Learning Microstructure-property Mapping Via Label-free 3D Convolutional Neural Network

Liangchao Zhu[sepa], Xuwei Wang[a], Weidong Zhong[a], Ming Li[a,*]

[a]*State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, 310027, China*

Abstract

Predicting the physical property of a class of microstructures is crucial in material design, structural simulation and design, etc. Property prediction may be conducted millions of times in these studies and is better derived instantly for computational efficiency. This issue is addressed in this study via building a mapping from a 3D microstructure to its effective material property, or called structure-property mapping, using a 3D convolutional neural network(CNN). Unlike the direct approach using labeled simulation data, the mapping is based on the physical knowledge of the structure-property relationship determined by its underlying PDE equations. The knowledge is embedded in the loss function of the CNN framework, which is designed and tested under several different formulations to improve its training convergence. Ultimately, the derived structure-property mapping can instantly predict the associated material property for a given microstructure and has far better generalization ability than the data-labeled approach, as demonstrated via numerical examples.

Keywords: Microstructure-property mapping; label-free; 3D convolutional neural network; homogenization

1. Introduction

In recent years, advancements in additive manufacturing have made it possible to fabricate complex structures at the microscale [1, 2]. At the same time, due to the excellent physical characteristics of porous structures, such as high strength-to-weight ratio [3] and enhanced damping [4], the optimal design of a porous structure or its internal microstructure has also attracted more and more attention in various industrial fields. Elaborate design or optimization of porous structures can obtain super-material properties that do not exist in nature. However, it is considerably difficult and computationally challenging to predict the physical behavior of these microstructures well. As the property prediction needs to be performed numerous times when optimizing a cellular structure, it becomes highly time-consuming and may impede the utilization of cellular structures in industrial applications.

The structure-property mapping can be obtained using the conventional prediction approaches, such as n-point correlations, asymptotic homogenization, machine learning, et al., as shown in Fig. 1. These conventional approaches consist of two main types: the analytical approach based on the statistical continuum theories [5], and the numerical approaches based on finite element (FE) methods. The analytical approaches are built on suitable statistical descriptions of the microstructures in the form of n-point spatial correlations, but they are insufficient to quantify the material property fully. Although the numerical approaches can effectively describe the microstructure's physical response under external loadings, their computational costs are significant.

In the past several years, some data-driven approaches (machine learning and deep learning) were also introduced to microstructure analysis to circumvent these difficulties, while all of these methods need to generate ground-truth labels because of the training process, which makes these approaches rather laborious.

*Corresponding author.
Email address: liming@cad.zju.edu.cn (Ming Li)

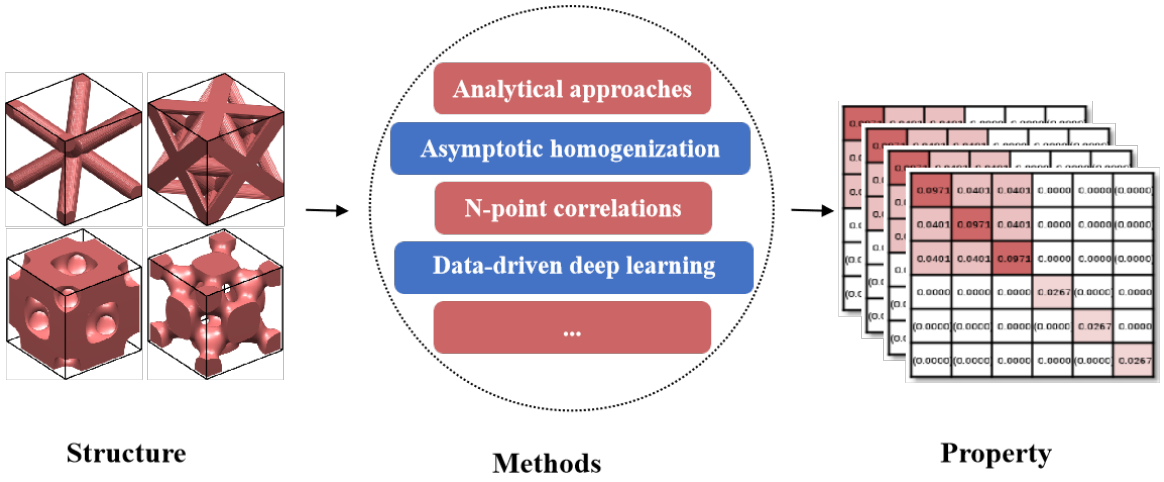


Figure 1: Conventional methods of structure-property prediction(e.g., predicating the homogenized elasticity tensors of the varying-diameter truss structures).

Physics-informed neural networks (PINN) [6] are typical neural networks trained to solve supervised learning tasks while respecting any given laws of physics described by general nonlinear partial differential equations(PDE). However, it has yet to be entirely clear which type of quantitative physical measure is appropriate for a specific learning task. In this study, we aim to resolve this issue by designing and testing various PINN measures, with or without data-driven measures, for the specific task of building a microstructure-property mapping that can instantly predict the effective property for an input microstructure.

We propose an approach to build a structure-property mapping derived based on the physical knowledge of the structure-property relationship determined by its underlying PDE equations. The knowledge is embedded in the loss function of a neural network and is carefully designed and tested using various physics-based discrete energy forms to improve its training convergence. In general, the n-point correlation functions, as described above, would involve selecting the neighbor points and weights to measure their relative priority to express the microstructure property. It has a very close similarity to the convolution involved in convolutional neural networks (CNNs), and we thus derive the mapping using CNNs. Ultimately, the derived structure-property mapping can instantly predict the associated material property for a given microstructure under study and has better generalization ability than the data-labeled approach, as demonstrated via numerical examples.

In summary, the study has the following main contributions:

- A plausible neural network framework to instantly predict structure-property is developed without using any labeled training data;
- The effectiveness of the energy-based loss function is validated by comparing its performance against other physics-driven loss functions, including PDE-based loss and weak-form-based loss, as well as data-driven approaches;
- Theoretical foundations behind the different performances of the different PINN loss functions were explored, in particular, by studying their associated matrix's condition numbers.

The paper is organized as follows. The problem of structure-property mapping and overview of the proposed approach is outlined in Section 3. Details on the label-free neural network for 3D microstructure-property mapping are explained in Section 4. Numerical results of the proposed approach are demonstrated, analyzed, and compared to other approaches in Section 5. Finally, we conclude the paper in Section 6.

2. Related work

Predicting the effective property of a microstructure has long been studied as a central topic in material studies [7, 8, 9, 10, 11, 12], mechanical industrials [13, 14, 15, 16, 17], and solid modeling communities [18].

The analytical approaches [19, 20] have demonstrated low computational costs for lower-order descriptors, but their progress toward higher-order descriptors is hindered by the need to accurately establish the associated Green's function-based kernel [21, 22]. Tan et al. [11] proposed a constrained finite element method for modeling the physical property. Although the numerical approaches usually have high prediction accuracy for highly heterogeneous materials and complex microstructures, the computational costs of these approaches play a dominant role in concurrent topology optimization that involves even millions of such FEA computations [24, 25].

Data-driven meta-models have been developed that have obtained practical usage in effectively combining accuracy and low computational costs [26, 27], relying on Machine Learning (ML) algorithms or data science toolsets. For example, offline computation strategies based on POD/PGD model reduction [28, 13, 29] or interpolation [30, 31] have been proposed via pre-computed elasticity tensors for some sampling data structures. On the other hand, the neural network method has been extended for micro-structure modeling [32]. Deep neural networks (DNNs) are emerging as one of the most promising candidates for their universal and high-order descriptions, facilitating instant structure-property mapping [8, 9, 24, 33, 34, 35]. For example, [24] computes the homogenized elasticity tensors for some data points of the varying-diameter truss structures and obtains a surrogate model offline for online interpolation. In these studies, the DNN training is commonly posed as a supervised learning problem, and a sufficient amount of training data has to be prepared for accurate prediction [8, 9]. These data-driven methods heavily rely on the ground-truth labels when training the model, which is time-consuming. In addition, the data-labeled approach usually tends to overfit the training data and suffers its loss of generalization ability, as will be further observed in our numerical examples in Section 5.

Recently, Yang et al. [36] introduced a novel CNN-based simulation for multi-material optimization, which does not rely on any labeled simulation data but is instead designed based on a physics-informed loss function without using labeled training data. In particular, a novel energy-based loss function is defined based on the principle of minimal potential energy. Such loss function was also used in the *PH-Net* [23] for predicting the homogeneous material properties for microstructures with a parallel hexahedral boundary shape. The newly developed 3D U-Net [37], instead of CNN, was used, where they introduced a shape-material transformation and voxel-material tensor to encode the microstructure type, base material, and boundary shape together as the input of the net.

In this study, the performance of the energy-based loss function is further explored and validated, e.g., the generalization of the models, by comparing it against other physics-driven models, such as PDE-based loss and weak-form loss, and data-driven approaches. In addition, relations between these loss functions and their associated matrix conditioner numbers are explored to lay a theoretical foundation of the PINN for microstructure-property prediction. More typical datasets of microstructures are also introduced in this study in comparison with [23].

3. Problem statement and approach overview

3.1. Problem statement

The problem of structure-property prediction takes a microstructure as input. The voxelization representation of the microstructure is taken due to its ability to produce uniform expression, compatibility with the conventional CNN, and ease of FE analysis.

Suppose ω is a 3D unit cell, divided into $m \times m \times m$ voxels e_{rst}, $1 \leq r, s, t \leq m$. A microstructure is defined on ω and represented by a material distribution parameter $\boldsymbol{\rho} \in \mathbb{R}^{m \times m \times m}$, where $0 \leq \boldsymbol{\rho}_{rst} \leq 1$ or $\boldsymbol{\rho}_{rst} = 0, 1$ represents a continuous or discrete density in voxel e_{rst}, see Fig. 3.

The linear elasticity problem is studied here, and the property of the structure $\boldsymbol{\rho}$ is defined as its effective elasticity tensor $\mathbf{D}^H(\boldsymbol{\rho})$ in the macro-scale, which measures the stress response to its strain tensor.

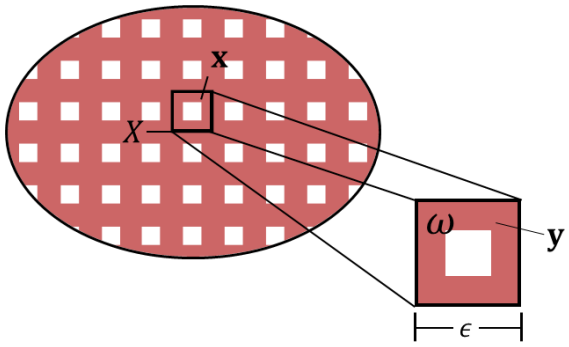
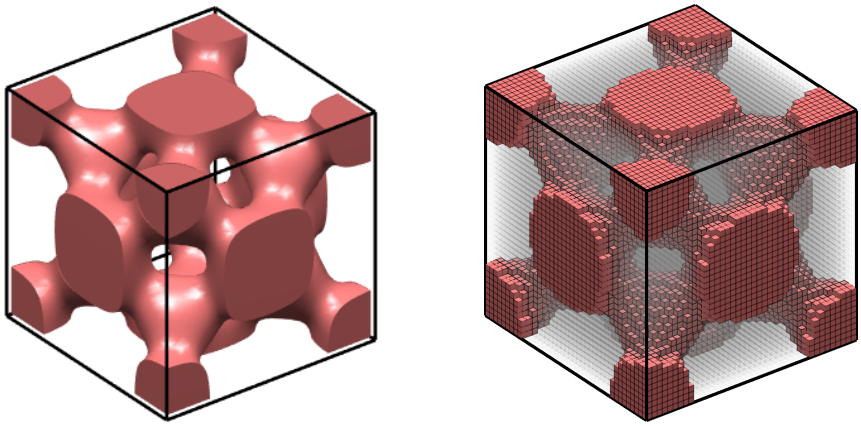


Figure 2: Macroscopic and microscopic scales in the asymptotic homogenization theory.



(a) A microstructure (b) Voxel representation

Figure 3: An input microstructures and its voxel representation.

Specifically, the generalized Hooke's law gives

$$\boldsymbol{\sigma} = \mathbf{D}^H(\boldsymbol{\rho})\boldsymbol{\varepsilon}, \tag{1}$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are the second-order stress and strain tensors, respectively, and $\mathbf{D}^H$ is the fourth-order elasticity tensor. The fourth-order elasticity tensor $\mathbf{D}^H_{ijkl}$ can also be written in a matrix form, as not detailed here.

In this study, we aim to construct the high-dimensional mapping from a set of microstructures $\{\boldsymbol{\rho}\}$ defined on the unit cell ω to the corresponding material properties, i.e., the effective elasticity tensor

$$\{\boldsymbol{\rho}\} \to \{\mathbf{D}^H(\boldsymbol{\rho})\}. \tag{2}$$

The convolutional neural network (CNN) is taken to build the mapping and is stated as follows:

(**Structure-Property Mapping Problem:**P_O) Construct the structure-property mapping via building the CNN model $\mathcal{N}_H$:

$$\hat{\mathbf{D}}^H(\boldsymbol{\rho}) = \mathcal{N}_H(\boldsymbol{\theta}, \boldsymbol{\rho}), \tag{3}$$

for predicting the effective elasticity tensor $\hat{\mathbf{D}}^H(\boldsymbol{\rho})$ for any input $\boldsymbol{\rho} \in p(\boldsymbol{\rho})$, where $p(\boldsymbol{\rho})$ is the associated density postulated by mathematical considerations or learned from data, $\boldsymbol{\theta}$ the parameters of the neural network $\mathcal{N}_H$, and $\hat{\mathbf{D}}$ represents the value of variable $\mathbf{D}$ outputted by the neural network.

In fact, various approaches [8, 9] have been developed to construct the DNN $\mathcal{N}_H$ for Problem P_O using data-driven approaches, which minimizes the loss function

$$\mathcal{L}_{label} = \frac{1}{N}\sum_{s=1}^{N}\left\|\overline{\mathbf{D}}^{H(s)} - \hat{\mathbf{D}}^{H}(\boldsymbol{\rho}^{(s)})\right\|_F^2, \tag{4}$$

where $\overline{\mathbf{D}}^{H(s)}$ is the effective elasticity tensor of s-th microstructure $\boldsymbol{\rho}^{(s)}$ pre-computed using traditional structure-property prediction approaches.

In this study, the model $\mathcal{N}_H(\boldsymbol{\theta}, \boldsymbol{\rho})$ will be generated without any labeled training data. The performance of the data-driven model is to be compared with the proposed label-free model in Section 5.

3.2. Asymptotic homogenization

The proposed label-free structure-property mapping relies on the theory of asymptotic homogenization [38, 39]. It assumes that $\omega(\boldsymbol{\rho})$ is periodically distributed within the space, and its area is relatively small enough as a homogenized point in the macroscopic scale.

Fig. 2 shows macroscopic and microscopic scales in the asymptotic homogenization theory of a porous structure, where X stands for the X-th element and ε means the size of the element. Let y represent the coordinate of a point in the micro-structure ω, and x be the coordinate of that point on the macroscopic scale. Correspondingly, we can obtain the relationship between the microscopic and macroscopic coordinates as:

$$x = X + y, 0 \leq y \leq \epsilon$$

Accordingly, with an asymptotic expansion of the governing equations, the asymptotic homogenization theory approximates the effective elasticity tensor $\mathbf{D}^H(\boldsymbol{\rho})$ for the cell $\boldsymbol{\rho}$ via equalizing their strain energy at the macro and micro scales, as stated below.

Lemma 1. *[40]Let $\mathbf{D}_{ijpq}(\boldsymbol{\rho}, \mathbf{y})$ be the elasticity tensor at every micropoint $\mathbf{y}$ for a cell ω; $\mathbf{D}_{ijpq}(\boldsymbol{\rho}, \mathbf{y})$ takes zero for void material. For each specific pair of $i, j, k, l \in \{1, 2, 3\}$, each component of the homogenized elasticity tensor $\mathbf{D}^H_{ijkl}(\boldsymbol{\rho})$ can be approximated as:*

$$\mathbf{D}^H_{ijkl}(\boldsymbol{\rho}) = \frac{1}{|\omega|}\int_{\omega}(\mathbf{D}_{ijkl}(\boldsymbol{\rho}, \mathbf{y}) - \mathbf{D}_{ijpq}(\boldsymbol{\rho}, \mathbf{y})\frac{\partial \boldsymbol{\mu}^{kl}_p}{\partial \mathbf{y}_q})d\omega, \tag{5}$$

where summation convention is used here for $p, q \in \{1, 2, 3\}$ and henceforth, and the periodic unit strain displacements $\boldsymbol{\mu}^{kl}_p$ are obtained via solving:

$$\frac{\partial}{\partial \mathbf{y}_j}(\mathbf{D}_{ijpq}(\boldsymbol{\rho}, \mathbf{y})\frac{\partial \boldsymbol{\mu}^{kl}_p}{\partial \mathbf{y}_q}) = \frac{\partial \mathbf{D}_{ijkl}(\boldsymbol{\rho}, \mathbf{y})}{\partial \mathbf{y}_j}, \tag{6}$$

or equivalently, solving the weak form

$$\int_{\omega}\mathbf{D}_{ijpq}(\boldsymbol{\rho}, \mathbf{y})\frac{\partial \boldsymbol{\mu}^{kl}_p}{\partial \mathbf{y}_q}\frac{\partial \mathbf{v}_i}{\partial \mathbf{y}_j}d\omega - \int_{\omega}\mathbf{D}_{ijkl}(\boldsymbol{\rho}, \mathbf{y})\frac{\partial \mathbf{v}_i}{\partial \mathbf{y}_j}d\omega = 0, \tag{7}$$

for all periodic test function $\mathbf{v}_i$.

Thus, as shown in Fig. 4, numerically calculating the homogenized tensor $\mathbf{D}^H(\boldsymbol{\rho})$ involves two steps:

1. Compute solutions $\boldsymbol{\mu}^{kl}$ to Eq. (7), via FE analysis or others;
2. Substitute the derived solutions $\boldsymbol{\mu}^{kl}$ into Eq. (5) to obtain $\mathbf{D}^H_{ijkl}(\boldsymbol{\rho})$ $(i, j \in \{1, 2, 3\})$.

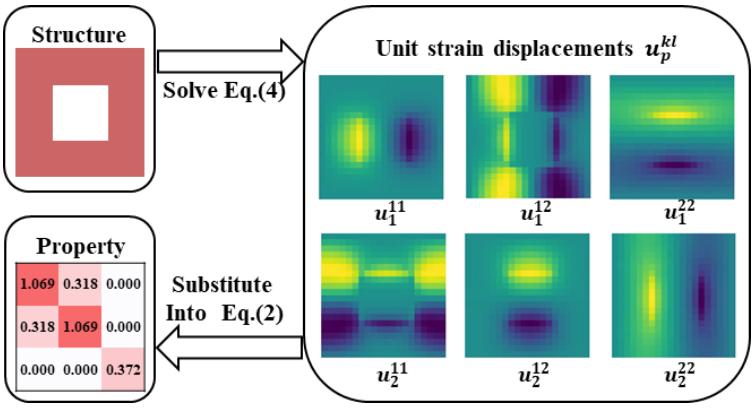


Figure 4: A 2D example of asymptotic homogenization procedure.

Different values of k, l determine different elements of the homogenized elasticity tensor $\mathbf{D}^H$. Considering the symmetry of the elasticity tensor ($\mathbf{D}_{ijkl} = \mathbf{D}_{jikl}, \mathbf{D}_{ijkl} = \mathbf{D}_{ijlk}, \mathbf{D}_{ijkl} = \mathbf{D}_{klij}$), six different combinations of (k, l) values are sufficient to get all the 21 independent components of the homogenized elasticity tensor in 3D case.

Asymptotic homogenization has been applied widely to predict the properties of different microstructures and is applicable to multi-material or high-contrast composite microstructures. However, its high computational costs will be prominent for high-resolution prediction or multi-scale design optimization problems, where the asymptotic homogenization has to be conducted millions of times during the optimization iterations. On the other hand, based on the asymptotic homogenization theory, our novel label-free neural network is to be developed next, which instantly builds the structure-property mapping.

3.3. Overview on label-free CNN approach

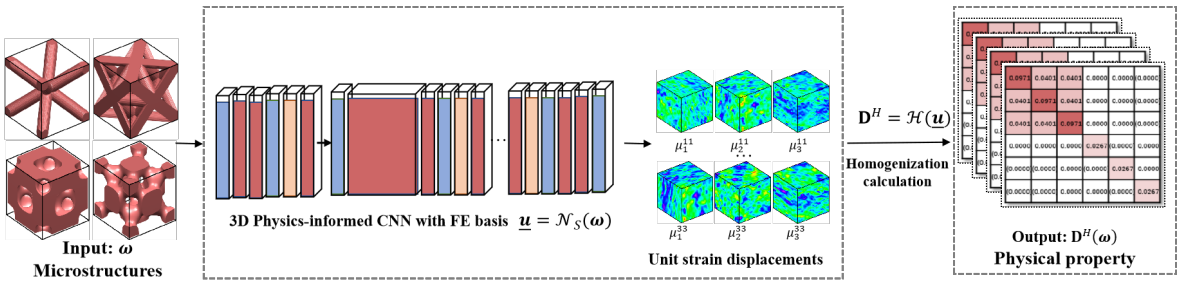


Figure 5: Framework of the label-free 3D microstructure-property mapping over convolution neural network (CNN).

Noticing that directly training a neural network on the elasticity tensor based on Eq. (5) is difficult to formulate, we instead derive the unit strain displacements $\boldsymbol{\mu}^{kl}$ via a label-free neural network based on Eqs. (6) or (7). The elasticity tensor can then be easily computed via Eq. (5). As illustrated in Fig. 5, the procedure consists of the following two main steps.

Firstly, a mapping between the structure $\boldsymbol{\rho}$ and the associated unit strain displacements is derived via constructing the following label-free neural network.

(**Label-free Structure-Property Mapping Problem** : P_S) Construct the simulation neural network model $\mathcal{N}_S$,

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}} = \{\{\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}^{kl}\} = \mathcal{N}_S(\boldsymbol{\theta}, \boldsymbol{\rho}), \ \boldsymbol{\rho} \in p(\boldsymbol{\rho}) \tag{8}$$

to predict the unit strain displacements $\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}} = \{\hat{\boldsymbol{\mu}}^{kl}_{\boldsymbol{\theta}}\}$ defined on the unit cell ω. Note $k, l = 1, 2, 3$, and 3 (2D) or 6 (3D) solution vectors are generated from Lemma 1.

The simulation model $\mathcal{N}_S$ is to be built using a 3D convolutional neural network (CNN) with a novel physics-informed loss function, as detailed in Section 4.

Next the effective elasticity tensor $\mathbf{D}^H(\boldsymbol{\rho})$ is derived using the predicted $\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}} = \{\hat{\boldsymbol{\mu}}^{kl}_{\boldsymbol{\theta}}\}$ based on Eq. (5), which ultimately gives the final mapping,

$$\hat{\mathbf{D}}^H(\boldsymbol{\rho}) = \mathcal{N}_H(\boldsymbol{\theta}, \boldsymbol{\rho}) = \mathcal{H}(\mathcal{N}_S(\boldsymbol{\theta}, \boldsymbol{\rho}), \boldsymbol{\rho}). \tag{9}$$

Both the forward propagation of the first part and the calculation of the second part can be accomplished in interactive time.

4. Label-free CNN on high-dimensional structure-property mapping

The 3D convolutional neural network (CNN) is applied for our microstructure-property mapping instead of the fully connected neural network. One reason is that the number of neural work parameters will explode as the resolution of the microstructure increases. More importantly, in constructing the microstructure-property mapping, the convolutional networks efficiently identify features from a great number of potential features automatically without feature engineering.

The simulation CNN takes the microstructure $\omega(\boldsymbol{\rho})$ as input and outputs the unit strain displacement fields $\hat{\boldsymbol{\mu}}$. To get a well-approximated solution, a CNN model has to be trained by adjusting the weights to minimize the pre-defined loss function. Instead of simply taking the loss function in (4), a physics-informed loss function based on (7) is carefully defined, as detailed below.

4.1. The physics-informed loss function

Let $\{\boldsymbol{\rho}^{(s)}\}_{s=1}^N$ be the training set (without any labels). The training of the neural network defined in **Problem**: P_S in (8) is equivalent to minimize the loss function defined as

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \{\boldsymbol{\rho}^{(s)}\}_{s=1}^N), \tag{10}$$

where

$$\mathcal{L}(\boldsymbol{\theta}, \{\boldsymbol{\rho}^{(s)}\}_{s=1}^N) = \frac{1}{N}\sum_{s=1}^{N} L(\boldsymbol{\rho}^{(s)}, \hat{\boldsymbol{\mu}}^{(s)}_{\boldsymbol{\theta}}) \tag{11}$$

for L being the loss of a single input.

It is observed in this study that different physics-informed forms for L may be defined, which may result in different prediction performances. Different previous studies mainly based on the differential form [41, 42]. In our work, we introduce a novel weak-form-based loss function and an energy-based loss function, which we will explain next.

PDE loss. In most previous studies, the original PDEs, together with their associated boundary conditions, are adopted to define the loss function [41, 42]. In this study, the PDE of the unit strain displacements is given in Eq. (6). It can be regarded as an elasticity problem under the body force, whose component on the i-direction is $\partial \mathbf{D}_{ijkl}(\boldsymbol{\rho})/\partial \mathbf{y}_j$. Accordingly, the following loss function is naturally considered,

$$L_{pde}(\boldsymbol{\rho}, \boldsymbol{\mu}) = \sum_{k,l}\int_{\omega}\left[\frac{\partial}{\partial \mathbf{y}_j}(\mathbf{D}_{ijpq}(\boldsymbol{\rho})\frac{\partial(\boldsymbol{\mu}^{kl}_p)}{\partial \mathbf{y}_q}) - \frac{\partial \mathbf{D}_{ijkl}(\boldsymbol{\rho})}{\partial \mathbf{y}_j}\right]^2 d\omega. \tag{12}$$

However, the body force equals zero within the domain of the same material. For the microstructure with explicit boundaries, each point in the unit cell is either solid or void such that $\partial \mathbf{D}_{ijkl}(\boldsymbol{\rho})/\partial \mathbf{y}_j$ is hard to calculate using numerical approaches. Based on the observations, the loss function defined as the residual norm of the PDE is thus not adopted here.

Weak form loss. Considering the weak form in Eq. (7), one can define the loss function as

$$L_{weak}(\boldsymbol{\rho},\boldsymbol{\mu}) = \sum_{k,l} \sum_{periodic\ \mathbf{v}} \left[\int_\omega \mathbf{D}_{ijpq}(\boldsymbol{\rho}) \frac{\partial \boldsymbol{\mu}_p^{kl}}{\partial \mathbf{y}_q} \frac{\partial \mathbf{v}}{\partial \mathbf{y}_j} d\omega - \int_\omega \mathbf{D}_{ijkl}(\boldsymbol{\rho}) \frac{\partial \mathbf{v}}{\partial \mathbf{y}_j} d\omega \right]^2. \tag{13}$$

Energy function loss. It is also noted that (7) is the first variation of the total potential energy function,

$$\Pi(\boldsymbol{\mu}^{kl}) = \int_\omega \frac{1}{2} \mathbf{D}_{ijpq}(\boldsymbol{\rho},\mathbf{y}) \frac{\partial \boldsymbol{\mu}_p^{kl}}{\partial \mathbf{y}_q} \frac{\partial \boldsymbol{\mu}_i^{kl}}{\partial \mathbf{y}_j} d\omega - \int_\omega \mathbf{D}_{ijkl}(\boldsymbol{\rho},\mathbf{y}) \frac{\partial \boldsymbol{\mu}_i^{kl}}{\partial \mathbf{y}_j} d\omega. \tag{14}$$

Computing solution to Eq. (7) is equivalent to solving the minimization problem,

$$\min_{\boldsymbol{\mu}^{kl}} \Pi(\boldsymbol{\mu}^{kl}). \tag{15}$$

Accordingly, the physics-inform loss function can also be defined in terms of the total potential energy function as follows, bendsoe1988generating

$$L_{eng}(\boldsymbol{\rho},\boldsymbol{\mu}) = \sum_{k,l} \int_\omega \frac{1}{2} \mathbf{D}_{ijpq}(\boldsymbol{\rho}) \frac{\partial \boldsymbol{\mu}_p^{kl}}{\partial \mathbf{y}_q} \frac{\partial \boldsymbol{\mu}_i^{kl}}{\partial \mathbf{y}_j} d\omega - \int_\omega \mathbf{D}_{ijkl}(\boldsymbol{\rho}) \frac{\partial \boldsymbol{\mu}_i^{kl}}{\partial \mathbf{y}_j} d\omega. \tag{16}$$

Performance of the CNN under different loss functions (14) or (16) will be tested and compared in Section 5.

4.2. Numerics combing finite element basis

Numerically implementing the loss functions Eqs. (13) and (16) in a neural network faces the following challenges. Firstly, innumerable options exist for $\mathbf{v}$ in Eq. (13) in the test function space and need to be carefully selected. Secondly, the gradient computation faces an accuracy problem because the displacement field has no analytical representation in the neural network. The gradient has to be approximated using numerical approaches. In fully connected neural network simulation approaches where the spatial coordinates are taken as the input, the gradient is computed using automatic differentiation of neural network [41, 42]. The CNN simulation approach [43] uses Sobel filters and correction matrices to approximate the gradients. Both approaches do not apply directly to the physical loss under study.

To resolve the above issues, the finite element basis is introduced here such that the periodic function $\mathbf{v}$ can be implemented as the FE basis, and the gradients of $\boldsymbol{\mu}_i^{kl}$ can be derived via the differentiation of the basis functions as we now explain.

Following the notations in Section 3, suppose the unit cell ω is discretized into $M = m^3$ finite elements of unit cubes. Following ideas in FE analysis, the solution $\boldsymbol{\mu}^{kl} = \{\boldsymbol{\mu}_e^{kl}\}$ can be taken as the assembly sum of those on each FE element e, where we specifically have

$$\boldsymbol{\mu}_e^{kl}(\mathbf{y}) = \sum_{n=1}^{n_b} N_n^e(\mathbf{y}) u_{e,n}^{kl}, \tag{17}$$

where $N_n^e(\mathbf{y})$ is the basis function corresponding to node n in element e, $u_{e,n}^{kl}$ is the unknown value at node n, and n_b is the number of nodes of element e.

This way, the gradient of $\boldsymbol{\mu}^{kl}$ can be computed via the gradients of the basis functions as

$$\frac{\partial \boldsymbol{\mu}_e^{kl}(\mathbf{y})}{\partial y_j} = \sum_{n=1}^{n_b} \frac{\partial N_n^e(\mathbf{y})}{\partial y_j} u_{e,n}^{kl}. \tag{18}$$

Following a classical finite element assembly process, we have the stiffness matrix and force vector given as follows,

$$\begin{aligned} \mathbf{K}(\boldsymbol{\rho}) &= \sum_{assemble} \int_e \mathbf{B}_e^T \mathbf{D}_e(\boldsymbol{\rho}) \mathbf{B}_e \ d\mathbf{y}, \\ \mathbf{f}^{kl}(\boldsymbol{\rho}) &= \sum_{assemble} \int_e \mathbf{B}_e^T \mathbf{D}_e(\boldsymbol{\rho}) \boldsymbol{\varepsilon}_0^{kl} \ d\mathbf{y}, \end{aligned} \tag{19}$$

where $\mathbf{D}_e$ is the elasticity tensor of element e, $\mathbf{B}_e$ is the strain-displacement matrix, and $\boldsymbol{\varepsilon}_0^{kl}$ corresponds to the macroscopic unit strain fields taken the following form for the 3D problem,

$$\begin{aligned}\boldsymbol{\varepsilon}_0^{kl} = (1,0,0,0,0,0)^T, \quad (0,1,0,0,0,0)^T, \quad (0,0,1,0,0,0)^T,\\ (0,0,0,1,0,0)^T, \quad (0,0,0,0,1,0)^T, \quad (0,0,0,0,0,1)^T.\end{aligned} \tag{20}$$

Accordingly, noticing that $\boldsymbol{\mu}$ contains $\boldsymbol{\mu}^{kl}$ for different $k,l = 1,2,3$, the discrete forms of Eq. (13) and (16) are derived as

$$L_{weak}(\boldsymbol{\rho},\boldsymbol{\mu}) = \sum_{k,l} \left\|\mathbf{K}(\boldsymbol{\rho})\boldsymbol{\mu}^{kl} - \mathbf{f}^{kl}(\boldsymbol{\rho})\right\|_2^2, \tag{21}$$

$$L_{eng}(\boldsymbol{\rho},\boldsymbol{\mu}) = \sum_{k,l} \left[\frac{1}{2}(\boldsymbol{\mu}^{kl})^T\mathbf{K}(\boldsymbol{\rho})\boldsymbol{\mu}^{kl} - (\boldsymbol{\mu}^{kl})^T\mathbf{f}^{kl}(\boldsymbol{\rho})\right]. \tag{22}$$

The assembly of the global stiffness matrix $\mathbf{K}(\boldsymbol{\rho})$ can be further avoided via calculation of the element-wise loss for parallel implementation,

$$\mathbf{K}(\boldsymbol{\rho})\boldsymbol{\mu}^{kl} = \sum_{assemble} \mathbf{K}^e(\boldsymbol{\rho}_e)\boldsymbol{\mu}_e^{kl}, \tag{23}$$

where $\mathbf{K}^e(\boldsymbol{\rho}_e) = \int_e \mathbf{B}_e^T\mathbf{D}_e(\boldsymbol{\rho}_e)\mathbf{B}_e\, d\mathbf{y}$ is the local stiffness matrix and $\boldsymbol{\mu}_e^{kl}$ is the displacement vector of element e.

The energy functional loss can also be easily computed as the sum of the element-wise values as

$$\begin{aligned}&\frac{1}{2}(\boldsymbol{\mu}^{kl})^T\mathbf{K}(\boldsymbol{\rho})\boldsymbol{\mu}^{kl} - (\boldsymbol{\mu}^{kl})^T\mathbf{f}^{kl}(\boldsymbol{\rho}) = \\ &\qquad \sum_{e=1}^{M}[\frac{1}{2}(\boldsymbol{\mu}_e^{kl})^T\mathbf{K}^e(\boldsymbol{\rho}_e)\boldsymbol{\mu}_e^{kl} - (\boldsymbol{\mu}_e^{kl})^T\mathbf{f}_e^{kl}(\boldsymbol{\rho})].\end{aligned} \tag{24}$$

Thus, solving a large-scale, sparse linear equation is avoided. In addition, as all the FE elements are of the same shape, no element integration is involved in this process, and the above loss computations are very efficient and different from FE analysis even though the FE basis is used. Moreover, the above computations are naturally conducted in parallel with GPUs in machine learning platforms like PyTorch, TensorFlow, et al.

Effective elasticity tensor computation with FE basis. Once the discretized unit strain displacements $\{\boldsymbol{\mu}_e^{kl}\}$ are obtained, the matrix form of the effective elasticity tensor can be computed as follows [44],

$$\mathbf{D}^H(\boldsymbol{\rho}) = \frac{1}{|\omega|}\sum_{e=1}^{M}\int_e (\mathbf{I} - \mathbf{B}_e\boldsymbol{\mu}_e)^T\mathbf{D}_e(\boldsymbol{\rho})(\mathbf{I} - \mathbf{B}_e\boldsymbol{\mu}_e)\, d\omega \tag{25}$$

where $\mathbf{I}$ is the identity matrix and $\boldsymbol{\mu}_e$ contains the three (2D) or six (3D) displacement vectors $\boldsymbol{\mu}_e^{kl}$ of element e.

4.3. Comparison between the physics-informed loss functions

The convergence behavior of the two different loss functions defined via Eqs. (13) and (16) is further analyzed below.

We have the gradient of the loss function with respect to the neural network parameter $\boldsymbol{\theta}$ as follows,

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial L}{\partial\hat{\boldsymbol{\mu}}^{kl}}\frac{\partial\hat{\boldsymbol{\mu}}^{kl}}{\partial\boldsymbol{\theta}}, \tag{26}$$

where for the weak form (13)

$$\frac{\partial L}{\partial \hat{\boldsymbol{\mu}}^{kl}} = \frac{\partial L_{weak}}{\partial \hat{\boldsymbol{\mu}}^{kl}} = 2\mathbf{K}^T\mathbf{K}\hat{\boldsymbol{\mu}}^{kl} - 2\mathbf{K}^T\mathbf{f}^{kl}, \tag{27}$$

or for the energy form (16)

$$\frac{\partial L}{\partial \hat{\boldsymbol{\mu}}^{kl}} = \frac{\partial L_{eng}}{\partial \hat{\boldsymbol{\mu}}^{kl}} = \mathbf{K}\hat{\boldsymbol{\mu}}^{kl} - \mathbf{f}^{kl}. \tag{28}$$

Solving the minimization problem is equivalent to solving the following linear equations,

$$\mathbf{K}\hat{\boldsymbol{\mu}}^{kl} = \mathbf{f}^{kl}, \tag{29}$$

or

$$(\mathbf{K}^T\mathbf{K})\hat{\boldsymbol{\mu}}^{kl} = \mathbf{K}^T\mathbf{f}^{kl}. \tag{30}$$

Noticing that the global stiffness matrix $\mathbf{K}$ is very sparse for the boundary value problem, the following property on the associated condition number is observed,

$$\text{cond}(\mathbf{K}^T(\boldsymbol{\rho})\mathbf{K}(\boldsymbol{\rho})) \gg \text{cond}(\mathbf{K}(\boldsymbol{\rho})), \tag{31}$$

where $\text{cond}(A)$ represents the condition number of a matrix A.

Correspondingly, Eq. (30) is generally expected to have much slower convergence behavior than Eq. (29), and we can expect that the neural network would perform better using L_{eng} than using L_{weak}. Results of our numerical examples in Section 5 are also consistent with the above theoretical analysis.

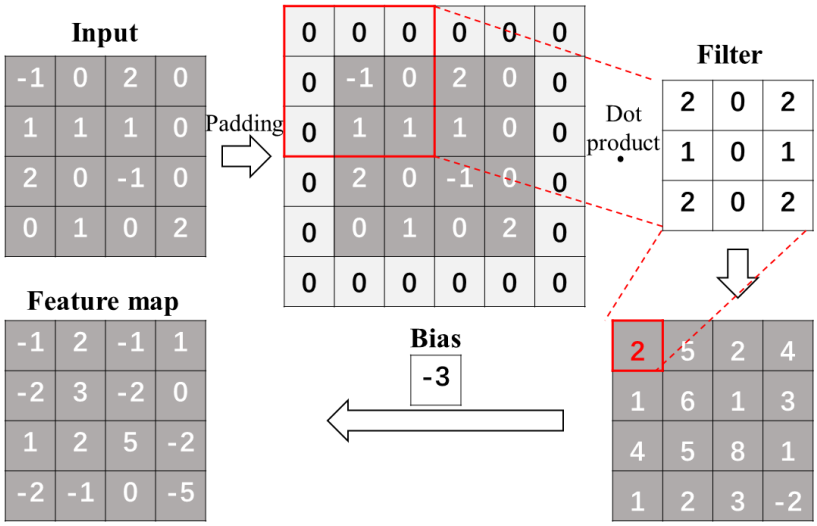


Figure 6: A example of 2D convolution computation with a 3×3 filter.

4.4. 3D convolutional neural network architecture

The convolutional neural network [45] is a class of deep learning neural networks containing convolution calculations, and is a powerful tool for solving problems in computer vision [46], natural language processing [47], etc. The convolutional layer is the core of convolutional neural networks. Fig. 6 shows a typical 2D convolution operation with a 3×3 filter. As a specialized kind of linear operation, convolution with each filter includes computing the dot product between the filter weights and an equal-sized subdomain of the input, putting together the results of the dot products, and adding the bias. The convolutional neural network extracts higher-order microstructure information as filters from the voxel microstructure information.

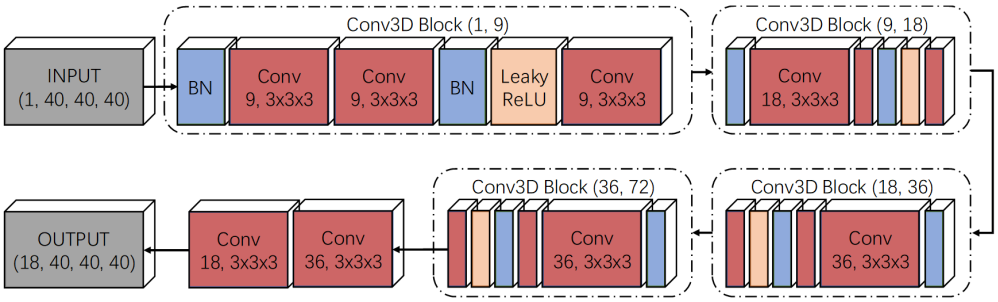


Figure 7: Architecture of our 3D CNN, which mainly consists of four Conv3D blocks and two 3D convolution layers.

The architecture of our 3D convolutional neural network is illustrated in Fig. 7. The 3D convolutional neural network mainly consists of four Conv3D blocks and two 3D convolutional layers. Different Conv3d blocks have a similar structure but different input and output channels. The numbers of the input and output channels of the 4 Conv3D blocks are (1,9), (9,18), (18,36), and (36,72), respectively. In detail, each Conv3D block is composed in sequence of the batch normalization layer, double $3 \times 3 \times 3$ convolutional layers, the batch normalization layer, the Leaky ReLU activation function with slope $a = 0.01$ and another $3 \times 3 \times 3$ convolutional layer. Zero-padding is used in all the convolutional layers such that the data shape of each channel is kept as $40 \times 40 \times 40$. In image processing, using multiple convolutional kernels of small size performs better than a single big-size convolutional kernel [48]. Our Conv3D block is designed based on this. The network has 14 convolutional layers with a total of 437642 parameters.

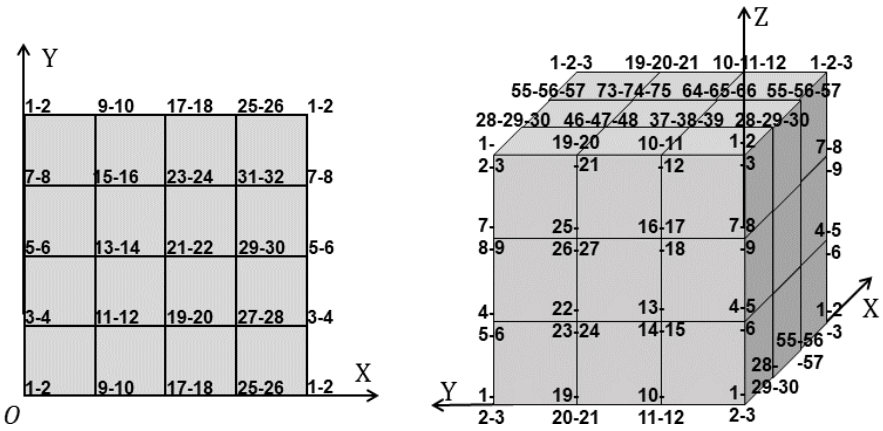


Figure 8: Degrees of freedom of 2D and 3D periodic FE elements.

Our CNN model takes as input the voxelized representation of a microstructure $\boldsymbol{\rho}$ and outputs the unit strain displacement fields $\boldsymbol{\mu}^{kl}$. Corresponding to the FE basis introduced for loss computation, our simulation neural network outputs the nodes' displacements. The $m \times m \times m$ voxel mesh has totally $(m+1)^3$ nodes. Each node corresponds to 3 degrees of freedom as the X-, Y-, and Z-direction displacements. However, the degrees of freedom on the boundaries of the unit cell are not independent because of the periodic boundary conditions. The square finite element mesh for the 2D unit cell and the cubic finite element mesh for the 3D case with periodic boundary conditions are illustrated in Fig. 8. In our case, for each input microstructure, the displacement fields $\boldsymbol{\mu}^{kl}$ for six different $\boldsymbol{\varepsilon}_0^{kl}$ are to be output, and each $\boldsymbol{\mu}^{kl}$ includes $m \times m \times m \times 3$ degrees of freedom. In summary, the output for each input microstructure $\omega(\boldsymbol{\rho})$ is 6×3 voxelization field of size $m \times m \times m$.

We compared our CNN model with PH-Net [23] in Table 1. From the comparisons in the table, we can see that the total parameters of PH-Net are approximately 14.3 times greater than ours, implying

higher computational costs under similar circumstances. It is worth noting that our approach addresses their deficiency in comparing loss functions as well. The utilization of the FE basis notably improves the computational efficiency of gradient and loss function computations.

Table 1: Comparison between PH-Net and our model.

	FE basis	**Physical loss function comparison**	**Data-driven loss function comparison**	**Total parameters**
PH-Net	×	×	×	6.26M
ours	√	√	√	0.0438M

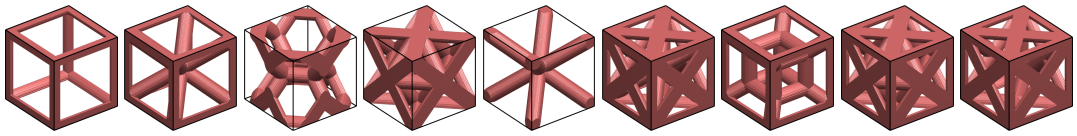
5. Numerical examples

5.1. Settings

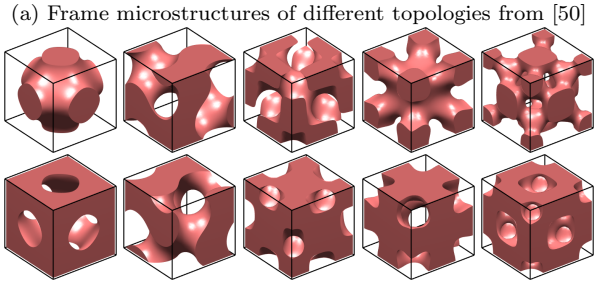
Platform and general settings. Our 3D microstructure-property mapping neural network model, as well as the other neural networks for comparison described below, have been implemented using PyTorch. A Python code of 3D numerical homogenization is implemented by our own based on the 2D one [49]. The homogenized results provide the reference effective elasticity tensors or are taken as labels for other data-driven neural networks. In this study, we aim to test and compare our method with different physics-informed loss functions (e.g., the weak form residual loss) and the data-driven approaches. Consequently, we need to control nonsignificant variables and set the solid materials with fixed values. The resolution of all unit cells is set as $40 \times 40 \times 40$, considering the computational budget of the 3D numerical homogenization computation. Although different resolutions may impact some test indices, such as training time, prediction time, and predictive accuracy, these influences are not the focus of the present study. The solid materials have Young's modulus $E = 1$ and the Poisson's ratio $\nu = 0.3$, which are the same as PH-Net [23]. The proposed method can be used for all kinds of materials, whether they can be manufactured or not at present. Validating the properties of the material through physical experiments is another critical topic in future work. All the experiments run on a PC of Ubuntu 16.04 with a 5GHz Intel core i9 CPU, 32GB RAM, and a single NVIDIA TITAN Xp GPU.

Datasets (microstructures). Only the input samples are needed to train our 3D microstructure-property mapping neural network model. In all the examples, the density of a void or solid element is set to $\rho = 10^{-9}$ or 1, and the value of the light-colored element varies within this range. The minimum density, $\rho_{\min} = 10^{-9}$, is established to prevent singularity in the global stiffness matrices during numerical homogenization. To train or validate the neural networks, voxelization sampling was conducted on the four types of microstructures, as depicted in Fig. 9.

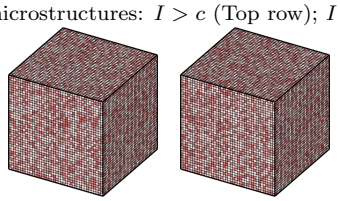
Dataset 1 (Frame) Widely used in topology optimization and additive manufacturing [24, 51]. In our experiment, 598 frame microstructures of 9 different topologies [50] are generated. For each topology, tens of microstructures are generated with different strut radii.
Dataset 2 (TPMS) Triply periodic minimal surface defined by an implicit function $I(x, y, z) = c$, and $I > c$ or $I \leq c$ each is regarded as the TPMS element as listed in Table 2. Widely used in porous structure modeling in 3D printing and tissue engineering fields [52, 53]. Different values of the shape parameters c generate the sampling microstructures while keeping each one-piece. A total number of 565 TPMS cells are included in this dataset.
Dataset 3 (LHS) Microstructures of random densities generated using the Latin hypercube Sampling (LHS) method [35, 54]. For a unit cell of resolution $m \times m \times m$, we sampled 8192 values in the m^3-dimensional space, resulting in microstructures consisting of random gray-scale voxels, i.e., $\boldsymbol{\rho}_{LHS} = \{\rho_{ijk}^{LHS}\}$, $\rho_{\min} \leq \rho_{ijk}^{LHS} \leq 1$.
Dataset 4 (P-LHS) Microstructures of penalized random densities for **Dataset 3 (LHS)**, or $\boldsymbol{\rho}^{(i)} = (\boldsymbol{\rho}_{LHS}^{(i)})^{d_i}$, $d_i \sim |\mathbf{N}(0, 1.2^2)|$ for the i-th microstructure sampled by LHS. The proposed label-free structure-property mapping relies on the theory of asymptotic homogenization which is based on the periodic boundary



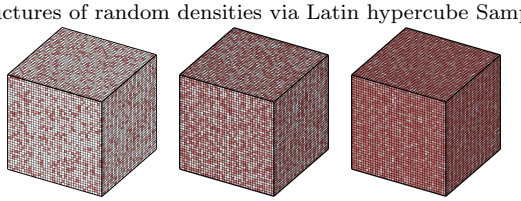
(a) Frame microstructures of different topologies from [50]



(b) TPMS-based microstructures: $I > c$ (Top row); $I \leq c$ (Bottom row);



(c) Microstructures of random densities via Latin hypercube Sampling (LHS)



(d) Microstructures of penalized LHS random densities (P-LHS)

Figure 9: Microstructures for training or validating the performance of the 3D homogenization surrogate model.

conditions, so both the microstructures in LHS and P-LHS are of periodic boundaries. While the volume faction of the LHS microstructure mostly falls around 50%, the volume fractions of the penalized microstructures vary from 16.43% to 99.97%, whose distribution is shown in Fig. 10. This dataset also includes 8192 microstructures.

We compared our datasets with those of PH-Net, as shown in Table 3. Our datasets have four types of microstructures, while PH-Net has only three. Furthermore, functions of five typical TPMS are defined in our Dataset 2, but PH-Net chooses TPMS as the microstructure with uniform samples in the volume fractions ranging from 2% to 33%. In contrast, the volume fraction of our dataset covers a wider range, varying from 16.43% to 99.97%. Overall, our datasets contain a greater variety of microstructures and a broader volume fraction in contrast with PH-Net.

Accuracy metrics. The predicted elasticity tensor and the one from numerical homogenization, or $\hat{\mathbf{D}}^H$ and $\overline{\mathbf{D}}^H$, are compared to demonstrate the accuracy of the approaches. The accuracy is measured by relative errors, calculated as follows for the (α, β)-th component of the elasticity tensor,

$$\delta_{\alpha\beta} = \begin{cases} \frac{\hat{\mathbf{D}}^H_{\alpha\beta} - \overline{\mathbf{D}}^H_{\alpha\beta}}{\overline{\mathbf{D}}^H_{\alpha\beta}}, & \text{if } |\overline{\mathbf{D}}^H_{\alpha\beta}| \geq \varepsilon, \\ 0, & \text{if } |\overline{\mathbf{D}}^H_{\alpha\beta}| < \varepsilon, \end{cases} \tag{32}$$

for $\alpha, \beta \in \{1, 2, \cdots, 6\}$ and ε being a small positive value. The mean relative error for a microstructure is defined as

$$\boldsymbol{\Delta}_m = \frac{1}{36} \sum_{\alpha=1}^{6} \sum_{\beta=1}^{6} \delta_{\alpha\beta} \tag{33}$$

Table 2: Functions of 5 typical TPMS. $X = 2\pi x - \pi$, $Y = 2\pi y - \pi$, $Z = 2\pi z - \pi$.

Type	**Function** $I(x, y, z)$
P	$\cos X + \cos Y + \cos Z = c$
G	$\sin X \cos Y + \sin Z \cos X + \sin Y \cos Z = c$
D	$\cos X \cos Y \cos Z - \sin X \sin Y \sin Z = c$
$I\text{-}WP$	$2(\cos X \cos Y + \cos Y \cos Z + \cos Z \cos X)$ $-(\cos 2X + \cos 2Y + \cos 2Z) = c$
$F\text{-}RD$	$-(\cos 2X \cos 2Y + \cos 2X \cos 2Z + \cos 2Y \cos 2Z)$ $+4 \cos X \cos Y \cos Z = c$

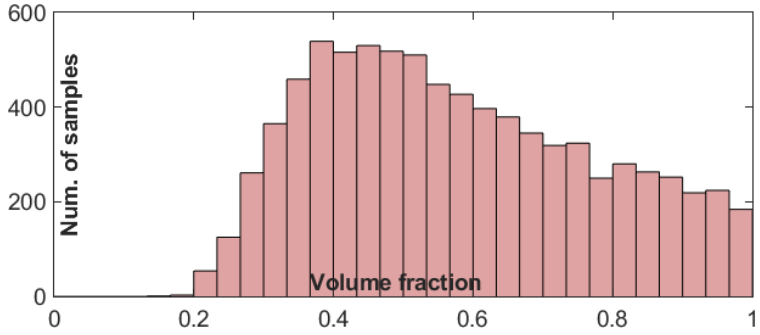


Figure 10: Distribution of volume fractions of microstructures in **Dataset 4**.

and the mean relative error for a dataset (training or validation set) is defined as

$$\boldsymbol{\Delta}_S = \frac{1}{N_S} \sum_{i=1}^{N_S} \boldsymbol{\Delta}_m^i, \tag{34}$$

where N_S is the number of microstructures in the training or validation set.

Trained models. Five neural network models are trained to demonstrate the approach's performance and its comparison with other approaches. **Models 1, 2, 3, 4** are independent models while **Model 1-A** is the fine-tuned from **Model 1**.

The common features of **Models 1, 2, 3, 4** are firstly introduced. The 3D CNN, as shown in Fig. 7, is adopted for **Models 1, 1-A, 2, 3** with the same hyperparameters and **Model 4** uses a slightly different one shown in Fig. 11 and to be explained. The frame microstructures are considerably common, and they

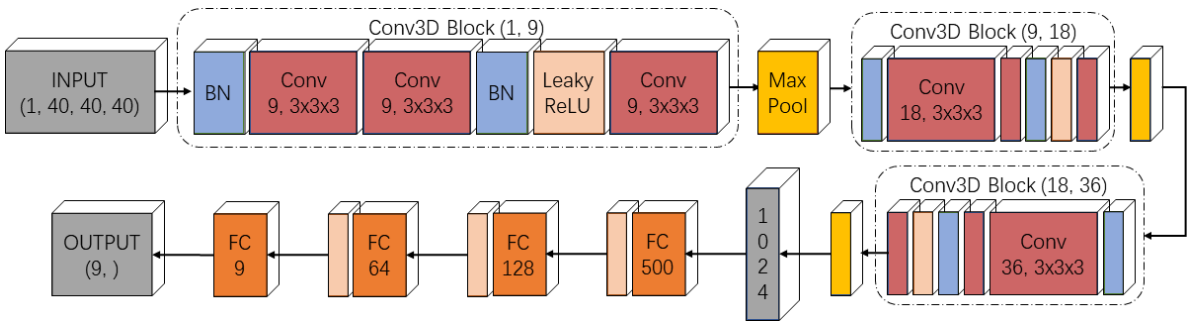


Figure 11: Architecture of the neural network for the elasticity tensor data-driven microstructure-property mapping model. The neural network mainly consists of three Conv3D Blocks, each followed by a max pooling layer and four fully connected layers.

Table 3: Comparison of microstructure datasets between PH-Net and our model.

	Frame/Truss	**Shell**	**LHS**	**P-LHS**	**Function numbers of typical TPMS**	**Range of volume fraction**
PH-Net	√	√	×	×	1	2%-33%
ours	√	×	√	√	5	16.43%-99.97%

have drawn the attention of many researchers. In our work, we selected frame microstructures from Dataset 1, which are pretty common, to train or validate these models for comparison. To make Dataset 1 more representative, we generated 598 frame microstructures using nine different topology structures, and tens of microstructures were generated with different strut radii for each topology. From **Dataset 1**, 512 samples are randomly picked as the training set, and the rest of the microstructures are adopted as the validation set for **Models 1, 2, 3, 4**. They are all trained for 6000 epochs using the SGD optimizer, where the batch size is eight, considering the GPU RAM. The learning rate is adaptively adjusted in training using the one cycle policy [55], where the maximum learning rate is 0.01.

The special characteristics of the models are listed as follows:
Model 1 (Ours): Trained using the potential energy loss function (22).
Model 1-A (Fine-tuned): Fine-tuned based on **Model 1** using additional 76 P and $F-RD$ TPMS elements.
Model 2 ($\|\mathbf{K}\boldsymbol{\mu}-\mathbf{f}\|_2^2$ loss): Trained using the weak form residual loss function (21).
Model 3 ($\boldsymbol{\mu}$ data-driven): Trained using pre-computed displacements $\overline{\boldsymbol{\mu}}^{kl}$ as labels and the loss function is defined as

$$\mathcal{L}_u = \frac{1}{N}\sum_{i=1}^{N} L_u(\boldsymbol{\rho}^{(i)}, \hat{\boldsymbol{\mu}}^{(i)}), \tag{35}$$

$$L_u(\boldsymbol{\rho}^{(i)}, \hat{\boldsymbol{\mu}}^{(i)}) = \sum_{k,l} \|(\overline{\boldsymbol{\mu}}^{kl})^{(i)} - \hat{\boldsymbol{\mu}}^{kl}(\boldsymbol{\rho}^{(i)})\|_2^2 \tag{36}$$

where $(\overline{\boldsymbol{\mu}}^{kl})^{(i)}$ is the pre-computed unit strain displacement field using FEA as the label of input $\boldsymbol{\rho}^{(i)}$.
Model 4 ($\mathbf{D}^H$ data-driven): Trained using the effective elasticity tensors as labels, and the loss function is defined in Eq. (4). The neural network architecture of the model is illustrated in Fig. 11. The neural network consists of three Conv3D blocks, each followed by a max pooling layer, one flattened layer, and four fully connected layers. The kernel size of the max pooling is two, such that the feature map is reduced to 1/2 after each max pooling layer. The max pooling and fully connected layers gradually reduce the data size. The number of parameters of the above neural network is 592643, larger than those of the 3D CNN in Fig. 7.

The output of the neural network is the nine components of the elasticity tensor as $\mathbf{D}_{11}^H, \mathbf{D}_{12}^H$, $\mathbf{D}_{13}^H$, $\mathbf{D}_{22}^H, \mathbf{D}_{23}^H$, $\mathbf{D}_{33}^H$, $\mathbf{D}_{44}^H, \mathbf{D}_{55}^H, \mathbf{D}_{66}^H$ considering the symmetry of the elasticity and that the values of the other components are 0 or very close to 0 for the microstructures in the experiments. We have also tested similar models outputting 21 and 36 components of elasticity tensors, and the one outputting nine components gave the best results.

The time of training, label preparation, and online prediction for these models are summarized in Table 4. The variations of loss functions (22), (21), (35), (4) corresponding to **Models 1, 2, 3, 4**, the training error and the validation error are used to show the overall performance of an approach, and are shown in Fig. 12. The training error represents the mean relative error, calculated as Eq. (34), for the training set including 512 frame microstructures, and the validation error indicates the mean relative error for the validation set consisting of 86 frame microstructures.

In the following, the overall performance of the proposed approach (**Models 1, 1-A**) is first explained in Section 5.2. After this, comparisons between the proposed and the one using a different loss function (**Model 2**) are given in Section 5.3. The comparisons to data-driven approaches (**Models 3, 4**) are given in Section 5.4.

Table 4: Time cost of homogenization surrogate modeling using our approach and data-driven approaches.

Approach	Label preparation	Training	Online predicting /computing
Ours	—	48.88 h (6000 epochs)	11.23 ms
Weak form residual loss	—	92.18 h (6000 epochs)	11.23 ms
Data-driven - displacements	34.66 h (512 samples)	28.91 h (6000 epochs)	11.23 ms
Data-driven - elasticity tensors	34.68 h (512 samples)	2.00 h (6000 epochs)	0.60 ms
Numerical homogenization	—	—	243.86 s*

(* In the Python implement of numerical homogenization, the global stiffness matrices are represented as sparse matrices, and the linear systems are solved using the conjugate gradient method (*scipy.sparse.linalg.cg*) while using the direct method (*scipy.sparse.linalg.spsolve*) costs more than half an hour for each microstructure.)

After training, given an input microstructure, it takes about 11.23 ms to output the effective elasticity tensor via forward propagation and the calculation via Eq. (25) using **Models 1, 1-A, 2, 3**, while it takes 0.60 ms using **Model 4**.

5.2. Overall performance

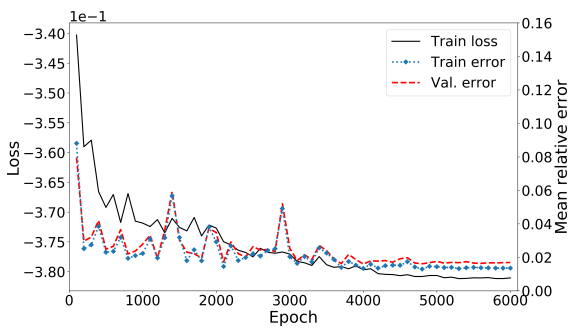
The overall performance of our approach **Model 1** is first explained, including its accuracy and generalization ability analysis.

Accuracy. Seeing from Fig. 12(a), after 6000 epochs, the mean relative error of the training set decreases to 1.4%, and the one of the validation set decreases to about 1.7%. Some critical components of the elasticity tensors as $\mathbf{D}^H_{11}, \mathbf{D}^H_{22}, \mathbf{D}^H_{33}$ are plotted as points in Fig. 13, where the x-coordinate indicates the reference value from numerical homogenization and the y-coordinate indicates the predicted value. The mean and the standard deviation values of the relative errors are marked in the upper left corners of the plots. The distribution of mean relative errors for microstructures in the validation set are also illustrated in Fig. 14(a), which tells that the mean relative errors of the predicted elasticity tensors are below 2% for most frame microstructures. The maximum mean relative error for frame microstructures is about 13%.

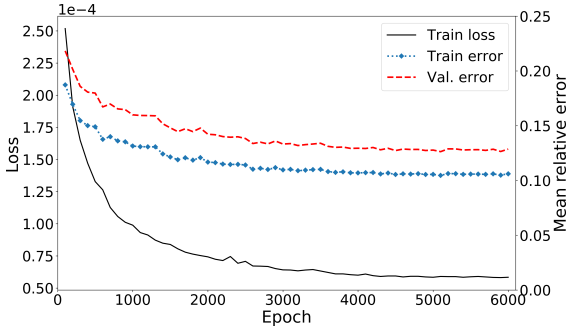
Generalization. Given any microstructure, the 3D microstructure-property mapping neural network model can output the corresponding elasticity tensor after training using the physics-informed loss function. In this sense, the trained CNN model can generalize to any input microstructure without additional training. On the other hand, the model is often expected to output the predicted elasticity tensor for a given input via forward propagation in real-time. Such generalization ability to predict the elasticity tensors of different microstructures is also evaluated here. To evaluate the generalization ability to predict the elasticity tensors of different microstructures, we tested on all the 565 TPMS elements from Dataset 2, 512 LHS microstructures randomly sampled from Dataset 3 and 512 P-LHS microstructures randomly sampled from Dataset 4. The results are analyzed as follows.

The relative errors in predicting the elasticity tensors of TPMS, LHS, and P-LHS microstructures are shown in Fig. 15(a). Except for several P and $F-RD$ type elements, the relative errors are all below 10%. For all LHS microstructures and most P-LHS microstructures, the relative errors are all below 1%. The mean relative errors for TPMS, LHS, and P-LHS microstructures are 4.4%, 0.66%, and 0.73%, respectively.

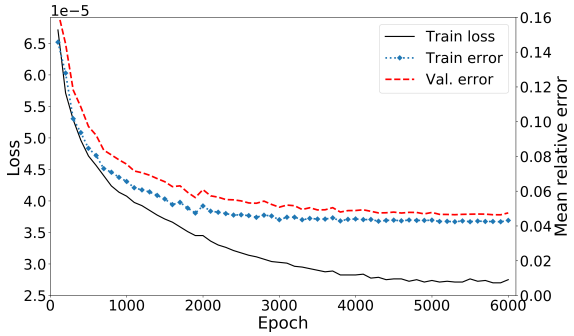
Fine tuning for different microstructures. **Model 1** is further trained to improve its performance for the P and $F-RD$ elements without good accuracy, resulting in **Model 1-A**. Instead of the expensive total



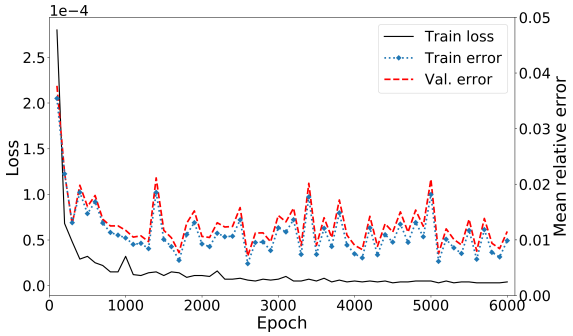
(a) **Model 1** trained using the potential energy functional loss (22)



(b) **Model 2** trained using the weak form residual loss (21)



(c) **Model 3** trained using displacements labels



(d) **Model 4** trained using elasticity tensor labels

Figure 12: Loss and mean relative error variations for different models on Test Set sampled from Dataset 1.

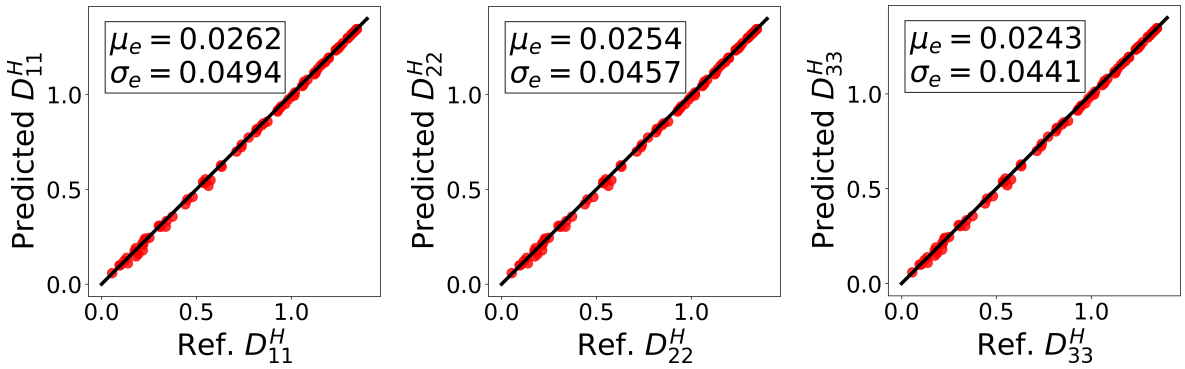


Figure 13: Scatter plots and relative error metrics of $\mathbf{D}^H_{11}, \mathbf{D}^H_{22}, \mathbf{D}^H_{33}$ predicted by **Model 1**.

re-training, the model is fine-tuned using the original 512 frame microstructures and one-third of the P and $F - RD$ elements (76 microstructures). The model is trained for another 500 epochs in 4.6 hours.

Using the fine-tuned model, the mean relative errors for TPMS, LHS, P-LHS, and frame microstructures are 3.1%, 0.64%, 0.76% and 1.6%, respectively, and the distribution of relative errors is also shown in Fig. 15(b).

5.3. Performance of loss functions

The performance of **Model 1** and **Model 2** under different loss functions was tested and summarized in the section. **Model 1** utilizes the potential energy functional loss (22), while **Model 2** employs the weak form residual loss (21).

Observing from Fig. 12(b), the value of the loss function finally reaches approximately 6×10^{-5} after 6000 epochs of training. The mean relative errors of the predicted elasticity tensors of the training and validation sets are 10.61% and 12.86%, respectively. The distribution of relative errors of frame validation is shown in Fig. 14(b). As can be seen from the results, the model trained by the energy functional loss is superior to the one trained by the weak form residual loss in both efficiency and accuracy.

5.4. Performance of physics-driven and data-driven approaches

The performance of the proposed approach is also compared with data-driven approaches that utilize displacements or effective elasticity tensors as labels, denoted as **Model 3** and **Model 4**, respectively. Computing the elasticity tensor of each microstructure takes approximately 243.86 seconds using numerical homogenization. The data-driven approach takes about 35 hours to obtain the elasticity tensors as labels for the training set of 512 samples.

5.4.1. Using displacement label

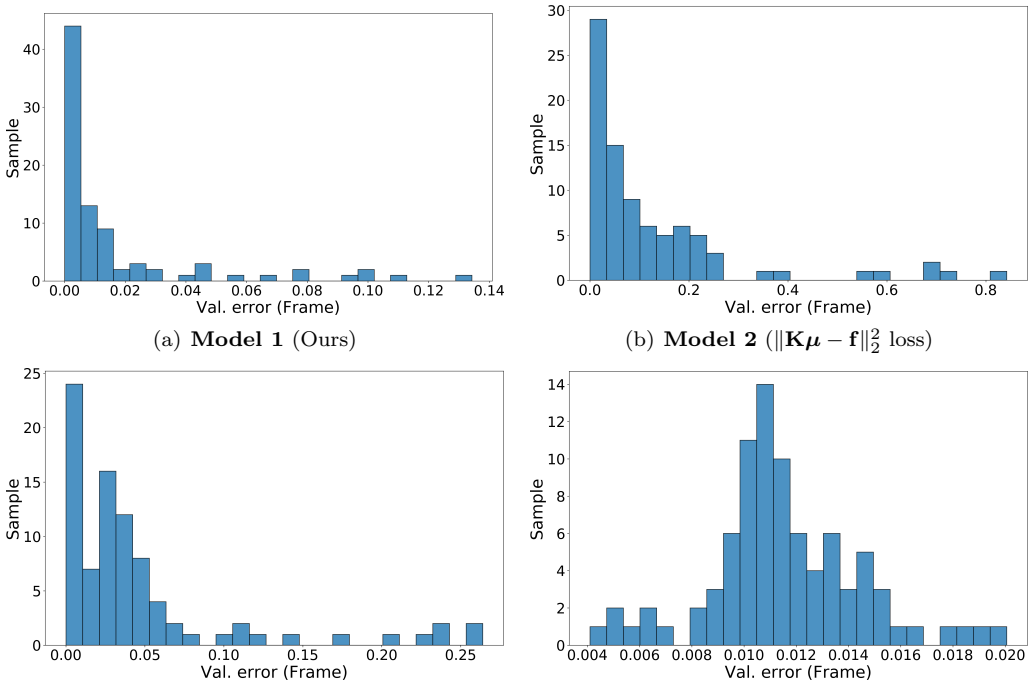
The training and validation errors in Fig. 12(c) are 4.3% and 4.7%, respectively, and the distribution of relative errors for frame validation are shown in Fig. 14(c). It was found that the displacement data-driven model obtains less accuracy than our physics-driven model for microstructures, either in the training or validation set.

To find the reason, we further compare the output $\hat{\boldsymbol{\mu}}^u$ of **Model 3** and the output $\hat{\boldsymbol{\mu}}^e$ of **Model 1**. For each microstructure in the training set, the outputted $\hat{\boldsymbol{\mu}}^u$ and $\hat{\boldsymbol{\mu}}^e$ are substituted into the energy functional function (22) and the displacement data-driven loss function (36). The differences as $L_u(\hat{\boldsymbol{\mu}}^e) - L_u(\hat{\boldsymbol{\mu}}^u)$ and $L_{eng}(\hat{\boldsymbol{\mu}}^e) - L_{eng}(\hat{\boldsymbol{\mu}}^u)$ are illustrated in Fig. 16. It can be seen that for most microstructures,

$$L_u(\hat{\boldsymbol{\mu}}^e) > L_u(\hat{\boldsymbol{\mu}}^u),$$

but for all microstructures

$$L_{eng}(\hat{\boldsymbol{\mu}}^e) \leq L_{eng}(\hat{\boldsymbol{\mu}}^u).$$



(a) **Model 1** (Ours) (b) **Model 2** ($\|\mathbf{K}\boldsymbol{\mu} - \mathbf{f}\|_2^2$ loss)

(c) **Model 3** ($\boldsymbol{\mu}$ data-driven) (d) **Model 4** ($\mathbf{D}^H$ data-driven)

Figure 14: Mean relative errors distribution of elasticity tensors of microstructures in the frame validation set of **Dataset 1** predicted by different models.

It indicates that the output of the displacement data-driven model is close to the displacement fields computed from FEA using the square error metric but is far from the minimum point of the energy functional. The fact tells that the displacement data-driven model needs to be better in terms of physical knowledge, which consequently results in effective elasticity tensors of not high accuracy.

5.4.2. Using effective elasticity tensor labels

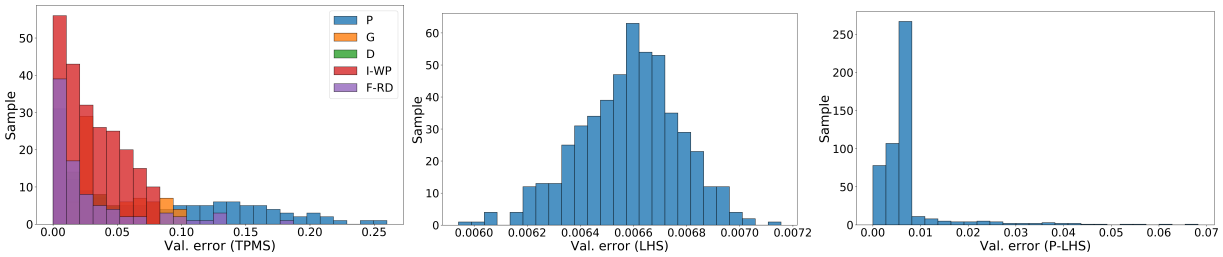
The performance of **Model 4**, which utilizes elasticity tensors as labels, is compared with **Model 1**. In this scenario, the neural network directly predicts the elasticity tensor instead of displacement fields.

Observing from Fig. 12(d), the trained model achieves lower mean relative errors on both the training and validation sets compared to those of our label-free approach, specifically 0.99% and 1.15%, respectively.

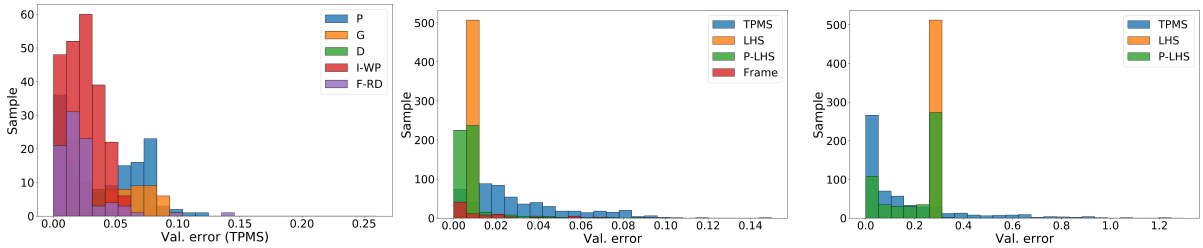
The relative errors for validation frame microstructures are all below 2%, as illustrated in Fig. 14(d). However, when the TPMS, LHS, and P-LHS microstructures are all input into **Model 4**, significantly higher mean relative errors are observed compared to those of our label-free approach, as depicted in Fig. 15(c). The prediction error of **Model 4** exceeds 20% for several microstructures, and the maximum relative error surpasses 100%.

The above tests show better generalization ability for different types of microstructures compared to the data-driven approaches. Furthermore, the generalization ability between microstructures of the same type is further compared between our approach and the elasticity tensor data-driven approach.We conducted the comparison experiments using Dataset 2 and Dataset 4, respectively. We trained models of our approach and the data-driven approach with 32 P-type TPMS microstructures from Dataset 2 and validated them with 68 different P-type TPMS microstructures. We sampled 128 P-LHS microstructures from **Dataset 4** to train two models, one based on the proposed approach and the other on the elasticity tensor data-driven approach, and another 1024 P-LHS microstructures were sampled for validation.

As seen from Fig. 17, both models get minimal relative errors on the training data after 5000-epoch training. However, the validation error of the data-driven model is far greater than the training error.



(a) **Model 1** (Ours)



(b) **Model 1-A** (Fine-tuned)

(c) **Model 4** ($\mathbf{D}^H$ data-driven)

Figure 15: Generalization of **Models 1, 1-A, 4** for 565 TPMS elements of different types as **Dataset 2**, 512 LHS microstructures randomly sampled from **Dataset 3** and 512 P-LHS microstructures randomly sampled from **Dataset 4**.

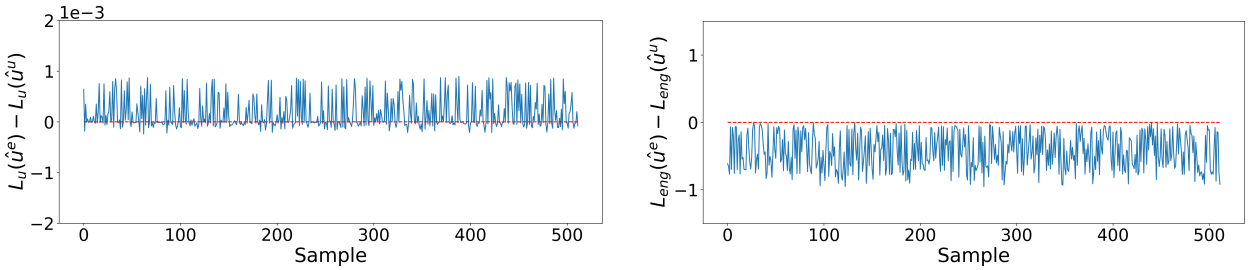


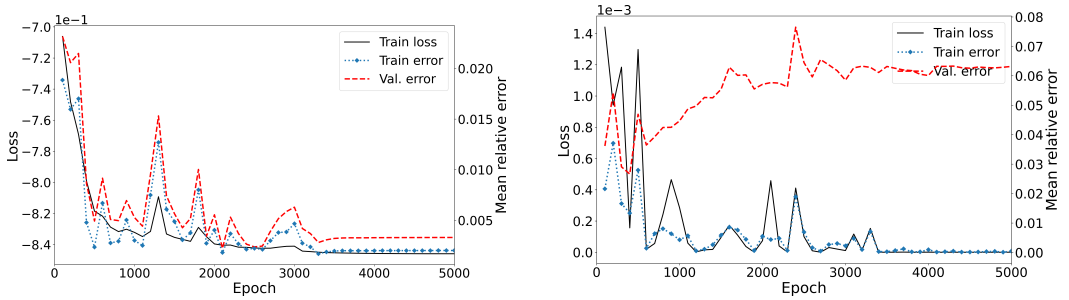
Figure 16: Differences of the displacements predicted by our model (**Model 1**) and the displacement data-driven model (**Model 3**) in terms of different metrics for microstructures in the training set.

In contrast, the variations of the validation error of our physics-driven model are similar to the training error. The validation error of the data-driven model tends to increase even as the training error decreases during the training procedure. Similar phenomena occur for other microstructures. For example, Fig. 18 illustrates the loss and relative error variations of models trained using our approach and the elasticity tensor data-driven approach with 32 P-type TPMS microstructures and validated with 68 different P-type TPMS microstructures. Our approach achieves a much smaller validation error compared to the data-driven one.

5.5. Experiments summation

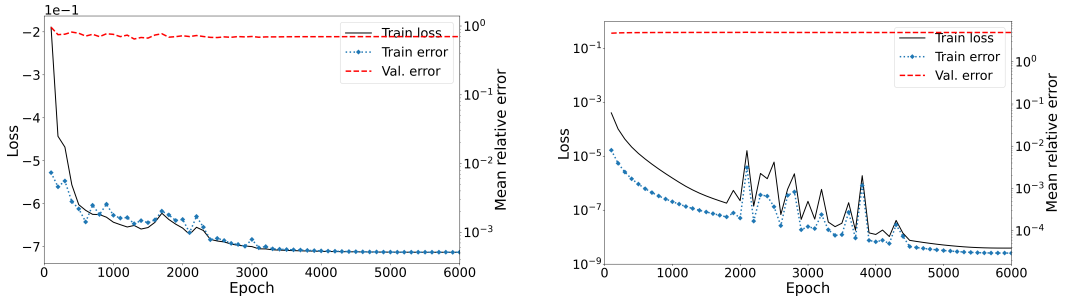
The overall performance of the proposed approach is demonstrated and compared to the approach using the weak form residual loss and the data-driven approaches. The time costs for different approaches are summarized in Table 4, and the mean relative errors of various methods for different microstructures are also summarized in Fig. 19.

Except for the elasticity tensor data-driven approach, our approach utilizing the energy functional loss surpasses other methods in terms of efficiency, accuracy, and generalization ability. The elasticity tensor data-driven approach achieves better accuracy in predicting the validation frame microstructures than ours, with mean relative errors of 1.15% and 1.7%, respectively. However, the relative errors of the elasticity tensors predicted by the data-driven model for different microstructures are much higher than those of our



(a) **Physics-driven** (Ours) (b) $\mathbf{D}^H$ **data-driven**

Figure 17: Loss and mean relative error variations in training (128 P-LHS microstructures) using our physics-driven and data-driven approaches.



(a) **Physics-driven** (Ours) (b) $\mathbf{D}^H$ **data-driven**

Figure 18: Loss and mean relative error variations in training (32 P-type TPMS microstructures) using our physics-driven and data-driven approaches.

physics-driven model. In summary, our approach establishes a 3D microstructure-property mapping neural network model with fair accuracy and far better generalization ability than data-driven approaches.

6. Conclusions

The study proposes a novel approach for constructing structure-property mapping using a label-free CNN model that does not require any labeled simulation data. This is achieved by leveraging observations on the characteristics of asymptotic approximation via FE analysis, statistical analysis via n-point correlation function, and the powerful feature extraction capabilities of CNNs. The CNN model, trained with a novel energy-based physics-informed loss function, ultimately learns the high-order hierarchical point correlation within the voxelization representation of an input microstructure. Various numerical examples demonstrate that the approach exhibits well-controlled errors and good generalization abilities across different microstructures compared to data-driven approaches.

Future work on this topic is extensive, with some aspects explained below. Firstly, further study is needed to understand why physics-informed models exhibit far better generalization ability than data-driven ones. Secondly, although the CNN model proves highly effective in instant property prediction once trained, similar to classical challenges in conventional deep learning, its validity and error control abilities lack theoretical study and merit further exploration. Additionally, extending the approach to nonlinear cases and exploring its applications in bi-scale simulation, inverse homogenization, or bi-scale topology optimization are also worthy of future research efforts.

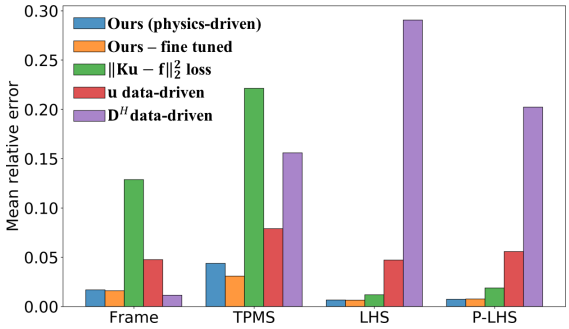


Figure 19: Mean relative errors of different models predicting elasticity tensors for different microstructure inputs.

Declarations

Conflicts of interest The authors declare no conflict of interest.
Data availability statement Data will be available upon request.

References

[1] B. Sun, X. Yan, P. Liu, Y. Xia, L. Lu, Parametric plate lattices: Modeling and optimization of plate lattices with superior mechanical properties, Additive Manufacturing (2023) 103626.
[2] D. Zhao, T. T. Gu, Y. Liu, S. Gao, M. Li, Constructing self-supporting structures in biscale topology optimization, The Visual Computer 38 (3) (2022) 1065–1082.
[3] H. Xu, R. Liu, A. Choudhary, W. Chen, A machine learning-based design representation method for designing heterogeneous microstructures, Journal of Mechanical Design 137 (5) (2015) 051403.
[4] I. S. Golovin, H. R. Sinning, Damping in some cellular metallic materials, Journal of Alloys and Compounds 355 (1) (2003) 2–9.
[5] H. Garmestani, S. Lin, B. Adams, S. Ahzi, Statistical continuum theory for large plastic deformation of polycrystalline materials, Journal of the Mechanics and Physics of Solids 49 (3) (2001) 589 – 607.
[6] 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, Journal of Computational physics 378 (2019) 686–707.
[7] Y. Liu, M. S. Greene, W. Chen, D. A. Dikin, W. K. Liu, Computational microstructure characterization and reconstruction for stochastic multiscale material design, Computer-Aided Design 45 (1) (2013) 65–76.
[8] Z. Yang, Y. C. Yabansu, R. Al-Bahrani, W.-k. Liao, A. N. Choudhary, S. R. Kalidindi, A. Agrawal, Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets, Computational Materials Science 151 (2018) 278–287.
[9] A. Cecen, H. Dai, Y. C. Yabansu, S. R. Kalidindi, L. Song, Material structure-property linkages using three-dimensional convolutional neural networks, Acta Materialia 146 (2018) 76–84.
[10] D. Chen, D. I. Levin, S. Sueda, W. Matusik, Data-driven finite elements for geometry and material design, ACM Transactions on Graphics (TOG) 34 (4) (2015) 1–10.
[11] S. Tan, T. Wong, Y. Zhao, W. Chen, A constrained finite element method for modeling cloth deformation, The Visual Computer 15 (1999) 90–99.
[12] F. Conde-Rodríguez, J.-C. Torres, Á.-L. García-Fernández, F.-R. Feito-Higueruela, A comprehensive framework for modeling heterogeneous objects, The Visual Computer 33 (2017) 17–31.
[13] L. Xia, B. Raghavan, P. Breitkopf, W. Zhang, A POD/PGD reduction approach for an efficient parameterization of data-driven material microstructure models, Computer Methods in Materials Science 13 (1-3) (2013) 219–225.
[14] M. Nesme, P. G. Kry, L. Jeřábková, F. Faure, Preserving topology and elasticity for embedded deformable models, in: ACM SIGGRAPH 2009 papers, 2009, pp. 1–9.
[15] L. Kharevych, P. Mullen, H. Owhadi, M. Desbrun, Numerical coarsening of inhomogeneous elastic materials, ACM Transactions on graphics (TOG) 28 (3) (2009) 1–8.
[16] R. Torres, A. Rodríguez, J. M. Espadero, M. A. Otaduy, High-resolution interaction with corotational coarsening models, ACM Trans. Graph. 35 (6) (Nov. 2016).
[17] J. Chen, H. Bao, T. Wang, M. Desbrun, J. Huang, Numerical coarsening using discontinuous shape functions, ACM Transactions on Graphics (TOG) 37 (4) (2018) 1–12.
[18] Z. Fang, B. Starly, W. Sun, Computer-aided characterization for effective mechanical properties of porous tissue scaffolds, Computer-Aided Design 37 (1) (2005) 65–72.
[19] E. Kröner, Statistical modelling, in: Modelling small deformations of polycrystals, Springer, 1986, pp. 229–291.

[20] S. Torquato, H. Haslach Jr, Random heterogeneous materials: microstructure and macroscopic properties, Appl. Mech. Rev. 55 (4) (2002) B62–B63.
[21] J.-C. Michel, H. Moulinec, P. Suquet, Effective properties of composite materials with periodic microstructure: a computational approach, Computer methods in applied mechanics and engineering 172 (1-4) (1999) 109–143.
[22] X. Liu, V. Shapiro, Homogenization of material properties in additively manufactured structures, Computer-Aided Design 78 (2016) 71–82.
[23] H. Peng, A. Liu, J. Huang, L. Cao, J. Liu, L. Lu, Ph-net: Parallelepiped microstructure homogenization via 3d convolutional neural networks, Additive Manufacturing 60 (2022) 103237.
[24] D. A. White, W. J. Arrighi, J. Kudo, S. E. Watts, Multiscale topology optimization using neural network surrogate models, Computer Methods in Applied Mechanics and Engineering 346 (2019) 1118–1135.
[25] J. P. Groen, J. Wu, O. Sigmund, Homogenization-based stiffness optimization and projection of 2D coated structures with orthotropic infill, Computer Methods in Applied Mechanics and Engineering 349 (2019) 722–742.
[26] M. Bessa, R. Bostanabad, Z. Liu, A. Hu, D. W. Apley, C. Brinson, W. Chen, W. K. Liu, A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality, Computer Methods in Applied Mechanics and Engineering 320 (2017) 633–667.
[27] Z. Liu, M. Fleming, W. K. Liu, Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials, Computer Methods in Applied Mechanics and Engineering 330 (2018) 547–577.
[28] L. Xia, P. Breitkopf, Multiscale structural topology optimization with an approximate constitutive model for local material microstructure, Computer Methods in Applied Mechanics and Engineering 286 (2015) 147–167.
[29] L. Zhu, M. Li, W. Xu, Direct design to stress mapping for cellular structures, Visual Informatics 3 (2) (2019) 69–80.
[30] J. P. Groen, O. Sigmund, Homogenization-based topology optimization for high-resolution manufacturable microstructures: Homogenization-based topology optimization for high-resolution manufacturable microstructures, International Journal for Numerical Methods in Engineering 113 (8) (2018) 1148–1163.
[31] J. Wu, W. Wang, X. Gao, Design and optimization of conforming lattice structures, IEEE Transactions on Visualization and Computer Graphics 27 (1) (2019) 43–56.
[32] Y. Holdstein, A. Fischer, Three-dimensional surface reconstruction using meshing growing neural gas (mgng), The Visual Computer 24 (2008) 295–302.
[33] M. Lefik, D. P. Boso, B. A. Schrefler, Artificial Neural Networks in numerical modelling of composites, Computer Methods in Applied Mechanics and Engineering 198 (21) (2009) 1785–1804.
[34] B. A. Le, J. Yvonnet, Q.-C. He, Computational homogenization of nonlinear elastic materials using neural networks: Neural networks-based computational homogenization, International Journal for Numerical Methods in Engineering 104 (12) (2015) 1061–1084.
[35] Z. Liu, C. Wu, M. Koishi, A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials, Computer Methods in Applied Mechanics and Engineering 345 (2019) 1138–1168.
[36] X. Yang, M. Li, L. Zhu, W. Zhong, Evolutionary discrete multi-material topology optimization using cnn-based simulation without labeled training data, in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 85376, American Society of Mechanical Engineers, 2021, p. V002T02A027.
[37] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation (2015). arXiv:1505.04597.
[38] G. Papanicolau, A. Bensoussan, J.-L. Lions, Asymptotic analysis for periodic structures, Vol. 5, Elsevier, 1978.
[39] E. S. Palencia, Non-homogeneous media and vibration theory, Springer-Verlag, 1980.
[40] M. P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, Computer Methods in Applied Mechanics and Engineering 71 (2) (1988) 197 – 224.
[41] C. Beck, S. Becker, P. Grohs, N. Jaafari, A. Jentzen, Solving stochastic differential equations and Kolmogorov equations by means of deep learning, arXiv:1806.00421 (Jun. 2018).
[42] J. Berg, K. Nyström, A unified deep artificial neural network approach to partial differential equations in complex geometries, Neurocomputing 317 (2018) 28–41.
[43] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, Journal of Computational Physics 394 (2019) 56–81.
[44] O. Sigmund, Materials with prescribed constitutive parameters: An inverse homogenization problem, International Journal of Solids and Structures 31 (17) (1994) 2313 – 2329.
[45] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognition 77 (2018) 354–377.
[46] R. Garg, V. K. BG, G. Carneiro, I. Reid, Unsupervised cnn for single view depth estimation: Geometry to the rescue, in: European Conference on Computer Vision, Springer, 2016, pp. 740–756.
[47] Y. Zhang, W. Chan, N. Jaitly, Very deep convolutional networks for end-to-end speech recognition, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017, pp. 4845–4849.
[48] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[49] E. Andreassen, C. S. Andreasen, How to determine composite material properties using numerical homogenization, Computational Materials Science 83 (2014) 488–495.
[50] G. Dong, Y. Tang, Y. F. Zhao, A 149 line homogenization code for three-dimensional cellular materials written in matlab, Journal of Engineering Materials and Technology 141 (1) (2019) 011005.
[51] L. Cheng, P. Zhang, E. Biyikli, J. Bai, S. Pilz, A. C. To, Integration of topology optimization with efficient design of additive manufactured cellular structures, in: Solid Freeform Fabrication Symposium, Austin, TX, 2015, pp. 10–12.

[52] D.-J. Yoo, Computer-aided porous scaffold design for tissue engineering using triply periodic minimal surfaces, International Journal of Precision Engineering and Manufacturing 12 (1) (2011) 61–71.
[53] N. Yang, Z. Quan, D. Zhang, Y. Tian, Multi-morphology transition hybridization CAD design of minimal surface porous structures for use in tissue engineering, Computer-Aided Design 56 (2014) 11–21.
[54] L. Dong, S. H. Gamal, S. N. Atluri, Stochastic macro material properties, through direct stochastic modeling of heterogeneous microstructures with randomness of constituent properties and topologies, by using trefftz computational grains (tcg), CMC: Computers, Materials & Continua 37 (1) (2013) 1–21.
[55] L. N. Smith, A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay, arXiv preprint arXiv:1803.09820 (2018).