



# Constructing boundary-identical microstructures via guided diffusion for fast multiscale topology optimization

Jingxuan Feng <sup>a,1</sup>, Lili Wang <sup>a,1</sup>, Xiaoya Zhai <sup>a</sup>\*, Kai Chen <sup>b</sup>, Wenming Wu <sup>c</sup>, Ligang Liu <sup>a</sup>, Xiao-Ming Fu <sup>a,\*</sup>

<sup>a</sup> School of Mathematical Sciences, University of Science and Technology of China, Hefei, 230026, People's Republic of China

<sup>b</sup> Zhongguancun Institute of Artificial Intelligence, Beijing 100194, People's Republic of China

<sup>c</sup> Hefei University of Technology, Hefei Anhui, 230026, People's Republic of China

## ARTICLE INFO

### Keywords:

Multiscale topology optimization  
Boundary-identical microstructures  
Large-range datasets  
Wide ranges of elastic modulus  
Self-conditioning diffusion model  
Active learning

## ABSTRACT

Hierarchical structures exhibit critical features across multiple scales. However, designing multiscale structures demands significant computational resources, and ensuring connectivity between microstructures remains a key challenge. To address these issues, **large-range, boundary-identical microstructure datasets** are successfully constructed, where the microstructures share the same boundaries and exhibit a wide range of elastic moduli. This approach enables highly efficient multiscale topology optimization. Central to our technique adopts a deep generative model, guided diffusion, to generate microstructures under the two conditions, including the specified boundary and homogenized elastic tensor. We generate the desired datasets using active learning approaches, where microstructures with diverse elastic moduli are iteratively added to the dataset, which is then retrained. After that, sixteen boundary-identical microstructure datasets with wide ranges of elastic modulus are constructed. We demonstrate the effectiveness and practicability of the obtained datasets over various multiscale design examples. Specifically, in the design of a mechanical cloak, we utilize macrostructures with  $30 \times 30$  elements and microstructures filled with  $256 \times 256$  elements. The entire reverse design process is completed within one minute, significantly enhancing the efficiency of the multiscale topology optimization.

## 1. Introduction

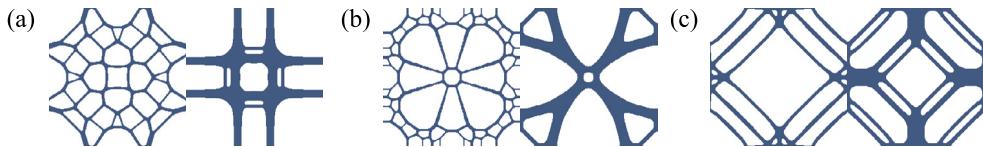
Microstructures are characterized by their unique physical properties derived from specialized topological structures. Heterogeneous multiscale systems composed of microstructures can achieve sophisticated functionalities, such as invisibility cloaks [1,2], soft robotics [3], and high thermal conductivity [4,5]. Despite advances in modern manufacturing technologies that enable the fabrication of these intricate systems spanning macroscopic and microscopic scales, designing multiscale systems remains a challenge, primarily involving intricate inverse design of microstructures and costly nested multiscale optimization processes.

Advances in computational capabilities and machine learning have spurred interest in data-driven multiscale optimization. The existing frameworks are broadly categorized into top-down methods [6–8] and bottom-up approaches [9–11]. Top-down methods first perform topology optimization to determine macroscopic property distributions, followed by populating the design with microstructures from a pre-generated database. Bottom-up approaches use microstructural parameters, such as volume fraction,

\* Corresponding authors.

E-mail addresses: [fjx0@mail.ustc.edu.cn](mailto:fjx0@mail.ustc.edu.cn) (J. Feng), [llw91@mail.ustc.edu.cn](mailto:llw91@mail.ustc.edu.cn) (L. Wang), [xiaoyazhai@ustc.edu.cn](mailto:xiaoyazhai@ustc.edu.cn) (X. Zhai), [fuxm@ustc.edu.edu](mailto:fuxm@ustc.edu.edu) (X.-M. Fu).

<sup>1</sup> These authors contributed equally to this work.



**Fig. 1.** Illustration of boundary connections. (a) The boundary is completely disconnected. (b) The boundary is partly connected. (c) The boundary is fully connected, known as boundary-identical microstructure sets.

as design variables, using machine learning models to predict effective properties and accelerate optimization. However, the parameterized representation in bottom-up approaches may limit the property space of microstructures. Thus, we adopt a top-down approach, which relies on a comprehensive microstructure database as the building blocks. The design process involves matching microstructures to desired properties while ensuring two key requirements: (1) Boundary Connectivity: Essential for periodic or graded arrangements, affecting the transmission of forces, waves, etc. Fig. 1 categorizes connections as unconnected, partially connected, or fully connected (identical boundary), with compatible microstructures offering improved performance [12]. (2) Elastic Moduli Range: A large-range dataset covering a wide range of elastic moduli enhances multiscale design by expanding the feasible design space and increasing alignment with required properties.

However, generating such a microstructure dataset is challenging. The reasons are twofold. First, the diversity of generated structures of previous structural optimization algorithms [7,12–19] is poor. Moreover, the hard connectivity constraint further limits the range of elastic moduli. Second, choosing a boundary for all microstructures to realize the goal of a wide range of elastic moduli is extremely difficult or clueless.

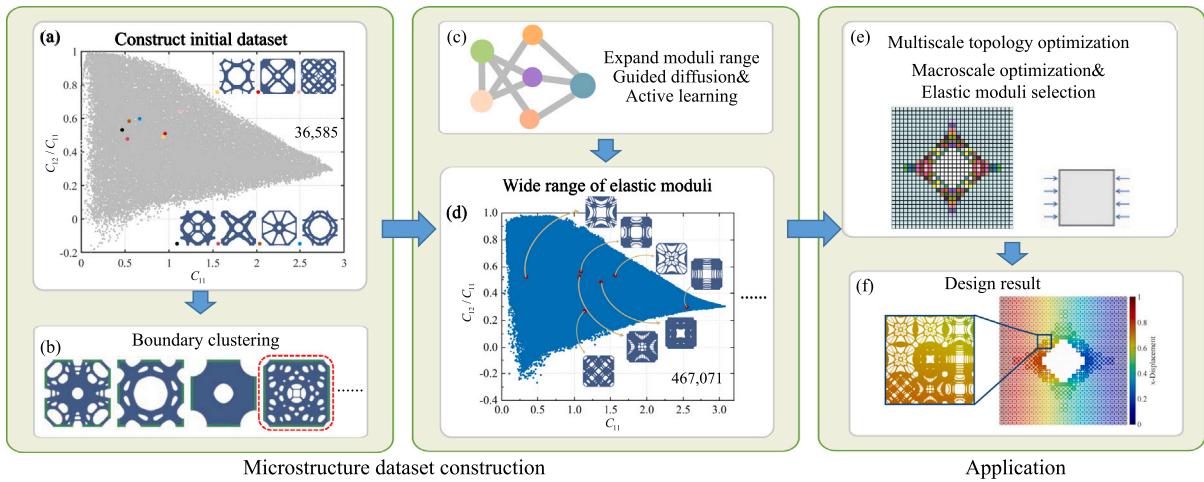
Three primary methods for generating compatible microstructures are parametric modeling [13–15], geometric interpolation [7,16,17], and connectivity topology optimization [12,18,19]. Parametric modeling adjusts predefined microstructures to meet property requirements through an iterative trial-and-error process. Geometric interpolation generates a microstructure sequence by interpolating between two predefined deformed structures. Connectivity topology optimization enforces connectivity constraints using gradient descent to achieve geometric continuity. However, they fail to ensure fully identical boundaries and may cause faults, stress concentrations, and manufacturing challenges. Furthermore, none of these methods effectively generate large-range microstructure datasets with a broad range of elastic moduli.

Identical boundary embedded large-range microstructure datasets are successfully constructed in this paper. Boundary connectivity constraints can be enforced using the same boundary and symmetry properties. In addition, we opt for cubic symmetric microstructures for simplicity of calculation. The key to our algorithm is using a deep generative model to achieve diverse microstructures with large elastic moduli coverage under boundary-identical and cubic symmetric constraints. Specifically, we adopt a self-conditioning diffusion model to generate microstructures under two conditions: (1) the predefined boundary and (2) the specified homogenized elastic tensor.

Training the diffusion model requires a dataset. However, constructing this dataset is our primary objective, creating a challenging chicken-and-egg problem. To this end, we propose a practical strategy. First, we train a self-conditioning diffusion model using a microstructure dataset constructed by an optimization algorithm that incorporates cubic symmetry constraints but excludes the boundary-identical condition. We also cluster the microstructure boundaries to generate a small set of seed boundaries. Then, we run the diffusion model for each seed boundary to generate diverse microstructures under various homogenized elastic tensors while fixing the seed boundary. These microstructures form a boundary-identical dataset corresponding to each seed boundary. Finally, we create the desired dataset by iteratively retraining a diffusion model on each boundary-identical dataset and adding the generated microstructures.

We are the first to construct large-range microstructure datasets with the same boundary and wide coverage of elastic moduli. The dataset is inherently conducive to multiscale design and effectively addresses a crucial issue in multiscale structures — the connectivity problem. We utilize the dataset for two multiscale design scenarios: a mechanical cloak design and a customized displacement design. By testing various boundary cases and hole configurations, we demonstrate the feasibility and versatility of the datasets. It is noteworthy that in the design of the mechanical cloak, we employ  $30 \times 30$  macro units and microstructures filled with  $256 \times 256$  elements. The entire reverse design process is accomplished within one minute, markedly enhancing the efficiency of multiscale optimization. Our pipeline is shown in Fig. 2. Code for this paper is available at <https://github.com/Schnabel-8/BDR-diffusion>, and the datasets are publicly available at <https://rec.ustc.edu.cn/share/dataset>. In summary, our technical contributions are as follows:

- Under the boundary-identical constraint, large-range microstructure datasets are successfully constructed. **This is proved to be feasible for the first time.**
- A fast multiscale optimization strategy using large-range, boundary-identical microstructure datasets achieves significant acceleration over state-of-the-art methods.



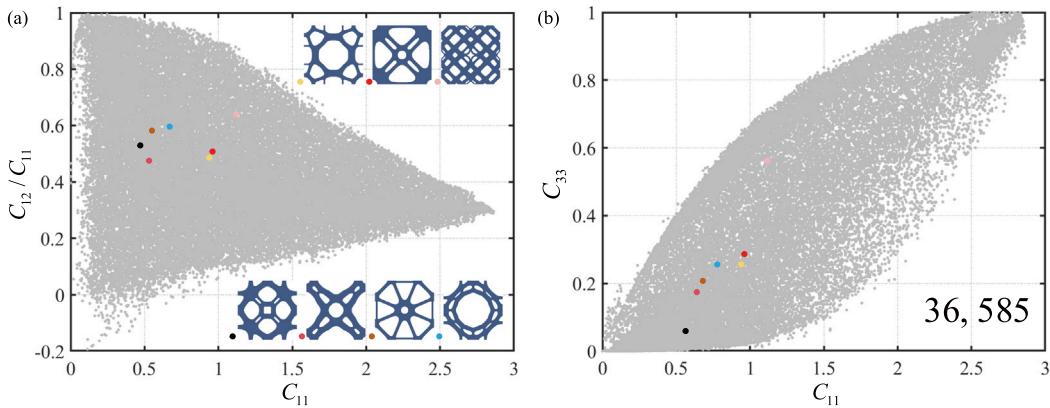
**Fig. 2.** The pipeline of constructing large-range boundary-identical microstructure datasets and their applications. (a) The cubic symmetric microstructure dataset without identical boundaries. (b) Representative boundaries derived from boundary clustering. (c) Generated microstructure datasets with the specified boundaries, which are expanded through guided diffusion and active learning. (d) A dataset with a wide range of elastic moduli. (e)–(f) The results of multiscale topology optimization.

## 2. Related works

**Microstructure connectivity constraints.** Three methods - parametric modeling [13–15], geometric interpolation [7,16,17], and topology optimization [12,18,19] — play crucial roles in generating microstructures with good connectivity. Parametric modeling effectively describes microstructure geometries for design purposes, yet achieving structures meeting specific performance criteria involves a trial-and-error process. Similarly, geometric interpolation ensures microstructural connectivity, but assessing its performance improvement remains challenging. Consequently, they are usually integrated with topology optimization techniques to achieve improved results. Zobazer and Sutradhar [14] introduce supershape-based parametric modeling [20] to ensure smooth connectivity at interfaces. Schumacher et al. [16] achieve connected microstructures by interpolating from a microstructure database. Shape interpolation methods are developed using signed distance functions or characteristic level set functions [7,17]. Du et al. [19] combine topology optimization with newly defined connectivity index constraints. Garner et al. [12] consider geometric connectivity and physical compatibility for the inverse homogenization method. Zhai et al. [18] propose a differentiable microstructure generation framework, reformulating the problem as a non-uniform heat diffusion process. Despite these advancements, they still need to ensure complete boundary connectivity. Moreover, the initial microstructure will influence the final design, leading to different structures and properties with varied initializations. While all three methods ensure microstructural connectivity, they impose varying limitations on the diversity of the compatibility dataset. Instead, we propose constructing cubic symmetric microstructures with identical boundaries and diverse geometries to ensure complete boundary connectivity.

**Microstructure dataset.** Microstructure datasets encompass geometric configurations, physical properties, and performance characteristics. They serve as essential tools for analysis and prediction goals [21–24], experimentation/simulation [25–28], and the development of models or algorithms in microstructure design [29–33]. Currently, microstructure design-oriented datasets consist primarily of density-based 2D and 3D datasets [34–36], along with truss-based datasets [37–39]. The construction of microstructure datasets is often based on tasks, where factors such as dataset size and property coverage can influence the effectiveness of downstream tasks. The methods employed for dataset generation can be broadly categorized into three main types: heuristic generation [35,39–41], perturbation generation [34,42–44], and simulation generation [25,27,45,46]. Shape-driven heuristics quickly generate data but lack control, leading to unpredictable and low-quality datasets. Effective property calculation demands costly finite element simulations, making high-quality dataset acquisition challenging. For multiscale design, datasets must have broad property coverage and diverse geometries to ensure compatibility during the filling process.

**AI for microstructures.** Neural networks have been successfully used for generating a variety of microstructures (c.f. the survey in [47]). Examples include obtaining the homogenization of composite structures [24,48,49], effective response of stress-strain curves [39,50–52], and extreme mechanical microstructures [22,53,54]. In terms of performance prediction, AI can establish predictive models by learning from a vast amount of microstructure data and their corresponding performance characteristics [24, 36,55]. This enables researchers to rapidly screen potential exceptional microstructure candidates and save significant time in experimentation and simulation. For example, Le et al. [48] propose a decoupled computation homogenization method for nonlinear elastic materials using neural networks. This method utilizes a neural network model to compute the effective potential. Deng et al. [56] design a network to predict the nonlinear response curves of hinged quadrilateral microstructures. Ma et al. [57] use residual networks instead of finite element analysis to compute strains and successfully employ genetic algorithms for the inverse



**Fig. 3.** Plots of  $C_{11}$  vs.  $C_{12}/C_{11}$  (a) and  $C_{11}$  vs.  $C_{33}$  (b) on a cubic symmetric microstructure dataset  $D_0$  without identical boundaries. The number of microstructures is 36,585.

design of microstructures with predetermined global strains under magnetic actuation. On the other hand, in the process of inverse design, AI models, particularly generative models, can effectively explore complex design spaces and discover new structures and material combinations that go beyond the limitations of human intuition [22]. Zheng et al. [39] utilize a graph-based VAE to design truss structures with customized mechanical properties in both linear and nonlinear states, including designs exhibiting exceptionally stiff, auxetic, pentamode-like, and tailored nonlinear behaviors. Wang et al. [46] propose a method using inverse homogenization generative adversarial networks (IH-GANs) for designing variable-density cellular structures. Zheng et al. [58] develop a GAN-based design method for auxetic microstructures and successfully use this method to design auxetic microstructures with predetermined Young's modulus and Poisson's ratio. Diffusion models are currently the mainstream approach for content generation. According to [59], diffusion models surpass previous methods such as GAN and VAE in terms of the quality of generated contents. Consequently, we employ conditional diffusion models for the inverse generation of microstructures.

### 3. Method

#### 3.1. Overview

*Goals and requirements.* Our goal is to construct a large-range microstructure dataset  $D$  satisfying the following requirements:

1. *Cubic symmetric*: each microstructure  $s$  is cubic symmetric.
2. *Identical boundary*: the microstructures in the dataset have an identical boundary  $B$ .
3. *Wide range of elastic moduli*: the physical properties of the microstructures cover a large space  $S$ .

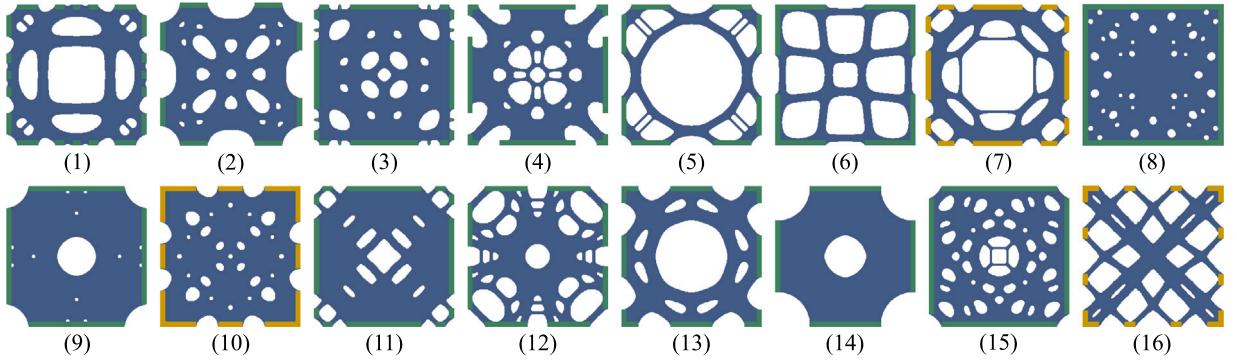
The first and second requirements are hard constraints to ensure that the microstructures can achieve full compatibility in multiscale design. The third requirement aims to increase the choice space in multiscale design significantly.

*Methodologies.* To ensure the diversity of microstructures, we use a deep generative model instead of the optimization methods. We first construct a dataset  $\hat{D}$  satisfying the first and second requirements (Section 3.2). Specifically, to reduce the difficulty of meeting the three requirements, we use a self-conditioning diffusion model [60] with property guidance, where the properties including the specified boundary  $B$  and homogenized elastic tensor  $C$ . Finally, to realize the third requirement, we propose a pipeline based on active learning that alternately runs the following two steps (Section 3.3).

1. *Training the self-conditioning diffusion model*: using the dataset satisfying the first and second requirements to train the self-conditioning diffusion model.
2. *Augmenting the dataset*: adding the microstructures generated by the self-conditioning diffusion model into the dataset.

#### 3.2. Initializing microstructure dataset

*Representation.* We discrete each microstructure  $s$  into  $256 \times 256$  elements denoted as a binary matrix  $A_s$ , where 0 represents void and 1 represents solid. The number of elements  $256 \times 256$  effectively represents diverse microstructures and offers satisfactory computational efficiency. We characterize the microstructure boundaries  $B_s$  through a binary vector  $b_s$  extracted from the first row of the matrix  $A_s$ .



**Fig. 4.** Illustration of sixteen boundaries derived by k-means clustering in dataset  $D_0$ . Green boundaries indicate that there are corresponding boundaries in  $D_0$ , while yellow boundaries indicate that no microstructures with these boundaries are found in  $D_0$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

*Dataset via optimization, perturbation and filtering.* We use the optimization method [61] and shape perturbation [62] to generate a cubic symmetric microstructure dataset  $D_0$  (shown in Fig. 3) without identical boundaries.

1. *Optimizing microstructures:* we generate microstructures by optimizing the bulk and shear modulus under various volume fractions via inverse homogenization method [61]. Specifically, we consider a volume fraction range of 0.2 to 0.9 and generate 16,000 microstructures.
2. *Perturbing microstructures:* utilizing the radial distortion model proposed by Kroon [62], each microstructure is subjected to perturbation. For each microstructure, we generate 30 different perturbed versions, resulting in a total of 480,000 perturbed microstructures. The perturbed microstructures are filtered to retain all fully connected microstructures while discarding the disconnected ones. There are about 400,000 microstructures left.
3. *Eliminating similarity:* We merge the perturbed microstructures with the optimized ones and refine the dataset to eliminate microstructures that are geometrically similar. Let the merged dataset be denoted as  $\bar{D}$ . We first perform uniform sampling within the coverage range of the elastic tensors in the dataset. The sampled dataset is referred as  $\bar{D}'$ . For each microstructure  $s$  in dataset  $\bar{D}'$ , we calculate its similarity to other structures within the dataset. Structure similarity (SS) is defined as:

$$\text{SS}(s) = \max \{ \text{IoU}(\mathcal{A}_s, \mathcal{A}_{s'}) : s' \in N(s, \bar{D}') \}, \quad (1)$$

where the set  $N(s, \bar{D}') = \{t \mid t \in \bar{D}', t \neq s \text{ and } \| (C_{11}(s), C_{12}(s), C_{13}(s)) - (C_{11}(t), C_{12}(t), C_{13}(t)) \|_2 < 0.5 \}$ , where  $C_{ij}(s)$  (for 2D cases,  $i, j = 1, 2, 3$ ) denotes the component  $C_{ij}$  of the elastic tensor matrix  $C$  for microstructure  $s$ .  $\text{IoU}(\mathcal{A}_s, \mathcal{A}_{s'})$  is the Intersection over Union operation between two microstructures ( $s$  and  $s'$ ), which is calculated as:

$$\text{IoU}(\mathcal{A}_s, \mathcal{A}_{s'}) = \frac{\mathcal{A}_s \cap \mathcal{A}_{s'}}{\mathcal{A}_s \cup \mathcal{A}_{s'}}. \quad (2)$$

Here,  $\mathcal{A}_s \cap \mathcal{A}_{s'} := \sum_{p,q} (\mathcal{A}_s(p, q) \wedge \mathcal{A}_{s'}(p, q))$ ,  $\mathcal{A}_s \cup \mathcal{A}_{s'} := \sum_{p,q} (\mathcal{A}_s(p, q) \vee \mathcal{A}_{s'}(p, q))$  ( $p, q = 1, 2, \dots, 256$ ).

After computing the similarity for each microstructure, we remove those with a similarity score greater than 0.9 from the dataset, thereby obtaining dataset  $D_0$  with 36,585 microstructures.

*Boundary clustering.* We perform clustering operations on the dataset  $D_0$  based on microstructure boundaries. When clustering these boundaries, we measure their similarity using the Euclidean distance. Specifically, for two boundary binary vectors  $\mathbf{b}$  and  $\mathbf{b}'$ , the boundary similarity (BS) is calculated as:

$$\text{BS}(\mathbf{b}, \mathbf{b}') = \sqrt{\sum_{i=1}^{256} (b_i - b'_i)^2}.$$

Specifically, we use the k-means clustering method to generate a set of seed boundaries  $\{\mathcal{B}_i^{\text{seed}}\}$ . To determine the number of classes, we first select 40 clusters, but the clustering results show several nearly identical boundaries. To address this, we gradually reduce the number of clusters until there are no longer highly similar boundaries in the clustering results. The final cluster number is 16 as shown in Fig. 4. Notably, the k-means clustering process is highly efficient, with all operations in this study completed in two minutes.

*Boundary-constrained generation.* Using the dataset  $D_0$ , we train a self-conditioning diffusion model that is capable of generating microstructures given specified elastic tensors and boundary types, as illustrated in Fig. 5. A detailed description of this model is provided in Section 3.3. For each seed boundary  $\mathcal{B}_i^{\text{seed}}$ , we run the generative model to generate various microstructures to form a dataset  $\hat{D}$  satisfying the boundary-identical and cubic symmetric constraints. Based on the generated results, the dataset  $\hat{D}$  has limitations regarding its elastic moduli space. To this end, we propose an iterative algorithm, an active learning strategy, to alternate model retraining and dataset augmentation.

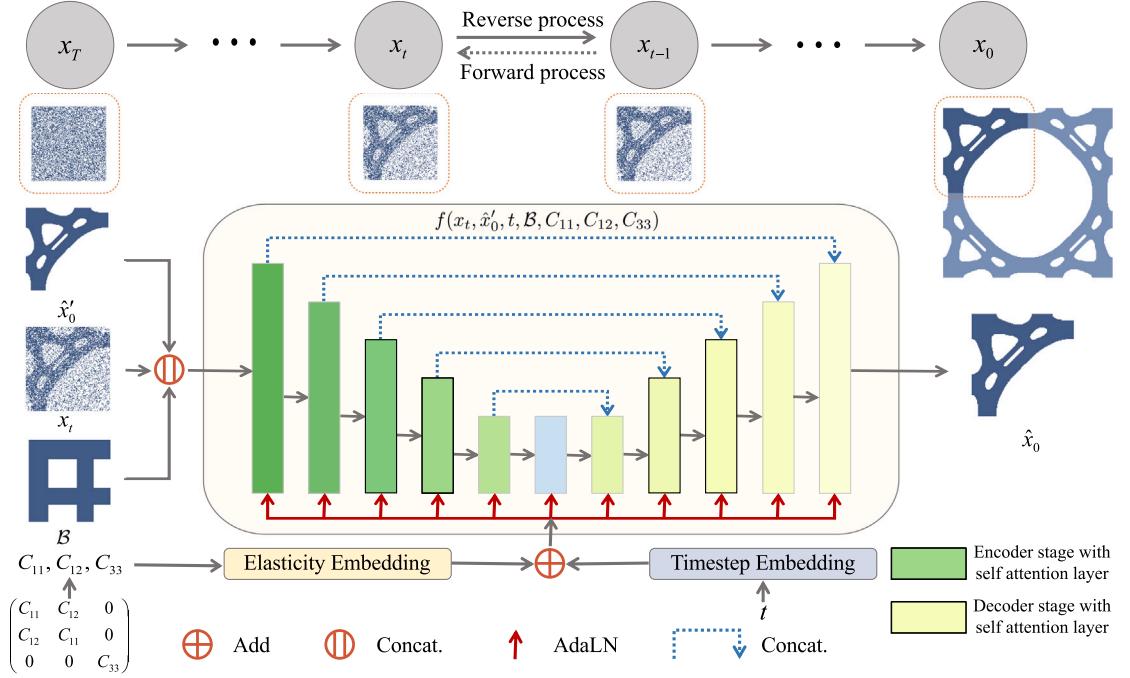


Fig. 5. Network architecture of the proposed self-conditioning diffusion model.

### 3.3. Self-conditioning diffusion model with property guidance

Given the cubic symmetry of the microstructure, we utilize the top-left quarter as a representative proxy for the entire structure. It facilitates computational resources during network training and microstructure generation. This portion is designated as  $x$  and is represented by the first 128 rows and 128 columns of matrix  $\mathcal{A}$ . Furthermore, to ensure that the generated microstructures exhibit cubic symmetry, we utilize symmetric Gaussian noise in the generation process.

*Diffusion model.* Diffusion models typically serve to denoise Gaussian noise  $x_T$  towards a data sample  $x_0$  in  $T$  steps, involving both forward and reverse processes during training (Fig. 5).

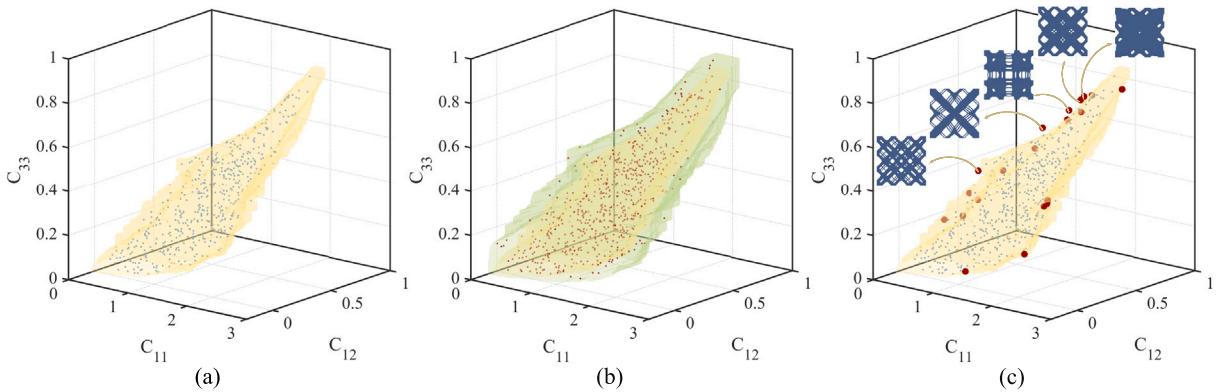
- **Forward Process:** This process begins with a data sample  $x_0$  and progressively generates a noisy sample  $x_t$  at each time step  $t$ . The addition of noise is achieved by sampling a Gaussian noise  $\epsilon \sim \mathcal{N}(0, I)$ .

$$x_t = \sqrt{\gamma_t} x_0 + \sqrt{1 - \gamma_t} \epsilon, \quad (3)$$

$\gamma_t$  is a noise schedule that gradually changes from 1 to 0. At each step, the model adds a bit of this Gaussian noise to the data sample, thereby gradually transforming the original data  $x_0$  into a noisy version  $x_t$  over  $T$  steps.

- **Reverse Process:** The reverse process commences from pure Gaussian noise  $x_T \sim \mathcal{N}(0, I)$ . The model is trained to learn the denoising sequence, effectively reversing the noising process step by step until it reaches  $x_0$ . In each step of this process, the model takes a noisy sample  $x_t$  and estimates the less noisy sample  $x_{t-1}$ . We generally use a neural network to predict this estimation. However, instead of directly learning the transition from  $x_t$  to  $x_{t-1}$ , we employ a network, denoted as  $f(x_t, t)$ , to predict  $x_0$  from  $x_t$ . We then estimate  $x_{t-1}$  based on  $x_t$  and the predicted  $\hat{x}_0$ . Following [63], the backbone of this denoising network is implemented as a U-Net architecture shown in Fig. 5.

*Self-conditioning.* Self-conditioning is a technique proposed by Chen et al. [64] to improve the generation quality of diffusion models. The core concept of self-conditioning involves utilizing the model's previously generated partial outputs as supplementary information to guide subsequent generation steps. This approach enhances the model's capacity to leverage conditional information, resulting in data that aligns more closely with the conditional input. The specific approach is as follows: we consider a slightly different denoising function,  $f(x_t, \hat{x}'_0, t)$ , which takes the concatenation of the previous estimation  $\hat{x}'_0$  for  $x_0$  and  $x_t$  as input to the network. Building on the work of Chen et al. [64], during the training process, we estimate  $\hat{x}'_0 = f(x_t, 0, t)$  with a 50% probability and use this estimate for self-conditioning. For the remaining instances, we set  $\hat{x}'_0$  to 0, effectively reverting to a modeling approach without self-conditioning.



**Fig. 6.** Illustration of the generation process: (a) Interior of the Signed Distance Function (SDF) representing the training set's property coverage. (b) The expanded SDF (green region) and the target elastic tensor (red points). (c) Microstructures generated by the network that lie beyond the coverage of the training set properties. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Conditions.** Our self-conditioning diffusion model aims to generate microstructures based on the given boundary and elastic tensor. For microstructures with cubic symmetry, their homogenized elastic tensor  $C^H = [C_{11}, C_{12}, C_{13}; C_{21}, C_{22}, C_{23}; C_{31}, C_{32}, C_{33}]$  can be fully determined by three independent components:  $C_{11}$ ,  $C_{12}$ , and  $C_{33}$ . Therefore, we only consider these three components as the conditions for elastic tensor. We incorporate them into the network using classifier-free guidance [65]. The implementation is detailed as follows: Initially, we normalize  $C_{11}$ ,  $C_{12}$ , and  $C_{33}$ . Following this normalization, the three quantities are encoded using Learnable Sinusoidal Embedding (LSE). The LSE utilized here is a type of learnable positional encoding that initializes with Sinusoidal Positional Embedding (SPE) [66]. The specific formula for this embedding is defined as:

$$PE_{(pos,2i)} = \sin(pos \cdot w_i \cdot 2\pi), \quad (4)$$

$$PE_{(pos,2i+1)} = \cos(pos \cdot w_i \cdot 2\pi). \quad (5)$$

Here,  $PE$  denotes Positional Embedding, and  $w_i$  represents a learnable parameter initialized as:  $w_i = \frac{1}{10000^{2i/d}}$ ,  $d$  stands for the dimensionality of the embedding vector.

Subsequently, the three encoded vectors are added to the timestep embedding, and the combined result is integrated into each layer of the U-Net through Adaptive Layer Normalization (AdaLN) [67]. According to [65], during training, we randomly set the elastic tensor condition to null with a probability of 20%, and during the generation process, the guidance scale is set to 1.

For the boundary constraint, we encode the boundary information into an image and then append this image to the network's input. Specifically, assuming that we want to use the boundary  $B$  of a particular microstructure as a condition, we first extract its top-left quarter and record the positions of the rows and columns where the upper and left boundaries lie. Then, a new binary image is created (default value 0) where the recorded boundary rows and columns are set to 1, establishing a boundary representation for the given microstructure.

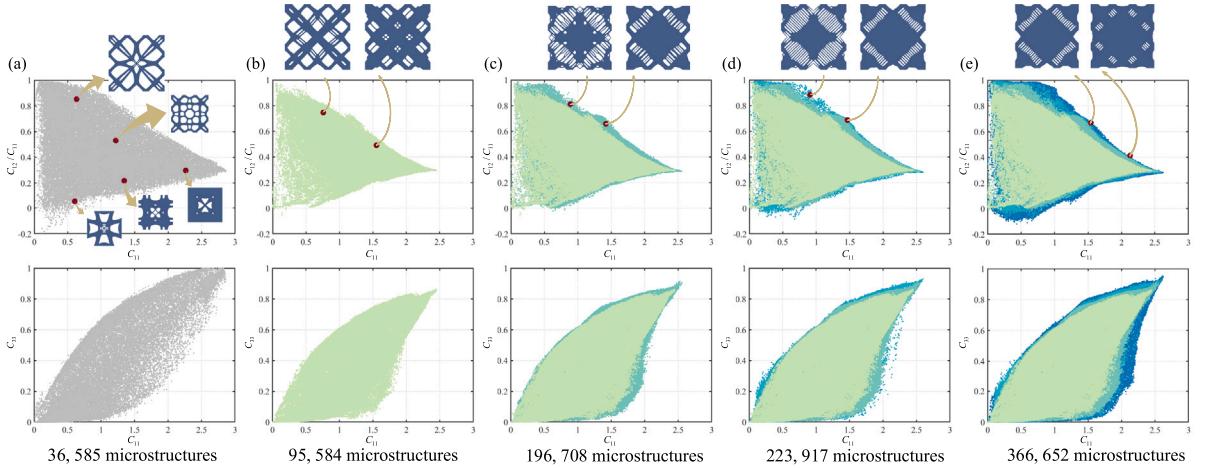
**Network architecture.** The Encoder and Decoder parts of our U-Net both consist of 5 stages. We set the number of resblocks for each stage as 1, and channel multiplications for each stage are set to 1, 2, 4, 8, and 8, respectively. The input and output feature dimensions for each layer are as follows: (32,32), (32,64), (64,128), (128,256), (256,256), (256,256), (256,256), (128,128), (128,64), (64,32), (32,32). Attention resolutions with respect to feature map sizes are set at 4 and 8. The diffusion step number is set to 1000 with a linear noise schedule during training.

**Training details.** The network training is based on the following denoising loss:

$$\mathcal{L}_{x_0} = E_{\epsilon \sim \mathcal{N}(0, I), t \sim \mathcal{U}(0, I)} \|f(x_t, \hat{x}'_0, t, B, C_{11}, C_{12}, C_{33}) - x_0\|^2. \quad (6)$$

We use the AdamW optimizer [68,69] to train the diffusion model. For each model mentioned below, we fix the learning rate to  $10^{-4}$ , batch size to 256, and train for 1000 epochs.

**Generation.** We use the diffusion model to generate diverse microstructures under two conditions: (1) the specified boundary and (2) the predefined elastic tensor matrix. To ensure the diversity of microstructures generated by the network, we utilize a set of elastic tensors with a broad range of values as conditions during sampling. To create this set of elastic tensors that encompasses a wide and reasonable range, we first compute the SDF values at each node of a Cartesian grid covering the property space of the training dataset [70,71], as illustrated in Fig. 6(a). Then, we set the SDF values to negative for the small neighborhood of nodes where the SDF is less than 0. This ensures that the sampling range is not limited to the coverage range of the training dataset. Finally, we perform sampling in the regions where the SDF is less than 0 to obtain the elastic tensors required for generation. Fig. 6(b) displays the expanded range of the SDF and the elastic tensors obtained from the sampling. We input these elastic tensors and the encoding of the target boundary together as conditions into the network and use 50 steps of DDPM sampling for microstructure generation. Fig. 6(c) shows microstructures generated by our generative model that lie outside the space of the training set.



**Fig. 7.** Overview of dataset generation framework: (a) Dataset  $D_0$  with microstructures exhibit diverse boundaries. (b) One identical boundary dataset constructed from the diffusion model with a single seed boundary. (c)–(d) Active learning process. (e) The final expanded dataset.

*Active learning strategy.* Building upon the aforementioned generation process, we can iteratively expand a boundary-identical dataset, as depicted in Fig. 7. This expansion involves incorporating the generated microstructures into the existing dataset to form an augmented training set, which is used to further train the network and produce subsequent generations. Through this iterative augmentation process, we can construct a large-range boundary-identical microstructure dataset.

### 3.4. Multiscale topology optimization

When a dataset with large-range identical boundaries, denoted as  $D$ , is constructed, multiscale topology optimization becomes significantly more efficient, as it can be implemented directly without concerns regarding connectivity issues. We illustrate the multiscale optimization model using the invisibility cloak [28] as an example. The reference design space is denoted as  $\Omega$ , and the mechanical cloak is defined at the inside of  $\Omega$  and marked  $\Omega_c$ . The properties of the cloak are evaluated in the remaining areas  $\Omega_s = \Omega / \Omega_c$ , and the stealth capabilities of the cloak are deemed better when the difference between the displacement and the reference displacement is minimized. The elastic tensors,  $C$ , are employed as the design variables. The objective function is defined as the deviation between the calculated displacement  $\mathbf{u}_o$  and the targeted displacement  $\mathbf{u}_t$  in  $\Omega_s$ . The constraints are determined by the projection of the equilibrium equation and the elastic tensor within the design space,  $S$ . The optimization model is formulated as follows:

$$\min_{\mathbf{C}} \|\mathbf{u}_o - \mathbf{u}_t\|_2^2 \quad (7)$$

$$\text{s.t. } \mathbf{K}(\mathbf{C})\mathbf{u} = \mathbf{F}, \quad (8)$$

$$\varphi(\mathbf{C}_e, f_S) \leq 0, \quad \forall e; \quad (9)$$

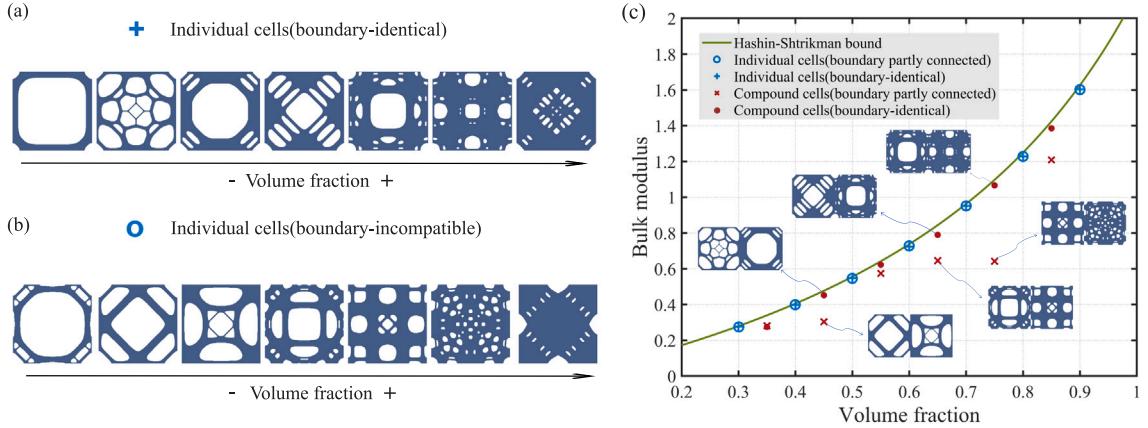
Here,  $\mathbf{K}(\mathbf{C})$  represents the stiffness matrix, and  $\mathbf{F}$  denotes the external force vector. The function  $\varphi(\cdot, \cdot)$  indicates that the elastic tensor  $\mathbf{C}_e$  of macrostructure element  $e$  ensures the SDF ( $f_S$ ) remains less than zero. Then, we find  $s_e \in D$  that minimizes:

$$\mathbf{s}_e = \arg \min_{s \in D} \|\mathbf{C}(s) - \mathbf{C}_e\|_2, \quad (10)$$

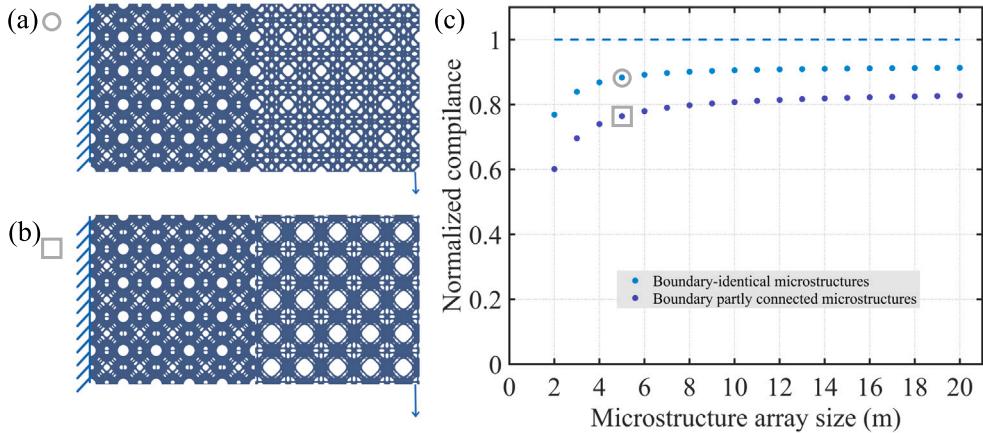
Finally, optimized microstructure  $\mathbf{s}_e$  is obtained and assembled into the final multiscale structure.

## 4. Numerical experiments and discussion

We validate the proposed microstructure datasets through various numerical examples. The training and inference of our network, as well as the full-scale finite element analysis for multi-scale structures, are conducted on a server equipped with 2 CPUs (Intel Xeon Silver 4316 2.30 GHz), 512 GB of RAM, and 8 Nvidia GeForce RTX 3090 GPUs. The multi-scale design process is carried out on a computer with 1 CPU (Intel i7-6700K 4.00 GHz) and 16 GB of RAM. The material has Young's modulus of  $E = 3$  and Poisson's ratio of  $\nu = 0.3$ . Linear elements are used to perform finite element analysis.



**Fig. 8.** (a) A set of individual cells with identical boundaries that achieve the Hashin–Shtrikman (HS) upper bound, with volume progressively increasing from left to right. (b) A set of individual cells that also reach the HS upper bound but have partially connected boundaries, similarly showing a progressive increase in volume from left to right. (c) The relationship between the bulk modulus and volume fraction for both sets of individual cells, as well as for compound cells formed by adjacent cells within the same group.



**Fig. 9.** A cantilever beam with two microstructural regions: (a) filled with boundary-identical structures (b) filled with partly connected boundary structures. (c) The relationship between normalized compliance and the array size  $m$ .

#### 4.1. Identical boundaries discussion

**Compound microstructures testing.** The boundary connectivity between microstructures plays a critical role in determining the mechanical properties of the resulting composite structures. To evaluate the mechanical compatibility of the boundary-identical microstructures we design, we compare two sets: one with boundary-identical microstructures and another with microstructures that are only partially connected at the boundaries.

We plot the bulk modulus for both individual cells and compound states alongside the theoretical Hashin–Shtrikman (HS) bounds. As illustrated in Fig. 8, while all selected single cells approach the HS bounds, the compound structures formed from microstructures with partial boundary connections exhibit a significant gap between their bulk modulus and the HS bounds. In contrast, compound structures composed of boundary-identical microstructures closely align with the HS bounds. This behavior arises from the degree of boundary connectivity, which directly influences mechanical performance. When boundaries are only partially connected, overall performance degrades.

**Full scale analysis.** The multiscale design, based on homogenization theory, assumes scale separation, meaning that the theory evaluates the properties of an infinitely periodic array of microstructures. However, achieving such an infinite array in practical manufacturing is not possible, leading to inevitable discrepancies between the properties predicted by homogenization theory and those of actual multiscale structures.

To examine the impact of boundary connectivity between structures, we conduct an experiment involving a cantilever beam composed of two regions, each filled with  $m \times m$  identical microstructures, as illustrated in Fig. 9. In Fig. 9(b), the microstructures

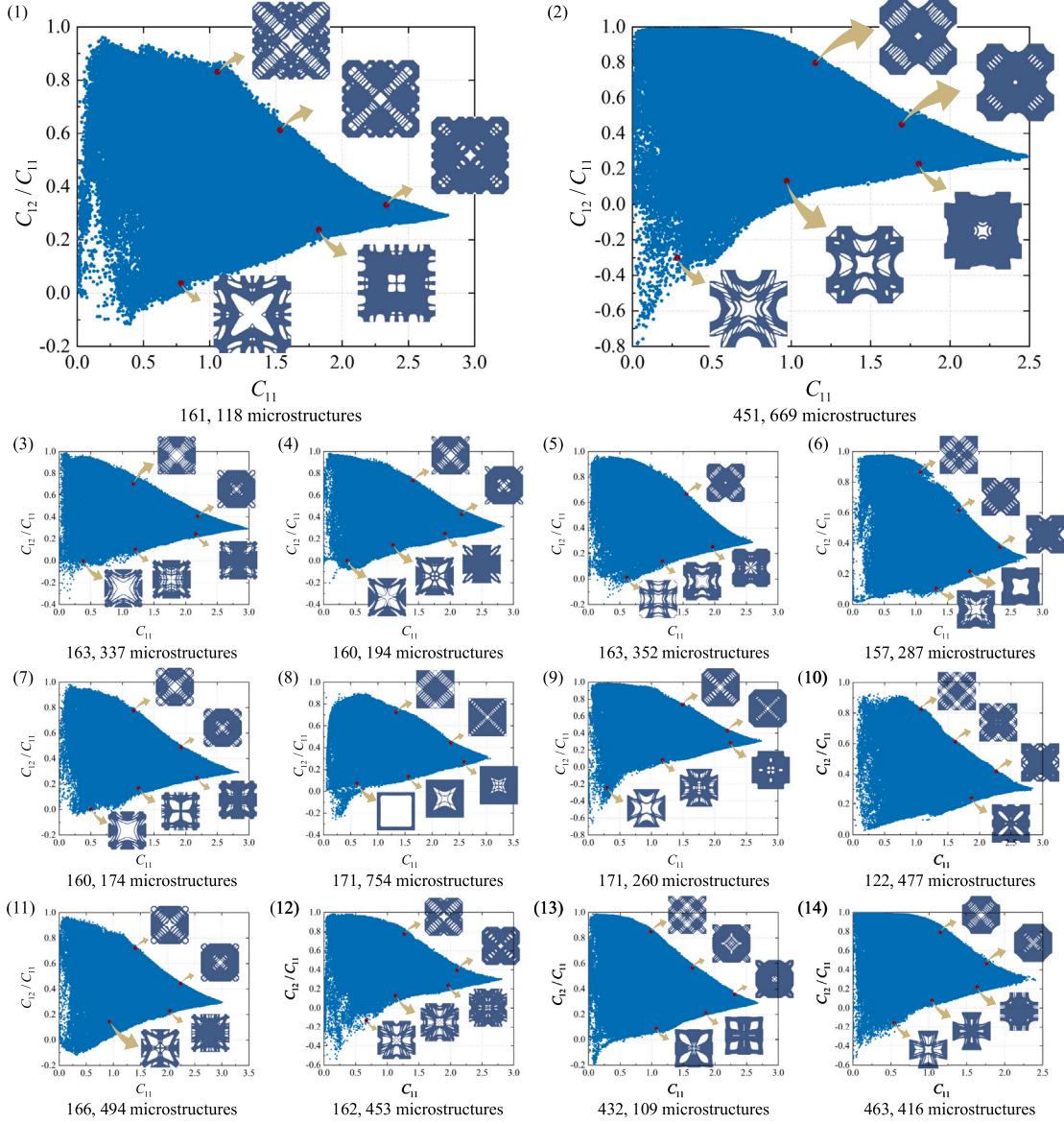


Fig. 10. The generated 14 large-range boundary-identical microstructure datasets.

are only partially connected at the boundaries, while in Fig. 9(a), the boundaries are identical. In both cases, the repeating number is 5. Importantly, the structures filling the same regions in both cantilever beams possess identical elastic tensors.

Fig. 9(c) plots the normalized compliance against  $m$ , where normalized compliance is defined as the ratio of compliance from homogenization analysis to that from full-scale analysis. As shown in Fig. 9(c), the normalized compliance approaches 1 as the resolution parameter  $m$  increases. Structures with identical boundaries exhibit higher normalized compliance, indicating that multiscale designs utilizing more boundary-identical microstructures yield properties closer to those predicted by homogenization theory.

**Boundary-identical microstructure datasets analysis.** We present the property coverage of 16 boundary-identical large-range microstructure datasets shown in Figs. 10 (datasets 1–14), 2 (dataset 15), and 7 (dataset 16). Each dataset contains over 100,000 microstructures. These datasets can be used for multiscale design or other data-driven researches. We conduct an analysis of the bulk modulus of all microstructures in the aforementioned datasets and compare them with the Hashin-Shtrikman bound (HS-bound) [72]. The structures achieve 98% or higher of the HS-bound. Details are shown in Fig. A.1. This indicates that our model has the potential to generate extreme materials with specified boundaries.

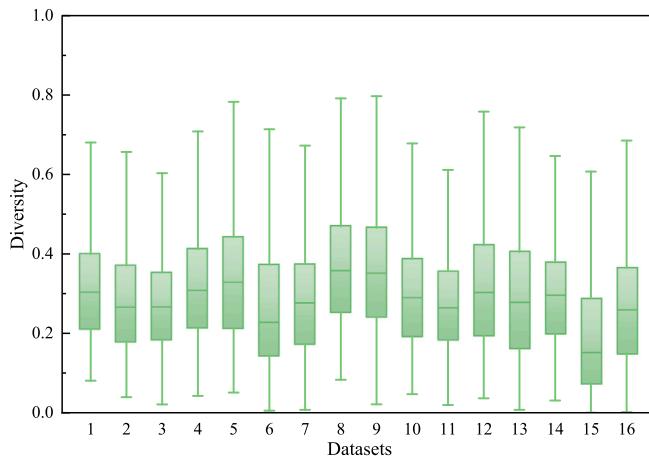


Fig. 11. The diversity statistics across the 16 datasets.

To further analyze the diversity of microstructures, we focus on the novelty of the generated microstructures relative to those in the dataset  $D_0$ . The diversity here is primarily assessed by comparing the novel data generated by the network with the microstructures contained in the dataset  $D_0$ . Thus, the structure diversity (SD) is defined as:

$$SD(s) = 1 - SS(s). \quad (11)$$

where  $SS(s)$  is the structure similarity defined in Eq. (1). The value signifies that as it approaches 0, the current dataset becomes more similar to the reference dataset  $D_0$ , whereas a value closer to 1 indicates greater novelty. Fig. 11 presents the diversity results across 16 datasets.

#### 4.2. Network performance analysis

To assess the capability of the self-conditioning diffusion model to generate microstructures under specified conditions, we perform conditional generation tests on a network that has been trained using the dataset  $D_0$ . This dataset comprises various microstructures characterized by diverse boundaries  $B$  and elastic tensors  $C$ . The diversity within the training data enhances the model's generalization ability and presents a significant challenge in generating microstructures that strictly conform to specified boundaries and elastic tensors. Consequently, conducting conditional generation tests on this trained network can effectively reveal the true performance of the model structure.

We evaluate the network's performance under all 16 boundaries, using two primary metrics: (1) the  $R^2$ -score between the generated elastic tensors and the target elastic tensors, which is defined as:

$$R^2(\mathbf{Y}, \mathbf{Y}') = 1 - \frac{\sum_{i=1}^N \|\mathbf{y}_i - \mathbf{y}'_i\|^2}{\sum_{i=1}^N \|\mathbf{y}_i - \bar{\mathbf{y}}\|^2}, \quad (12)$$

where  $\mathbf{Y}$  and  $\mathbf{Y}'$  are the generated and target elastic tensor matrix, respectively, each with the dimension of  $k \times N$ .  $N (= 1000)$  is the number of generated microstructures.  $\mathbf{y}_i$  is the  $i$ th column of  $\mathbf{Y}$ . For  $k = 1$ ,  $\mathbf{y}_i$  corresponds to a single elastic tensor component ( $C_{11}$  or  $C_{12}$  or  $C_{33}$ ) of the  $i$ th generated microstructure. When  $k = 3$ ,  $\mathbf{y}_i$  is the elastic tensor vector ( $[C_{11}, C_{12}, C_{33}]$ ) of the  $i$ th generated microstructure.  $\mathbf{y}'$  follows a similar explanation.  $\bar{\mathbf{y}}$  is the mean of the generated properties.

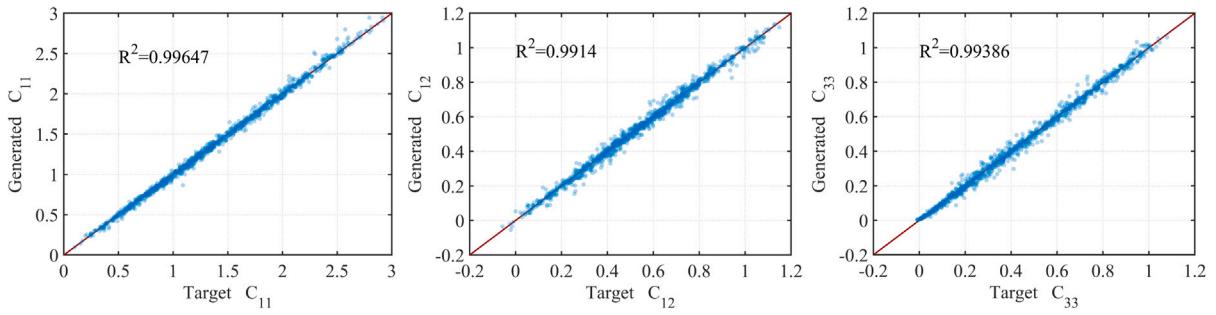
(2) The degree of match (DoM) between the generated boundaries and the specified boundaries, which is calculated as:

$$DoM(\mathbf{b}, \mathbf{b}') = 1 - \frac{\|\mathbf{b} - \mathbf{b}'\|_2}{\|\mathbf{b}\|_2},$$

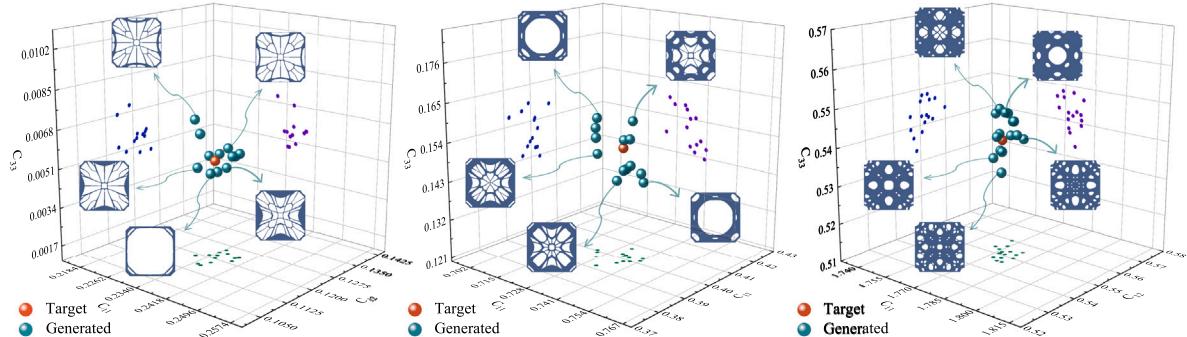
where  $\mathbf{b}$  represents the target boundary, and  $\mathbf{b}'$  is the boundary of the generated microstructure. The results indicate that the model performs consistently well across all 16 boundaries without exhibiting significant overfitting to any particular boundary. The values of  $R^2$ -score and DoM are shown in Tables A.1–A.2.

Furthermore, the test results for the 15th dataset are visualized in Fig. 12. It shows the  $R^2$ -scores corresponding to three components of the elastic tensor:  $C_{11}$ ,  $C_{12}$ , and  $C_{33}$ . These results demonstrate that the microstructures generated by the network exhibit good agreement with the target elastic tensors.

Fig. 13 illustrates the diversity of the network's generative capabilities, demonstrating that the proposed model can generate multiple structures with similar elastic tensors but distinctly different geometric configurations. By keeping the boundary conditions constant and using the same target elastic tensor across multiple generations, the results emphasize the model's ability to capture a one-to-many mapping from material properties to geometry. This highlights the model's capacity to produce a broad range of geometrically distinct microstructures that exhibit comparable elastic properties.



**Fig. 12.** The  $R^2$ -score demonstrates the disparity between the properties of the microstructures generated by the network and the target properties.



**Fig. 13.** The generated results for three different predefined elastic tensors under the 15th boundary.

#### 4.3. Ablation study

To investigate the impact of the self-conditioning mechanism and the encoding of elastic tensors on the network's generation performance, we design two ablation studies that focus on these aspects.

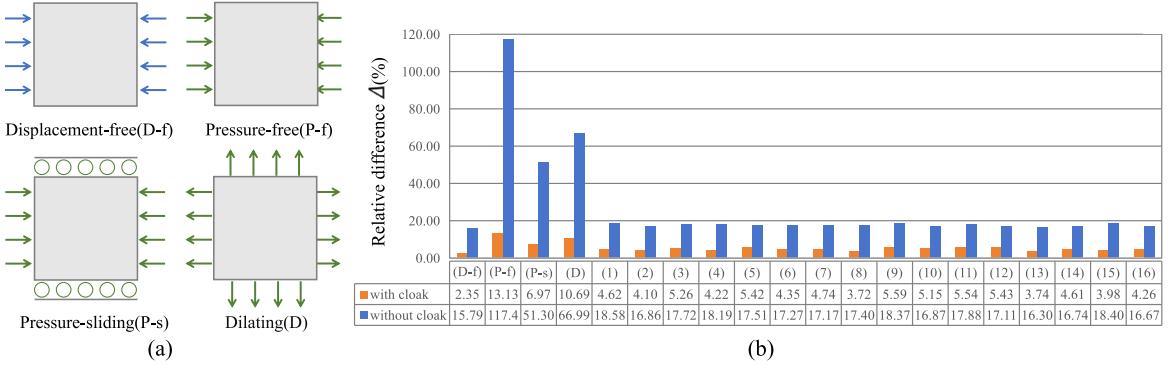
*The ablation study on self-conditioning.* Self-conditioning has yielded excellent results in image generation tasks. To verify its effectiveness in generating microstructures, we conduct the following experiments. For the baseline model, we use the architecture detailed in Fig. 5 and train a variant without the self-conditioning mechanism as a control. The training dataset,  $\mathcal{D}_0$ , is selected as described in Section 3.2. Generation tests are conducted under all 16 boundaries, evaluating accuracy using the  $R^2$ -score for elastic tensors and the alignment between generated and specified boundaries.

Compared to the baseline model, the model lacking self-conditioning exhibits an average decrease of 0.0516 in  $R^2$ -score, with a maximum reduction of 0.1120 (Table A.1). Moreover, there is an average decrease of 0.0255 in the DoM, with a maximum reduction of 0.7000 (Table A.2). These test results indicate that removing the self-conditioning mechanism leads to a significant reduction in generation accuracy. This underscores the critical role of self-conditioning in enhancing the precision of the model's output.

*The ablation study on elastic tensor embedding.* In our network, elastic tensor embeddings are integrated into each U-Net layer via AdaLN. An ablation study evaluates their impact on model performance, using the original network as the baseline. Three variants omit embeddings from the high, middle, or low layers, corresponding to the upsampling path, bottleneck, and downsampling path, respectively. All models are trained on dataset  $\mathcal{D}_0$ , with sampling experiments conducted across the sixteen specified boundaries.

Compared to the baseline model, the three variant models show the following reductions in  $R^2$ -score: an average decrease of 0.0221, 0.0398, and 0.0426, with maximum reductions of 0.8800, 0.1470, and 0.1600 (Table A.1). Regarding the value of DoM, the average decreases are 0.0149, 0.0163, and 0.0187, with corresponding maximum reductions of 0.0380, 0.0580, and 0.0510 (Table A.2).

Our findings suggest that omitting elastic tensor embeddings from certain layers negatively impacts the model's generative accuracy, particularly in its ability to align with the target elastic tensor. This effect is most pronounced in specific boundary sampling experiments. To maintain the accuracy and robustness of the model, it is essential to include elastic tensor embeddings in every layer of the U-Net.



**Fig. 14.** (a) Four initial settings. (b) Relative displacement differences of the cloak design result under different boundary conditions, void shapes, and datasets, where designs D-f, P-f, P-s, and D are the four designs presented in Fig. 15, and designs (1) to (16) are the sixteen designs illustrated in Fig. 16. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

#### 4.4. Active learning analysis

Figs. 6 and 7 illustrate the iterative process of expanding the property space. To further quantify the computation, we first define the coverage ratio of the space and then calculate these ratios achieved through the active learning strategy.

Given that the elastic tensors we consider have only three independent components  $C_{11}$ ,  $C_{12}$ , and  $C_{33}$ , we conceptualize a three-dimensional space defined by these components, denoted as  $C_{11} \times C_{12} \times C_{33}$ . The elastic tensors of all microstructures in each dataset can be visualized as a point cloud  $P$  within this 3D space. Assuming the value ranges for the three elastic tensors are  $[C_{11}^{\min}, C_{11}^{\max}]$ ,  $[C_{12}^{\min}, C_{12}^{\max}]$ , and  $[C_{33}^{\min}, C_{33}^{\max}]$  respectively, we divide the cuboid in this 3D space into smaller cubes using a resolution of  $N_1 \times N_2 \times N_3$ , labeled as  $\{c_i\}_{i=1}^{N_1 \times N_2 \times N_3}$ . We use the following metric to express the coverage of elastic tensors:

$$\text{Range} = \frac{\#\{c_i | c_i \cap P \neq \emptyset, 1 \leq i \leq N_1 \times N_2 \times N_3\}}{N_1 \times N_2 \times N_3}.$$

This ratio indicates the proportion of small cubes that contain at least one point from the point cloud  $P$ , reflecting the extent to which the elastic tensor space is covered by the dataset. In practical computations, we set the following parameters:  $C_{11}^{\min} = 0$ ,  $C_{11}^{\max} = 3.5$ ,  $C_{12}^{\min} = -0.5$ ,  $C_{12}^{\max} = 1.5$ ,  $C_{33}^{\min} = 0$ ,  $C_{33}^{\max} = 1.5$ , resolution:  $N_1 = 350$ ,  $N_2 = 200$ ,  $N_3 = 150$ . These minimum and maximum values for the elastic tensor components are determined based on the statistical properties of all microstructures in the dataset. In the selection of resolution, we refer to the value ranges of each component to ensure that the scale intervals on the three axes are identical. Additionally, we appropriately chose the order of magnitude for the resolution to ensure that the partitioning of the elastic tensor space can effectively capture the coverage of the elastic tensors without being overly sparse or excessively dense.

On average, each iteration enhances the dataset's elastic tensor coverage by 6.83%, with a maximum expansion of 13.87%. Detailed statistics are provided in Table A.3. These results highlight the effectiveness of active learning methods in iteratively expanding datasets.

#### 4.5. Multiscale topology optimization

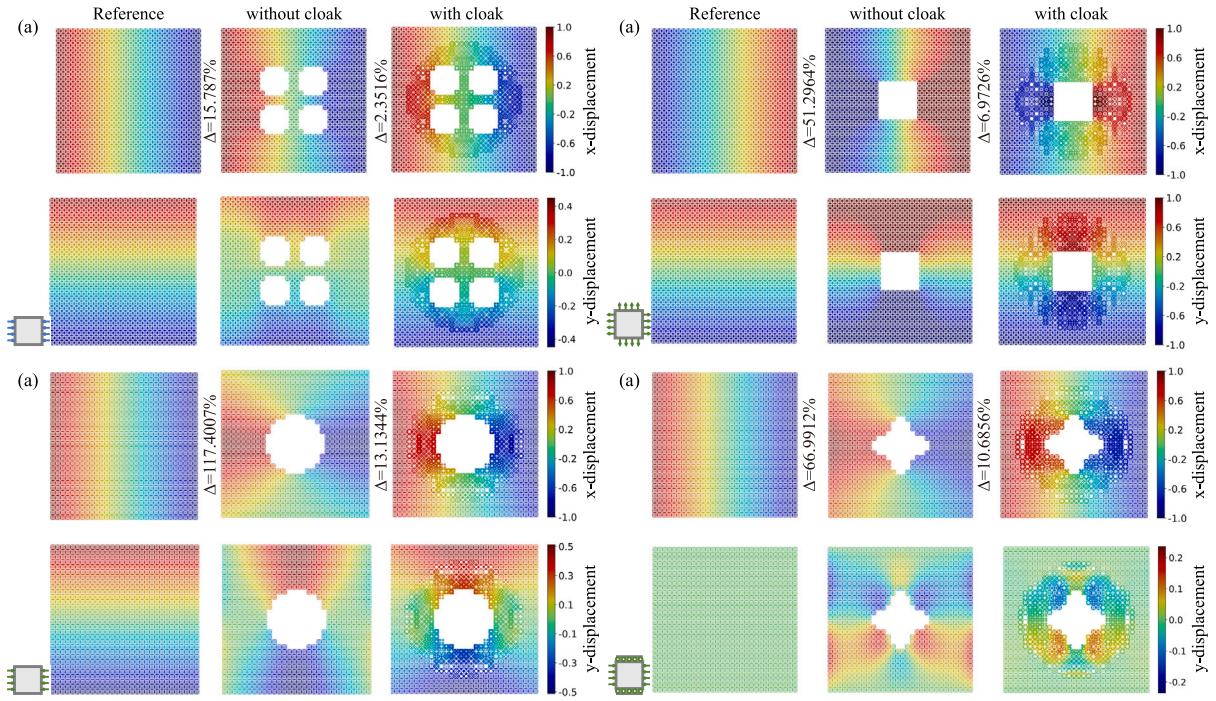
##### 4.5.1. Mechanical cloak design

For mechanical cloak design, the reference structures (without voids and cloak) are composed of  $30 \times 30$  periodically tessellated base cells chosen in the microstructure dataset. The mechanical cloak is defined in a circle of radius 12 at the center of the design domain.

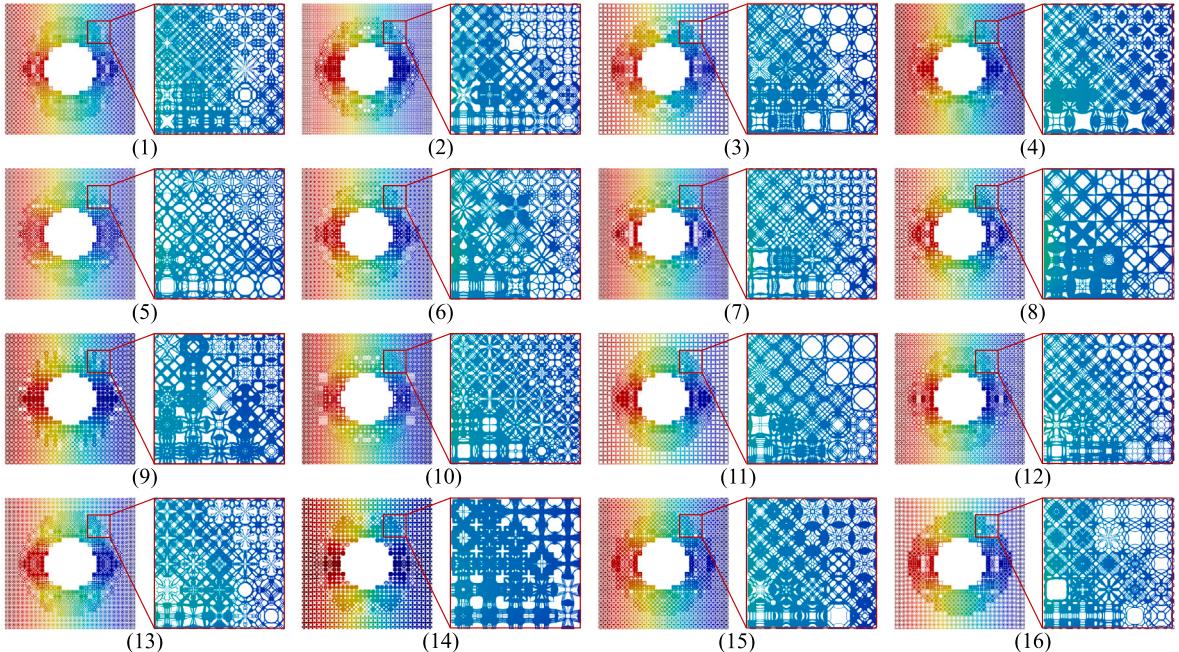
As shown in Fig. 14(a), four different boundaries are employed. Notably, the blue arrow signifies that the designated displacement represents the boundary condition  $u_{bc}$ , while the green arrow indicates that the specified force serves as the boundary condition  $p_{bc}$ . Fig. 14(b) illustrates the relative displacement difference before and after the incorporation of the mechanical cloak. For a quantitative measure of the distortion of the displacement field, the relative displacement difference  $\Delta$  is formulated as

$$\Delta = \frac{\sqrt{\sum_{\Omega_s} (\mathbf{u}_o - \mathbf{u}_t)^2}}{\sqrt{\sum_{\Omega_s} (\mathbf{u}_t)^2}}, \quad (13)$$

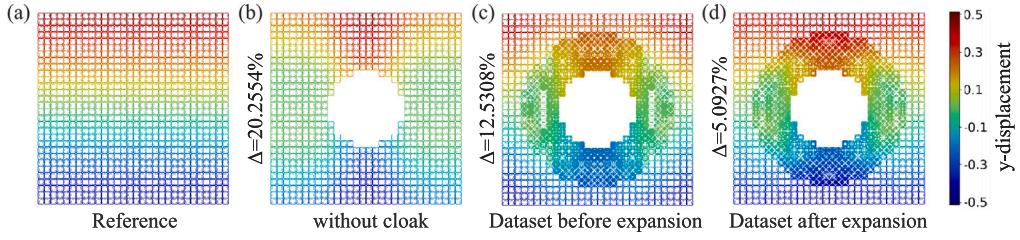
where  $\mathbf{u}_o$  and  $\mathbf{u}_t$  are the node displacement of mechanical cloak and reference structures. The difference, as indicated by the red bar with the cloak applied, is significantly lower compared to the blue bar without the cloak, especially for the second boundary



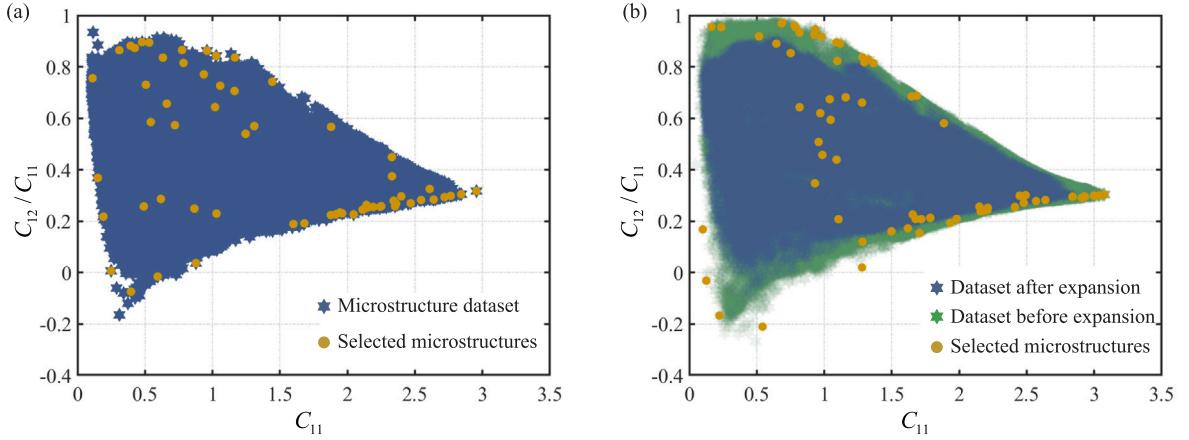
**Fig. 15.** Mechanical cloak design results under different boundary conditions and void shapes using microstructure dataset (1). (a) The results of cloak structure for four circular voids under displacement-free boundary condition (b) The results of cloak structure for circular void under pressure-free boundary condition (c) The results of cloak structure for square void under dilating boundary condition (d) The results of cloak structure for star void under pressure-sliding boundary condition.



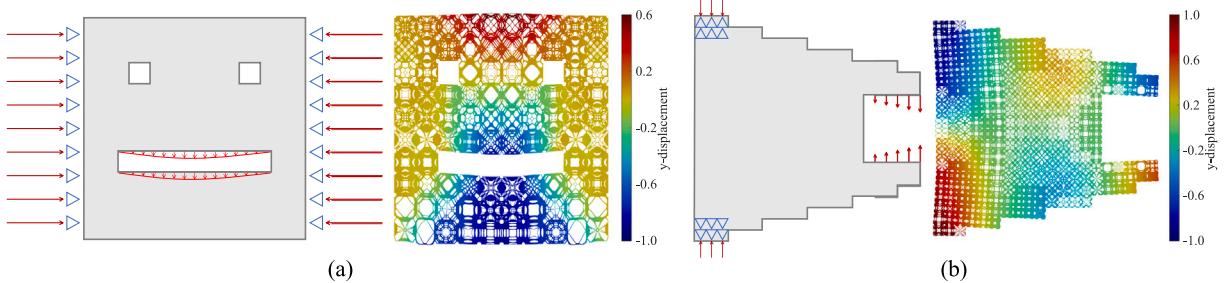
**Fig. 16.** Cloak design results under displacement-free boundaries and circular void using 16 different datasets. The red box shows a partial enlargement of the cloak. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 17.** Mechanical cloaks designed using different datasets, where colors indicate the  $y$ -direction displacement. (a) The original structure, (b) Structure with void, (c) Structure with a cloak designed using the dataset before expansion, (d) Structure with a cloak designed using the final dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 18.** Range of properties for different microstructure datasets and the properties of selected microstructures for mechanical cloak design: (a) dataset before expansion (b) dataset after expansion.

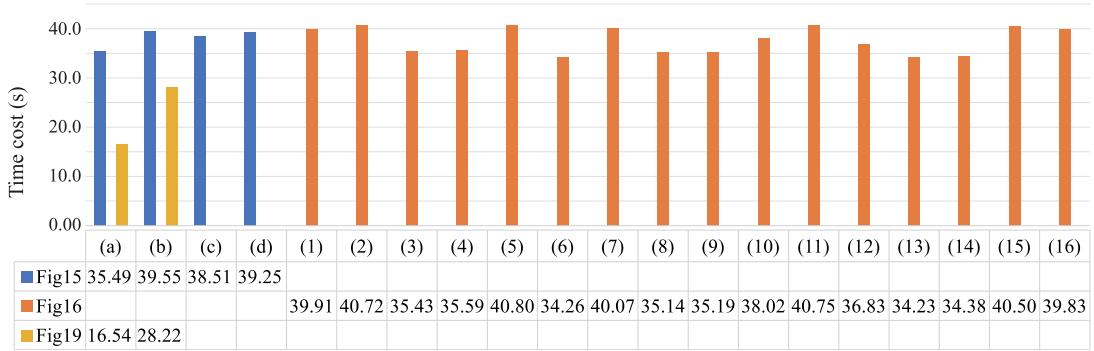


**Fig. 19.** The target settings and displacement simulation of optimized results for (a) smiley face filled with  $10 \times 10$  macro unit cells and (b) gripper filled with  $20 \times 20$  macro unit cells.

(13.13% vs. 117.40%).

Fig. 15 depicts the outcomes of the cloak design for four distinct boundary conditions using the microstructure dataset (1) shown in Fig. 2. In Fig. 15(b) and (d), it is evident that under the pressure-free and dilating boundary conditions, the displacements in the  $x$  and  $y$  directions in the results without the cloak significantly deviate from those of the reference structures. The addition of the cloak structure notably reduces the displacement gap. The dataset generated by our method shown in Fig. 2 is capable of addressing complex deformation scenarios while maintaining full boundary connectivity, thus reducing the error of the above four examples to less than 13.2% .

To validate the design capabilities of mechanical cloaks for additional datasets, we conducted tests on 16 datasets shown in Fig. 10 under displacement-free boundary conditions, each featuring a hollow circular hole in the center. The results and statistics are shown in Fig. 16 and Fig. 14, respectively. The results obtained from the analysis of 16 datasets revealed a maximum difference of 5.6%. The generated datasets encompass a broad spectrum and has the potential to facilitate the design of mechanical cloaks.



**Fig. 20.** The statistical results for the optimization time of multiscale designs in Figs. 15, 16, 19 include optimizing the elastic modulus of macroscopic elements and searching for the microstructure corresponding to the target modulus.

To evaluate the impact of dataset iterative expansion on multiscale design outcomes, we conduct the following experiment: We select the 15th dataset, both before and after expansion, to perform identical mechanical cloak designs. Before filling in the mechanical cloaks, the relative displacement difference was  $\Delta = 20.2554\%$ . Through multiscale topology optimization, two mechanical cloak designs as shown in Fig. 17 are obtained. The design using the 15th dataset before expansion obtains a relative displacement difference of  $\Delta = 12.5308\%$ , while the expansion dataset yields a difference of  $\Delta = 5.0927\%$ .

The positions of elastic tensors for the microstructures used in the mechanical cloaks are shown in Fig. 18. Most of microstructures are located at the periphery. The cloak designed with the expanded dataset outperforms that of the pre-expansion dataset, likely due to the latter's narrower property coverage, which limited further performance improvements. This highlights the benefit of broader property coverage for this problem and confirms that dataset coverage significantly impacts design performance.

#### 4.5.2. Customized displacement design

For the second application, two examples (a face and a gripper) are designed to show the effect of customized displacement design. For the multiscale structure design of the face, we opt for a combination of  $10 \times 10$  unit cells and utilize the microstructure dataset presented in Fig. 2 for the selection and assembly of microstructures. When pressure is applied to both sides of the face, the mouth will exhibit a downward curve, creating the appearance of a smile expression. Likewise, in the gripper design, a grid of  $20 \times 20$  unit cells is chosen, and the microstructure dataset (5) in Fig. 10 served as the microstructure library for optimal design. When the gripper is compressed vertically on the left side, the structure on the right side contracts towards the central axis. Simulation results indicate that the optimized outcomes align closely with the specified displacement shown in Fig. 19. Applying the calculation formula in Eq. (13), the relative differences for the face and gripper are 12.19% and 8.70%, respectively.

#### 4.5.3. Optimization efficiency

The present study underscores a notable enhancement in the algorithm's efficiency, evidenced by a substantial reduction in the computation time. The computational time for multiscale topology optimization is primarily allocated to two tasks: (1) optimizing the elastic modulus of macroscopic elements and (2) searching for the microstructure corresponding to the target modulus. Specifically, the algorithm has achieved a remarkable improvement, all tested examples are completed in one minute. The specific time is shown in Fig. 20. In the data-driven approach [29] for computing top-down multiscale problems, conducting a single-cell search to ensure connectivity imposes a considerable time burden.

## 5. Conclusion and future work

This paper introduces 16 large-range cubic symmetric microstructure datasets with identical boundaries and a wide range of elastic moduli generated through a deep generative model. The self-conditioning diffusion model derives microstructures under the specified boundary and the pre-defined homogenized elastic tensor. Initially, an inverse homogenization approach is utilized to generate the optimized dataset. Subsequently, the microstructure boundaries of the dataset are clustered to categorize it into 16 cases. Then, through the expansion process, a series of datasets with a broad range of elastic moduli and identical boundaries are generated. We showcase the data generation capabilities of this generative model by designing two multiscale structures: mechanical cloaks and structures with specified displacements. By employing numerous examples featuring varying numbers and shapes of voids and diverse boundaries, we illustrate that our method excels in multiscale systems. This superiority is validated through numerical simulations and computational efficiency. Besides, this results in a substantial reduction in calculation time.

**Table A.1**Generation accuracy evaluated by the  $R^2$ -score of different models under 16 boundaries.

Methods	1	2	3	4	5	6	7	8
Baseline	0.958	0.950	0.962	0.957	0.968	0.971	0.985	0.977
w/o self-conditioning	0.870	0.924	0.939	0.920	0.925	0.921	0.941	0.936
w/o low-level layers elastic tensor embedding	0.896	0.917	0.960	0.944	0.951	0.970	0.985	0.974
w/o middle layers elastic tensor embedding	0.906	0.875	0.948	0.956	0.922	0.969	0.918	0.973
w/o high-level layers elastic tensor embedding	0.840	0.881	0.961	0.955	0.911	0.934	0.984	0.970
Methods	9	10	11	12	13	14	15	16
Baseline	0.963	0.967	0.971	0.941	0.950	0.956	0.994	0.975
w/o self-conditioning	0.914	0.932	0.859	0.899	0.847	0.914	0.935	0.943
w/o low-level layers elastic tensor embedding	0.963	0.950	0.883	0.936	0.902	0.948	0.975	0.937
w/o middle layers elastic tensor embedding	0.923	0.939	0.883	0.934	0.803	0.914	0.971	0.974
w/o high-level layers elastic tensor embedding	0.957	0.914	0.811	0.924	0.859	0.919	0.970	0.973

Three-dimensional microstructures find more widespread applications, and we intend to explore the extension of the current results to three-dimensional microstructures in future work. This extension poses two potential challenges: (1) Generating voxel-based 3D microstructure libraries demands significant computational time and memory resources. Balancing generation efficiency with effectiveness becomes a critical consideration. (2) Classifying boundaries for 3D microstructures is notably more complex than for their 2D counterparts, which presents a challenging task in determining reasonable boundary categories for 3D microstructures.

#### CRediT authorship contribution statement

**Jingxuan Feng:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization. **Lili Wang:** Writing – review & editing, Writing – original draft, Visualization, Data curation, Conceptualization. **Xiaoya Zhai:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Conceptualization. **Kai Chen:** Writing – review & editing, Conceptualization. **Wenming Wu:** Writing – review & editing, Supervision. **Ligang Liu:** Supervision, Funding acquisition. **Xiao-Ming Fu:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Conceptualization.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62402467, No. 61972368, No. 62025207), the National Key R&D Program of China (No. 2024YFA1016300), the Youth Innovation Key Research Funds for the Central Universities, China (No. YD0010002010), the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDB0640000).

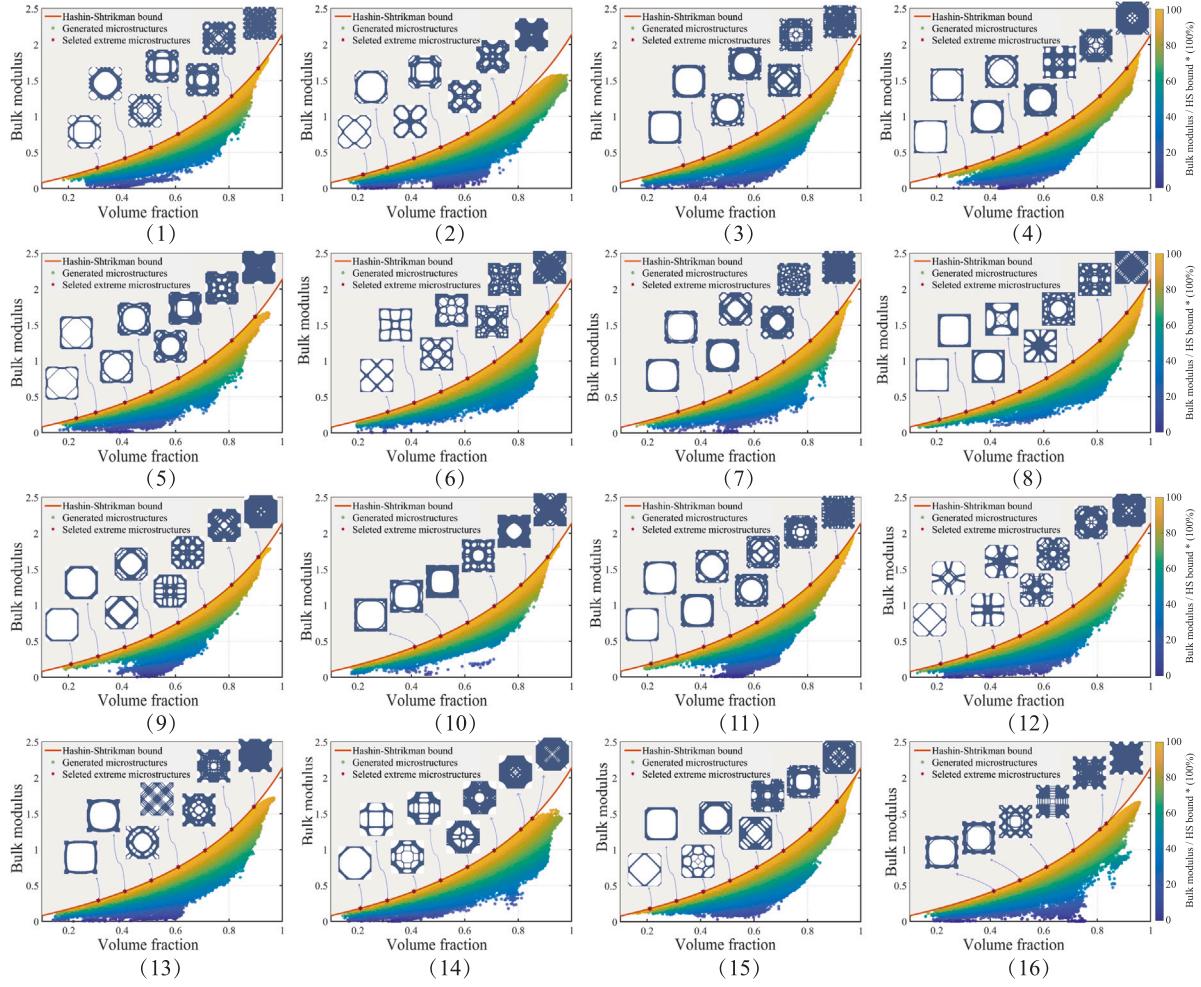
#### Appendix A. Analysis of bulk modulus for the datasets

We analyze the bulk modulus of the 16 datasets constructed in this study and compared them with the Hashin-Shtrikman bound. The results of this comparison are presented in Fig. A.1.

#### Appendix B. Additional results

**Ablation study results.** The study on self-conditioning and elastic tensor embedding examines a base model alongside four variations, each omitting either self-conditioning or elastic tensor embedding at the low, mid, or high-level layers. The models are evaluated based on specified target elastic tensors and 16 boundary conditions. Table A.1 reports the  $R^2$  scores, measuring the correlation between the generated and target elastic tensors. Table A.2 highlights the alignment between the boundaries of the generated microstructures and the corresponding target boundaries.

**Elastic tensor coverage ratios over expansion iterations.** Table A.3 illustrates changes in coverage ratios for the 16 microstructure datasets with identical boundaries during iterative expansion processes using active learning method. It shows improvements in overall coverage expansion. As shown, the expansion iterations result in the effective expansion of the elastic tensor coverage within the microstructure datasets.



**Fig. A.1.** The bulk modulus of the microstructures in the 16 datasets. The color bar indicates the proportion of the microstructure's bulk modulus to the corresponding HS-bound. The red line represents HS-bound and the red star represents the selected microstructures that reach 98% HS-bound. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table A.2**

Generation accuracy evaluated by the DoM value of different models under 16 boundaries.

Methods	1	2	3	4	5	6	7	8
Baseline	0.985	0.993	0.990	0.987	0.991	0.989	0.997	0.994
w/o self-conditioning	0.962	0.976	0.974	0.978	0.973	0.987	0.989	0.985
w/o low-level layers elastic tensor embedding	0.951	0.989	0.986	0.980	0.968	0.982	0.985	0.957
w/o middle layers elastic tensor embedding	0.967	0.981	0.985	0.983	0.974	0.989	0.976	0.992
w/o high-level layers elastic tensor embedding	0.939	0.975	0.983	0.971	0.986	0.978	0.984	0.990
Methods	9	10	11	12	13	14	15	16
Baseline	0.992	0.987	0.991	0.989	0.995	0.994	0.998	0.996
w/o self-conditioning	0.953	0.975	0.921	0.934	0.962	0.926	0.988	0.977
w/o low-level layers elastic tensor embedding	0.986	0.989	0.953	0.971	0.985	0.980	0.983	0.984
w/o middle layers elastic tensor embedding	0.934	0.988	0.963	0.982	0.951	0.987	0.975	0.980
w/o high-level layers elastic tensor embedding	0.979	0.965	0.942	0.980	0.944	0.988	0.984	0.981

**Table A.3**

The statistics of coverage ratios for datasets 1–16 from iterations 0 to 3.

Iterations	1	2	3	4	5	6	7	8
0	0.00785	0.01378	0.00900	0.00848	0.00774	0.00689	0.00778	0.00821
1	0.00845	0.01501	0.00945	0.00909	0.00844	0.00778	0.00863	0.00886
2	0.00884	0.01594	0.00983	0.00990	0.00903	0.00813	0.00923	0.00941
3	0.00915	0.01633	0.01035	0.01018	0.00941	0.00847	0.00942	0.00968
Iterations	9	10	11	12	13	14	15	16
0	0.00858	0.00519	0.00827	0.00765	0.01685	0.01269	0.01601	0.01304
1	0.00961	0.00591	0.00923	0.00825	0.01824	0.01416	0.01796	0.01471
2	0.01015	0.00636	0.00969	0.00920	0.01916	0.01491	0.01968	0.01560
3	0.01059	0.00662	0.01032	0.00986	0.01990	0.01554	0.02006	0.01633

## Data availability

Data will be made available on request.

## References

- [1] U. Leonhardt, Optical conformal mapping, *Sci.* 312 (5781) (2006) 1777–1780.
- [2] A. Greenleaf, Y. Kurylev, M. Lassas, U. Leonhardt, G. Uhlmann, Cloaked electromagnetic, acoustic, and quantum amplifiers via transformation optics, *Proc. Natl. Acad. Sci.* 109 (26) (2012) 10169–10174.
- [3] A. Rafsanjani, Y. Zhang, B. Liu, S.M. Rubinstein, K. Bertoldi, Kirigami skins make a simple soft actuator crawl, *Science Robotics* 3 (15) (2018) eaar7555.
- [4] G. Feng, G. Pan, Y. Feng, X. Zhang, L. Qiu, Topology optimization of controllable porous microstructure with maximum thermal conductivity, *Int. J. Heat Mass Transfer* 220 (2024) 124990.
- [5] J. Dirker, J.P. Meyer, Topology optimization for an internal heat-conduction cooling scheme in a square domain for high heat flux applications, *J. Heat Transfer* 135 (11) (2013) 111010.
- [6] C. Chu, G. Graf, D.W. Rosen, Design for additive manufacturing of cellular structures, *Comput.-Aided Des. Appl.* 5 (5) (2008) 686–696.
- [7] A.D. Cramer, V.J. Challis, A.P. Roberts, Microstructure interpolation for macroscopic design, *Struct. Multidiscip. Optim.* 53 (2016) 489–500.
- [8] Y. Han, W.F. Lu, A novel design method for nonuniform lattice structures based on topology optimization, *J. Mech. Des.* 140 (9) (2018) 091403.
- [9] L. Cheng, J. Bai, A.C. To, Functionally graded lattice structure topology optimization for the design of additive manufactured components with stress constraints, *Comput. Methods Appl. Mech. Engrg.* 344 (2019) 334–359.
- [10] X. Liu, L. Gao, M. Xiao, Y. Zhang, Kriging-assisted design of functionally graded cellular structures with smoothly-varying lattice unit cells, *Comput. Methods Appl. Mech. Engrg.* 390 (2022) 114466.
- [11] D. Li, N. Dai, Y. Tang, G. Dong, Y.F. Zhao, Design and optimization of graded cellular structures with triply periodic level surface-based topological shapes, *J. Mech. Des.* 141 (7) (2019) 071402.
- [12] E. Garner, H.M. Kolken, C.C. Wang, A.A. Zadpoor, J. Wu, Compatibility in microstructural optimization for additive manufacturing, *Addit. Manuf.* 26 (2019) 65–75.
- [13] Y. Wang, L. Wang, Z.-d. Ma, T. Wang, Parametric analysis of a cylindrical negative Poisson's ratio structure, *Smart Mater. Struct.* 25 (3) (2016) 035038.
- [14] S.T. Zobaer, A. Sutradhar, An energy-based method for interface connectivity of incompatible microstructures through parametric modeling, *Comput. Methods Appl. Mech. Engrg.* 370 (2020) 113278.
- [15] Y. Zhao, Y. Wang, J. Hao, Y. Wang, K. Wang, S. Tai, Study on mechanical properties of cellular structures with negative Poisson's ratio based on the development of abaqus plug-in tool, *Compos. Struct.* 322 (2023) 117348.
- [16] C. Schumacher, B. Bickel, J. Rys, S. Marschner, C. Daraio, M. Gross, Microstructures to control elasticity in 3d printing, *ACM Trans. Graph.* 34 (4) (2015).
- [17] Y. Wang, F. Chen, M.Y. Wang, Concurrent design with connectable graded microstructures, *Comput. Methods Appl. Mech. Engrg.* 317 (2017) 84–101.
- [18] X. Zhai, W. Wang, F. Chen, J. Wu, Topology optimization of differentiable microstructures, *Comput. Methods Appl. Mech. Engrg.* 418 (2024) 116530.
- [19] Z. Du, X.-Y. Zhou, R. Picelli, H.A. Kim, Connecting microstructures for multiscale topology optimization with connectivity index constraints, *J. Mech. Des.* 140 (11) (2018).
- [20] J. Gielis, A generic geometric transformation that unifies a wide range of natural and abstract shapes, *Am. J. Bot.* 90 (3) (2003) 333–338.
- [21] Y. Xu, H. Weng, X. Ju, H. Ruan, J. Chen, C. Nan, J. Guo, L. Liang, A method for predicting mechanical properties of composite microstructure with reduced dataset based on transfer learning, *Compos. Struct.* 275 (2021) 114444.
- [22] D. Chen, M. Skouras, B. Zhu, W. Matusik, Computational discovery of extremal microstructure families, *Sci. Adv.* 4 (1) (2018) eaao7005.
- [23] L. Wang, S. Tao, P. Zhu, W. Chen, Data-driven topology optimization with multiclass microstructures using latent variable gaussian process, *J. Mech. Des.* 143 (3) (2021) 031708.
- [24] H. Peng, A. Liu, J. Huang, L. Cao, J. Liu, L. Lu, Ph-net: Parallelepiped microstructure homogenization via 3d convolutional neural networks, *Addit. Manuf.* 60 (2022) 103237.
- [25] P. Steinmetz, Y.C. Yabansu, J. Hötzter, M. Jainta, B. Nestler, S.R. Kalidindi, Analytics for microstructure datasets produced by phase-field simulations, *Acta Mater.* 103 (2016) 192–203.
- [26] A.L. Pilchak, J. Shank, J.C. Tucker, S. Srivatsa, P.N. Fagin, S.L. Semiatin, A dataset for the development, verification, and validation of microstructure-sensitive process models for near-alpha titanium alloys, *Integrating Mater. Innov.* 5 (2016) 259–276.
- [27] 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, *Comput. Mater. Sci.* 151 (2018) 278–287.
- [28] L. Wang, J. Boddapati, K. Liu, P. Zhu, C. Daraio, W. Chen, Mechanical cloak via data-driven aperiodic metamaterial design, *Proc. Natl. Acad. Sci.* 119 (13) (2022) e2122185119.
- [29] L. Wang, Y.-C. Chan, F. Ahmed, Z. Liu, P. Zhu, W. Chen, Deep generative modeling for mechanistic-based learning and design of metamaterial systems, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113377.
- [30] J. Rade, A. Balu, E. Herron, J. Pathak, R. Ranade, S. Sarkar, A. Krishnamurthy, Algorithmically-consistent deep learning frameworks for structural topology optimization, *Eng. Appl. Artif. Intell.* 106 (2021) 104483.
- [31] W. Ahmad, G. Wang, Y. Yan, Ann-based inverse goal-oriented design method for targeted final properties of materials, *Appl. Sci.* 12 (7) (2022) 3420.

- [32] D. Patel, D. Bielecki, R. Rai, G. Dargush, Improving connectivity and accelerating multiscale topology optimization using deep neural network techniques, *Struct. Multidiscip. Optim.* 65 (4) (2022) 126.
- [33] J. Seo, R.K. Kapania, Development of deep convolutional neural network for structural topology optimization, *AIAA J.* 61 (3) (2023) 1366–1379.
- [34] L. Wang, Y.-C. Chan, Z. Liu, P. Zhu, W. Chen, Data-driven metamaterial design with laplace-beltrami spectrum as shape-dna, *Struct. Multidiscip. Optim.* 61 (2020) 2613–2628.
- [35] Y.-C. Chan, F. Ahmed, L. Wang, W. Chen, Metaset: Exploring shape and property spaces for data-driven metamaterials design, *J. Mech. Des.* 143 (3) (2021) 031707.
- [36] D. Lee, Y.-C. Chan, W.W. Chen, L. Wang, A. van Beek, W. Chen, T-metaset: Task-aware acquisition of metamaterial datasets through diversity-based active learning, *J. Mech. Des.* 145 (3) (2022) 031704.
- [37] N. Korshunova, I. Papaioannou, S. Kollmannsberger, D. Straub, E. Rank, Uncertainty quantification of microstructure variability and mechanical behavior of additively manufactured lattice structures, *Comput. Methods Appl. Mech. Engrg.* 385 (2021) 114049.
- [38] J.-H. Bastek, S. Kumar, B. Telgen, R.N. Glaesener, D.M. Kochmann, Inverting the structure–property map of truss metamaterials by deep learning, *Proc. Natl. Acad. Sci.* 119 (1) (2022) e2111505119.
- [39] L. Zheng, K. Karapiperis, S. Kumar, D.M. Kochmann, Unifying the design space and optimizing linear and nonlinear truss metamaterials by generative modeling, *Nature Commun.* 14 (1) (2023) 7563.
- [40] Y.-C. Chan, F. Ahmed, L. Wang, W. Chen, Metaset: An automated data selection method for scalable data-driven design of metamaterials, in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 84003, American Society of Mechanical Engineers, 2020, V11AT11A021.
- [41] J. Wang, S. Zhu, L. Chen, T. Liu, H. Liu, Z. Lv, B. Wang, X. Tan, Data mining from a hierarchical dataset for mechanical metamaterials composed of curved-sides triangles, *Compos. Struct.* 319 (2023) 117153.
- [42] J. Löffner, F. Fritzen, Data-driven microstructure property relations, *Math. Comput. Appl.* 24 (2) (2019) 57.
- [43] A.E. Robertson, C. Kelly, M. Buzzy, S.R. Kalidindi, Local–global decompositions for conditional microstructure generation, *Acta Mater.* 253 (2023) 118966.
- [44] G. Lambard, K. Yamazaki, M. Demura, Generation of highly realistic microstructural images of alloys from limited data with a style-based generative adversarial network, *Sci. Rep.* 13 (1) (2023) 566.
- [45] Y. Li, B. Holmedal, B. Liu, H. Li, L. Zhuang, J. Zhang, Q. Du, J. Xie, Towards high-throughput microstructure simulation in compositionally complex alloys via machine learning, *CALPHAD* 72 (2021) 102231.
- [46] J. Wang, W.W. Chen, D. Da, M. Fuge, R. Rai, Ih-gan: A conditional generative model for implicit surface-based inverse design of cellular structures, *Comput. Methods Appl. Mech. Engrg.* 396 (2022) 115060.
- [47] D. Lee, W. Chen, L. Wang, Y.-C. Chan, W. Chen, Data-driven design for metamaterials and multiscale systems: A review, *Adv. Mater.* (2023) 2305254.
- [48] B. Le, J. Yvonnet, Q.-C. He, Computational homogenization of nonlinear elastic materials using neural networks, *Internat. J. Numer. Methods Engrg.* 104 (12) (2015) 1061–1084.
- [49] X. Liu, S. Tian, F. Tao, W. Yu, A review of artificial neural networks in the constitutive modeling of composite materials, *Composites B* 224 (2021) 109152.
- [50] M. Yuan, S. Paradiso, B. Meredig, S.R. Niezgoda, Machine learning–based reduce order crystal plasticity modeling for icme applications, *Integrating Mater. Manuf. Innov.* 7 (2018) 214–230.
- [51] X. Long, M. Mao, C. Lu, R. Li, F. Jia, Modeling of heterogeneous materials at high strain rates with machine learning algorithms trained by finite element simulations, *J. Micromechanics Mol. Phys.* 6 (01) (2021) 2150001.
- [52] Y. Wu, W. Yu, S. Shen, Developing a variable charge potential for hf/nb/ta/ti/zr/o system via machine learning global optimization, *Mater. Des.* 230 (2023) 111999.
- [53] L. Wu, L. Liu, Y. Wang, Z. Zhai, H. Zhuang, D. Krishnaraju, Q. Wang, H. Jiang, A machine learning-based method to design modular metamaterials, *Extreme Mech. Lett.* 36 (2020) 100657.
- [54] A. Challapalli, J. Konlan, G. Li, Inverse machine learning discovered metamaterials with record high recovery stress, *Int. J. Mech. Sci.* 244 (2023) 108029.
- [55] E.J. Kautz, Predicting material microstructure evolution via data-driven machine learning, *Patterns* 2 (7) (2021).
- [56] B. Deng, A. Zareei, X. Ding, J.C. Weaver, C.H. Rycroft, K. Bertoldi, Inverse design of mechanical metamaterials with target nonlinear response via a neural accelerated evolution strategy, *Adv. Mater.* 34 (41) (2022) 2206238.
- [57] C. Ma, Y. Chang, S. Wu, R.R. Zhao, Deep learning-accelerated designs of tunable magneto-mechanical metamaterials, *ACS Appl. Mater. Interfaces* 14 (29) (2022) 33892–33902.
- [58] X. Zheng, T.-T. Chen, X. Guo, S. Samitsu, I. Watanabe, Controllable inverse design of auxetic metamaterials using deep learning, *Mater. Des.* 211 (2021) 110178.
- [59] P. Dhariwal, A. Nichol, Diffusion models beat gans on image synthesis, *Adv. Neural Inf. Process. Syst.* 34 (2021) 8780–8794.
- [60] X.-Y. Zheng, H. Pan, P.-S. Wang, X. Tong, Y. Liu, H.-Y. Shum, Locally attentional sdf diffusion for controllable 3d shape generation, *ACM Trans. Graph. (SIGGRAPH)* 42 (4) (2023).
- [61] L. Xia, P. Breitkopf, Design of materials using topology optimization and energy-based homogenization approach in matlab, *Struct. Multidiscip. Optim.* 52 (6) (2015) 1229–1241.
- [62] D.-J. Kroon, Pinch and spherize filter, 2009, <https://www.mathworks.com/matlabcentral/fileexchange/22573-pincho-and-spherize-filter>.
- [63] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, *Adv. Neural Inf. Process. Syst.* 33 (2020) 6840–6851.
- [64] T. Chen, R. Zhang, G. Hinton, Analog bits: Generating discrete data using diffusion models with self-conditioning, 2022, arXiv preprint [arXiv:2208.04202](https://arxiv.org/abs/2208.04202).
- [65] J. Ho, T. Salimans, Classifier-free diffusion guidance, in: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications, 2021.
- [66] A. Vaswani, Attention is all you need, *Adv. Neural Inf. Process. Syst.* (2017).
- [67] J. Xu, X. Sun, Z. Zhang, G. Zhao, J. Lin, Understanding and improving layer normalization, *Adv. Neural Inf. Process. Syst.* 32 (2019).
- [68] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [69] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2017, arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101).
- [70] C. Green, Improved alpha-tested magnification for vector textures and special effects, in: ACM SIGGRAPH 2007 Courses, 2007, pp. 9–18.
- [71] Y. Mishchenko, A fast algorithm for computation of discrete euclidean distance transform in three or more dimensions on vector processing architectures, *Signal Image Video Process.* 9 (1) (2015) 19–27.
- [72] Z. Hashin, S. Shtrikman, A variational approach to the theory of the elastic behaviour of multiphase materials, *J. Mech. Phys. Solids* 11 (2) (1963) 127–140.