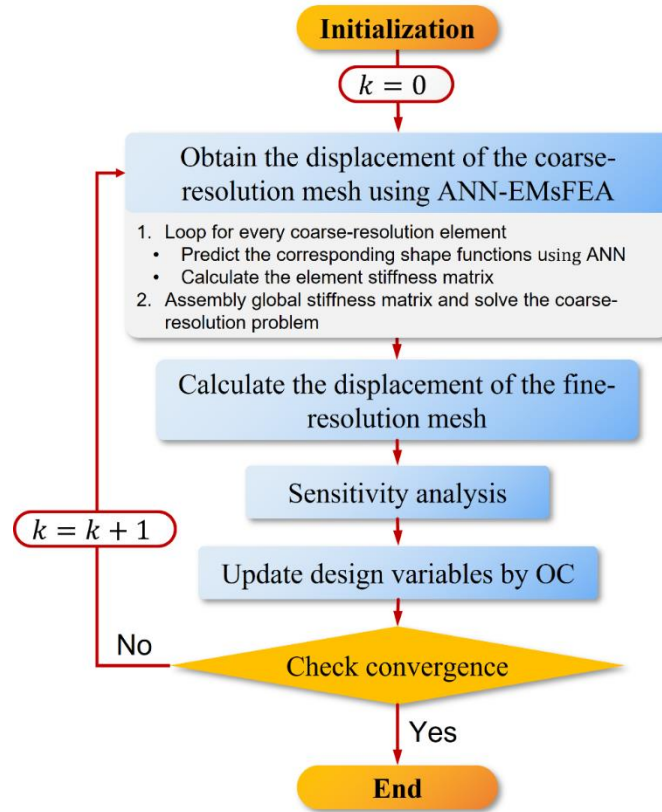


Fig. 4. The proposed PIML-enhanced topology optimization procedure.

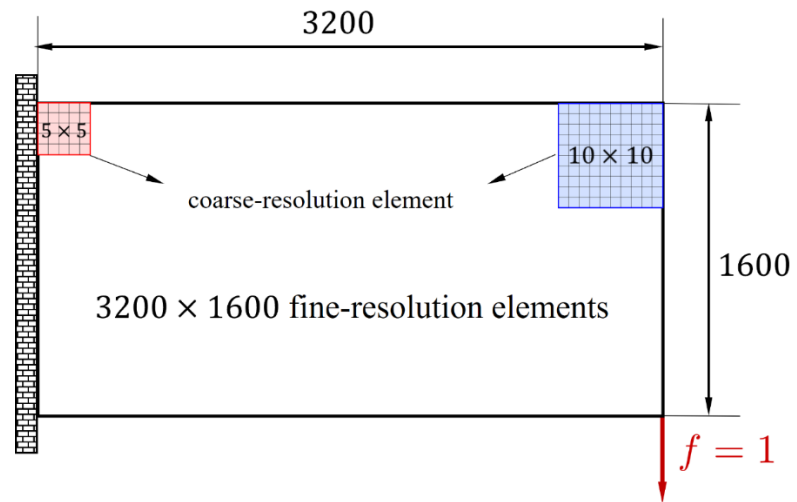


Fig. 5. The problem setting of the short cantilever beam example.

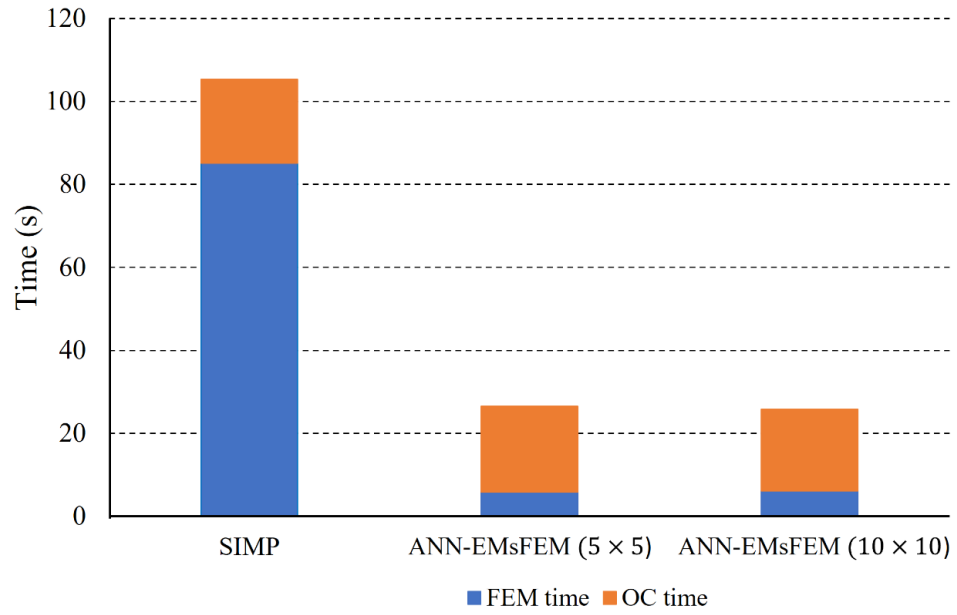


Fig. 6. A comparison of the computational times at a typical iteration step (short cantilever beam example).

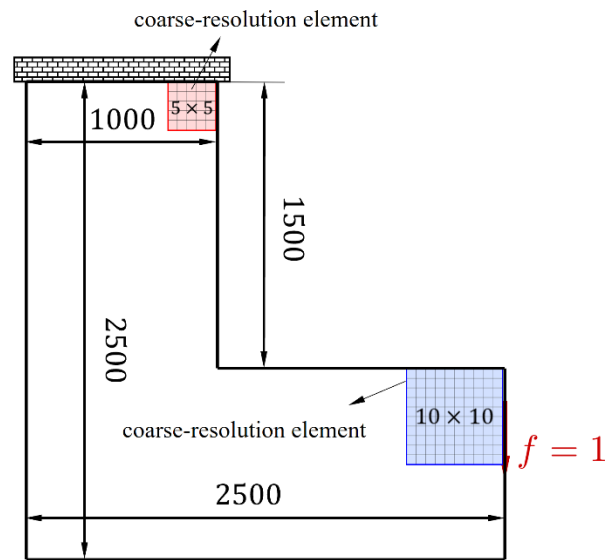


Fig. 7. The problem setting of the L-shaped beam example.

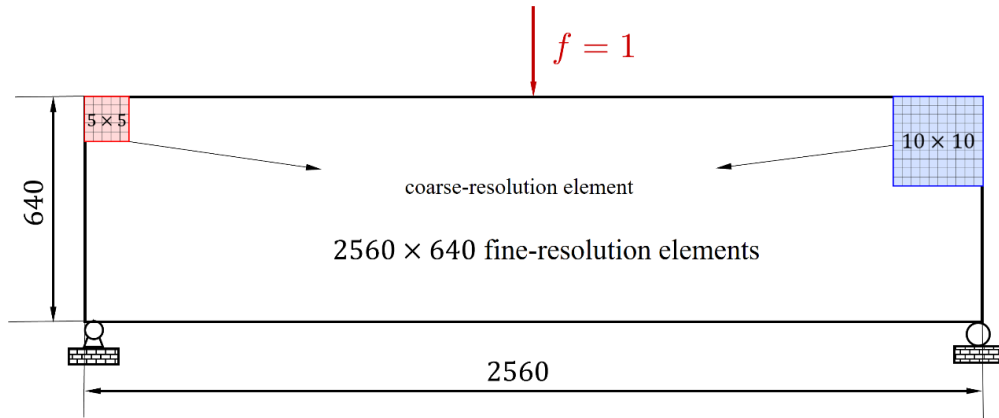


Fig. 8. The problem setting of the MBB beam example.

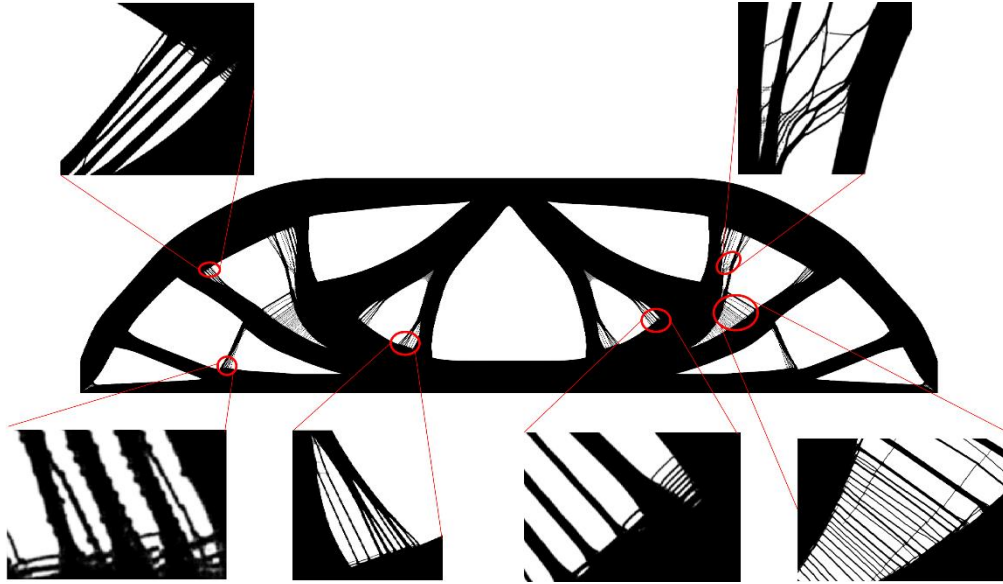


Fig. 9. The optimized structure of the MBB beam with 200 million fine-resolution elements.

## Tables

Table 1. The optimized short cantilever beam structures obtained by different methods, the average time for one iteration step ( $t_{it}$ ), the optimized structural compliances calculated by ANN-EMsFEM ( $C_{ANN-EMs}$ ), EMsFEM ( $C_{EMs}$ ) and direct fine-resolution calculation ( $C_f$ ), respectively.



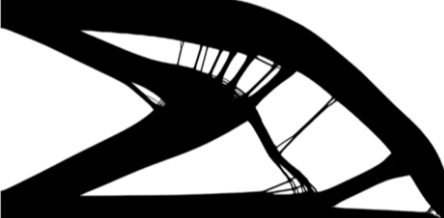
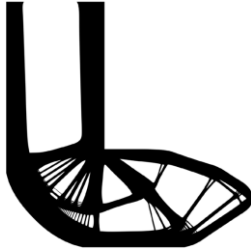
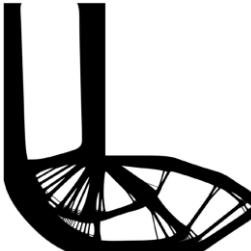
Method	Optimized structure	$t_{it}(s)$	$C_{ANN-EMs}$	$C_{EMs}$	$C_f$
SIMP		105.29	/	/	69.30
PIML- enhanced ( $m = 5 \times 5$ )		26.83	66.43	66.45	69.30
PIML- enhanced ( $m = 10 \times 10$ )		25.51	64.76	65.33	69.56



Table 2. The optimized L-shaped beam structures obtained by different methods, the average time for one iteration step ( $t_{it}$ ), the optimized structural compliances calculated by ANN-EMsFEM ( $C_{ANN-EMs}$ ), EMsFEM ( $C_{EMs}$ ) and direct fine-resolution calculation ( $C_f$ ), respectively.

Method	Optimized structure	$t_{it}(s)$	$C_{ANN-EMs}$	$C_{EMs}$	$C_f$
SIMP		68.45			153.02
PIML-enhanced ( $m = 5 \times 5$ )		20.33	151.99	152.17	153.08
PIML-enhanced ( $m = 10 \times 10$ )		20.34	148.38	153.02	154.74

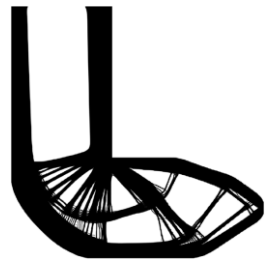
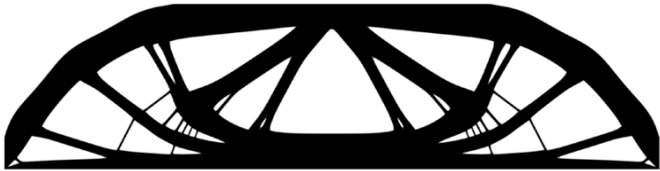
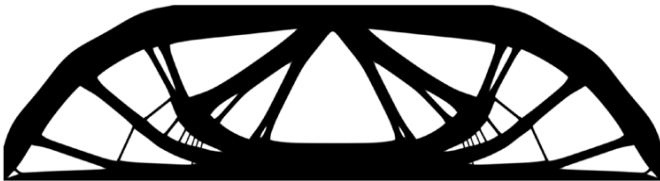
					
--	--	--	--	--	--

Table 3. The optimized MBB structures obtained by different methods, the average time for an iteration step ( $t_{it}$ ), the optimized structural compliances calculated by ANN-EMsFEM ( $C_{ANN-EMs}$ ), EMsFEM ( $C_{EMs}$ ) and direct fine-resolution calculation ( $C_f$ ), respectively.

Method	Optimized structure	$t_{it}(s)$	$C_{ANN-EMs}$	$C_{EMs}$	$C_f$
SIMP		28.05	/	/	38.37
PIML- enhanced ( $m = 5 \times 5$ )		8.76	36.08	36.11	38.39

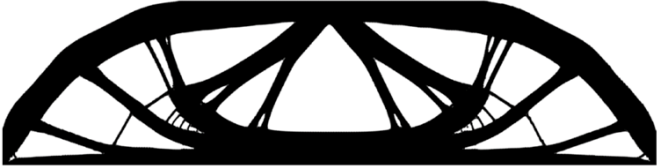
PIML- enhanced ( $m = 10 \times 10$ )		8.91	34.68	35.31	38.70
---	--	------	-------	-------	-------

Table 4. The breakdown of the computational times at some representative iteration steps

(MBB beam example discretized with 200 million fine-resolution elements).

Iteration number	Total time (s)	ANN time (s)	EMsFEM time (s)	OC time (s)
1	1956.93	1108.60	106.11	742.21
10	1293.41	482.42	83.63	727.36
22	970.79	30.44	77.82	862.53
383	926.31	34.29	85.35	806.66