Computer Methods in Applied Mechanics and Engineering 425 (2024) 116864

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Topology optimization of irregular multiscale structures with tunable responses using a virtual growth rule

Yingqi Jia [a], Ke Liu [b,*], Xiaojia Shelly Zhang [a,c,d,**]

[a] Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA
[b] Department of Advanced Manufacturing and Robotics, Peking University, Beijing 100871, China
[c] Department of Mechanical Science and Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA
[d] National Center for Supercomputing Applications, USA

ARTICLE INFO

Keywords:
Multiscale structures
Tunable responses
Topology optimization
Virtual growth
Inverse design
Irregular architecture

ABSTRACT

Many applications demand tunable structural responses through tailored organic microstructural distributions and spatially varied material properties. Notable progress has been made in discovering optimized designs using periodic material patterns and fixed material phases to achieve unusual structural responses. To enable the capability of exploring non-periodic material architectures with continuous material phase design space, we propose a topology optimization methodology that leverages a virtual growth rule for designing unique multiscale structures with tunable responses and irregular architectures, while naturally ensuring manufacturability. Our approach exploits the virtual growth algorithm to create a material database, delineating constitutive relations between the homogeneous frequency hints of building blocks responsible for generating microstructures and the resultant homogenized microstructural elasticity tensors. We then employ a neural network to yield a continuous and differentiable constitutive relation. Subsequently, a topology optimization framework is introduced to optimize both the macroscale material layout and the local frequency hints for building block distribution. Finally, we generalize the virtual growth algorithm to account for optimized heterogeneous frequency hints and grow irregular yet optimized structures at the microscale. We present four examples to showcase our proposed approach in programming several types of responses, including displacement cloaking, tunable strain energy density, and global structural stiffness, in both two and three dimensions. The optimized multiscale structures, characterized by their stochastic and irregular architectures, demonstrate programmed responses that closely match the desired targets. These structures also ensure microstructural connectivity and offer the flexibility to select building blocks with guaranteed minimal features. Consequently, we leverage such connectivity and minimal features to manifest the manufacturability of the optimized structures by 3D printing. Our proposed computational strategy, which precisely realizes programmed structural responses in multiscale structures with irregular architectures and facilitates manufacturing feasibility, can be beneficial for applications that prioritize structures exemplifying disorderedness, non-uniformity, and heterogeneity.

* Corresponding author.
** Corresponding author at: Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA.
E-mail addresses: liuke@pku.edu.cn (K. Liu), zhangxs@illinois.edu (X.S. Zhang).

https://doi.org/10.1016/j.cma.2024.116864
Received 20 September 2023; Received in revised form 14 January 2024; Accepted 16 February 2024
Available online 25 March 2024

1. Introduction

Tunable structural responses are useful in numerous fields, including electromagnetics [1,2], bioengineering [3,4], optics [5], and mechanical engineering [6,7]. In the field of mechanical engineering, researchers have successfully achieved a wide array of unconventional structural responses [8–12] through various approaches. To achieve certain intricate yet desired structural responses, it calls for complex structural geometries and/or the tailored distribution of material properties [13–17].

To discover the structural geometries and material property distributions for achieving desired structural responses, topology optimization [18,19] is a robust computational approach employed among various design methods. Classical topology optimization [20,21] focuses on maximizing a structural performance of interest while adhering to specific constraints to determine the optimal material layout. Over time, this computational approach has been implemented for programming complex structural behaviors within the multi-physics contexts. These intricate behaviors often encompass nonlinearities, anisotropies, and multistabilities, which demand synergistic utilization of multiple materials. In order to optimize the distribution of these multiple materials, the field of multi-material topology optimization [22–27] has seen significant advancements in recent years. Such an approach concurrently optimizes both the topological layout and material phases, and the latter encompasses various aspects, including material types [16,28], fiber directions in composites [29], magnetic orientations in soft materials [30], and more. The optimized structures resulting from this approach possess a finite number of discrete material phases with well-defined boundaries [14,30].

Featuring a continuous and enlarged design space of material phases, topology optimization of multiscale structures (see [31] for a comprehensive review) has also garnered increasing attention in recent years. In this approach, the microscale typically comprises a single isotropic material phase (constituent or base material), while diverse macroscale effective material properties are achieved through an array of unit cells with varying topology and/or geometries. The topology optimization of multiscale structures can be broadly categorized into full-scale and multiscale methods, and the latter explicitly assumes the separation of length scales [31]. The full-scale method, exemplified by single-scale topology optimization for stiffness maximization [20], is conducted on a fine mesh, naturally resulting in multiscale structures due to the necessity for optimal structures to span multiple scales [31]. Alternatively, full-scale methods can be implemented through local control strategies such as pattern repetition [32–34] and local volume constraints [24,35]. The former enforces subdomain repetition, while the latter imposes an upper bound on volume fraction in nearby regions for each material point in the design domain. These full-scale methods generate multiscale structures with unit cells naturally connected to their neighbors, but the computational cost can be high due to direct structural analysis at a fine scale.

To mitigate the computational cost in designing multiscale structures, multiscale methods have gained popularity. This approach decouples the topology optimization problem into two subproblems: one macroscale subproblem for determining the distribution of effective material properties to achieve desired structural responses, and one microscale subproblem for designing unit cells to attain the desired effective material properties. This decoupling allows the macroscale subproblem to be solved on a coarse mesh, reducing computational costs compared to the full-scale method. As for the microscale subproblem, it is bridged to the macroscale using homogenization theory [36,37], linking microscale topology and/or geometries of unit cells to macroscale effective material properties. Implementing the multiscale method can be achieved through hierarchical formulations [38–45] that solve the two subproblems either concurrently or sequentially, allowing for the design of free-form multiscale structures.

For enhanced manufacturability, recent advancements in multi-scale methods have focused on leveraging predefined primitive unit cells such as trusses, spinodals, gyroids, and ranked laminates. These unit cells can either remain unchanged [46] or be parameterized with geometric parameters (e.g., thickness of members) [47–55] and/or transformation parameters (e.g., rotation and elongation) [56–63]. Following the specification of primitive unit cells, the next step involves relating unit cell parameters to their effective properties through analytical or numerical homogenization. The latter is often accompanied by fitting functions or machine learning techniques to create a continuous and differentiable surrogate model [15,64–67]. The macroscale subproblem is then solved based on the derived relationships to find the desired unit cell parameters. Subsequently, these optimized unit cell parameters are used to construct the multiscale structures, with de-homogenization techniques [58,61,68] potentially involved for certain primitive unit cells such as ranked laminates. Ensuring good connectivity between neighboring unit cells poses a challenge during construction. One approach is to restrict the topology of primitive unit cells, such as adding a bounding box, to ensure they can always be connected to their neighbors. Alternatively, connectivity issues can be successfully addressed through additional constraints or considerations, including pseudo-load [47], unified formulations with non-linear diffusion [47], geometric interpolation between optimized unit cells [69], connectivity index [70], optimization of extended domain covering adjacent cells [51], etc.

Despite notable advances in existing studies, our work introduces a novel topology optimization framework (Fig. 1) aimed at engineering irregular (non-periodic) multiscale structures with controllable responses. In comparison to existing studies, our proposed framework seeks to achieve several objectives simultaneously. First, leveraging the versatile virtual growth algorithm [71] ensures local connections between neighboring unit cells without relying on additional constraints, thus avoiding limitations on achievable effective material properties. Second, our framework yields stochastic and highly aperiodic multiscale structures, which may exhibit enhanced failure resistance [55,72] and may be preferable in certain scenarios due to aesthetic considerations compared to deterministic structures. Third, while most of the existing work in mechanical engineering focuses on global compliance minimization, we aim to program three representative types of structural responses — node-wise mechanical displacement cloaking, element-wise strain energy density distribution, and global maximized structural stiffness (Fig. 1(b)). Finally, our proposed framework generates structures with a uniform and explicitly controllable microscale feature size that promotes manufacturability.

To achieve these objectives, we employ our proposed methodology to design the material layout and property distribution as shown in Fig. 1(c)–(d). During the optimization of the property distribution, rather than directly specifying microstructures (as in classical multiscale topology optimization), our approach starts with a finite set of building blocks capable of forming seamless

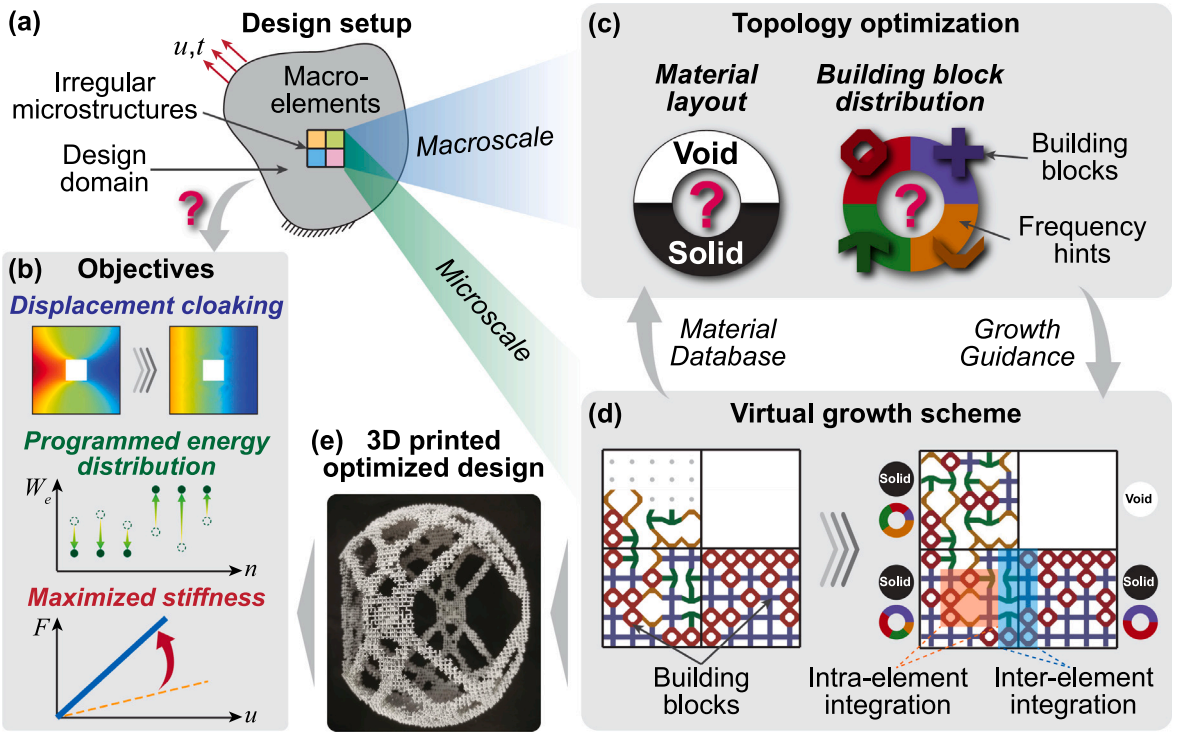

Fig. 1. Illustration of the proposed topology optimization methodology using a virtual growth rule. (a) The design domain, comprising macro-elements and subjected to applied displacement loading (u) and traction loading (t). (b) Objectives in this study: achieving mechanical displacement cloaking, programming strain energy density distribution, and maximizing overall structural stiffness. Here, W_e represents the strain energy density of macro-element e, n denotes control macro-elements, and F is the resultant reaction force. (c) Topology optimization for macroscale material layout and building block distribution, determined by building block types and their frequency hints. (d) Virtual growth algorithm used to generate irregular multiscale structures following optimized material layout and building block distribution. (e) A 3D printed optimized design (see details in Section 4.4) generated by the proposed methodology.

microstructures based on prescribed frequency hints. Here, the frequency hint of a building block is a scalar field variable, defined as the desired probability of the building block appearing at a material point. We then sample varying frequency hints and utilize the virtual growth scheme to create the corresponding microstructures. Consequently, we construct a material database specifying the constitutive relationship between the microstructural property and the frequency hints of building blocks. We optimize the property configuration by distributing these building blocks through controlling their frequency hints at different locations. Based on the optimized material layout and building block distribution, we employ the extended virtual growth algorithm to generate optimized irregular multiscale structures at the microscale (Fig. 1(d)). Owing to the extended virtual growth algorithm, the generated multiscale structures exhibit stochastic, self-sustaining ensembles, seamlessly integrating both within and between macro-elements. Furthermore, these self-sustaining ensembles are constructed from given building blocks, carefully selected to meet the minimum feature size required for fabrication. This inherent self-supporting characteristic, combined with the established minimum feature size, significantly enhances the feasibility of manufacturing these optimized structures using 3D printing, as shown in Fig. 1(e).

The remainder of this paper is organized as follows. Section 2 briefly reviews the classical virtual growth algorithm presented in [71], providing readers with a convenient overview. We then introduce its extended version, one of the four interconnected components of our proposed methodology. Section 3 presents the other three components: the creation of a discrete material database, training of a neural network, and topology optimization. In Section 4, we employ four examples to demonstrate the tunable responses facilitated by our methodology and extend the discussion to the three-dimensional (3D) case. Finally, in Section 5, we offer several concluding remarks. The paper also includes four appendices. Appendix A introduces the normalization projection scheme used in topology optimization to enforce the equality constraint for the frequency hints with non-trivial bounds. Appendix B details the sensitivity analysis and its verification as part of topology optimization in our proposed methodology. Appendix C complements the first example by considering rounded void and designable regions. Lastly, Appendix D presents the proposed two-grid projection scheme, which enables topology optimization on an unstructured grid while implementing the virtual growth algorithm on a structured grid.

2. Classical and extended virtual growth algorithms

The virtual growth algorithm constitutes one of the pivotal components of our proposed topology optimization methodology. This section provides a concise overview of the classical virtual growth algorithm as presented in [71], ensuring the reader's familiarity

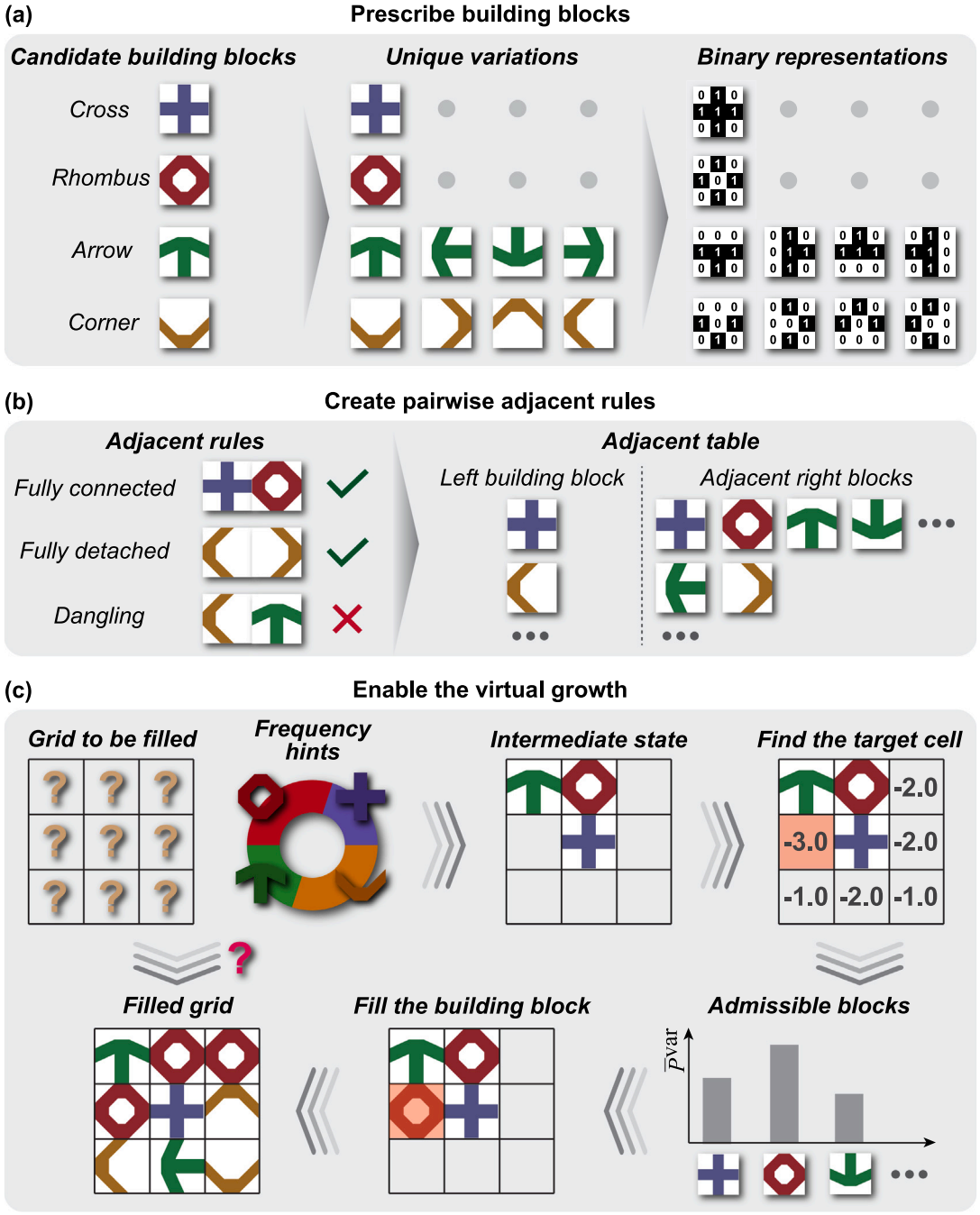

Fig. 2. Illustration of the classical virtual growth algorithm. (a) Prescribing building blocks. (b) Creating adjacent rules for each pair of building block variations. (c) The virtual growth process. The numbers shown in the figure represent cell entropies calculated using Eq. (3). The variable $\overline{P}^{var}$ denotes the normalized probability of selecting a specific building block variation.

with its foundations. Subsequently, we introduce the extended version of this algorithm, which serves as the primary method employed in this work.

2.1. Classical virtual growth algorithm

The virtual growth algorithm, as introduced in [71], is a pioneering graphical theory for generating irregular and stochastic architected patterns. This process relies on a set of adjacent rules applied to a limited number of building blocks. In Fig. 2(a), we

initiate the process by specifying M^{block} ($M^{\text{block}} \geq 1$) simple building blocks. These building blocks are only required to be capable of connecting through specific rotational configurations. For our illustration, we consider $M^{\text{block}} = 4$ types of two-dimensional (2D) building blocks: `cross`, `rhombus`, `arrow`, and `corner`, with each block rotated at 90, 180, and 270 degrees to yield unique variations. Subsequently, we derive binary representations for these variations, each represented by a 3×3 matrix where each element is either 0 (void) or 1 (solid).

To determine whether two building block variations can be adjacent (Fig. 2(b)), we establish a condition: two variations can be adjacent if and only if they are either fully connected or fully detached, with no dangling parts. This condition can be automatically checked using their binary representations. Let $\mathbf{L}_{3\times3}$ and $\mathbf{R}_{3\times3}$ denote the binary representations of a given left and right building block variation, respectively. The two variations can be adjacent if and only if $\|L_{*3} - R_{*1}\|_2 = 0$, where L_{*3} represents the third column of $\mathbf{L}$, and R_{*1} represents the first column of $\mathbf{R}$. By examining all pairs of building block variations, we construct an adjacency table that specifies all right building block variations that can be adjacent to a given left building block variation.

With the adjacent table in hand, we can now initiate the virtual growth process (Fig. 2(c)). Our objective is to populate a given empty grid with prescribed building blocks, adhering to both the adjacent rules and the desired frequency hints. We achieve this target iteratively as follows: In each iteration, we traverse all cells of the grid, determining the building block variations present in neighboring cells. We compute the admissible building block variations for the current cell ($\mathcal{A}^{\text{var}}$) based on the adjacency table (Fig. 2(b)). Subsequently, we calculate the normalized probability ($\overline{P}_i^{\text{var}}$) of selecting the ith type of admissible building block variation as

$$\overline{P}_i^{\text{var}} = \frac{P_i^{\text{var}}}{\sum_{j\in\mathcal{A}^{\text{var}}} P_j^{\text{var}}} \tag{1}$$

where P_i^{var} is the probability distribution of the ith type of building block variation in a given cell and computed as

$$\overline{\xi}_i^{\text{var}} = \frac{N_i^{\text{var}} + \left(N^{\text{cell}} - \sum_{j=1}^{M^{\text{var}}} N_j^{\text{var}}\right) P_i^{\text{var}}}{N^{\text{cell}}} \quad \Rightarrow \quad P_i^{\text{var}} = \frac{\overline{\xi}_i^{\text{var}} N^{\text{cell}} - N_i^{\text{var}}}{N^{\text{cell}} - \sum_{j=1}^{M^{\text{var}}} N_j^{\text{var}}}, \tag{2}$$

where $\overline{\xi}_i^{\text{var}} = \overline{\xi}/M^{\text{u}}$ is the desired frequency distribution of the ith type of building block variation, and $\overline{\xi}$ denotes the prescribed frequency distribution of the corresponding original building block. The symbol M^{u} stands for the number of unique variation types of the corresponding building block. The symbol M^{var} is the number of unique variation types of all building blocks. The parameter N^{cell} corresponds to the number of cells in the grid, and N_i^{var} represents the number of filled building block variations of type i.

Next, we compute the cell entropy as

$$S = \sum_{i\in\mathcal{A}^{\text{var}}} -\overline{P}_i^{\text{var}} \log \overline{P}_i^{\text{var}} \tag{3}$$

and select the target cell with the lowest entropy. After identifying the target cell, we randomly insert one admissible building block variation based on the normalized probability distribution, $\overline{P}_i^{\text{var}}$ for $i \in \mathcal{A}^{\text{var}}$. This process repeats until all cells are filled, resulting in the generation of an irregular architected pattern. Such a virtual growth process is valuable for creating irregular and stochastic architected mechanical materials, providing precise control over properties via the input frequency hints of building blocks (for additional details, see Section 3.1 and [71]).

2.2. Extented virtual growth algorithm for heterogeneous frequency hints

While the classical virtual growth algorithm described in [71] primarily focuses on a single set of intuition-based frequency hints, this section introduces an extension to accommodate multiple/heterogeneous optimization-guided frequency hints. This extended version also permits the inclusion of passive solid and void regions within the final design. For instance, as illustrated in Fig. 1(d), we present a grid featuring four macro-elements with distinct prescribed frequency hints. Notably, the top-right macro-element is designated as a passive void region, where no building blocks are allowed. The design created through the extended virtual growth algorithm effortlessly fulfills these conditions. Compared to the classical virtual growth algorithm, the extended version also brings notable improvements in enhancing computational efficiency, promoting a higher likelihood of finding a feasible solution, and eliminating global disconnection as follows.

First, note that the existence of a solution to the virtual growth problem discussed herein cannot be guaranteed. However, given the stochastic nature of this virtual growth problem, we can employ a strategy of multiple algorithm retries until a feasible solution is obtained. In the classical version, each retry begins anew, incurring a potentially high computational cost, especially when dealing with a large number of cells to fill, as is the case in the problems addressed in this study. In contrast, the extended version of the algorithm introduces an innovative approach. It retains the information regarding filled building blocks and allows for the reversion to a specific iteration in cases where no solution is found. Leveraging the inherent stochasticity of the algorithm, this approach allows us to restart the algorithm from the chosen iteration, significantly reducing computational costs.

Second, during each iteration, the classical algorithm extensively traverses all unfilled cells to compute entropies, which is a time-consuming process. To mitigate computational costs, we have optimized this step by exclusively calculating the entropies of front cells. These front cells, defined as unfilled cells with at least one filled neighboring cell, consistently offer fewer choices for admissible building blocks, resulting in lower entropies when compared to cells at a distance. See a detailed procedure in Algorithm 1.

Finally, the extended algorithm leverages the PyVista package [73] to verify global connectivity, mitigating the occasional occurrence of global disconnection. While virtual growth inherently ensures local integration between neighboring building blocks, there are instances where these blocks might inadvertently create a self-enclosed section detached from the main structure. By introducing global connectivity checks, we ensure the structural integrity of the entire design.

Algorithms 2–3 provide a comprehensive breakdown of the preparation and iteration phases of the extended virtual growth algorithm. To implement this algorithm, we have developed an in-house Python code [74]. This code is specifically designed to optimize computational efficiency by converting certain while-loops and for-loops into matrix and vector operations. Detailed computational time is reported in Section 4.

Algorithm 1: Identifying the front cells and computing their entropies

Inputs: iteration counter, $n^{\texttt{iter}}$; the number of all cells, $N^{\texttt{cell}}$; the set of designable cells, $D^{\texttt{cell}}$; probability matrix, $\mathbf{P}^{\texttt{var}}$ with $P_{ij}^{\texttt{var}}$ representing the probability of the building block variation of type j to be chosen in macro-element i;
if $n^{\texttt{iter}} = 1$ **then**
  Let $\mathcal{F}^{\texttt{cell}} \leftarrow D^{\texttt{cell}}$ be the set of front cells;
else
  Initialize the set of front cells, $\mathcal{F}^{\texttt{cell}} \leftarrow \emptyset$;
  for $\texttt{cell} \in D^{\texttt{cell}}$ **do**
    Find all neighboring cells of the current cell;
    if `cell` is unfilled and at least one neighboring cell is filled **then**
      $\mathcal{F}^{\texttt{cell}} \leftarrow \mathcal{F}^{\texttt{cell}} \cup \{n^{\texttt{cell}}\}$;
    end
  end
end
for $\texttt{cell} \in \mathcal{F}^{\texttt{cell}}$ **do**
  Find all building block variations filled in neighboring cells if any;
  According to the adjacent table in Fig. 2(b), find the adjacency building block variations of neighboring building block variations and then take their intersection as the admissible building block variations of the current cell;
  if no admissible building block variations exist **then**
    Break the algorithm;
  end
  Find the corresponding macro-element of the current cell and then the normalized probabilities of the admissible building block variations of the current cell based on $\mathbf{P}^{\texttt{var}}$ and Eq. (1);
  Use Eq. (3) to compute the current cell's entropy (S).
end

3. Proposed virtual-growth-based topology optimization

This section introduces the proposed topology optimization methodology, utilizing the extended virtual growth algorithm outlined in Algorithms 2–3. The methodology consists of four interconnected components: the creation of a discrete material database, training of a neural network model, topology optimization, and the virtual growth of structures. In the previous section, we introduced the virtual growth algorithm. The remaining three components will be discussed in the following subsections.

3.1. Creation of a discrete material database

To facilitate subsequent topology optimization, it is imperative to establish the constitutive relationship between the frequency hints of building blocks, denoted as $\{\bar{\xi}_1, \bar{\xi}_2, \bar{\xi}_3, \bar{\xi}_4\}$, and the effective (homogenized) elasticity tensor of the generated material (microstructure), represented as $\mathbb{C}$. This task is accomplished through a discrete material database, which is detailed in Fig. 3(a). Initially, we prescribe four building blocks for both 2D and 3D cases, adhering to the descriptions provided in [71]. The selected building blocks should represent different connectivities. For example, in the 2D case, the `cross` building block can be connected from all four directions while the `arrow` building block can only be connected from three directions. In addition, the selected building blocks should yield microstructures that cover a relatively large material property space as shown in Fig. 3(b) such that the desired macroscopic structural responses can be achieved.

Upon prescribing the building blocks, we then perform uniform sampling of frequency hints on the hyperplane defined by $\bar{\xi}_1 + \bar{\xi}_2 + \bar{\xi}_3 + \bar{\xi}_4 = 1$. This process yields 800 data points for the 2D case and 200 data points for the 3D case. With these sampled frequency hints, we utilize the classical virtual growth algorithm detailed in Section 2.1 to generate architected materials. Specifically, we create square materials consisting of 40×40 building blocks for the 2D case and cubic materials with $10 \times 10 \times 10$

Algorithm 2: Extended virtual growth algorithm — preparation phase

Inputs: the number of macro-elements of the input grid, $N^{\texttt{elem}}$; the number of building block types, $M^{\texttt{block}}$; the frequency hint matrix, $\mathbf{F}_{N^{\texttt{elem}}\times M^{\texttt{block}}}$ with F_{ij} representing the prescribed frequency distribution of the j-th type of building block in macro-element i; the set of all passive solid macro-elements, $\mathcal{S}^{\texttt{elem}}$; the set of all passive void macro-elements, $\mathcal{V}^{\texttt{elem}}$;
Initialize the building block type counter, $m^{\texttt{block}} \leftarrow 1$; the vector of building block variation counters, $\mathbf{m}^{\texttt{var}} \leftarrow \mathbf{0}$;
while $m^{\texttt{block}} \leq M^{\texttt{block}}$ **do**
  Find unique variations of the $m^{\texttt{block}}$-th type of building block;
  Store the number of unique variations in $m^{\texttt{var}}_{m^{\texttt{block}}}$;
  $m^{\texttt{block}} \leftarrow m^{\texttt{block}} + 1$;
end
Compute the total number of all unique building block variations, $M^{\texttt{var}} = \sum_{m^{\texttt{block}}=1}^{M^{\texttt{block}}} m^{\texttt{var}}_{m^{\texttt{block}}}$;
Initialize the augmented frequency hint matrix, $\widehat{\mathbf{F}}_{N^{\texttt{elem}}\times M^{\texttt{var}}} \leftarrow \mathbf{O}$, with $\widehat{F}_{ij}$ representing the frequency distribution of the j-th type of building block variation in macro-element i; the building block type counter, $m^{\texttt{block}} \leftarrow 1$; matrix column counter, $n^{\texttt{col}} \leftarrow 1$; end column indicator, $N^{\texttt{col}} \leftarrow 0$;
while $m^{\texttt{block}} \leq M^{\texttt{block}}$ **do**
  $N^{\texttt{col}} \leftarrow N^{\texttt{col}} + m^{\texttt{var}}_{m^{\texttt{block}}}$;
  while $n^{\texttt{col}} \leq N^{\texttt{col}}$ **do**
    $\widehat{F}_{*n^{\texttt{col}}} \leftarrow F_{*m^{\texttt{block}}}/m^{\texttt{var}}_{m^{\texttt{block}}}$;
    $n^{\texttt{col}} \leftarrow n^{\texttt{col}} + 1$;
  end
  $m^{\texttt{block}} \leftarrow m^{\texttt{block}} + 1$;
end
Set each cell with the same solid, void, and designable attribute as the macro-element that the cell resides in;
Figure out the set of all cells in the grid, $\mathcal{A}^{\texttt{cell}}$; all passive solid cells, $\mathcal{S}^{\texttt{cell}}$; all passive void cells, $\mathcal{V}^{\texttt{cell}}$;
Let $\mathcal{P}^{\texttt{cell}} = \mathcal{S}^{\texttt{cell}} \cup \mathcal{V}^{\texttt{cell}}$ denote all passive cells and $\mathcal{D}^{\texttt{cell}} = \mathcal{A}^{\texttt{cell}} \setminus \mathcal{P}^{\texttt{cell}}$ be the designable cells.
Compute the number of all cells as $N^{\texttt{cell}} = n(\mathcal{A}^{\texttt{cell}})$ and the number of cells to be filled as $N^{\texttt{fill}} = n(\mathcal{D}^{\texttt{cell}})$, where $n(\cdot)$ represents the cardinal number of a set.

building blocks for the 3D case. For each set of frequency hints, we generate 100 samples, resulting in a total of 80,000 2D materials and 20,000 3D materials.

Upon generating the architected materials, we proceed to evaluate their effective elasticity tensors, denoted as $\mathbb{C}$, utilizing homogenization techniques [36]. For the sake of clarity and consistency, we adopt the numerical homogenization approach outlined in [37]. This approach requires the global stiffness matrix of the material as input, which we compute via the finite element analysis (FEA) [75]. Specifically, for the 2D case, we employ the first-order quadrilateral Lagrange element and make use of the plane stress assumption. In the 3D case, we opt for frame elements to enhance computational efficiency. Following the numerical homogenization process, we calculate the average elasticity tensor over 100 samples for each set of frequency hints. Consequently, we compile a discrete material database that comprises 800 frequency–elasticity pairs for the 2D case and 200 pairs for the 3D case.

The compiled material database encompasses a broad spectrum of material properties by utilizing predefined building blocks and sampled frequency hints. In Fig. 3(b) and (c), we depict the 2D and 3D material properties in the $E^{\texttt{ave}}/E^{\texttt{solid}}$–$\nu^{\texttt{ave}}$ space, respectively, following the convention in [71]. Here, the variable $E^{\texttt{solid}} = 1163$ MPa represents the Young's modulus of the constituent solid material. The variables $E^{\texttt{ave}}$ and $\nu^{\texttt{ave}}$ denote the effective Young's modulus and Poisson's ratio defined as

$$E^{\texttt{ave}} = \frac{1}{2}\left(\frac{1}{\mathbb{S}_{1111}} + \frac{1}{\mathbb{S}_{2222}}\right), \quad \nu^{\texttt{ave}} = -\frac{1}{2}\left(\frac{\mathbb{S}_{2211}}{\mathbb{S}_{1111}} + \frac{\mathbb{S}_{1122}}{\mathbb{S}_{2222}}\right)$$

in 2D and

$$E^{\texttt{ave}} = \frac{1}{3}\left(\frac{1}{\mathbb{S}_{1111}} + \frac{1}{\mathbb{S}_{2222}} + \frac{1}{\mathbb{S}_{3333}}\right), \quad \nu^{\texttt{ave}} = -\frac{1}{6}\left(\frac{\mathbb{S}_{2211}}{\mathbb{S}_{1111}} + \frac{\mathbb{S}_{1122}}{\mathbb{S}_{2222}} + \frac{\mathbb{S}_{3322}}{\mathbb{S}_{2222}} + \frac{\mathbb{S}_{2233}}{\mathbb{S}_{3333}} + \frac{\mathbb{S}_{3311}}{\mathbb{S}_{1111}} + \frac{\mathbb{S}_{1133}}{\mathbb{S}_{3333}}\right)$$

in 3D. Here, $\mathbb{S} = \mathbb{C}^{-1}$ represents the fourth-order homogenized compliance tensor.

In Fig. 3(b) and (c), the dots denote data points averaged over 100 samples corresponding to the same set of frequency hints, and the insets display the frequency hints of several representative data points. The boundary of the shaded area indicates the extreme values achieved by the generated samples. Based on Fig. 3(b) and (c), we observe that the selected building blocks yield a broad material property space spanning $E^{\texttt{ave}}/E^{\texttt{solid}} \in [0.012, 0.272]$ and $\nu^{\texttt{ave}} \in [-0.541, 0.628]$ in 2D, as well as $E^{\texttt{ave}}/E^{\texttt{solid}} \in [0.003, 0.036]$ and $\nu^{\texttt{ave}} \in [0.008, 0.144]$ in 3D. Moreover, the frequency hints significantly influence the achieved material properties. For instance, in the 2D case, a higher frequency of the `cross` building block results in larger $E^{\texttt{ave}}/E^{\texttt{solid}}$ values, while a greater frequency of the `rhombus` building blocks enables larger $\nu^{\texttt{ave}}$ values. These observations stem from the higher tensile/compressive resistance of the `cross` building blocks in both horizontal and vertical directions, contrasting with the `rhombus` building block's tendency to exhibit lateral contraction when expanded in the orthogonal direction. Harnessing the extensive range of material properties provided by the compiled database enables precise control over diverse structural responses, as demonstrated in Section 4.

Algorithm 3: Extended virtual growth algorithm — iteration phase

Inputs: the maximum number of trials, $N^{\texttt{try}}$; the number of macro-elements of the input grid, $N^{\texttt{elem}}$; the total number of all unique building block variations, $M^{\texttt{var}}$; the number of cells to be filled, $N^{\texttt{fill}}$; rolling back rate, $r \in (0, 1)$;
Initialize the trial counter, $n^{\texttt{try}} \leftarrow 1$; the iteration for keeping the histories, $N^{\texttt{keep}} \leftarrow 1$;
while $n^{\texttt{try}} \leq N^{\texttt{try}}$ **do**
  Clear the filled building block variations since iteration $N^{\texttt{keep}}$;
  Calculate the matrix for storing the number of filled building block variations, $\mathbf{C}^{\texttt{var}}_{N^{\texttt{elem}} \times M^{\texttt{var}}}$ with $C^{\texttt{var}}_{ij}$ representing the number of building block variations of type j filled in macro-element i;
  Initialize the iteration counter, $n^{\texttt{iter}} \leftarrow N^{\texttt{keep}}$;
  while $n^{\texttt{iter}} \leq N^{\texttt{fill}}$ **do**
    Based on $\mathbf{C}^{\texttt{var}}$ and Eq. (2), compute the probability matrix, $\mathbf{P}^{\texttt{var}}$;
    Deal with non-positive probabilities by setting $P^{\texttt{var}}_{ij} = 10^{-6}$ if $P^{\texttt{var}}_{ij} \leq 0$ for $i = 1, 2, ..., N^{\texttt{elem}}$ and $j = 1, 2, ..., M^{\texttt{var}}$;
    Normalize the probability matrix by setting $P^{\texttt{var}}_{ij} = P^{\texttt{var}}_{ij} / \sum_{k=1}^{M^{\texttt{var}}} P^{\texttt{var}}_{ik}$ for $i = 1, 2, ..., N^{\texttt{elem}}$ and $j = 1, 2, ..., M^{\texttt{var}}$;
    Identify the set of front cells, $\mathcal{F}^{\texttt{cell}}$, and compute their entropies using Algorithm 1;
    if Algorithm 1 is successful **then**
      Let $\mathcal{I} \leftarrow \texttt{successful}$ be the algorithm indicator;
      Find the target cell with the lowest entropy;
      Find the admissible building block variations of the target cell;
      Compute the normalized probabilities of the admissible building block variations based on $\mathbf{P}^{\texttt{var}}$ and Eq. (1);
      Randomly select an admissible building block variation based on their normalized probabilities;
      Fill the selected building block variation into the target cell and update $\mathbf{C}^{\texttt{var}}$;
    else
      Let $\mathcal{I} \leftarrow \texttt{failed}$ be the algorithm indicator;
      break;
    end
    $n^{\texttt{iter}} \leftarrow n^{\texttt{iter}} + 1$;
  end
  if the algorithm indicator, $\mathcal{I}$, is $\texttt{successful}$ **then**
    Check the global connectivity of the generated structure with the PyVista package [73];
    if disconnection exists in the generated structure **then**
      $N^{\texttt{keep}} \leftarrow 1$;
      $n^{\texttt{try}} \leftarrow n^{\texttt{try}} + 1$;
    else
      Break the algorithm because it is successfully finished;
    end
  else
    $N^{\texttt{keep}} \leftarrow \max\{1, \lfloor n^{\texttt{iter}} \times (1 - r) \rfloor\}$ where $\lfloor \cdot \rfloor$ represents the greatest integer that is smaller than $(\cdot)$;
    $n^{\texttt{try}} \leftarrow n^{\texttt{try}} + 1$;
  end
end

3.2. Training of a neural network model to establish the constitutive relationship

The previous material database records discrete relationships between the frequency hints of building blocks and the elasticity tensors of generated architected materials. However, for the sensitivity analysis in subsequent topology optimization, a continuous and differentiable relationship is essential. To capture this intricate relationship, similar to many other data-driven-based methods such as [15,64–67], we employ a fully connected neural network, as illustrated in Fig. 4(a). This neural network comprises one input layer ($\mathbb{I}$) for the frequency hints (comprising four components, $\bar{\xi}_1$–$\bar{\xi}_4$), two hidden layers ($\mathbb{H}_1$ and $\mathbb{H}_2$) with m and p nodes, respectively, and one output layer ($\mathbb{O}$) for N independent components of the material's elasticity tensor (C_1–C_N, where $N = 6$ for the 2D case and $N = 21$ for the 3D case). Between the $\mathbb{I}$ and $\mathbb{H}_1$ layers, as well as the $\mathbb{H}_1$ and $\mathbb{H}_2$ layers, we employ the rectified linear activation unit (ReLU), denoted as $\text{ReLU}(\cdot) = \max\{\cdot, 0\}$, as a nonlinear activation function. This configuration defines the mathematical formulation of the neural network as

$$\begin{cases} \mathbb{I} \to \mathbb{H}_1: & \boldsymbol{\zeta}_1 = \mathbf{W}_1 \boldsymbol{\zeta} + \mathbf{b}_1, \quad \mathbf{h}_1 = \text{ReLU}(\boldsymbol{\zeta}_1) \\ \mathbb{H}_1 \to \mathbb{H}_2: & \boldsymbol{\zeta}_2 = \mathbf{W}_2 \mathbf{h}_1 + \mathbf{b}_2, \quad \mathbf{h}_2 = \text{ReLU}(\boldsymbol{\zeta}_2) \\ \mathbb{H}_2 \to \mathbb{O}: & \mathbf{C} = \mathbf{W}_3 \mathbf{h}_2 + \mathbf{b}_3 \end{cases} \tag{4}$$

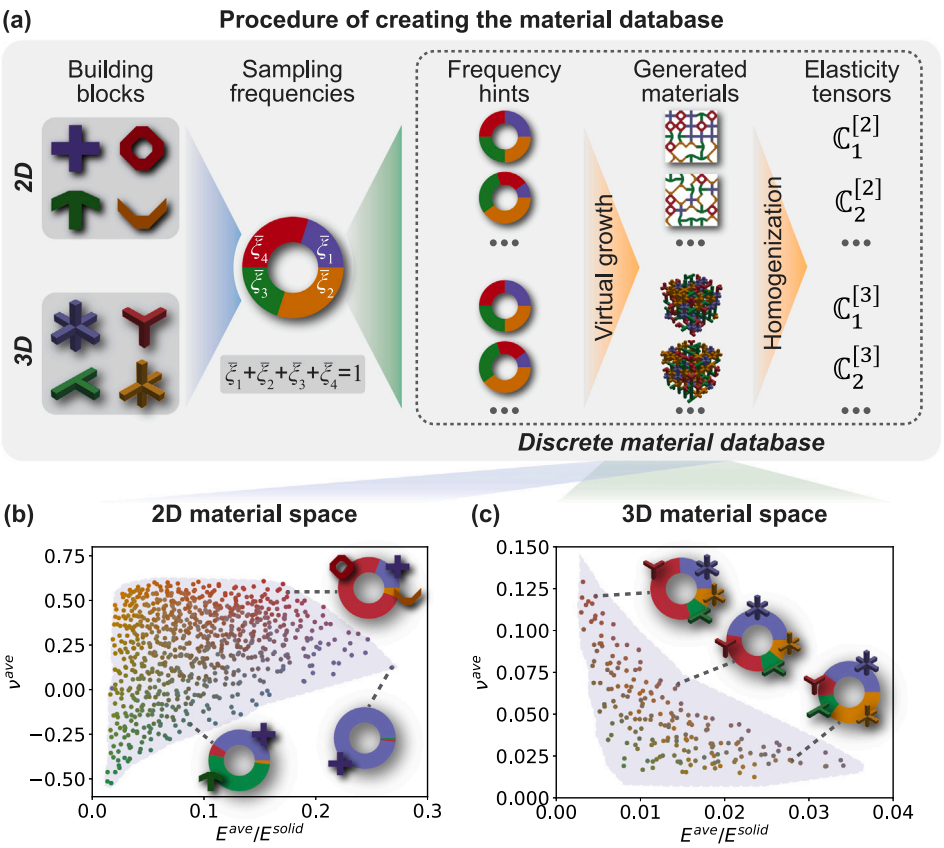

Fig. 3. Development of a discrete material database. (a) The creation process involves prescribing building blocks, sampling frequency hints, $\{\bar{\xi}_1, \bar{\xi}_2, \bar{\xi}_3, \bar{\xi}_4\}$, on the hyperplane $\bar{\xi}_1 + \bar{\xi}_2 + \bar{\xi}_3 + \bar{\xi}_4 = 1$, using the classical virtual growth algorithm [71] to generate architected materials, and finally employing numerical homogenization to evaluate the elasticity tensors of these materials. The variable $\mathbb{C}_i^{[d]}$ represents the ith elasticity tensor in a d-dimensional space. (b) and (c) show the 2D and 3D material property space, respectively, which are achieved by the materials generated from the prescribed building blocks. The dots on the plots symbolize data points averaged over 100 samples associated with the same set of frequency hints, with the color of the dots indicating the respective frequency hints. Insets provide a visual representation of the frequency combination for selected data points, and the boundary of the shaded area delineates the extreme values attained by the generated samples. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\boldsymbol{\zeta} = [\bar{\xi}_1, \bar{\xi}_2, \bar{\xi}_3, \bar{\xi}_4]^\top$ represents the input frequency hints. Intermediate variables during the forward pass of the neural network are stored in vectors, $\boldsymbol{\zeta}_1$, $\boldsymbol{\zeta}_2$, $\mathbf{h}_1 = [h_{1,1}, h_{1,2}, \ldots, h_{1,m}]^\top$, and $\mathbf{h}_2 = [h_{2,1}, h_{2,2}, \ldots, h_{2,p}]^\top$. The weighting matrices, $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$, and bias vectors, $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$, represent the model parameters to be optimized. The vector $\mathbf{C}$ stores the independent components of the material's elasticity tensor, C_1–C_N.

To optimize the model parameters ($\mathbf{W}_1$–$\mathbf{W}_3$ and $\mathbf{b}_1$–$\mathbf{b}_3$), we employ the mean squared error (MSE) of the independent components of the elasticity tensor as the loss function, which is denoted as $\mathcal{L}$. The goal is to minimize $\mathcal{L}$ using the Adam optimizer [76]. Before embarking on the optimization of model parameters, it is essential to establish reasonable model hyperparameters. These hyperparameters include the number of nodes in the two hidden layers (m and p) and the learning rate (α). Setting appropriate hyperparameters promotes faster convergence and improved training performance. To determine these hyperparameters, we randomly test 100 combinations of hyperparameters, with m and $p \in \{4, 8, 16, 32, 64, 128, 256, 512\}$ and $\alpha \in \{10^{-5.0}, 10^{-4.5}, 10^{-4.0}, 10^{-3.5}, 10^{-3.0}, 10^{-2.5}, 10^{-2.0}\}$. Each neural network training run encompasses 200 epochs, and we record the final loss ($\mathcal{L}$) for each combination. Based on the test results, we determine that the optimal hyperparameters for training the 2D database are $m = p = 256$ and $\alpha = 10^{-2}$. For training the 3D database, we find that $m = 512$, $p = 256$, and $\alpha = 10^{-2}$ yield the best performance.

During formal training, we employ a holdout validation strategy to ensure the neural network's accuracy and generalizability (Fig. 4(a)). Specifically, we randomly allocate 800 2D data into two sets: 720 for training and 80 for testing. For the 3D data, we split the 200 samples into 160 for training and 40 for testing. The neural network undergoes 1000 epochs of training, with a learning rate schedule: 10^{-2} for epochs 1–200, 10^{-3} for epochs 201–500, and finally, 10^{-4} for epochs 501–1000. The neural network is implemented and trained using an in-house Python code [74] and the PyTorch package [77]. Fig. 4(b) highlights the good performance of the neural network through a comparison between the true and predicted values across both the training and

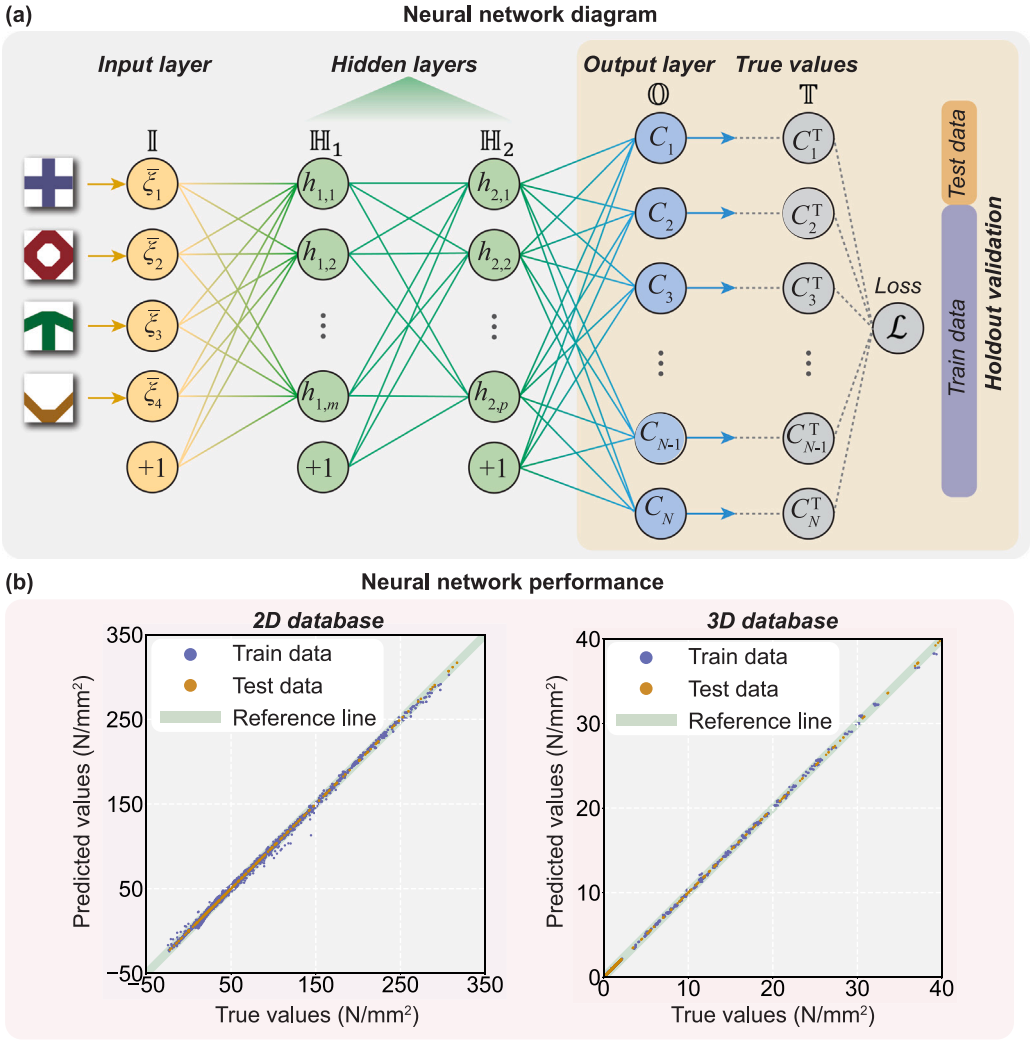

Fig. 4. Neural network diagram for predicting the elasticity tensor based on the input frequency hints. Panel (a) shows a fully connected neural network containing one input layer ($\mathbb{I}$), two hidden layers ($\mathbb{H}_1$ and $\mathbb{H}_2$), and one output layer ($\mathbb{O}$). The predicted independent components of the elasticity tensor (C_1 to C_N) from the output layer are compared with their corresponding ground truth values (C_1^{T} to C_N^{T}), and the disparity is quantified using the mean squared error (MSE), denoted as $\mathcal{L}$. Additionally, a holdout validation strategy is employed to ensure the neural network's accuracy and generalizability by partitioning the data into training and testing sets. Panel (b) highlights the outstanding performance of the neural network by presenting a comparison between the true values and the predicted values across both the training and testing datasets within the framework of 2D and 3D databases.

testing datasets in the context of 2D and 3D databases. Notably, most data points cluster closely around the reference line, signifying the neural network's strong predictive capabilities for the elasticity tensor. Furthermore, based on the test data, the neural network achieves MSEs of 2.538 MPa in 2D and 0.134 MPa in 3D, significantly smaller than their respective mean values of 62.542 MPa and 2.687 MPa.

3.3. Topology optimization framework

Based on the trained neural network model, we now introduce the proposed topology optimization framework for designing irregular multiscale structures with tunable responses. This subsection outlines the proposed topology optimization framework across three key aspects: design space parameterization, elasticity interpolation, and the mathematical formulation for topology optimization.

3.3.1. Design space parameterization

Classical topology optimization [20,21] is concerned with determining the optimal material layout (i.e., the presence or absence of material) to achieve desired performance while adhering to certain constraints. In our work, we extend this concept by optimizing

both the material layout and the distribution of building blocks, as depicted in Fig. 1(c). This extension enables us to create optimized irregular multiscale structures with tunable responses. To facilitate this optimization, we employ a parametric approach to describe the design space. This description involves using two distinct sets of continuous, dimensionless field variables: a density field, $\rho(\mathbf{X}) \in [0,1]$, and multiple frequency fields, $\xi_i(\mathbf{X}) \in [0,1]$ for $i = 1, 2, \dots, M^{\text{block}}$. Here, M^{block} represents the number of building block types ($M^{\text{block}} = 4$ in this work), and $\mathbf{X}$ signifies the position vector of a material point within the design domain, Ω, using a Cartesian coordinate system.

To mitigate the infamous checkerboard phenomenon and reduce the dependency of the design variables (ρ and ξ_1–$\xi_{M^{\text{block}}}$) on mesh size, we apply a linear filter [78] to these design variables. This step results in the filtered density ($\widetilde{\rho}(\mathbf{X})$) and frequency ($\widetilde{\xi}_1(\mathbf{X})$–$\widetilde{\xi}_{M^{\text{block}}}(\mathbf{X})$) fields, which are expressed as

$$\widetilde{\rho}(\mathbf{X}) = \frac{\int_\Omega w_\rho(\mathbf{X},\mathbf{X}')\rho(\mathbf{X}')\,\mathrm{d}\mathbf{X}'}{\int_\Omega w_\rho(\mathbf{X},\mathbf{X}')\,\mathrm{d}\mathbf{X}'}, \quad \widetilde{\xi}_i(\mathbf{X}) = \frac{\int_\Omega w_\xi(\mathbf{X},\mathbf{X}')\xi_i(\mathbf{X}')\,\mathrm{d}\mathbf{X}'}{\int_\Omega w_\xi(\mathbf{X},\mathbf{X}')\,\mathrm{d}\mathbf{X}'} \quad \text{for} \quad i = 1, 2, \dots, M^{\text{block}}, \tag{5}$$

where the parameters $w_\rho(\mathbf{X},\mathbf{X}') = \max\{0, R_\rho - |\mathbf{X}-\mathbf{X}'|\}$ and $w_\xi(\mathbf{X},\mathbf{X}') = \max\{0, R_\xi - |\mathbf{X}-\mathbf{X}'|\}$ are weighting factors, and parameters R_ρ and R_ξ are the prescribed filter radii.

To ensure a 0–1 solution of the density field (ρ), we further apply a Heaviside projection [21] to the filtered density field ($\widetilde{\rho}$) and obtain the projected (physical) density field ($\overline{\rho}$) as

$$\overline{\rho}(\mathbf{X}) = \frac{\tanh(\beta\theta) + \tanh(\beta(\widetilde{\rho}(\mathbf{X}) - \theta))}{\tanh(\beta\theta) + \tanh(\beta(1-\theta))}, \tag{6}$$

where $\theta = 0.5$ is a prescribed threshold value, and β is a sharpness parameter. As for the filtered frequency fields, they need to adhere to the axiom of probability: $\sum_{i=1}^{M^{\text{block}}} \widetilde{\xi}_i(\mathbf{X}) = 1$. Moreover, we need to prescribe tighter bounds to the frequency fields in certain scenarios, i.e., $\xi_i \in [\varepsilon, 1-\varepsilon] \subseteq [0,1]$ for $i = 1, 2, \dots, M^{\text{block}}$ and some $\varepsilon \in [0, 0.5)$, for the investigation purpose and for promoting a feasible solution of the virtual growth algorithm. To achieve this, we propose to apply a normalization projection (see detailed discussion in Appendix A) to the filtered frequency fields ($\widetilde{\xi}_1$–$\widetilde{\xi}_{M^{\text{block}}}$) and obtain the projected (physical) ones ($\overline{\xi}_1$–$\overline{\xi}_{M^{\text{block}}}$) as

$$\overline{\xi}_i(\mathbf{X}) = \frac{\widetilde{\xi}_i(\mathbf{X})}{\sum_{j=1}^{M^{\text{block}}} \widetilde{\xi}_j(\mathbf{X})} \in [0,1] \quad \text{for } i = 1, 2, \dots, M^{\text{block}} \text{ and } M^{\text{block}} \geq 2. \tag{7}$$

We shall now elucidate the significance of the physical density ($\overline{\rho}$) and frequency ($\overline{\xi}_1$–$\overline{\xi}_{M^{\text{block}}}$) fields. Fig. 5(a) and (b) showcase the optimized physical density and frequency fields, respectively, for the design of Case 2 in Section 4.3. The physical density, $\overline{\rho}$, serves as a descriptor of the material arrangement within the structure, where $\overline{\rho}(\mathbf{X}) = 1$ signifies the solid regions, while $\overline{\rho}(\mathbf{X}) = 0$ designates void spaces. The physical frequency fields, $\overline{\xi}_i(\mathbf{X})$ for $i = 1, 2, 3, 4$, represent the frequency distribution of the ith type of building block, respectively. In essence, they quantify the likelihood of encountering a building block of type i at position $\mathbf{X}$. Through the parameterization of the design space utilizing the physical density ($\overline{\rho}$) and physical frequency ($\overline{\xi}_1$–$\overline{\xi}_4$) fields, we achieve an effective means of quantifying both the material arrangement and the distribution of building blocks across the design domain. Specifically, following the topology optimization in Eq. (10), which will be introduced later, we supply the optimized physical density and frequency fields to the extended virtual growth algorithm in Section 2.2. Subsequently, this algorithm is employed to generate the optimized multiscale structure, as illustrated in Fig. 5(c). During this process, topology optimization identifies the optimal material property distribution, achieved by the optimized physical density and frequency fields. The virtual growth algorithm then utilizes these optimized fields to actualize the multiscale design, enabling seamless collaboration between topology optimization and the virtual growth algorithm.

3.3.2. Elasticity interpolation

With the design space parameterized by the physical density ($\overline{\rho}$) and frequency ($\overline{\xi}_1$–$\overline{\xi}_{M^{\text{block}}}$) fields, we can now interpolate the material's elasticity tensor ($\mathbb{C}$) for varying values of these fields. The interpolation rule is defined as

$$\mathbb{C} = \left[\varepsilon_\rho + (1-\varepsilon_\rho)\overline{\rho}^{p_\rho}\right] \cdot \left[(1-\alpha)\cdot \mathbb{C}_d(\overline{\xi}_1, \overline{\xi}_2, \dots, \overline{\xi}_{M^{\text{block}}}) + \alpha \cdot \mathbb{C}_p\right], \tag{8}$$

which is essentially similar to those in [79–81]. Specifically, the first term involves the interpolation of material existence through the application of a modified SIMP rule [82,83] to the physical density field. Simultaneously, the second term employs a weighted sum to interpolate designable and passive regions. In Eq. (8), the parameter ε_ρ is a small positive number ($\varepsilon_\rho = 10^{-6}$ in this work) that prevents singularity of the global stiffness matrix in the FEA. The parameter p_ρ is a power that penalizes intermediate values of the physical density field $\overline{\rho} \in (0,1)$ and promotes a 0–1 solution. The function $\alpha(\mathbf{X})$ characterizes the designable (Ω_d) and passive ($\Omega_p = \Omega \setminus \Omega_d$) regions and is defined as

$$\alpha(\mathbf{X}) = \begin{cases} 0, & \text{if } \mathbf{X} \in \Omega_d \\ 1, & \text{if } \mathbf{X} \in \Omega_p \end{cases}.$$

The variables $\mathbb{C}_p(\mathbf{X})$ and $\mathbb{C}_d(\mathbf{X})$ are the elasticity tensors of the passive and designable materials, respectively. Recall that $\mathbb{C}_d(\mathbf{X})$ can be computed from the trained neural network in Eq. (4).

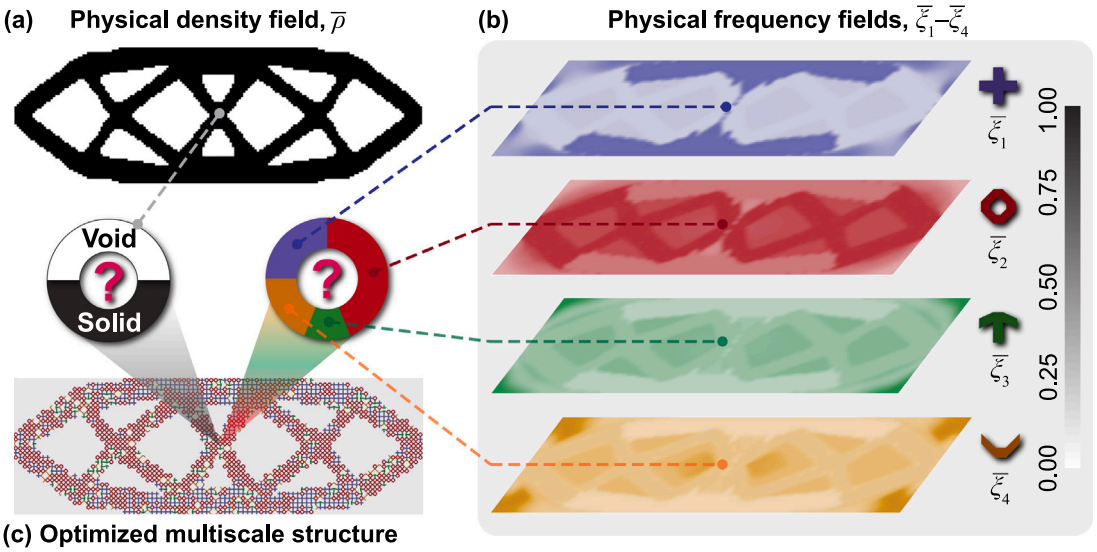

Fig. 5. Illustration of the design space parameterization. (a)–(b) Optimized physical density and frequency fields for the design exemplified in Case 2 of Section 4.3. (c) Resultant optimized multiscale structure.

Next, the static equilibrium equations, based on the balance of linear momentum [75], are established as

$$\begin{cases} \mathrm{Div}(\mathbb{C} : \mathbf{E}) + \overline{\mathbf{b}} = \mathbf{0}, & \mathbf{X} \in \Omega \\ \mathbf{u} = \overline{\mathbf{u}}, & \mathbf{X} \in \partial\Omega_D \\ (\mathbb{C} : \mathbf{E}) \cdot \mathbf{N} = \overline{\mathbf{t}}, & \mathbf{X} \in \partial\Omega_{\mathcal{N}} \end{cases}, \tag{9}$$

where the variable $\mathbf{E} = (\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathsf{T}})/2$ represents the infinitesimal strain tensor, and $\mathbf{u}$ denotes the displacement field. The variables $\overline{\mathbf{b}}$ and $\overline{\mathbf{t}}$ correspond to the prescribed body force and traction vectors acting on the design domain (Ω) and its Neumann boundary ($\partial\Omega_{\mathcal{N}}$), respectively. The variable $\overline{\mathbf{u}}$ is the displacement field prescribed on the Dirichlet boundary ($\partial\Omega_D$). The symbol $\mathbf{N}$ denotes the unit outward normal vector of the boundary $\partial\Omega_{\mathcal{N}}$.

3.3.3. Topology optimization formulation

In accordance with the previously established design space parameterization and elasticity interpolation, we now present the proposed mathematical formulation for topology optimization as

$$\begin{cases} \underset{\rho,\ \xi_1,\ \xi_2,\ \ldots,\ \xi_{M^{\mathrm{block}}}}{\text{minimize}} : & J(\rho, \xi_1, \xi_2, \ldots, \xi_{M^{\mathrm{block}}}); \\ \text{subjected to:} & g_1 = \dfrac{1}{|\Omega|}\displaystyle\int_\Omega \overline{\rho}\,\mathrm{d}\mathbf{X} - \overline{V} \le 0, \\ & g_2 = J_S - \overline{J}_S \ge 0, \\ & 0 \le \rho \le 1, \\ & \varepsilon \le \xi_i \le 1 - \varepsilon, \quad \text{for } i = 1, 2, \ldots, M^{\mathrm{block}}; \\ \text{with:} & \text{equilibrium equations in Eq. (9).} \end{cases} \tag{10}$$

Here, we recall both $\rho(\mathbf{X}) \in [0,1]$ and $\xi_i(\mathbf{X}) \in [0,1]$ for $i = 1, 2, \ldots, M^{\mathrm{block}}$ are macroscale design variables, which are field variables defined over the entire domain. Their corresponding physical variables, $\overline{\rho}(\mathbf{X}) \in [0,1]$ and $\overline{\xi}_i(\mathbf{X}) \in [0,1]$, describe the macroscale material existence and the frequency hints of building blocks throughout the design domain, respectively. The variable J is the objective function that will be introduced below. The symbols g_1 and g_2[1] represent the volume and total strain energy constraint functions, respectively. Here $\overline{V}$ is the upper bound of the material volume fraction, and J_S and $\overline{J}_S$ are the total strain energy (defined in Eq. (13)) and its lower bound, respectively. Again, the parameter $\varepsilon \in [0, 0.5)$ is a user-defined value determining the bounds of the frequency variables (ξ_1–$\xi_{M^{\mathrm{block}}}$).

We emphasize that the proposed topology optimization formulation is capable of controlling diverse structural responses. In this study, we primarily illustrate how to program three representative types of responses — node-wise mechanical displacement

[1] The total strain energy constraint, g_2, is only activated in Case 2 of Section 4.2.

cloaking, the manipulation of element-wise strain energy density, and the maximization of global structural stiffness. In particular, to measure the error in displacement fitting, we define

$$J = J_{\mathcal{U}} = \|\boldsymbol{\theta}_{\mathcal{U}} \odot (\mathbf{U} - \overline{\mathbf{U}}) \oslash \overline{\mathbf{U}}\|_p, \tag{11}$$

where $\boldsymbol{\theta}_{\mathcal{U}}$ signifies an indicator vector with $\theta_{\mathcal{U},i} = 1$ if the ith nodal degree-of-freedom (DOF) is controlled and $\theta_{\mathcal{U},i} = 0$ otherwise. The variable $\mathbf{U}$ denotes the global displacement vector, while $\overline{\mathbf{U}}$ is the target displacement vector. The operator $\|\cdot\|_p$ is employed to gauge the p-norm of a vector. The operators $\odot$ and $\oslash$ denote element-wise (Hadamard) vector multiplication and division, respectively.

In a similar vein, we quantify the fitting error of strain energy density through

$$J = J_{\mathcal{W}} = \|\boldsymbol{\theta}_{\mathcal{W}} \odot (\mathbf{W} - \overline{\mathbf{W}})\|_p, \tag{12}$$

where $\boldsymbol{\theta}_{\mathcal{W}}$ is an indicator vector with $\theta_{\mathcal{W},e} = 1$ if macro-element e is under control and $\theta_{\mathcal{W},e} = 0$ otherwise. The variable $\mathbf{W}$ is a vector with W_e representing the strain energy density averaged over macro-element e. Meanwhile, the variable $\overline{\mathbf{W}}$ is the target strain energy density vector. The average strain energy density of macro-element e can be computed as

$$W_e = \frac{1}{2|\Omega_e|} \int_{\Omega_e} \mathbf{E} : \mathbb{C} : \mathbf{E} \mathrm{d}\mathbf{X},$$

where Ω_e represents the domain occupied by macro-element e.

Finally, to quantify the overall structural stiffness, we define the (minus) total strain energy as

$$J = -J_S = -\frac{1}{2} \int_{\Omega} \mathbf{E} : \mathbb{C} : \mathbf{E} \mathrm{d}\mathbf{X}. \tag{13}$$

Note that we put a minus sign here because maximizing the total strain energy results in maximized structural stiffness under displacement loading.

To solve the topology optimization problem outlined in Eq. (10), we employ the adjoint method [21] for evaluating the relevant sensitivities. This sensitivity evaluation involves the backward propagation of a neural network (Fig. 4(a)), and we provide a detailed description of this intricate sensitivity analysis in Appendix B. Furthermore, we utilize the method of moving asymptotes (MMA) [84] to update the design variables (ρ and ξ_1–$\xi_{M^{\mathtt{block}}}$). The solution procedure for the topology optimization problem is presented in Algorithm 4. To implement this solution procedure, we utilize an in-house Python code [74] while making use of the open-source FEniCSx [85] and PyTorch [77] packages. The FEniCSx package facilitates parallel computation for the FEA and a portion of the sensitivity analysis, while the PyTorch package enables both the forward pass of the neural network (Fig. 4(a)) and the remaining sensitivity analysis. Appendix B also briefly discusses the combined utilization of automatic differentiation from these two packages to carry out the complete sensitivity analysis.

Algorithm 4: Solution procedure for the topology optimization problem in Eq. (10)

Inputs: optimization iteration counter, $n \leftarrow 1$; maximum iteration, N^{max}; the initial values of design variables, $\rho \leftarrow \rho_0$ and $\xi_i \leftarrow \xi_{i,0}$ for $i = 1, 2, ..., M^{\mathtt{block}}$;
while $n \leq N^{max}$ **do**
  Apply the linear filter (Eq. (5)) on the design variables, ρ and ξ_1–$\xi_{M^{\mathtt{block}}}$, to obtain the filtered variables, $\widetilde{\rho}$ and $\widetilde{\xi}_1$–$\widetilde{\xi}_{M^{\mathtt{block}}}$;
  Apply the Heaviside projection (Eq. (6)) on the filtered density variable, $\widetilde{\rho}$, to obtain the physical one, $\overline{\rho}$;
  Apply the normalization projection (Eq. (7)) on the filtered frequency variables, $\widetilde{\xi}_1$–$\widetilde{\xi}_{M^{\mathtt{block}}}$, to obtain the physical ones, $\overline{\xi}_1$–$\overline{\xi}_{M^{\mathtt{block}}}$;
  Substitute the physical variables ($\overline{\rho}$ and $\overline{\xi}_1$–$\overline{\xi}_{M^{\mathtt{block}}}$) into Eq. (8) to evaluate the interpolated elasticity tensor, $\mathbb{C}$;
  Use the interpolated elasticity tensor, $\mathbb{C}$, to establish the static equilibrium equations in Eq. (9) and use FEA to numerically solve the global displacement vector, $\mathbf{U}$;
  Based on the global displacement vector, $\mathbf{U}$, evaluate the function values, J, g_1, and g_2, and perform the sensitivity analysis (see Appendix B) to obtain the function gradients with respect to the design variables, ρ and ξ_1–$\xi_{M^{\mathtt{block}}}$;
  Based on the function values and gradients, utilize the MMA optimizer to update the design variables, ρ and ξ_1–$\xi_{M^{\mathtt{block}}}$;
  Update the iteration counter, $n \leftarrow n + 1$;
end

4. Examples

In this section, we present four sample examples to demonstrate the application of the proposed topology optimization methodology for designing irregular multiscale structures with customizable responses. We specifically program three types of structural responses — node-wise displacement, element-wise strain energy density, and global structural stiffness. All computations are performed using our in-house Python code on a workstation equipped with an AMD Ryzen Threadripper PRO 3995WX CPU featuring 64 cores and 128 threads, an NVIDIA Quadro RTX 4000 graphics card, and 256 GB of memory.

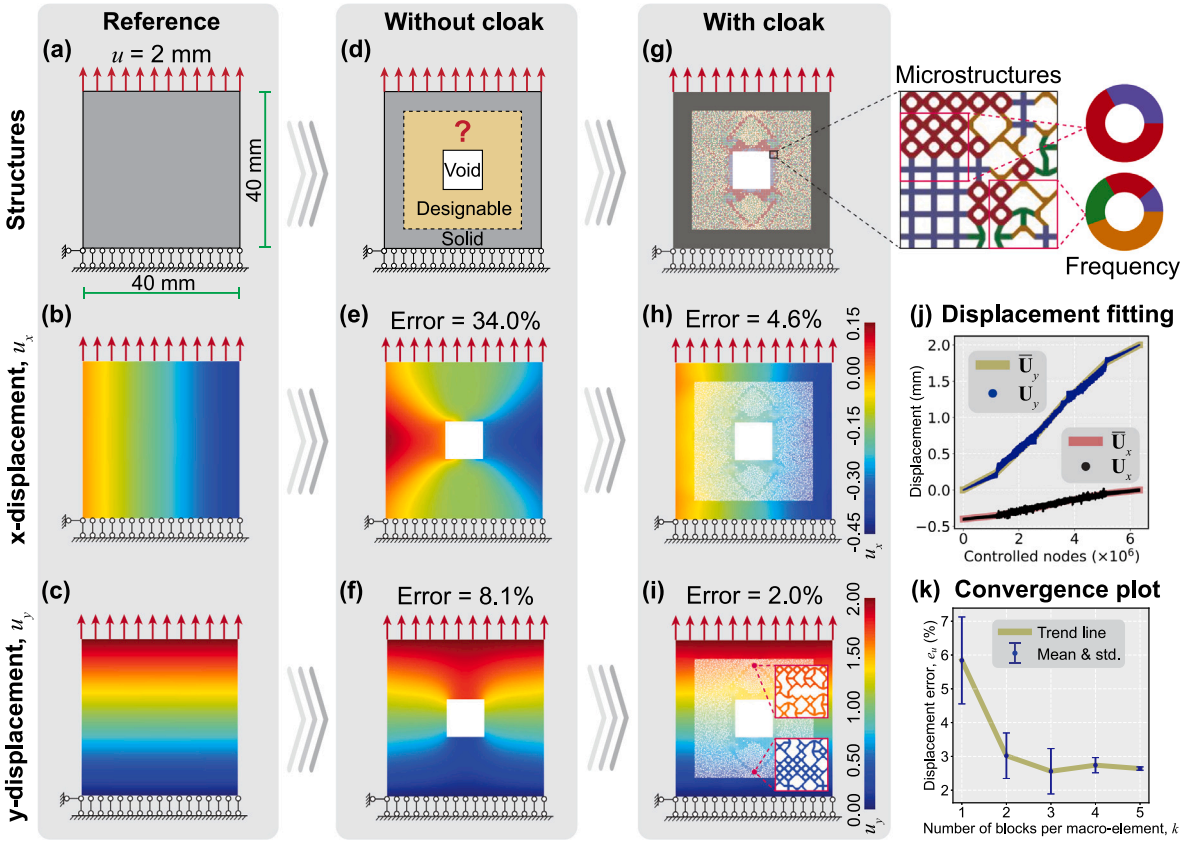

Fig. 6. Mechanical cloaking achieved by programming the displacement field. (a) The design setup of the reference domain. (b)–(c) Displacements field in the horizontal (x-) and vertical (y-) directions. (d) The structure with a void in the middle. (e)–(f) Distorted displacement fields due to the void. (g) Optimized irregular multiscale structure. The inset shows the zoom-in view of the microstructure and the associated frequency hints of building blocks. (h)–(i) Recovered displacement fields evaluated at the microscale. The insets show the zoom-in views of the microstructural displacement fields. (j) Displacement fitting effect of the optimized structure, evaluated at the microscale. (k) A convergence plot demonstrating the relationship between the displacement fitting error and the number of building blocks (in each direction) per macro-element.

4.1. Mechanical displacement cloaking

4.1.1. Programming the structural displacement response

In this subsection, we showcase the capability of the proposed virtual-growth-based topology optimization methodology in programming node-wise displacement responses. Our objective is to achieve a mechanical cloaking effect [15,86] within a structure, making the displacement field around a void indistinguishable from the surrounding material (Fig. 6). In Fig. 6(a), we present the reference structure subjected to uniaxial tension with an applied displacement loading of u=2 mm. This reference structure is composed of an isotropic linear elastic material with Young's modulus $E = 50$ MPa and Poisson's ratio $\nu = 0.2$. The domain is discretized using a structured grid consisting of 50×50 first-order quadrilateral Lagrange macro-elements. By employing a standard FEA, we compute the displacement fields in both horizontal and vertical directions, denoted as u_x and u_y respectively (Fig. 6(b)–(c)).

Subsequently, we introduce a square void within the structure, as illustrated in Fig. 6(d). Such a void can be utilized for concealing objects in specific applications. As expected, the removal of material leads to significant alterations in the displacement fields (Fig. 6(e)–(f)), both in the horizontal and vertical directions (u_x and u_y, respectively), when compared to the reference structure (Fig. 6(b)–(c)). To quantitatively assess these displacement changes, we calculate the relative errors, denoted as e_x and e_y, for the horizontal and vertical directions, respectively. These relative errors are given by

$$e_x = \frac{\|\mathbf{U}_x - \overline{\mathbf{U}}_x\|_2}{\|\overline{\mathbf{U}}_x\|_2} \quad \text{and} \quad e_y = \frac{\|\mathbf{U}_y - \overline{\mathbf{U}}_y\|_2}{\|\overline{\mathbf{U}}_y\|_2}.$$

Here, $\mathbf{U}_x$ and $\mathbf{U}_y$ represent the global displacement vectors in the horizontal and vertical directions, respectively, within the displacement control region (encompassing the entire structure excluding the void). Meanwhile, $\overline{\mathbf{U}}_x$ and $\overline{\mathbf{U}}_y$ are the corresponding global displacement vectors of the reference structure. The computed relative errors are $e_x = 34.0\%$ and $e_y = 8.1\%$, highlighting the notable changes in the displacement fields due to the void's presence.

To achieve the desired mechanical cloaking effect, we undertake the optimization of the materials surrounding the void (as depicted in the designable region in Fig. 6(d)). The primary goal is to minimize the displacement errors, e_x and e_y. We initiate this process with four building block types, namely `cross`, `rhombus`, `arrow`, and `corner` in Fig. 2. Our methodology focuses on optimizing their frequency hints by adjusting the frequency variables, ξ_1–ξ_4, while keeping the density variable fixed at $\rho \equiv 1$. The key optimization parameters employed in this process are as follows. The p-norm parameter for the displacement error in Eq. (11) is set as $p = 2$. We maintain an allowable volume fraction of $\overline{V} = 1$, a frequency field range of [0, 1] ($\epsilon = 0$), and a filter radius of $R_\xi = 10^{-4}$ mm for frequency fields. Finally, we cap the optimization process with a maximum of $N^{max} = 400$ iterations.

In the current example, we conduct topology optimization using parallel computation across 4 threads, resulting in a total computational time of 0.568 h. Utilizing the optimized frequency fields, we apply the extended virtual growth algorithm to construct the irregular multiscale structure displayed in Fig. 6(g). In this structure, we arrange 3×3 building blocks within each macro-element. Upon closer inspection (as seen in the zoom-in view), we can observe the seamless integration of microstructures with their corresponding optimized frequency hints. To verify the achieved mechanical cloaking effect, we perform standard FEA for the optimized structure in Fig. 6(g). The resulting displacement fields in the horizontal and vertical directions, evaluated at the microscale, are presented in Fig. 6(h)–(i), respectively. Notably, the displacement fields of the optimized structure closely align with those of the reference structure in Fig. 6(b)–(c). This alignment is further substantiated by the relative errors, $e_x = 4.6\%$ and $e_y = 2.0\%$, as well as a direct quantitative comparison (Fig. 6(j)) between the actual displacement, $\mathbf{U}_x$ ($\mathbf{U}_y$), and the target displacement, $\overline{\mathbf{U}}_x$ ($\overline{\mathbf{U}}_y$). Moreover, to emphasize that this promising displacement fitting effect is not confined to specific domain geometries, Appendix C presents displacement fields for a structure with a disk-shaped void and a ring-shaped designable region. In this case, we also observe a good agreement between the actual and target displacements, remarking the versatility of the proposed approach. We also stress that our primary focus here is to showcase displacement tunability. As a result, we concentrate solely on achieving the mechanical cloaking effect in a single loading direction. To extend this cloaking effect to multiple directions [87], one can integrate multiple load cases into the proposed methodology.

4.1.2. A convergence study

In Fig. 6(j), we indeed observe a satisfactory displacement fitting effect, demonstrating the successful realization of mechanical cloaking. However, it is valuable to conduct a convergence study to assess the robustness of the proposed methodology. This investigation is particularly pertinent because the underlying virtual growth algorithm is stochastic, and the structural response also relies on the number of building blocks filled within each macro-element. Furthermore, the proposed multiscale topology optimization framework implicitly assumes that the microstructures locally satisfy periodic boundary conditions. This assumption may be invalid if adjacent microstructures have significantly different design variables or if the microstructures are near the boundaries of the optimized design. In this work, linear filters (Eq. (5)) for the design variables are employed, if needed, to ensure a smooth variation across the design domain, resulting in adjacent unit cells exhibiting similar material properties and adhering to periodic boundary conditions. However, the microstructures near the boundaries of the optimized structure may not strictly conform to periodic boundary conditions, potentially causing slight discrepancies between microscale and macroscale structural responses. We have also addressed this issue through the following convergence study.

To conduct the convergence study, we introduce the parameter k, where k denotes the number of building blocks per macro-element in each direction, taking values from the set $\{1, 2, 3, 4, 5\}$. For each k value, we generate 5 samples using the same optimized frequency fields by executing the extended virtual growth algorithm multiple times. Subsequently, we discretize these samples to ensure they have a similar number of DOFs, followed by conducting standard FEA to evaluate the displacement field. To quantitatively evaluate the convergence of our approach, we compute the relative displacement error at the microscale using Eq. (11). In Fig. 6(k), we present the mean values and standard deviations of the relative errors, denoted as e_u, for different values of k. Additionally, we provide a trend line fitted according to the mean values. The following observations can be made from this analysis. First, the mean error decreases as the number of building blocks per macro-element (k) increases. This reduction in error is attributed to the presence of more building blocks, which results in a more pronounced contrast between the length scales of the macro-element and the building block. This contrast, in turn, leads to higher precision in the homogenized material properties and subsequently less error in the performance prediction. Second, the standard deviation, which quantifies the variability in performance among different samples for the same k value (essentially, the reproducibility of the optimized design), also decreases with larger values of k. This reduction is consistent with the law of large numbers, as a larger number of filled building blocks tends to converge the actual frequency hints of building blocks towards the desired one. Last, the trend line fitted to the mean values indicates that the mean error converges when $k \geq 3$. These observations highlight the practical use of the method, with larger k values leading to improved convergence in terms of displacement error.

Table 1 presents a comprehensive overview of our convergence study, offering detailed statistics that include the average number of DOFs, relative displacement errors, virtual growth time required for sample generation, and computational time for FEA across different block numbers. It is noteworthy that FEA tasks involve the discretization of building blocks into sufficiently small finite elements to minimize discretization errors, resulting in FEA models with 11–14 million DOFs. To efficiently manage such computationally demanding tasks, we employ parallel computation utilizing 12 threads within our in-house Python code, resulting in FEA computational time of approximately 150 s. Finally, we conclude that our proposed methodology has successfully delivered an optimized irregular multiscale structure with tunable node-wise displacement responses. These tunable responses are not only accurate but also highly reproducible, as demonstrated by the findings of this convergence study.

Table 1
Statistics of the convergence study for the optimized structure in Fig. 6(g).

Block numbers	DOF numbers	Displacement errors	Growth time (s)	FEA time (s)
1 × 1	13,040,550	5.84%	5.2	153.6
2 × 2	11,939,651	3.02%	31.0	137.6
3 × 3	12,734,537	2.56%	197.2	146.2
4 × 4	13,512,658	2.74%	338.2	157.0
5 × 5	13,808,400	2.64%	1194.0	162.0

4.2. Programming the strain energy density

In this subsection, we illustrate how to program another crucial type of response — element-wise strain energy density. Through this example, we also demonstrate the simultaneous optimization of both density and frequency variables and address the challenges posed by an unstructured grid. Fig. 7(a) shows the design setup including the design domain, boundary conditions, and control regions. The design domain consists of a centrally fixed disk, with rotational displacement loading applied at a magnitude of $u = 1$ mm around the disk. The control region is in the shape of a Taiji symbol, featuring two interconnected, anti-symmetric areas. Our objective is to manipulate the strain energy density, denoted as W, within these two regions to achieve values of $\overline{W} = 0.05$ and 0.15 N/mm^2, respectively. We also require that the outer ring-shaped region remains solid ($\rho \equiv 1$).

To program the strain energy density, we consider two distinct cases — optimizing frequency variables exclusively in Case 1 and optimizing both density and frequency variables in Case 2. Subsequently, we leverage the proposed methodology to design irregular multiscale structures. For topology optimization, we discretize the design domain using an unstructured grid consisting of 16,144 first-order quadrilateral Lagrange macro-elements, each with a size $h = 0.3$ mm. The remaining topology optimization parameters are as follows. The parameter governing the p-norm of the fitting error of the strain energy density in Eq. (12) is $p = 2$. The allowable volume fraction is $\overline{V} = 1$ in Case 1 and $\overline{V} = 0.7$ in Case 2. The frequency field range is [0, 1] ($\varepsilon = 0$). The filter radii of the density and frequency fields are $R_\rho = 4$ mm and $R_\xi = 10^{-4}$ mm, respectively. The parameters used in the interpolated elasticity in Eq. (8) are $\varepsilon_\rho = 10^{-6}$ and $p_\rho = 3$. The maximum optimization iteration is $N^{max} = 400$. Additionally, for Case 2, the sharpness parameter in the Heaviside projection (Eq. (6)) is initially set as $\beta = 1$ and then doubled every 40 iterations until $\beta = 256$. Here, we activate the strain energy constraint, g_2, in the topology optimization formulation (Eq. (10)) for Case 2 to ensure an adequate level of structural stiffness. The introduction of this constraint serves to prevent potential macroscopic structural disconnection, floating members, and thin joints. However, the necessity of this minimum strain energy constraint might vary, especially if the target strain energy density is sufficiently large. In this regard, we set the lower bound for the total strain energy as $\overline{J}_S = 150$ N · mm, determined through a trial-and-error process. It is important to emphasize that this lower bound should be less than the strain energy of the corresponding stiffness maximization design.

Note that an unstructured grid is used in topology optimization to yield smooth boundaries of control regions. However, a structured grid with square elements in 2D and cubic elements in 3D is required when creating the material database. To address this requirement, we propose a two-grid projection scheme as depicted in Appendix D. Specifically, this scheme involves projecting the optimized density and frequency fields obtained from an unstructured grid used in topology optimization onto a structured grid. Subsequently, these projected fields guide the virtual growth process. With the implementation of this proposed two-grid projection scheme, we present the optimized irregular multiscale structure for Case 1 in Fig. 7(b). In this particular instance, the density variable is held constant at $\rho \equiv 1$, while the frequency variables, ξ_1–ξ_4, are subjected to optimization. The generated structure is composed of identical building blocks as shown in Fig. 2, achieving seamless integration. By performing FEA on the optimized structure, we illustrate the strain energy density field in Fig. 7(c). Notably, the strain energy densities within the two control regions align with the target values, as indicated by the matching colors to the design setup in Fig. 7(a). This alignment is further evidenced by a direct comparison between the actual and target strain energy densities of macro-elements within the two control regions, as demonstrated in Fig. 7(d).

Compared to classical multi-material topology optimization [16,28], which rely on finite and discrete candidate materials, our proposed approach operates within a vast and continuous material space, as exemplified by the attainable material properties presented in [71]. Consequently, our proposed methodology possesses the capability to program intricate responses, as evidenced by the complex strain energy density distribution illustrated in Fig. 7(c). To further underscore the robust programmability of strain energy density, we concentrate on the design of a lightweight structure in Case 2, wherein the total material utilization is restricted to 70% of that in Case 1. Fig. 7(e) showcases the resulting irregular multiscale structure generated based on the optimized density and frequency fields. Despite employing fewer materials, this structure also successfully attains the target strain energy densities, as evidenced by the strain energy density distribution depicted in Fig. 7(f), and the comparison between the actual and target strain energy densities presented in Fig. 7(g). To perform topology optimization, we utilize parallel computation across 4 threads for both cases, resulting in computational time of 3.15 and 4.40 h, respectively. For clarity and ease of comprehension, we visually elucidate the concept of structural growth in Fig. 7(h). This illustration demonstrates the incremental assembly of building blocks into the grid, with each unit seamlessly fitting in accordance with the optimized density and frequency fields.

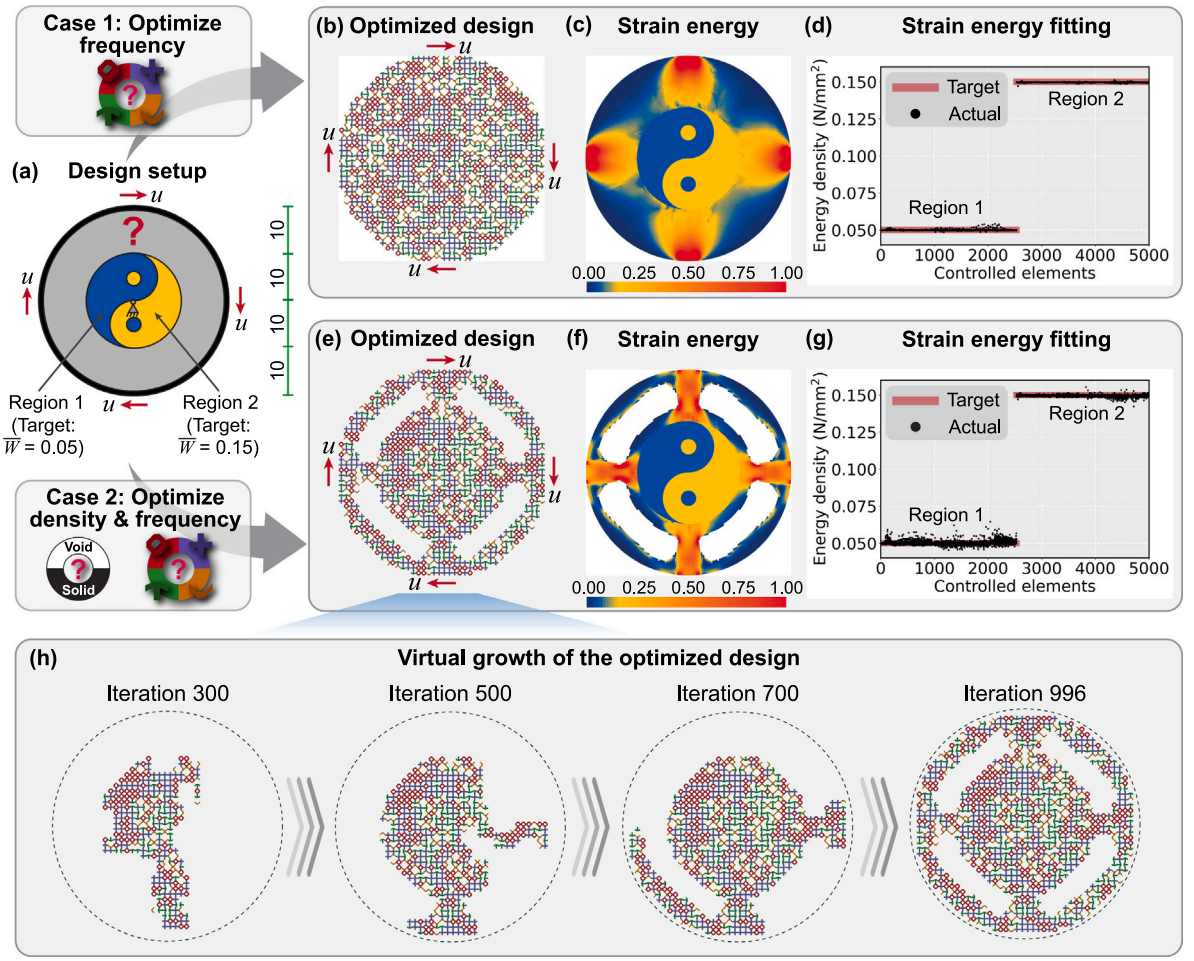

Fig. 7. Programming the strain energy density on a rotational disk. (a) Design setup including the design domain, boundary conditions, and the control regions. (b)–(d) Optimized design, strain energy density distribution (unit: N/mm^2), and the fitting effect of the strain energy density of Case 1. (e)–(g) The counterparts of Case 2. (h) The virtual growth process of the optimized structure in (e). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.3. Planar beam with maximized structural stiffness

Our previous discussions have primarily centered on programming local structural responses, such as node-wise displacement and element-wise strain energy density. In this subsection, our focus shifts towards optimizing a global response — the structural stiffness — by maximizing the total strain energy. Fig. 8(a) depicts the design domain of a planar beam subjected to three-point bending with a prescribed displacement loading of $u = 5$ mm. Our objective is to maximize the overall structural stiffness through the optimization of both the density and frequency fields. To accomplish this goal, we discretize the design domain using a structured grid consisting of 180×60 first-order quadrilateral Lagrange macro-elements. Subsequently, we employ the proposed methodology, which encompasses both topology optimization and the virtual growth algorithm, to facilitate the optimization process.

To explore the impact of the lower bound (ϵ) applied to the frequency variables on structural performance, we examine two distinct cases, each featuring different ϵ values: $\epsilon = 0$ in Case 1 and $\epsilon = 0.15$ in Case 2, and the resulting ranges of frequency fields are [0, 1] and [0.15, 0.85], respectively. The remaining topology optimization parameters are as follows. The allowable volume fraction is $\overline{V} = 0.5$. The filter radii of the density and frequency fields are $R_\rho = R_\xi = 4.0$ mm in Case 1 and $R_\rho = R_\xi = 2.0$ mm in Case 2. The parameters used in the interpolated elasticity in Eq. (8) are $\epsilon_\rho = 10^{-6}$ and $p_\rho = 3$. The sharpness parameter in the Heaviside projection (Eq. (6)) is initially set as $\beta = 1$ and then doubled every 40 iterations until $\beta = 256$. Finally, the maximum optimization iteration is $N^{max} = 400$.

In this example, we leverage parallel computing across 8 threads to accelerate the topology optimization process. This optimization effort yields a total runtime of 1.69 h for Case 1 and 1.72 h for Case 2, and approximately 77% of the computational time is due to the solution update with an MMA optimizer [84]. Utilizing the optimized density and frequency fields, we employ the same set of building blocks as shown in Fig. 2 to generate the optimized structures for both cases (see digital rendering in

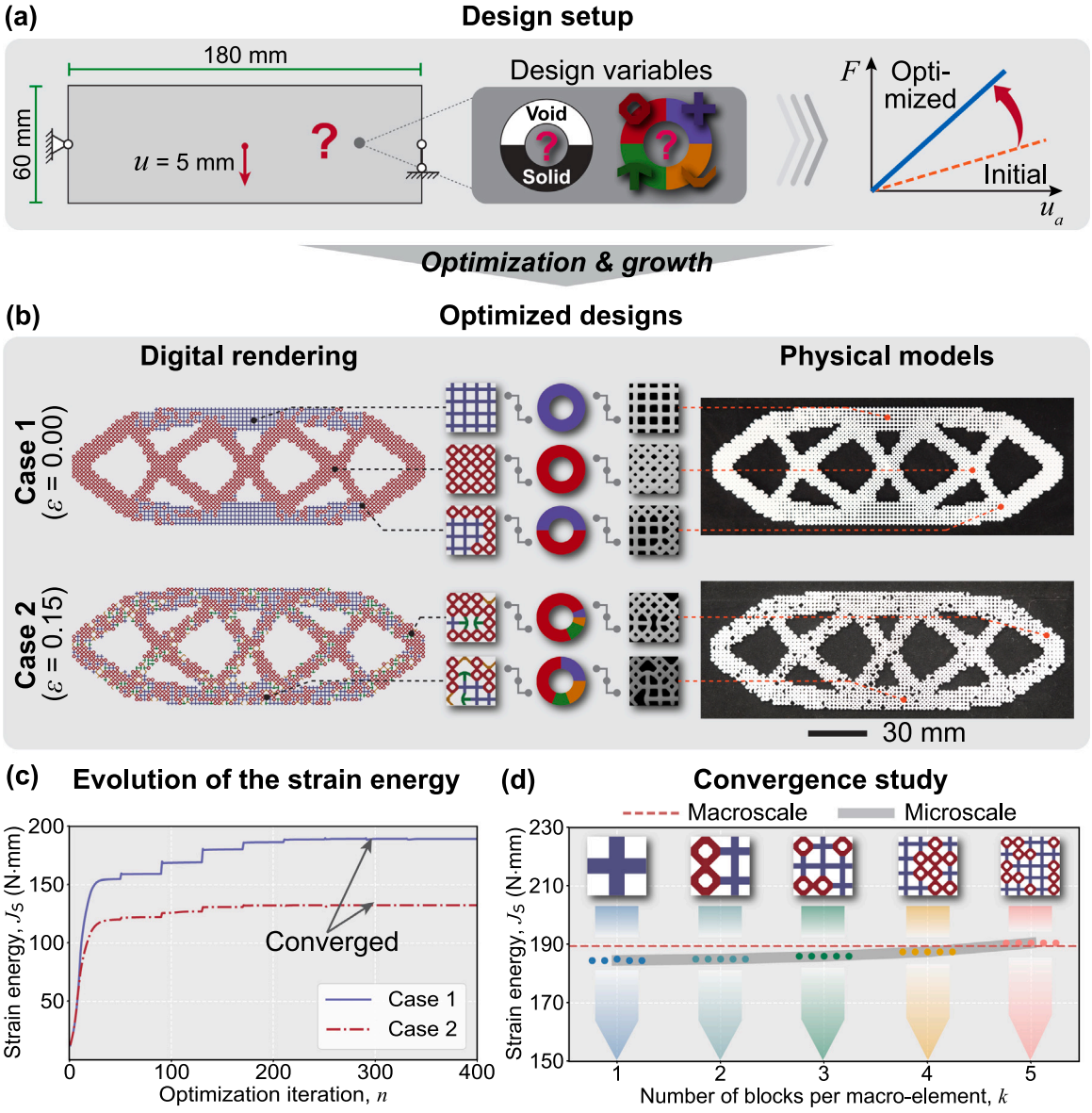

Fig. 8. Planar beam with maximized overall structural stiffness. (a) Design setup including the design domain, design variables, and the optimization objective. (b) The digital rendering and physical models of the optimized designs. The zoom-in views demonstrate the microstructures and the associated frequency hints of building blocks at different locations. (c) The histories of the strain energies of two cases during topology optimization. (d) A convergence study for investigating the relationship between the total strain energy and the number of building blocks (in each direction) per macro-element. The dots represent the strain energy of different samples, evaluated at the microscale, and the dashed line represents the macroscale value.

Fig. 8(b)). Upon comparing the two optimized structures, it becomes evident that in Case 1, a clear preference emerges for two types of building blocks — namely, the `cross` building blocks, which cluster in the top and bottom members, and the `rhombus` building blocks, which concentrate in the middle inclined members. This preference arises from a fundamental structural analysis, where the principal stress aligns horizontally in the top and bottom members, and diagonally in the middle members. As a result, the selection of `cross` and `rhombus` building blocks constructs an efficient load path, thereby maximizing the overall stiffness. Conversely, the optimized structure in Case 2 exhibits a mixed utilization of all four building block types. This variation arises due to the imposition of a lower bound (and also an upper bound) on the frequency variables, which leads to a more diverse selection of building blocks.

Fig. 8(b) additionally provides physical models of the optimized structures in both cases. These physical models are created using a commercial masked-stereolithography (m-SLA) process, specifically employing the Elegoo Saturn 2 printer and Elegoo water-washable photopolymer resins. When comparing the digital renderings with the physical models of the optimized structures, we observe that all the geometrical features are faithfully preserved in the physical models. Furthermore, the virtual growth algorithm

Table 2
Statistics of the convergence study for the optimized structure of Case 1 in Fig. 8(b).

Block numbers	DOF numbers	Strain energy errors	Growth time (s)	FEA time (s)
1 × 1	24,825,370	−2.59%	24.1	273.6
2 × 2	23,342,659	−2.37%	273.0	251.2
3 × 3	26,506,099	−1.85%	1318.0	299.2
4 × 4	29,734,792	−1.06%	3914.0	344.2
5 × 5	29,579,150	0.53%	9300.0	369.6

plays a pivotal role in achieving seamless connections between structural members at the joints, contributing to the structural integrity and functionality of the physical models. This successful translation from digital rendering to physical realization remarks the manufacturability of the designs obtained through our proposed methodology.

Fig. 8(c) shows the evolution of total strain energies (J_S) of two optimized structures during topology optimization. The strain energies start to converge after around 200 optimization iterations for both cases. Notably, the optimized structure in Case 1 exhibits a higher strain energy (higher stiffness), thus demonstrating superior mechanical efficiency. This result stems from the imposition of fewer constraints on the bounds of the frequency variables, allowing the structure to freely select the most effective building blocks. However, we emphasize that imposing tighter bounds for the frequency variables is necessary for certain problems (not shown here) to find a feasible solution within the virtual growth algorithm. Additionally, it is crucial to avoid overly tight bounds for the frequency variables, as those bounds can severely restrict the design space. Based on our tests, setting $\epsilon \in [0, 0.15]$, i.e., establishing frequency bounds from [0, 1] to [0.15, 0.85], has proven effective for most examples.

To further study the characteristics of the optimized structure in Case 1, we conduct a convergence study that examines how the number of building blocks influences the total strain energy (structural stiffness). Similar to the investigation depicted in Fig. 6(k), we employ the extended virtual growth algorithm to generate five samples for varying numbers of building blocks (in each direction) per macro-element, denoted as $k = 1, 2, 3, 4, 5$. These samples are generated based on the same optimized physical density and frequency fields. Subsequently, we perform standard FEA at the microscale using parallel computation with 12 threads to assess the actual total strain energy, as illustrated in Fig. 8(d). Fig. 8(d) reveals that the microscale total strain energy remains consistent for structures with the same k value, indicating good reproducibility. Moreover, the microscale total strain energy exhibits an increasing trend with higher values of k, ultimately converging to the macroscale strain energy determined through the evaluation of homogenized material properties. Detailed statistics of the convergence study are presented in Table 2, encompassing information on block numbers, DOF numbers, the relative errors between microscale and macroscale total strain energies, and the computational time required for both the virtual growth process and FEA. We highlight that our in-house Python code, equipped with parallel computing capabilities, efficiently manages the computationally intensive FEA tasks encompassing 24 to 30 million DOFs with a relatively short computational time of approximately 300 s.

Despite the promising convergence trend, it is important to conduct a comprehensive investigation comparing the newly proposed multiscale structures (Fig. 8(b)) with the corresponding single-scale structure (Fig. 9(a)). The single-scale design in Fig. 9(a) is optimized under identical boundary conditions in Fig. 8(a). Utilizing the classical topology optimization formulation from [21], this single-scale design maintains an allowable volume fraction equivalent to the actual microscale solid material of Case 1 in Fig. 8(b), denoted as $\overline{V} = 0.267$. The comparison between Fig. 8(b) and Fig. 9(a) shows that the macroscale topology of the proposed multiscale designs aligns with the single-scale structure. This alignment suggests that the proposed framework effectively optimizes the macroscale topology. Additionally, Fig. 9(b) demonstrates the evolution of the strain energy for the single-scale structure, which converges to 429.5 N·mm after 400 optimization iterations. In contrast, Cases 1 and 2 of the multiscale structures in Fig. 8(b) exhibit converged strain energies of 189.5 N·mm and 132.5 N·mm, respectively. These values are 56% and 69% lower than the strain energy of the single-scale structure. Such performance differences are anticipated as multiscale structures that utilize prescribed primitive unit cells either explicitly or implicitly impose local volume constraints compared to the single-scale designs, thereby limiting the attainable material properties within the Hashin–Strikman bounds. However, despite the performance gap, multiscale structures remain preferable in specific scenarios, such as situations where their porosity can enhance failure resistance [55,72,88]. We also emphasize that the current example is presented solely to illustrate the application of the proposed formulation in controlling global structural responses of multiscale structures.

4.4. Rotational spherical shell with maximized structural stiffness

In this concluding illustration, we further extend the proposed methodology to the optimization of structural responses within the realm of 3D designs. Our focus centers on the maximization of structural stiffness, achieved by optimizing the overall strain energy with our established methodology. Fig. 10(a) provides a comprehensive depiction of the design configuration, encompassing the design domain, boundary conditions, design variables, and the primary optimization objective. The design domain is a spherical shell featuring an outer radius of $R_{\text{out}} = 50$ mm and an inner radius of $R_{\text{in}} = 45$ mm. Boundary conditions are imposed by restraining the upper and lower regions of the domain, while a rotational displacement loading of $u = 1$ mm is applied to the midplane. In this context, our principal aim is the enhancement of structural stiffness by optimizing the material layout and the distribution of 3D building blocks.

To work towards our optimization objective, we employ an unstructured grid to discretize the design domain. This grid consists of 190,080 first-order hexahedral Lagrange elements, each with a size of $h = 1$ mm. The remaining topology optimization parameters

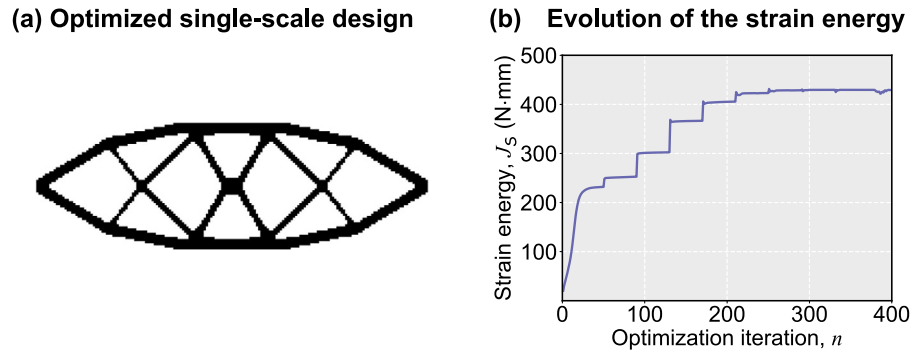

Fig. 9. Single-scale structure with maximized overall structural stiffness. (a) Optimized single-scale structure. (b) The evolution of the strain energy.

are as follows. The allowable volume fraction is $\overline{V} = 0.15$. The range of the frequency field is [0.15, 0.85] ($\varepsilon = 0.15$). The filter radii of the density and frequency fields are $R_\rho = R_\xi = 3.5$ mm. The parameters used in the interpolated elasticity in Eq. (8) are $\varepsilon_\rho = 10^{-6}$ and $p_\rho = 3$. The sharpness parameter in the Heaviside projection (Eq. (6)) is initially set as $\beta = 1$ and then doubled every 40 iterations until $\beta = 256$. Finally, the maximum optimization iteration is $N^{max} = 400$.

In this example, we harness the computational power of parallel computing across 4 threads to expedite the topology optimization process, culminating in a total runtime of 28.9 h. Fig. 10(b) presents the optimization histories of the objective function (total strain energy), J_S, and the actual material volume fraction, V. Both two values exhibit convergence tendencies after approximately 300 iterations, with the actual material volume fraction (V) reaching its predefined upper threshold of $\overline{V} = 0.15$. Subsequently, based on the optimized density and frequency fields, we employ the extended virtual growth algorithm to generate the optimized irregular multiscale structure. Fig. 10(c) demonstrates the digital rendering of such a virtual growth process. We observe the gradual "growth" of the structure, ultimately resulting in its complete formation after a total of 11,867 iterations. Additionally, Fig. 10(c) displays two sectional views to better visualize the generated structure, which exhibits an organic material layout and tailored distribution of building blocks.

To substantiate the manufacturability of this intricately optimized structure, we present a 3D printed physical model, offering various perspectives in Fig. 10(d). Once again, the physical model faithfully reproduces the free-form material layout and tailored distribution of building blocks, effectively preserving the geometrical intricacies observed in the digital rendering in Fig. 10(c). In the zoomed-in view on the right-hand side of Fig. 10(d), a meticulous examination reveals the seamless integration of all types of building blocks in constructing the microstructures within the generated design. The 3D printing details are as follows. To ensure the successful production and detailed visualization of this structure, we enlarge the design by a factor of two in all dimensions and then divide it into four distinct parts. Subsequently, each segment is crafted using the Elegoo Saturn 2 printer, accompanied by Elegoo water-washable photopolymer resins, chosen for their printing precision and ease of printing. Following the printing process, we assemble the individual parts using Loctite, a high-quality superglue from Henkel Corporation, guaranteeing structural integrity. Lastly, to enhance the visual impact, we apply a layer of metallic paint, specifically Krylon Fusion All-In-One by Krylon Products Group, to the assembled structure. This comprehensive workflow ensures that the final result not only captures the geometrical features of the digital design but also presents it in an aesthetically pleasing and tangible form.

5. Conclusions and discussion

In this study, we present a multiscale topology optimization approach centered around a virtual growth scheme, designed for optimizing non-periodic multiscale structures with tunable responses, while naturally ensuring manufacturability. The proposed methodology comprises four interconnected components: the generation of a discrete material database, the training of a neural network model, topology optimization, and the generalized virtual growth algorithm. These components collaborate harmoniously to optimize material distribution through a density variable, ρ, and building block distribution through frequency hints, ξ_1–$\xi_{M^{\text{block}}}$. Subsequently, these optimized variables guide the virtual growth process, resulting in structures with customizable responses. We substantiate the capabilities of this methodology through four representative examples. Specifically, we demonstrate the programming of three distinct response types, i.e., node-wise displacement, element-wise strain energy density, and global stiffness, in both 2D and 3D. Our results reveal a good alignment between the actual and target responses, and we verify the reproducibility of this alignment through two convergence studies. These convergence studies also demonstrate that the optimized multiscale structures remain independent of the number of building blocks in each macro-element. Furthermore, the designs are also unaffected by the macro-element size, thanks to the incorporation of linear filters in (5). To demonstrate the natural connections among building blocks and the manufacturability of the generated architectures with complex features, we 3D printed prototypes in 2D and 3D.

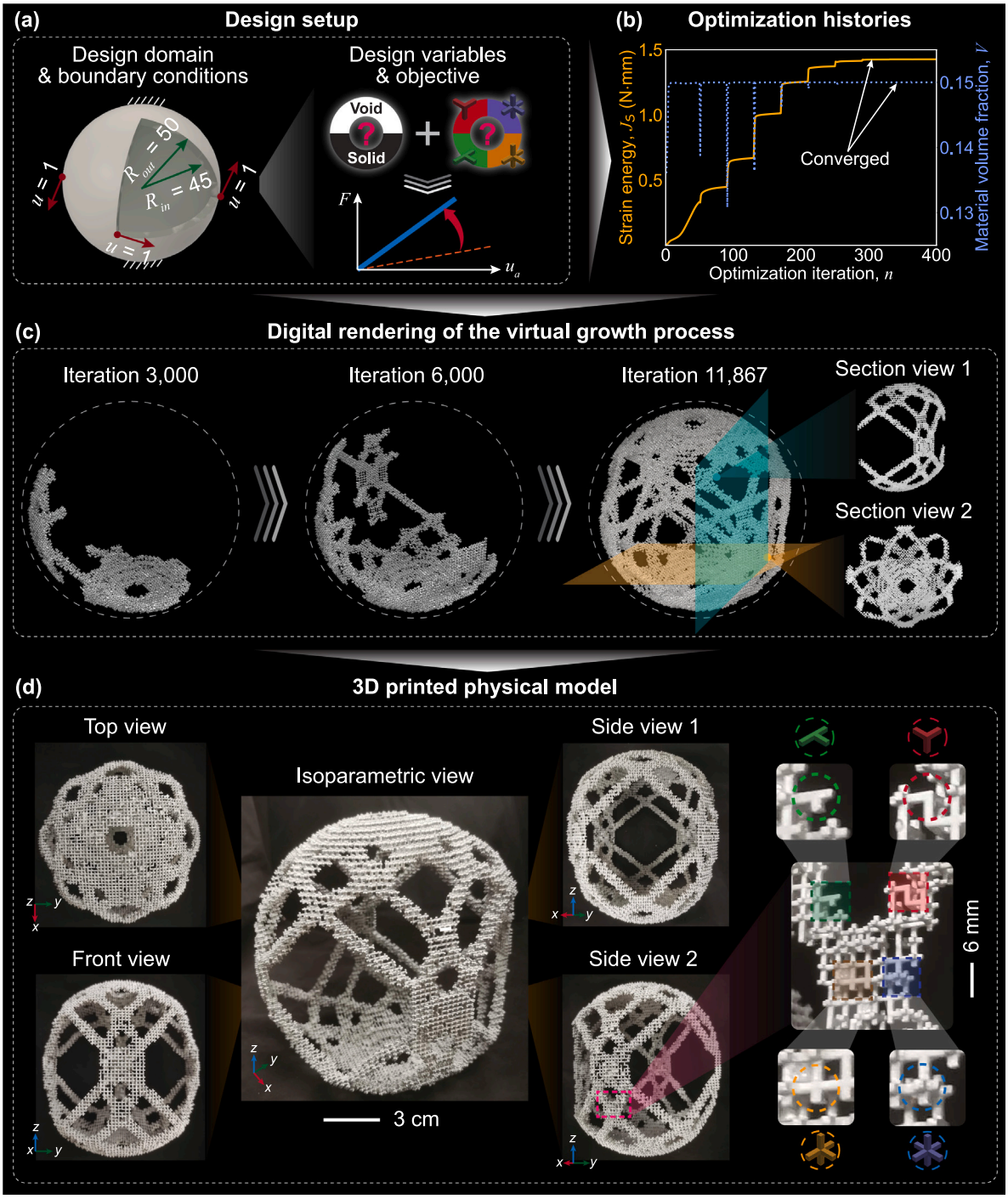

Fig. 10. 3D rotational spherical shell with maximized stiffness. (a) Design setup including the design domain, boundary conditions, design variables, and the optimization objective. (b) Optimization histories of the objective function (total strain energy), J_S, and the actual material volume fraction, V. (c) Digital rendering of the virtual growth process. Two section views correspond to the final iteration of the generated structure. (d) 3D printed physical model of the generated structure, showcased from various angles. On the right, the inset demonstrates the microstructure constructed from the prescribed building blocks.

We highlight that our approach does not impose strict limitations on either the quantity or the geometrical configurations of the building blocks, thereby enabling the creation of intricate aperiodic multiscale structures. Furthermore, owing to the extended virtual growth algorithm, the generated multiscale structures exhibit stochastic, self-sustaining ensembles composed of given building blocks. These building blocks are chosen to ensure a manufacturable minimum feature size, avoiding any delicate or protruding elements. This inherent self-supporting property, combined with the established minimum feature size, greatly enhances the feasibility of manufacturing these optimized structures via 3D printing and demonstrates the practical utility of our approach. Our proposed methodology holds promise for a wide array of applications, including flexible electronics and vibration-dampening systems.

Despite the efficacy of the proposed methodology for designing irregular multiscale structures, it is essential to emphasize that this framework can still be further improved. For instance, in the current study, we treat the unique rotated variations of building blocks in Fig. 2 as equivalent. While this approach couples the frequencies of the rotated variations with the frequency of the building block, reducing the number of design variables, it may result in a larger standard deviation of the macroscale structural performance. As part of future work, we believe it is meaningful to directly treat the frequency of each unique rotated variation as a separate design variable. This adjustment may further reduce the standard deviation of the macroscale structural performance, despite the convergence studies in Figs. 6(k) and 8(d).

Furthermore, the structural responses of the optimized designs are stochastic, and the current study primarily focuses on controlling the average responses. In doing so, we assume that the probability distribution of the homogenized material properties possesses unimodality and zero skewness, as indicated in Fig. 3 in [71]. Despite the convergence studies in Figs. 6(k) and 8(d), it would be meaningful to theoretically prove this assumption. Additionally, to capture the stochastic nature of the design problem, it would be interesting to investigate how the geometries of the building blocks and adjacency rules affect the probability distribution of material properties, and how the stochasticity of material properties further influences the randomness of structural responses.

It is also important to acknowledge that the building blocks utilized in the current study feature fixed solid material volume fractions and unchanged geometries. Furthermore, the generated unit cells are oriented solely along the horizontal and vertical directions, without allowance for rotation and elongation. These constraints on building blocks and unit cells may result in suboptimal multiscale designs for stiffness maximization. Looking ahead, we aspire to extend the proposed framework to incorporate freely rotated and elongated unit cells together with building blocks allowing variations in material fractions and geometries. This extension represents an ongoing area of exploration within the field of multiscale structures and metamaterials.

CRediT authorship contribution statement

Yingqi Jia: Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing, Software. **Ke Liu:** Conceptualization, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Xiaojia Shelly Zhang:** Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

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.

Data availability

Data will be made available on request.

Acknowledgments

Authors X.S.Z. and Y.J. acknowledge the support from U.S. National Science Foundation (NSF) EAGER Award CMMI-2127134, NSF CAREER Award CMMI-2047692, NSF Award CMMI-2245251, and Air Force Office of Scientific Research (AFOSR YIP) FA9550-23-1-0297. Author K.L. acknowledges the support from the National Natural Science Foundation of China through grant 12372159. The information provided in this paper is the sole opinion of the authors and does not necessarily reflect the view of the sponsoring agencies.

Appendix A. Normalization projection with non-trivial bounds

As previously emphasized, physical frequency fields need to satisfy the probability axiom, specifically the equality constraint $\sum_{i=1}^{M^{\text{block}}} \overline{\xi}_i = 1$. Furthermore, to facilitate viable solutions during the virtual growth process and to explore the influence of frequency variables on the performance of resulting generated structures such as those illustrated in Case 2 of Fig. 8(b), it is imperative to impose non-trivial lower and upper bounds, denoted as $[p, q] \subseteq [0, 1]$, on these physical variables. The equality constraint with trivial bounds, $[0, 1]$, is commonly required in topology optimization involving multi-materials [16,28], multi-fiber orientations [29,89], and multi-magnetic orientations [30], where the optimized design variables are required to be discrete (0–1 solution). In this work,

we permit the frequency variables to assume non-discrete values ($\bar{\xi}_1$–$\bar{\xi}_{M^{\text{block}}} \in (0,1)$), taking into account their physical significance within the context of frequency distribution.

Efficiently enforcing the equality constraint often involves the use of two commonly employed methods — (modified) hypercube-to-simplex projection (HSP) [90] and bi-value coding parameterization (BCP) [91]. When applying the modified HSP approach, it is necessary to define $M^{\text{block}}-1$ frequency variables and then use Eq. (5) to derive $M^{\text{block}}-1$ filtered variables. Subsequently, these filtered variables are projected onto $M^{\text{block}}-1$ physical variables, $\bar{\xi}_i \in [0,1]$ for $i=1,2,\dots,M^{\text{block}}-1$. Next, we directly define the last physical variable as $\bar{\xi}_{M^{\text{block}}} = 1-\sum_{i=1}^{M^{\text{block}}-1}\bar{\xi}_i$, and the equality constraint is automatically satisfied. However, the modified HSP fails to uniformly assign the non-trivial bounds to these physical variables. For example, let $[p,q] \subseteq [0,1]$ be the bounds of the first $M^{\text{block}}-1$ physical frequency variables. We then derive the bounds of the last physical frequency variable as $[\max\{1-(M^{\text{block}}-1)q,0\}, 1-(M^{\text{block}}-1)p]$, which is unequal to $[p,q]$.

Alternatively, we can use BCP to enforce the equality constraint. In this approach, only $\log_2 M^{\text{block}}$ frequency variables need to be defined, with $\xi_i \in [-1,1]$ for $i=1,2,\dots,\log_2 M^{\text{block}}$. For simplicity, we deactivate the linear filter in Eq. (5) and directly apply the BCP projection. Consequently, this method results in M^{block} physical variables, all adhering to the same non-trivial bounds $[p,q] \subseteq [0,1]$, while simultaneously ensuring compliance with the equality constraint. However, BCP restricts the number of building block types, M^{block}, to be the power of 2. In addition, the physical variables are closely coupled and thus cannot be interchangeable because fewer frequency variables are projected onto a larger number of physical ones.

In light of the aforementioned constraints and to overcome the limitations of both HSP and BCP methods, we propose a normalization scheme as delineated in Eq. (7) (see similar projection schemes in [81,92]). This projection guarantees that the physical variables, $\bar{\xi}_1,\bar{\xi}_2,\dots,\bar{\xi}_{M^{\text{block}}}$, adhere to the equality constraint, $\sum_{i=1}^{M^{\text{block}}}\bar{\xi}_i = 1$. Moreover, these physical variables share the same non-trivial bounds as follows. Let $[\varepsilon, 1-\varepsilon] \subseteq [0,1]$ be the bounds of frequency variables, $\xi_1,\xi_2,\dots,\xi_{M^{\text{block}}}$, with $\varepsilon \in [0,0.5)$ of choice. We deactivate the linear filter in Eq. (5) for simplicity and derive the bounds for the physical variables as $[\varepsilon/[M^{\text{block}}-1+(2-M^{\text{block}})\varepsilon], (1-\varepsilon)/[1+(M^{\text{block}}-2)\varepsilon]] \subseteq [0,1]$ by noticing

$$\bar{\xi}_i = \frac{\xi_i}{\xi_i + \sum_{j=1,j\neq i}^{M^{\text{block}}}\xi_j} \begin{cases} \geq \dfrac{\varepsilon}{\varepsilon + \sum_{j=1,j\neq i}^{M^{\text{block}}}\xi_j} \geq \dfrac{\varepsilon}{\varepsilon+(M^{\text{block}}-1)(1-\varepsilon)} = \dfrac{\varepsilon}{M^{\text{block}}-1+(2-M^{\text{block}})\varepsilon} \\ \leq \dfrac{1-\varepsilon}{1-\varepsilon+\sum_{j=1,j\neq i}^{M^{\text{block}}}\xi_j} \leq \dfrac{1-\varepsilon}{1-\varepsilon+(M^{\text{block}}-1)\varepsilon} = \dfrac{1-\varepsilon}{1+(M^{\text{block}}-2)\varepsilon} \end{cases}.$$

To accomplish the sensitivity analysis, we provide the derivative of the physical variable, $\bar{\xi}_i$, with respect to the frequency variable, ξ_j, as

$$\frac{\mathrm{d}\bar{\xi}_i}{\mathrm{d}\xi_j} = \frac{\delta_{ij}\sum_{k=1}^{M^{\text{block}}}\xi_k - \xi_i}{(\sum_{k=1}^{M^{\text{block}}}\xi_k)^2} = \begin{cases} \dfrac{1}{\sum_{k=1}^{M^{\text{block}}}\xi_k} - \dfrac{\xi_i}{(\sum_{k=1}^{M^{\text{block}}}\xi_k)^2}, & \text{for } i=j \\ -\dfrac{\xi_i}{(\sum_{k=1}^{M^{\text{block}}}\xi_k)^2}, & \text{for } i\neq j \end{cases}, \tag{A.1}$$

where δ_{ij} is the Kronecker delta with $\delta_{ij}=1$ if $i=j$ and $\delta_{ij}=0$ otherwise.

Appendix B. Sensitivity analysis and verification

As demonstrated in Algorithm 4, sensitivity analysis constitutes a pivotal step in the execution of the proposed topology optimization method. This approach incorporates a neural network, as depicted in Fig. 4(a), necessitating the inclusion of neural network backpropagation within the framework of the sensitivity analysis. In this section, we elucidate the efficient handling of this intricate sensitivity analysis process.

Fig. B.11 illustrates the computational graph encompassing both forward pass and backward propagation throughout the iterative process of topology optimization. This computational process is composed of two key elements: design space parameterization and FEA. The forward pass of these components assembles the global residual vector, $\mathbf{R}$, and calculates the function value, J. The primary objective at hand is to assess the sensitivity (gradient) of a given objective or constraint function (J) concerning the design variables (ρ and ξ_1–$\xi_{M^{\text{block}}}$), $\mathrm{d}J/\mathrm{d}\rho$ and $\mathrm{d}J/\mathrm{d}\xi_1$–$\mathrm{d}J/\mathrm{d}\xi_{M^{\text{block}}}$. To achieve this, we can follow backpropagation in Fig. B.11 and decompose the sensitivities utilizing chain rules as

$$\frac{\mathrm{d}J}{\mathrm{d}\rho} = \frac{\mathrm{d}J}{\mathrm{d}\bar{\rho}}\cdot\frac{\mathrm{d}\bar{\rho}}{\mathrm{d}\widetilde{\rho}}\cdot\frac{\mathrm{d}\widetilde{\rho}}{\mathrm{d}\rho} \quad \text{and} \quad \frac{\mathrm{d}J}{\mathrm{d}\xi_j} = \sum_{i=1}^{N}\sum_{k=1}^{M^{\text{block}}}\frac{\mathrm{d}J}{\mathrm{d}C_i}\cdot\frac{\mathrm{d}C_i}{\mathrm{d}\bar{\xi}_k}\cdot\frac{\mathrm{d}\bar{\xi}_k}{\mathrm{d}\widetilde{\xi}_j}\cdot\frac{\mathrm{d}\widetilde{\xi}_j}{\mathrm{d}\xi_j}, \quad j=1,2,\dots,M^{\text{block}},$$

and we only need to compute these decomposed terms. Specifically, we compute $\mathrm{d}\bar{\rho}/\mathrm{d}\widetilde{\rho}$, $\mathrm{d}\widetilde{\rho}/\mathrm{d}\rho$, and $\mathrm{d}\widetilde{\xi}_1/\mathrm{d}\xi_1$–$\mathrm{d}\widetilde{\xi}_{M^{\text{block}}}/\mathrm{d}\xi_{M^{\text{block}}}$ by following conventions [93,94]; compute $\mathrm{d}\bar{\xi}_k/\mathrm{d}\widetilde{\xi}_j$ for $k=1,2,\dots,M^{\text{block}}$ and $j=1,2,\dots,M^{\text{block}}$ following Eq. (A.1). We now focus on the computation of $\mathrm{d}J/\mathrm{d}\bar{\rho}$, $\mathrm{d}J/\mathrm{d}C_1$–$\mathrm{d}J/\mathrm{d}C_N$, and $\mathrm{d}C_i/\mathrm{d}\bar{\xi}_k$ for $i=1,2,\dots,N$ and $k=1,2,\dots,M^{\text{block}}$ as follows.

First, we evaluate $\mathrm{d}J/\mathrm{d}\bar{\rho}$ with the adjoint method [21] to reduce the computational cost. Specifically, we rewrite any given function, J, in its Lagrangian form as

$$\widehat{J} = J + \lambda\cdot\mathbf{R}$$

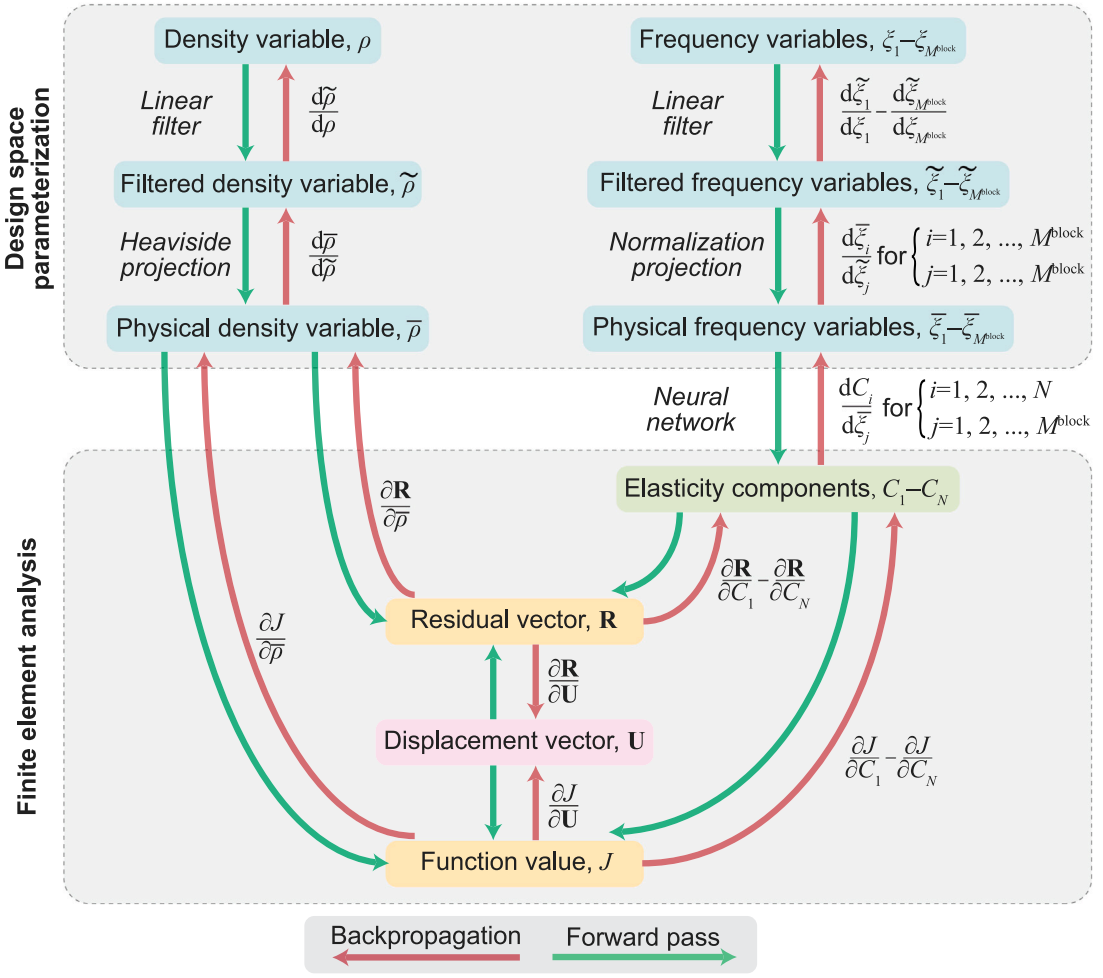

Fig. B.11. The computational graph for sensitivity analysis. This comprehensive computational framework encompasses design space parameterization and FEA, each featuring forward pass and backward propagation.

where λ is an adjoint vector to be determined and $\lambda_i = 0$ if DOF i is fixed in FEA. Taking the derivative of J with respect to the physical density variable, $\overline{\rho}$, we obtain

$$\frac{d\hat{J}}{d\overline{\rho}} = \frac{\partial J}{\partial \overline{\rho}} + \frac{\partial J}{\partial \mathbf{U}} \cdot \frac{\partial \mathbf{U}}{\partial \overline{\rho}} + \lambda \cdot \left(\frac{\partial \mathbf{R}}{\partial \overline{\rho}} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \cdot \frac{\partial \mathbf{U}}{\partial \overline{\rho}} \right).$$

Regrouping terms yields

$$\frac{d\hat{J}}{d\overline{\rho}} = \frac{\partial J}{\partial \overline{\rho}} + \lambda \cdot \frac{\partial \mathbf{R}}{\partial \overline{\rho}} + \left[\frac{\partial J}{\partial \mathbf{U}} + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \right)^{\mathsf{T}} \cdot \lambda \right] \cdot \frac{\partial \mathbf{U}}{\partial \overline{\rho}}.$$

We eliminate the computationally expensive term, $\partial \mathbf{U}/\partial \overline{\rho}$, by enforcing

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \right)^{\mathsf{T}} \cdot \lambda = -\frac{\partial J}{\partial \mathbf{U}}, \tag{B.1}$$

and we then have

$$\frac{dJ}{d\overline{\rho}} = \frac{d\hat{J}}{d\overline{\rho}} = \frac{\partial J}{\partial \overline{\rho}} + \lambda \cdot \frac{\partial \mathbf{R}}{\partial \overline{\rho}},$$

which can be expressed in a tensorial form as

$$\frac{dJ}{d\overline{\boldsymbol{\rho}}} = \frac{\partial J}{\partial \overline{\boldsymbol{\rho}}} + \left(\frac{\partial \mathbf{R}}{\partial \overline{\boldsymbol{\rho}}} \right)^{\mathsf{T}} \cdot \lambda \tag{B.2}$$

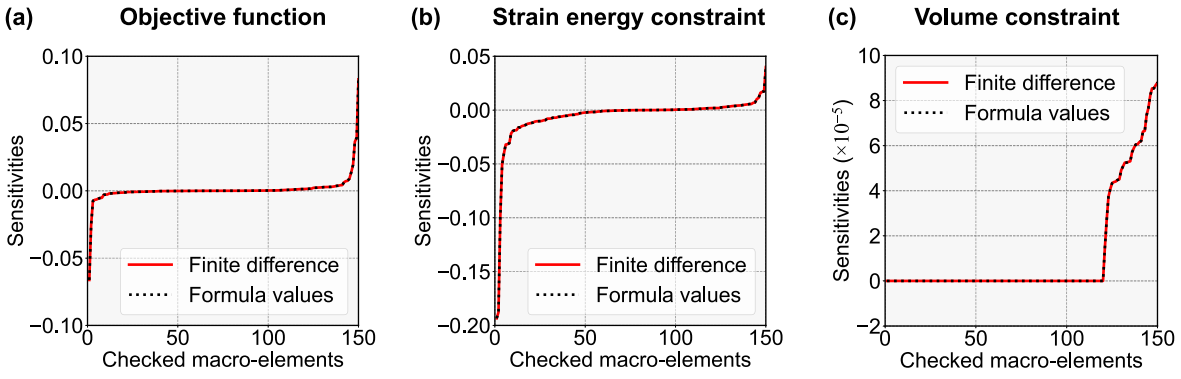

Fig. B.12. The comparisons between the sensitivities of the (a) objective function, (b) strain energy constraint, and (c) volume constraint of the optimized structure in Fig. 7(e) computed by the derived formula and forward finite difference. Sensitivity values are sorted to enhance visual clarity.

where $\overline{\boldsymbol{\rho}}$ is the vector form of the physical density variable defined as

$$\overline{\boldsymbol{\rho}} = \begin{bmatrix} \overline{\rho}_1 \\ \overline{\rho}_2 \\ \vdots \\ \overline{\rho}_{N^{\mathtt{elem}}} \end{bmatrix} = \begin{bmatrix} \overline{\rho}(\mathbf{X}_1) \\ \overline{\rho}(\mathbf{X}_2) \\ \vdots \\ \overline{\rho}(\mathbf{X}_{N^{\mathtt{elem}}}) \end{bmatrix}$$

where $N^{\mathtt{elem}}$ represents the number of macro-elements. The variables $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{N^{\mathtt{elem}}}$ represent the centers of finite elements $1, 2, \ldots, N^{\mathtt{elem}}$, respectively. Note that the adjoint vector, $\boldsymbol{\lambda}$, can be solved from the adjoint equation in (B.1). Therefore, Eq. (B.2) relies on four terms: $\partial J/\partial\overline{\boldsymbol{\rho}}$, $\partial J/\partial\mathbf{U}$, $\partial\mathbf{R}/\partial\overline{\boldsymbol{\rho}}$, and $\partial\mathbf{R}/\partial\mathbf{U}$, as also reflected in Fig. B.11. To efficiently compute these four terms, we leverage the automatic differentiation provided by the open-source software, FEniCSx [85]. In a similar fashion, we can compute $\mathrm{d}J/\mathrm{d}C_1$–$\mathrm{d}J/\mathrm{d}C_N$ in their tensorial forms as $\mathrm{d}J/\mathrm{d}\mathbf{C}_1$–$\mathrm{d}J/\mathrm{d}\mathbf{C}_N$, where $\mathbf{C}_1$–$\mathbf{C}_N$ are the vector forms of the elasticity components with their ith items representing the elasticity components of macro-element i in FEA.

Next, we evaluate $\mathrm{d}C_i/\mathrm{d}\overline{\xi}_j$ for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M^{\mathtt{block}}$. To achieve this, we utilize the automatic differentiation in another open-source software, PyTorch [77]. PyTorch forms a computational graph when constructing a neural network and traces the neural network's topology for automatic differentiation during backpropagation. Such automatic differentiation is sufficient to compute $\mathrm{d}C_i/\mathrm{d}\overline{\xi}_j$ for one macro-element. However, a grid for topology optimization can contain thousands to millions of macro-elements [19]. To perform such automatic differentiation in batches for multiple macro-elements, we further leverage transforms, `vmap` and `jacrev`, in the functorch package [95]. Finally, we can rewrite the complete sensitivities in the tensorial form as

$$\frac{\mathrm{d}J}{\mathrm{d}\rho} = \left(\frac{\mathrm{d}\widetilde{\rho}}{\mathrm{d}\rho}\right)^{\mathsf{T}} \cdot \left(\frac{\mathrm{d}\overline{\rho}}{\mathrm{d}\widetilde{\rho}}\right)^{\mathsf{T}} \cdot \frac{\mathrm{d}J}{\mathrm{d}\overline{\rho}} \quad \text{and} \quad \frac{\mathrm{d}J}{\mathrm{d}\xi_j} = \sum_{i=1}^{N} \sum_{k=1}^{M^{\mathtt{block}}} \left(\frac{\mathrm{d}\widetilde{\xi}_j}{\mathrm{d}\xi_j}\right)^{\mathsf{T}} \cdot \left(\frac{\mathrm{d}\overline{\xi}_k}{\mathrm{d}\widetilde{\xi}_j}\right)^{\mathsf{T}} \cdot \left(\frac{\mathrm{d}\mathbf{C}_i}{\mathrm{d}\overline{\xi}_k}\right)^{\mathsf{T}} \cdot \frac{\mathrm{d}J}{\mathrm{d}\mathbf{C}_i}, \quad j = 1, 2, \ldots, M^{\mathtt{block}}, \tag{B.3}$$

where $\overline{\boldsymbol{\xi}}_j$ is the vector form of the physical frequency variable j defined as

$$\overline{\boldsymbol{\xi}}_j = \begin{bmatrix} \overline{\xi}_{j,1} \\ \overline{\xi}_{j,2} \\ \vdots \\ \overline{\xi}_{j,N^{\mathtt{elem}}} \end{bmatrix} = \begin{bmatrix} \overline{\xi}_j(\mathbf{X}_1) \\ \overline{\xi}_j(\mathbf{X}_2) \\ \vdots \\ \overline{\xi}_j(\mathbf{X}_{N^{\mathtt{elem}}}) \end{bmatrix}$$

with its eth component ($\overline{\xi}_{j,e}$) representing the physical frequency variable j of macro-element e in FEA.

To verify the aforementioned sensitivity analysis, we use the optimized design in Fig. 7(e) as an example for conducting the sensitivity verification. Specifically, we undertake a random sampling of 30 macro-elements for all design variables, ρ and ξ_1–ξ_4. We then use Eq. (B.3) to compute the formula-based sensitivity values during the final optimization iteration. In parallel, we execute a forward finite difference approach for the same set of macro-elements to recompute the sensitivities. Fig. B.12 shows the comparisons between the formula-based and the finite difference values of the objective function, strain energy constraint, and volume constraint. The relative 2-norm errors of the three sets of sensitivities are 4.896×10^{-6}, 1.697×10^{-6}, and 1.190×10^{-7}, respectively, underscoring the precision and accuracy of the sensitivity analysis previously outlined.

Appendix C. Supplementary analysis of the mechanical displacement cloaking example

This section presents supplementary results (Fig. C.13) for the mechanical displacement cloaking example discussed in Section 4.1. In contrast to the structure outlined in Section 4.1, Fig. C.13(d) demonstrates a modification where a disk-shaped void

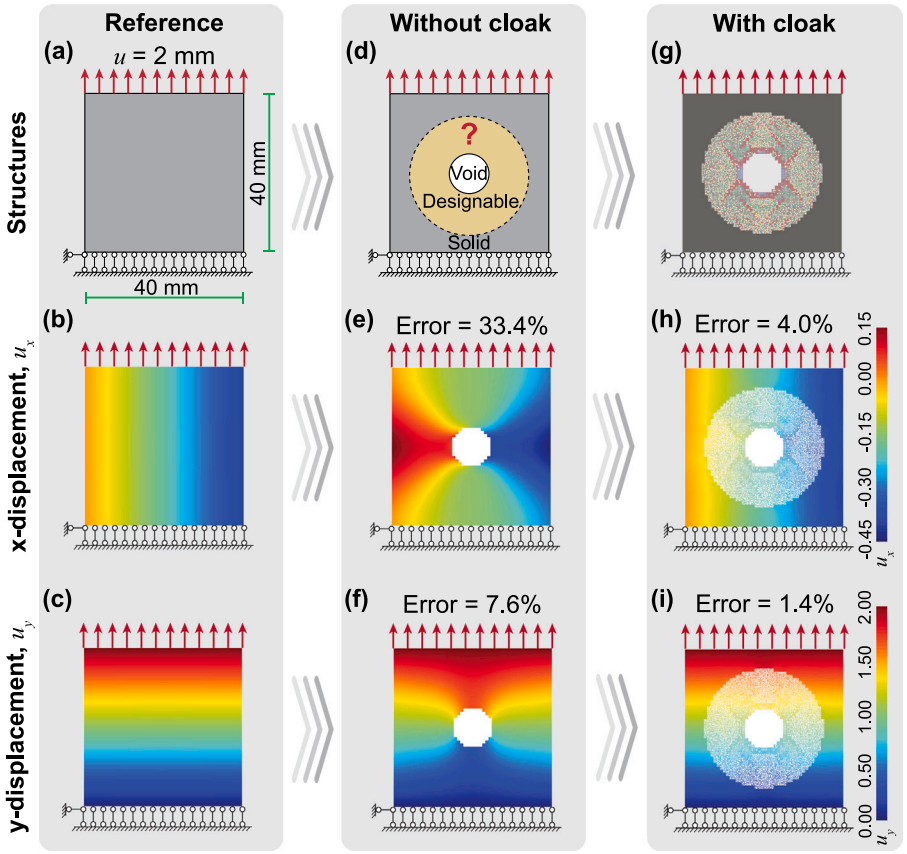

Fig. C.13. Supplementary results of the mechanical displacement cloaking example in Section 4.1. Herein, we dig a disk-shaped void in the structure with a ring-shaped designable region, and the mechanical displacement cloaking effect is still achievable, as quantified by the relative errors, $e_x = 4.0\%$ and $e_y = 1.4\%$.

has been introduced into the structure, and a ring-shaped designable region has been defined. Comparing this configuration to the reference structure (Fig. C.13(b)–(c)), we observe that the introduction of the void region results in significant alterations in the displacement fields in both directions (Fig. C.13(e)–(f)). This shift is quantified by the relative errors, $e_x = 33.4\%$ and $e_y = 7.6\%$. To recover the displacement fields, we utilize the proposed methodology to optimize an irregular multiscale structure (Fig. C.13(g)). Subsequently, we analyze its displacement fields (Fig. C.13(h)–(i)) using standard FEA. The results indicate a satisfactory mechanical cloaking effect, as evidenced by the relative errors, $e_x = 4.0\%$ and $e_y = 1.4\%$.

Appendix D. Proposed two-grid projection scheme

The virtual growth algorithm in [71] constrains the building blocks and the cells containing them to square shapes in 2D and cubic shapes in 3D. However, in scenarios involving irregular design domains as depicted in Figs. 7 and 10, it is often preferable to employ an unstructured grid for reducing the zigzagged boundaries of the optimized designs. To facilitate the application of the virtual growth algorithm to these unstructured grids, we introduce a two-grid projection scheme as illustrated in Fig. D.14. In this scheme, we project the optimized physical density and frequency fields from an unstructured background grid, used for topology optimization, onto a structured foreground grid, employed for virtual growth (Fig. D.14(a)–(b)). Subsequently, we implement the virtual growth algorithm on the foreground mesh using the projected fields to generate optimized irregular architectural structures (Fig. D.14(b)–(c)). Optionally, we can further refine the generated structures to align with the original boundaries. Consequently, these generated structures conform to the optimized physical density and frequency fields defined on the unstructured background grid at the macroscale, while the building blocks adhere to a structured foreground grid at the microscale.

In this proposed scheme, a pivotal step involves determining the projected fields for the foreground grid (Fig. D.14(a)–(b)). To accomplish this goal, we iterate over all foreground macro-elements, allowing each foreground macro-element to inherit the optimized fields from the background grid. Specifically, we require that foreground macro-element i

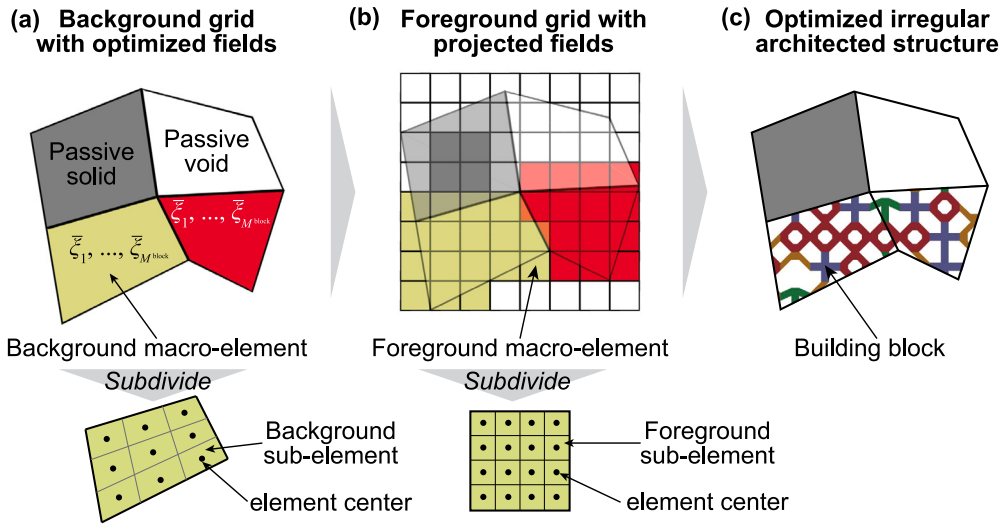

Fig. D.14. Illustration of the proposed two-grid projection scheme. (a) An unstructured background grid used for topology optimization. This grid consists of one passive solid, one passive void, and two designable macro-elements with optimized fields. Each background macro-element can be further divided into multiple sub-elements. (b) A foreground grid with projected fields. Each foreground macro-element can also be further divided into multiple sub-elements. (c) Optimized irregular multiscale structure generated based on the foreground grid.

- is a passive solid if this macro-element is completely within passive solid background macro-elements,
- is a passive void if this macro-element does not intersect with any designable background macro-elements and this macro-element is not passive solid,
- and otherwise inherits the optimized fields from background macro-element q with q computed as

$$q = \underset{j}{\operatorname{argmax}} \frac{|\Omega_i^{\mathtt{fore}} \cap \Omega_j^{\mathtt{back}}|}{|\Omega_i^{\mathtt{fore}}|}$$

where $\Omega_i^{\mathtt{fore}}$ and $\Omega_j^{\mathtt{back}}$ are the spaces occupied by foreground macro-element i and background macro-element j, respectively. These rules maintain the design's integrity after the cutting (Fig. D.14(c)).

The rules outlined above involve the computation of intersection areas between arbitrary polygons in 2D and intersection volumes between arbitrary polyhedra in 3D. To handle these intricate calculations, we initiate the process by identifying all solid and void background macro-elements, $\mathcal{S}$ and $\mathcal{V}$, respectively. Subsequently, each background macro-element is divided into $N_{\mathtt{sub}}^{\mathtt{back}}$ sub-elements, and each foreground macro-element undergoes subdivision into $N_{\mathtt{sub}}^{\mathtt{fore}}$ sub-elements. Furthermore, we determine the center coordinates of these sub-elements using the PyVista package [73]. Moving forward, for a given foreground macro-element, we iterate through its sub-elements and utilize the GeoPandas [96] and Shapely [97] packages to ascertain whether the center of a specific sub-element lies within the background grid. If it does, we proceed to identify the closest background sub-element and subsequently the associated background macro-element. This information is stored in a vector $\mathbf{R}$ with $R_i = j$ signifying that foreground sub-element i is within background macro-element j, while $R_i = 0$ indicates that foreground sub-element i is not within the background grid. For later use, we convert the vector $\mathbf{R}$ to a set $\mathcal{U}$ by retaining its unique components. Finally, we can mathematically reformulate the above rules as follows: foreground macro-element i

- is a passive solid if $\mathcal{U} \subseteq \mathcal{S}$,
- is a passive void if $\mathcal{U} \nsubseteq \mathcal{S}$ and $\mathcal{U} \subseteq \mathcal{S} \cup \mathcal{V} \cup \{0\}$,
- and otherwise inherits the optimized fields from background macro-element q with q defined as

$$q = \text{Mode}(\mathbf{R}) \quad \text{and} \quad q \notin \mathcal{S} \cup \mathcal{V} \cup \{0\}.$$

Algorithm 5 shows the detailed procedure of the proposed two-grid projection scheme. Notably, this proposed scheme imposes no restrictions on the geometry of either the background or foreground grid, enabling the projection of optimized fields from a structured/unstructured background grid onto another structured/unstructured foreground grid. To implement this algorithm, we have utilized an in-house Python code [74] augmented by the previously mentioned PyVista [73], GeoPandas [96], and Shapely [97] extensions. These extensions significantly enhance computational efficiency, replacing the need for certain while-loops in Algorithm 5.

Algorithm 5: Two-grid scheme for projecting the optimized fields

Inputs: the number of foreground macro-elements, $N^{\mathtt{fore}}$; the number of sub-elements in each background macro-element, $N_{\mathtt{sub}}^{\mathtt{back}}$; the number of sub-elements in each foreground macro-element, $N_{\mathtt{sub}}^{\mathtt{fore}}$;
Find all solid and void background macro-elements, $\mathcal{S}$ and $\mathcal{V}$, respectively;
Subdivide each background macro-element into $N_{\mathtt{sub}}^{\mathtt{back}}$ sub-elements;
Subdivide each foreground macro-element into $N_{\mathtt{sub}}^{\mathtt{fore}}$ sub-elements;
Initialize the foreground macro-element count, $n^{\mathtt{fore}} \leftarrow 1$;
while $n^{\mathtt{fore}} \leq N^{\mathtt{fore}}$ **do**
  Initialize a relation vector, $\mathbf{R} \leftarrow \mathbf{0} \in \mathbb{R}^{N_{\mathtt{sub}}^{\mathtt{fore}}}$. Here $R_i = j$ represents that sub-element i of the current foreground grid ($n^{\mathtt{fore}}$) is within background macro-element j; $R_i = 0$ if sub-element i does not belong to any background macro-element;
  Initialize the foreground sub-element count, $n_{\mathtt{sub}}^{\mathtt{fore}} \leftarrow 1$;
  while $n_{\mathtt{sub}}^{\mathtt{fore}} \leq N_{\mathtt{sub}}^{\mathtt{fore}}$ **do**
    if foreground sub-element $n_{\mathtt{sub}}^{\mathtt{fore}}$ is within the background grid **then**
      Find the nearest background sub-element and then its associated background macro-element, k;
      $R_{n_{\mathtt{sub}}^{\mathtt{fore}}} \leftarrow k$;
    end
    $n_{\mathtt{sub}}^{\mathtt{fore}} \leftarrow n_{\mathtt{sub}}^{\mathtt{fore}} + 1$;
  end
  Let $\mathcal{U}$ denote unique components in $\mathbf{R}$;
  if $\mathcal{U} \subseteq \mathcal{S}$ **then**
    Foreground macro-element $n^{\mathtt{fore}}$ is solid;
  else if $\mathcal{U} \nsubseteq \mathcal{S}$ and $\mathcal{U} \subseteq \mathcal{S} \cup \mathcal{V} \cup \{0\}$ **then**
    Foreground macro-element $n^{\mathtt{fore}}$ is void;
  else
    Remove all components in $\mathbf{R}$ if they are also in $\mathcal{S} \cup \mathcal{V} \cup \{0\}$;
    Foreground macro-element $n^{\mathtt{fore}}$ inherits the optimized density and frequency fields of background macro-element $q = \mathrm{Mode}(\mathbf{R})$;
  end
  $n^{\mathtt{fore}} \leftarrow n^{\mathtt{fore}} + 1$;
end

References

[1] R.C. Rumpf, J.J. Pazos, J.L. Digaum, S.M. Kuebler, Spatially variant periodic structures in electromagnetics, Phil. Trans. R. Soc. A 373 (2049) (2015) 20140359, http://dx.doi.org/10.1098/rsta.2014.0359.

[2] M. Vinyas, On frequency response of porous functionally graded magneto-electro-elastic circular and annular plates with different electro-magnetic conditions using HSDT, Compos. Struct. 240 (2020) 112044, http://dx.doi.org/10.1016/j.compstruct.2020.112044.

[3] S. Arabnejad, B. Johnston, M. Tanzer, D. Pasini, Fully porous 3D printed titanium femoral stem to reduce stress-shielding following total hip arthroplasty, J. Orthop. Res. 35 (8) (2017) 1774–1783, http://dx.doi.org/10.1002/jor.23445.

[4] G. Theocharidis, H. Yuk, H. Roh, L. Wang, I. Mezghani, J. Wu, A. Kafanas, M. Contreras, B. Sumpio, Z. Li, E. Wang, L. Chen, C.F. Guo, N. Jayaswal, X.-L. Katopodi, N. Kalavros, C.S. Nabzdyk, I.S. Vlachos, A. Veves, X. Zhao, A strain-programmed patch for the healing of diabetic wounds, Nat. Biomed. Eng. 6 (10) (2022) 1118–1133, http://dx.doi.org/10.1038/s41551-022-00905-2.

[5] N.I. Zheludev, Y.S. Kivshar, From metamaterials to metadevices, Nature Mater. 11 (11) (2012) 917–924, http://dx.doi.org/10.1038/nmat3431.

[6] M. Skouras, B. Thomaszewski, S. Coros, B. Bickel, M. Gross, Computational design of actuated deformable characters, ACM Trans. Graph. 32 (4) (2013) 82:1–82:10, http://dx.doi.org/10.1145/2461912.2461979.

[7] A. Rafsanjani, K. Bertoldi, A.R. Studart, Programming soft robots with flexible mechanical metamaterials, Science Robotics 4 (29) (2019) eaav7874, http://dx.doi.org/10.1126/scirobotics.aav7874.

[8] S. Sepehri, H. Jafari, M. Mosavi Mashhadi, M.R. Hairi Yazdi, M.M. Seyyed Fakhrabadi, Tunable elastic wave propagation in planar functionally graded metamaterials, Acta Mech. 231 (8) (2020) 3363–3385, http://dx.doi.org/10.1007/s00707-020-02705-8.

[9] J. Bauer, J.A. Kraus, C. Crook, J.J. Rimoli, L. Valdevit, Tensegrity metamaterials: Toward failure-resistant engineering systems through delocalized deformation, Adv. Mater. 33 (10) (2021) 2005647, http://dx.doi.org/10.1002/adma.202005647.

[10] G.X. Gu, M. Takaffoli, M.J. Buehler, Hierarchically enhanced impact resistance of bioinspired composites, Adv. Mater. 29 (28) (2017) 1700060, http://dx.doi.org/10.1002/adma.201700060.

[11] J. Mueller, J.R. Raney, K. Shea, J.A. Lewis, Architected lattices with high Stiffness and toughness via multicore–shell 3D printing, Adv. Mater. 30 (12) (2018) 1705001, http://dx.doi.org/10.1002/adma.201705001.

[12] R. Ma, L. Wu, D. Pasini, Contact-driven snapping in thermally actuated metamaterials for fully reversible functionality, Adv. Funct. Mater. 33 (16) (2023) 2213371, http://dx.doi.org/10.1002/adfm.202213371.

[13] A. Clausen, F. Wang, J.S. Jensen, O. Sigmund, J.A. Lewis, Topology optimized architectures with programmable Poisson's ratio over large deformations, Adv. Mater. 27 (37) (2015) 5523–5527, http://dx.doi.org/10.1002/adma.201502485.

[14] W. Li, F. Wang, O. Sigmund, X.S. Zhang, Digital synthesis of free-form multimaterial structures for realization of arbitrary programmed mechanical responses, Proc. Natl. Acad. Sci. 119 (10) (2022) e2120563119, http://dx.doi.org/10.1073/pnas.2120563119.

[15] 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, http://dx.doi.org/10.1073/pnas.2122185119.
[16] W. Li, Y. Jia, F. Wang, O. Sigmund, X.S. Zhang, Programming and physical realization of extreme three-dimensional responses of metastructures under large deformations, Internat. J. Engrg. Sci. 191 (2023) 103881, http://dx.doi.org/10.1016/j.ijengsci.2023.103881.
[17] G. Felsch, N. Ghavidelnia, D. Schwarz, V. Slesarenko, Controlling auxeticity in curved-beam metamaterials via a deep generative model, Comput. Methods Appl. Mech. Engrg. 410 (2023) 116032, http://dx.doi.org/10.1016/j.cma.2023.116032.
[18] O. Sigmund, K. Maute, Topology optimization approaches: A comparative review, Struct. Multidiscip. Optim. 48 (6) (2013) 1031–1055, http://dx.doi.org/10.1007/s00158-013-0978-6.
[19] C. Wang, Z. Zhao, M. Zhou, O. Sigmund, X.S. Zhang, A comprehensive review of educational articles on structural and multidisciplinary optimization, Struct. Multidiscip. Optim. 64 (2021) 2827–2880, http://dx.doi.org/10.1007/s00158-021-03050-7.
[20] M.P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, Comput. Methods Appl. Mech. Engrg. 71 (2) (1988) 197–224, http://dx.doi.org/10.1016/0045-7825(88)90086-2.
[21] M.P. Bendsoe, O. Sigmund, Topology Optimization: Theory, Methods, and Applications, Springer Science & Business Media, 2003.
[22] P. Liu, Y. Luo, Z. Kang, Multi-material topology optimization considering interface behavior via XFEM and level set method, Comput. Methods Appl. Mech. Engrg. 308 (2016) 113–133, http://dx.doi.org/10.1016/j.cma.2016.05.016.
[23] X.S. Zhang, G.H. Paulino, A.S. Ramos, Multimaterial topology optimization with multiple volume constraints: Combining the ZPR update with a ground-structure algorithm to select a single material per overlapping set, Internat. J. Numer. Methods Engrg. 114 (10) (2018) 1053–1073, http://dx.doi.org/10.1002/nme.5736.
[24] Z. Zhao, X.S. Zhang, Design of graded porous bone-like structures via a multi-material topology optimization approach, Struct. Multidiscip. Optim. 64 (2) (2021) 677–698, http://dx.doi.org/10.1007/s00158-021-02870-x.
[25] X.S. Zhang, H. Chi, G.H. Paulino, Adaptive multi-material topology optimization with hyperelastic materials under large deformations: A virtual element approach, Comput. Methods Appl. Mech. Engrg. 370 (2020) 112976, http://dx.doi.org/10.1016/j.cma.2020.112976.
[26] J. Tian, M. Li, Z. Han, Y. Chen, X.D. Gu, Q.J. Ge, S. Chen, Conformal topology optimization of multi-material ferromagnetic soft active structures using an extended level set method, Comput. Methods Appl. Mech. Engrg. 389 (2022) 114394, http://dx.doi.org/10.1016/j.cma.2021.114394.
[27] F. Jiang, L. Chen, J. Wang, X. Miao, H. Chen, Topology optimization of multimaterial distribution based on isogeometric boundary element and piecewise constant level set method, Comput. Methods Appl. Mech. Engrg. 390 (2022) 114484, http://dx.doi.org/10.1016/j.cma.2021.114484.
[28] W. Li, F. Wang, O. Sigmund, X.S. Zhang, Design of composite structures with programmable elastic responses under finite deformations, J. Mech. Phys. Solids 151 (2021) 104356, http://dx.doi.org/10.1016/j.jmps.2021.104356.
[29] R.D. Kundu, X.S. Zhang, Stress-based topology optimization for fiber composites with improved stiffness and strength: Integrating anisotropic and isotropic materials, Compos. Struct. 320 (2023) 117041, http://dx.doi.org/10.1016/j.compstruct.2023.117041.
[30] C. Wang, Z. Zhao, X.S. Zhang, Inverse design of magneto-active metasurfaces and robots: Theory, computation, and experimental validation, Comput. Methods Appl. Mech. Engrg. 413 (2023) 116065, http://dx.doi.org/10.1016/j.cma.2023.116065.
[31] J. Wu, O. Sigmund, J.P. Groen, Topology optimization of multi-scale structures: A review, Struct. Multidiscip. Optim. 63 (3) (2021) 1455–1480, http://dx.doi.org/10.1007/s00158-021-02881-8.
[32] W. Zhang, S. Sun, Scale-related topology optimization of cellular materials and structures, Internat. J. Numer. Methods Engrg. 68 (9) (2006) 993–1011, http://dx.doi.org/10.1002/nme.1743.
[33] X. Huang, Y.M. Xie, Optimal design of periodic structures using evolutionary topology optimization, Struct. Multidiscip. Optim. 36 (6) (2008) 597–606, http://dx.doi.org/10.1007/s00158-007-0196-1.
[34] Y. Liu, Z. Li, P. Wei, W. Wang, Parameterized level-set based topology optimization method considering symmetry and pattern repetition constraints, Comput. Methods Appl. Mech. Engrg. 340 (2018) 1079–1101, http://dx.doi.org/10.1016/j.cma.2018.04.034.
[35] J. Wu, N. Aage, R. Westermann, O. Sigmund, Infill optimization for additive manufacturing—approaching bone-like porous structures, IEEE Trans. Vis. Comput. Graphics 24 (2) (2018) 1127–1140, http://dx.doi.org/10.1109/TVCG.2017.2655523.
[36] B. Hassani, E. Hinton, A review of homogenization and topology optimization I—homogenization theory for media with periodic structure, Comput. Struct. 69 (6) (1998) 707–717, http://dx.doi.org/10.1016/S0045-7949(98)00131-X.
[37] A. Vigliotti, D. Pasini, Stiffness and strength of tridimensional periodic lattices, Comput. Methods Appl. Mech. Engrg. 229–232 (2012) 27–43, http://dx.doi.org/10.1016/j.cma.2012.03.018.
[38] H. Rodrigues, J. Guedes, M. Bendsoe, Hierarchical optimization of material and structure, Struct. Multidiscip. Optim. 24 (1) (2002) 1–10, http://dx.doi.org/10.1007/s00158-002-0209-z.
[39] P.G. Coelho, P.R. Fernandes, J.M. Guedes, H.C. Rodrigues, A hierarchical model for concurrent material and topology optimisation of three-dimensional structures, Struct. Multidiscip. Optim. 35 (2) (2008) 107–115, http://dx.doi.org/10.1007/s00158-007-0141-3.
[40] L. Liu, J. Yan, G. Cheng, Optimum structure with homogeneous optimum truss-like material, Comput. Struct. 86 (13) (2008) 1417–1425, http://dx.doi.org/10.1016/j.compstruc.2007.04.030.
[41] B. Niu, J. Yan, G. Cheng, Optimum structure with homogeneous optimum cellular material for maximum fundamental frequency, Struct. Multidiscip. Optim. 39 (2) (2009) 115–132, http://dx.doi.org/10.1007/s00158-008-0334-4.
[42] J. Deng, J. Yan, G. Cheng, Multi-objective concurrent topology optimization of thermoelastic structures composed of homogeneous porous material, Struct. Multidiscip. Optim. 47 (4) (2013) 583–597, http://dx.doi.org/10.1007/s00158-012-0849-6.
[43] L. Xia, P. Breitkopf, Concurrent topology optimization design of material and structure within FE2 nonlinear multiscale analysis framework, Comput. Methods Appl. Mech. Engrg. 278 (2014) 524–542, http://dx.doi.org/10.1016/j.cma.2014.05.022.
[44] X. Guo, X. Zhao, W. Zhang, J. Yan, G. Sun, Multi-scale robust design and optimization considering load uncertainties, Comput. Methods Appl. Mech. Engrg. 283 (2015) 994–1009, http://dx.doi.org/10.1016/j.cma.2014.10.014.
[45] J. Yan, X. Guo, G. Cheng, Multi-scale concurrent material and structural design under mechanical and thermal loads, Comput. Mech. 57 (3) (2016) 437–446, http://dx.doi.org/10.1007/s00466-015-1255-x.
[46] A. Clausen, N. Aage, O. Sigmund, Exploiting additive manufacturing infill in topology optimization for improved buckling load, Engineering 2 (2) (2016) 250–257, http://dx.doi.org/10.1016/J.ENG.2016.02.006.
[47] S. Zhou, Q. Li, Design of graded two-phase microstructures for tailored elasticity gradients, J. Mater. Sci. 43 (15) (2008) 5157–5167, http://dx.doi.org/10.1007/s10853-008-2722-y.
[48] D. Li, W. Liao, N. Dai, G. Dong, Y. Tang, Y.M. Xie, Optimal design and modeling of gyroid-based functionally graded cellular structures for additive manufacturing, Comput. Aided Des. 104 (2018) 87–99, http://dx.doi.org/10.1016/j.cad.2018.06.003.
[49] C. Wang, J.H. Zhu, W.H. Zhang, S.Y. Li, J. Kong, Concurrent topology optimization design of structures and non-uniform parameterized lattice microstructures, Struct. Multidiscip. Optim. 58 (1) (2018) 35–50, http://dx.doi.org/10.1007/s00158-018-2009-0.
[50] C. Imediegwu, R. Murphy, R. Hewson, M. Santer, Multiscale structural optimization towards three-dimensional printable structures, Struct. Multidiscip. Optim. 60 (2) (2019) 513–525, http://dx.doi.org/10.1007/s00158-019-02220-y.
[51] E. Garner, H.M.A. Kolken, C.C.L. Wang, A.A. Zadpoor, J. Wu, Compatibility in microstructural optimization for additive manufacturing, Addit. Manuf. 26 (2019) 65–75, http://dx.doi.org/10.1016/j.addma.2018.12.007.

[52] C. Wang, X. Gu, J. Zhu, H. Zhou, S. Li, W. Zhang, Concurrent design of hierarchical structures with three-dimensional parameterized lattice microstructures for additive manufacturing, Struct. Multidiscip. Optim. 61 (3) (2020) 869–894, http://dx.doi.org/10.1007/s00158-019-02408-2.
[53] V.-N. Hoang, P. Tran, V.-T. Vu, H. Nguyen-Xuan, Design of lattice structures with direct multiscale topology optimization, Compos. Struct. 252 (2020) 112718, http://dx.doi.org/10.1016/j.compstruct.2020.112718.
[54] E.D. Sanders, A. Pereira, G.H. Paulino, Optimal and continuous multilattice embedding, Sci. Adv. 7 (16) (2021) eabf4838, http://dx.doi.org/10.1126/sciadv.abf4838.
[55] F.V. Senhora, E.D. Sanders, G.H. Paulino, Optimally-tailored spinodal architected materials for multiscale design and manufacturing, Adv. Mater. 34 (26) (2022) 2109304, http://dx.doi.org/10.1002/adma.202109304.
[56] C.A. Soto, A.R. Díaz, On the modelling of ribbed plates for shape optimization, Struct. Optim. 6 (3) (1993) 175–188, http://dx.doi.org/10.1007/BF01743510.
[57] M.P. Bendsøe, Optimal shape design as a material distribution problem, Struct. Optim. 1 (4) (1989) 193–202, http://dx.doi.org/10.1007/BF01650949.
[58] J.P. Groen, O. Sigmund, Homogenization-based topology optimization for high-resolution manufacturable microstructures, Internat. J. Numer. Methods Engrg. 113 (8) (2018) 1148–1163, http://dx.doi.org/10.1002/nme.5575.
[59] S.D. Larsen, O. Sigmund, J.P. Groen, Optimal truss and frame design from projected homogenization-based topology optimization, Struct. Multidiscip. Optim. 57 (4) (2018) 1461–1474, http://dx.doi.org/10.1007/s00158-018-1948-9.
[60] J.P. Groen, J. Wu, O. Sigmund, Homogenization-based stiffness optimization and projection of 2D coated structures with orthotropic infill, Comput. Methods Appl. Mech. Engrg. 349 (2019) 722–742, http://dx.doi.org/10.1016/j.cma.2019.02.031.
[61] G. Allaire, P. Geoffroy-Donders, O. Pantz, Topology optimization of modulated and oriented periodic microstructures by the homogenization method, Comput. Math. Appl. 78 (7) (2019) 2197–2229.
[62] F.C. Stutz, J.P. Groen, O. Sigmund, J.A. Bærentzen, Singularity aware de-homogenization for high-resolution topology optimized structures, Struct. Multidiscip. Optim. 62 (5) (2020) 2279–2295, http://dx.doi.org/10.1007/s00158-020-02681-6.
[63] L. Wang, Z. Liu, D. Da, Y.-C. Chan, W. Chen, P. Zhu, Generalized de-homogenization via sawtooth-function-based mapping and its demonstration on data-driven frequency response optimization, Comput. Methods Appl. Mech. Engrg. 395 (2022) 114967, http://dx.doi.org/10.1016/j.cma.2022.114967.
[64] D.A. White, W.J. Arrighi, J. Kudo, S.E. Watts, Multiscale topology optimization using neural network surrogate models, Comput. Methods Appl. Mech. Engrg. 346 (2019) 1118–1135, http://dx.doi.org/10.1016/j.cma.2018.09.007.
[65] D. Da, Y.-C. Chan, L. Wang, W. Chen, Data-driven and topological design of structural metamaterials for fracture resistance, Extreme Mech. Lett. 50 (2022) 101528, http://dx.doi.org/10.1016/j.eml.2021.101528.
[66] 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 (031704) (2022) http://dx.doi.org/10.1115/1.4055925.
[67] Z. Zhou, Y. Zhu, X. Guo, Machine learning based asymptotic homogenization and localization: Predictions of key local behaviors of multiscale configurations bearing microstructural varieties, Internat. J. Numer. Methods Engrg. 124 (3) (2023) 639–669, http://dx.doi.org/10.1002/nme.7136.
[68] O. Pantz, K. Trabelsi, A post-treatment of the homogenization method for shape optimization, SIAM J. Control Optim. 47 (3) (2008) 1380–1398.
[69] A.D. Cramer, V.J. Challis, A.P. Roberts, Microstructure interpolation for macroscopic design, Struct. Multidiscip. Optim. 53 (3) (2016) 489–500, http://dx.doi.org/10.1007/s00158-015-1344-7.
[70] Z. Du, X.-Y. Zhou, R. Picelli, H.A. Kim, Connecting microstructures for multiscale topology optimization with connectivity index constraints, J. Mech. Des. 140 (111417) (2018) http://dx.doi.org/10.1115/1.4041176.
[71] K. Liu, R. Sun, C. Daraio, Growth rules for irregular architected materials with programmable properties, Science 377 (6609) (2022) 975–981, http://dx.doi.org/10.1126/science.abn1459.
[72] M. Zaiser, S. Zapperi, Disordered mechanical metamaterials, Nat. Rev. Phys. 5 (11) (2023) 679–688, http://dx.doi.org/10.1038/s42254-023-00639-3.
[73] C.B. Sullivan, A.A. Kaszynski, PyVista: 3D plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK), J. Open Source Softw. 4 (37) (2019) 1450, http://dx.doi.org/10.21105/joss.01450.
[74] Oliphant, E. Travis, Python for scientific computing, Comput. Sci. Eng. 9 (3) (2007) 10–20.
[75] T.J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Courier Corporation, 2012.
[76] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, 2017, http://dx.doi.org/10.48550/arXiv.1412.6980, arXiv:arXiv:1412.6980.
[77] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019).
[78] B. Bourdin, Filters in topology optimization, Internat. J. Numer. Methods Engrg. 50 (9) (2001) 2143–2158, http://dx.doi.org/10.1002/nme.116.
[79] O. Sigmund, S. Torquato, Design of materials with extreme thermal expansion using a three-phase topology optimization method, J. Mech. Phys. Solids 45 (6) (1997) 1037–1067, http://dx.doi.org/10.1016/S0022-5096(96)00114-7.
[80] L.V. Gibiansky, O. Sigmund, Multiphase composites with extremal bulk modulus, J. Mech. Phys. Solids 48 (3) (2000) 461–498, http://dx.doi.org/10.1016/S0022-5096(99)00043-5.
[81] J. Stegmann, E. Lund, Discrete material optimization of general composite shell structures, Internat. J. Numer. Methods Engrg. 62 (14) (2005) 2009–2027, http://dx.doi.org/10.1002/nme.1259.
[82] M.P. Bendsøe, O. Sigmund, Material interpolation schemes in topology optimization, Arch. Appl. Mech. 69 (9) (1999) 635–654, http://dx.doi.org/10.1007/s004190050248.
[83] O. Sigmund, Design of multiphysics actuators using topology optimization–Part II: two-material structures, Comput. Methods Appl. Mech. Engrg. 190 (49) (2001) 6605–6627, http://dx.doi.org/10.1016/S0045-7825(01)00252-3.
[84] K. Svanberg, The method of moving asymptotes—a new method for structural optimization, Internat. J. Numer. Methods Engrg. 24 (2) (1987) 359–373, http://dx.doi.org/10.1002/nme.1620240207.
[85] M.W. Scroggs, J.S. Dokken, C.N. Richardson, G.N. Wells, Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes, ACM Trans. Math. Software 48 (2) (2022) 18:1–18:23, http://dx.doi.org/10.1145/3524456.
[86] Y. Lu, L. Tong, Concurrent multiscale topology optimization of metamaterials for mechanical cloak, Comput. Methods Appl. Mech. Engrg. 409 (2023) 115966, http://dx.doi.org/10.1016/j.cma.2023.115966.
[87] E.D. Sanders, M.A. Aguiló, G.H. Paulino, Optimized lattice-based metamaterials for elastostatic cloaking, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 477 (2253) (2021) 20210418, http://dx.doi.org/10.1098/rspa.2021.0418.
[88] Y. Jia, O. Lopez-Pamies, X.S. Zhang, Controlling the fracture response of structures via topology optimization: from delaying fracture nucleation to maximizing toughness, J. Mech. Phys. Solids 173 (2023) 105227, http://dx.doi.org/10.1016/j.jmps.2023.105227.
[89] X.S. Zhang, H. Chi, Z. Zhao, Topology optimization of hyperelastic structures with anisotropic fiber reinforcement under large deformations, Comput. Methods Appl. Mech. Engrg. 378 (2021) 113496, http://dx.doi.org/10.1016/j.cma.2020.113496.
[90] Y. Zhou, T. Nomura, K. Saitou, Multi-component topology and material orientation design of composite structures (MTO-C), Comput. Methods Appl. Mech. Engrg. 342 (2018) 438–457, http://dx.doi.org/10.1016/j.cma.2018.07.039.
[91] T. Gao, W. Zhang, P. Duysinx, A bi-value coding parameterization scheme for the discrete optimal orientation design of the composite laminate, Internat. J. Numer. Methods Engrg. 91 (1) (2012) 98–114, http://dx.doi.org/10.1002/nme.4270.
[92] C.Y. Kiyono, E.C.N. Silva, J.N. Reddy, A novel fiber optimization method based on normal distribution function with continuously varying fiber path, Compos. Struct. 160 (2017) 503–515, http://dx.doi.org/10.1016/j.compstruct.2016.10.064.

[93] C. Talischi, G.H. Paulino, A. Pereira, I.F.M. Menezes, PolyTop: A Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes, Struct. Multidiscip. Optim. 45 (3) (2012) 329–357, http://dx.doi.org/10.1007/s00158-011-0696-x.
[94] F. Ferrari, O. Sigmund, A new generation 99 line Matlab code for compliance topology optimization and its extension to 3D, Struct. Multidiscip. Optim. 62 (4) (2020) 2211–2228, http://dx.doi.org/10.1007/s00158-020-02629-w.
[95] R.Z. Horace He, Functorch: JAX-like composable function transforms for pytorch, 2021, https://github.com/pytorch/functorch.
[96] K. Jordahl, J. Van den Bossche, M. Fleischmann, J. Wasserman, J. McBride, J. Gerard, J. Tratner, M. Perry, A. Garcia Badaracco, C. Farmer, et al., Geopandas/geopandas: v0.8.1, 2020, Zenodo.
[97] S. Gillies, The shapely user manual, 2013, URL https://pypi.org/project/Shapely.