

# A Computational Framework to design 3D stiffness gradient acoustic metamaterials for impedance matching

Rayehe Karimi Mahabadi <sup>a,b</sup>, Doksoo Lee <sup>c</sup>, Alexander C. Ogren <sup>d</sup>,  
Chiara Daraio <sup>d</sup>, Wei Chen <sup>c</sup>, Cynthia Rudin <sup>e,f</sup>, L. Catherine Brinson <sup>a,\*</sup>

<sup>a</sup> Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA

<sup>b</sup> Department of Civil and Environmental Engineering, Duke University, Durham, NC, USA

<sup>c</sup> Department of Mechanical Engineering, Northwestern University, Evanston, IL, USA

<sup>d</sup> Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA, USA

<sup>e</sup> Department of Computer Science, Duke University, Durham, NC, USA

<sup>f</sup> Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA

## ARTICLE INFO

### Keywords:

Metamaterial  
Acoustic impedance matching  
Gradient stiffness  
Optimization  
Differentiable machine learning  
Surrogate modeling

## ABSTRACT

Acoustic waves play a crucial role in various applications, including medical imaging, non-destructive testing, and sonar systems. One of the significant challenges in these applications is impedance matching, which is essential for minimizing reflections and maximizing the transfer of acoustic energy between different media. Acoustic metamaterials offer a promising solution to this challenge. In addition to impedance control, gradient stiffness can enhance structural efficiency and enable spatial control of wave propagation, making it a valuable feature in acoustic metamaterial design. In this paper, we present our developed computational method to design 3D stiffness gradient acoustic metamaterials for impedance matching. The key steps in our approach include generating initial designs using a periodic covariance function to provide unit cells that are both periodic on the boundaries and randomly formed inside the unit cell. Furthermore, we integrated manufacturing constraints into the design process, ensuring that the structures are interconnected for fabrication. We propose two computational optimization algorithms: GenUnit, based on a non-dominated sorting genetic algorithm (NSGA-II), and MLMatch, which leverages differentiable machine learning. The two approaches are not separate contributions but complementary components of a unified framework. GenUnit requires no training data and directly interfaces with physics-based simulations, making it highly accurate but slower for large-scale exploration. In contrast, MLMatch is data-hungry during training but, once trained, enables near-instantaneous inference and broad design-space coverage. Together, they form a hybrid strategy: MLMatch rapidly explores the global design space, and GenUnit provides local refinement with high-fidelity accuracy. This balance between training cost, inference time, and precision is the motivation for including both methods in the same study. We applied this dual-algorithm framework to generate two metallic-based metamaterial designs that match the acoustic impedance of water while exhibiting a controlled gradient in stiffness (from stiff to soft). The stiffness gradient is particularly advantageous in applications where one side of the structure must interface with soft or sensitive surfaces, such as human tissue or delicate components. This work paves the way for improved materials in various acoustic applications, particularly in ultrasound devices, by providing better impedance matching and thereby improving the efficiency of acoustic energy transfer.

\* Corresponding author.

E-mail address: [cate.brinson@duke.edu](mailto:cate.brinson@duke.edu) (L.C. Brinson).

<https://doi.org/10.1016/j.cma.2025.118571>

Received 29 June 2025; Received in revised form 4 November 2025; Accepted 6 November 2025

Available online 21 November 2025

0045-7825/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

## 1. Introduction

Acoustic waves have a wide range of applications in communication, medicine, agriculture, and industry [1]. Acoustic impedance is a physical property that describes the resistance of a medium to the propagation of acoustic waves [2,3]. At interfaces between two domains with different impedances, there is a power loss and wave reflection [4,5]. Since the early 20<sup>th</sup> century, when Bell Laboratories discovered the crucial role of impedance matching in enabling transcontinental telephone communication [6], the concept of impedance matching has remained a hot topic in maximizing energy transmission between two media [7].

The practical significance of impedance matching is evident in many acoustic applications. For instance, in medical and non-destructive testing applications, transducers are used to generate and receive acoustic waves [8]. However, there is an impedance mismatch between the transducer and the target domain, such as human body tissue, resulting in power loss and a prolonged operational time [9]. Another example is underwater sensing devices that require an impedance that is matched to water for optimal performance [10]. Similarly, in acoustic lens design [11], impedance matching of the lens with the environment impacts the focus [12]. Therefore, a critical point in improving the performance of acoustic devices is impedance matching [13] to help maximize power transfer. Acoustic metamaterials [14,15] can serve as a material platform to achieve impedance matching.

Metamaterials are artificially engineered structures that support either exceptional properties or emergent behavior that are not readily achievable in nature [16,17]. A distinct aspect of metamaterial design is that their special functionalities stem from their engineered structure and material composition [7,18–20]. Design of acoustic metamaterials has been essential in identifying the architected material systems that support superior, often exotic, sound propagation behaviors [21]. A non-exhaustive list includes impedance matching [22], bandgap design [23], negative refraction [24], nonreciprocal propagation [25], and acoustic cloaking [26].

Despite the promise of acoustic metamaterials, reverse-engineering of them presents a grand challenge. Conventional design approaches to metamaterials include metaheuristic optimization [27,28], Bayesian optimization [29–33], topology optimization [16,34–39], among others; see Campbell et al. [40], Elsayy et al. [41], or Schneider et al. [42] for a review. Recent years have seen the emergence and widespread adoption of machine learning for design of metamaterials, evidenced by review articles reported from different communities [43–48]. Key approaches therein include, but are not limited to, (i) generative modeling that is powerful in both distilling compact-yet-expressive design representations out of high dimensional data and identifying one-to-many solutions [49–55]; (ii) differentiable machine learning that enables gradient-based solution procedures through Automatic Differentiation and backpropagation [51,56,57], (iii) reinforcement learning that trains decision-making agents interacting with environments through cumulative reward maximization [58,59].

Metamaterials design for acoustic impedance matching has mainly relied on geometric motifs conceived through either human intuition or biomimetic inspiration. Examples include coiled-up structures [60], labyrinthine geometries [61], tapered labyrinthine designs [62], helical hole designs [63], gradient spiral-shaped metamaterials [64], coned-structure metamaterials [8], and MUT-type structures [13]. While these designs have proven effective for specific applications, there is a notable gap in the literature for a systematic approach to design metamaterials that can achieve impedance matching across diverse target domains. This gap highlights the need for a more inclusive computational framework that can be applied generally rather than being limited to specific target cases.

Beyond acoustic impedance, there are additional material properties that remain underexplored in the design of acoustic metamaterials, yet have been shown to significantly enhance performance in broader structural applications. For instance, gradient stiffness, widely adopted in aerospace and mechanical structures, has been demonstrated to improve load bearing efficiency, enhance buckling resistance, and enable spatial control of wave propagation [65]. Despite its clear mechanical advantages, gradient stiffness has not been systematically incorporated into acoustic metamaterial design. This gap highlights the necessity for multiobjective computational frameworks that jointly consider acoustic impedance and mechanical performance metrics, enabling the design of functional materials that meet both wave manipulation and structural reliability criteria.

Here, we propose a computational framework to generate voxelated metamaterial designs with gradient stiffness that simultaneously achieves impedance matching with target materials. This framework algorithmically ensures periodicity and inter-connectivity in designs. Benefiting from multiobjective optimization, the framework generates designs with desired properties, specifically stiffness and acoustic impedance. Our approach addresses a broad design space, unconstrained by geometric intuition, enabling exploration of novel, data-driven architectures. The inverse design framework integrates two complementary computational algorithms: GenUnit, which is based on a non-dominated sorting genetic algorithm II (NSGA-II), and MLMatch, which leverages differentiable machine learning models. These algorithms can be used independently or in tandem, depending on the application requirements. GenUnit does not require a training phase and operates directly through physics-based simulations, offering high accuracy at the cost of longer inference times. In contrast, MLMatch involves a significant initial training cost but enables rapid design evaluations post-training. Using these algorithms, we propose optimized designs to achieve impedance matching with the target domain and desired stiffness. To showcase the ability of our framework, we use it to generate designs out of a metallic alloy with gradient stiffness and acoustic impedance matching that of water. To the best of our knowledge, although there are polymers that can mimic the acoustic impedance of water, no studies have successfully achieved water-like impedance using metals. Additionally, we incorporate a stiffness gradient in the metamaterial structure which is particularly beneficial for applications requiring soft contact at one interface and structural rigidity at the other. For example, in medical ultrasound devices, a graded stiffness allows seamless acoustic coupling with soft

biological tissues while maintaining mechanical stability on the transducer side. This dual functionality enhances performance, user safety, and device durability, making the designs especially suited for biomedical and precision sensing applications.

## 2. Methodology

In this section, we detail our framework for generating metamaterials that achieve impedance matching with a target material and exhibit a gradient in stiffness. We present two complementary algorithms in our framework for metamaterial design: (i) a population-based optimization method we name GenUnit, which leverages NSGA-II to evolve voxelated unit cell structures toward optimal acoustic impedance and stiffness tradeoffs, and (ii) a machine learning-based inverse design method, named MLMatch, which rapidly generates design candidates for new target properties. While GenUnit systematically explores the design space to identify Pareto-optimal solutions, MLMatch generalizes beyond observed data to propose solutions in real-time. These two approaches balance cost and scalability: GenUnit is computationally efficient and well-suited for practical optimization, whereas MLMatch demands higher computational effort but enables broader, on-demand design generation across varying targets. Together, these methods enable both deep exploration and rapid prediction, significantly enhancing design flexibility and performance scalability. We describe the generation of our unit cells (Section 2.1) and the methods used to ensure connectivity within the unit cell design (Section 2.2). Subsequently, we outline the procedures for calculating the impedance and stiffness of our structure (Section 2.3). Finally, we present the optimization methods employed, GenUnit (Section 2.4) and MLMatch (Section 2.5).

### 2.1. Initialization of periodic unit cells

To generate random, but sensible unit cells, we sample a Gaussian process with a periodic covariance function. This is done by combining one-dimensional periodic covariance functions in each direction through multiplication.

Let  $\mathbf{x} = (x, y, z)$  and  $\mathbf{x}' = (x', y', z')$  denote the spatial coordinates within the unit cell. The vector  $\mathbf{L} = (\ell_x, \ell_y, \ell_z)$  represents the characteristic length scales in each spatial direction, where each  $\ell_i$  defines the feature size of the material along the  $i$ -th axis. The parameter  $p$  defines the period, i.e., the side length of the unit cell in each spatial dimension. Finally,  $\sigma_f^2$  controls the global variance of the Gaussian process. The one-dimensional periodic covariance function is defined as [66]

$$k_{\text{per-1D}}(x, x', \ell) \propto \sigma_f^2 \exp\left(-\frac{2}{\ell^2} \sin^2\left(\pi \frac{|x - x'|}{p}\right)\right), \quad (1)$$

and thus the full three-dimensional periodic covariance function is

$$k_{\text{per-3D}}(\mathbf{x}, \mathbf{x}', \mathbf{L}) \propto k_{\text{per-1D}}(x, x', \ell_x) k_{\text{per-1D}}(y, y', \ell_y) k_{\text{per-1D}}(z, z', \ell_z). \quad (2)$$

Because the covariance is defined as a product of one-dimensional periodic kernels, the correlation between two points is high only if they are close in all coordinate directions relative to the corresponding length scales. In this sense, the kernel behaves like a logical “and” across  $x$ ,  $y$ , and  $z$ . This does not restrict the choice of length scales: for example, selecting a small  $\ell_x$  and a large  $\ell_y$  yields features that are short in  $x$  but elongated in  $y$ , consistent with the intended anisotropy of the covariance.

We define a stationary, periodic Gaussian process over a 3D spatial domain:

$$f(\mathbf{x}) \sim \mathcal{GP}_{\text{per-3D}}(m(\mathbf{x}), k_{\text{per-3D}}(\mathbf{x}, \mathbf{x}'; \mathbf{L})), \quad (3)$$

where  $m(\mathbf{x}) = 0.5$  is the constant mean function,  $\mathbf{x}$  and  $\mathbf{x}'$  are spatial positions within the unit cell such that  $\mathbf{x}, \mathbf{x}' \in [0, p]^3$ , and  $\mathbf{L} = (\ell_x, \ell_y, \ell_z)$  defines the correlation length scales in the  $x$ ,  $y$ , and  $z$  directions, respectively. The kernel  $k_{\text{per-3D}}$  is constructed as a product of one-dimensional periodic kernels:

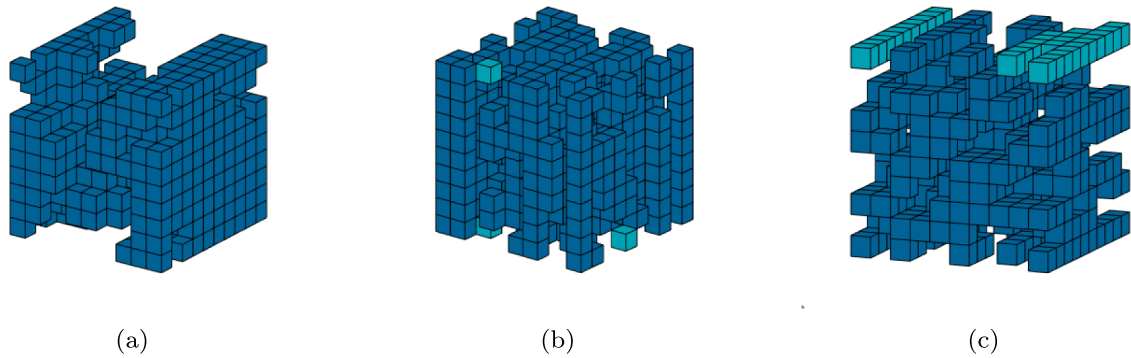
$$k_{\text{per-3D}}(\mathbf{x}, \mathbf{x}'; \mathbf{L}) = \sigma_f^2 \prod_{i \in \{1,2,3\}} \exp\left(-\frac{2 \sin^2(\pi(\mathbf{x}_i - \mathbf{x}'_i)/p)}{\ell_i^2}\right). \quad (4)$$

This formulation allows spatial correlation and periodicity in all three directions, and corresponds to a factorized Gaussian process where the covariance function decomposes along the coordinate axes.

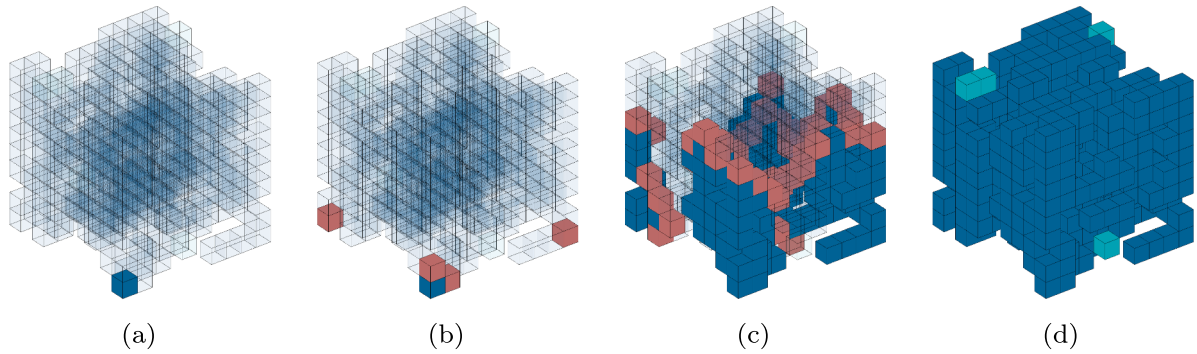
To generate a unit cell design, we draw a sample  $f(\mathbf{x})$  from the Gaussian process  $\mathcal{GP}_{\text{per-3D}}$ . We do this on a regular 3D voxel grid spanning the unit cell. Letting  $\mathbf{x}^j$  denote the spatial coordinates of each voxel in this grid, the sample drawn from the GP is given as a 3D array  $f(\mathbf{x}^j)$ . Importantly,  $f(\mathbf{x})$  statistically obeys both the length scales and the continuous periodicity encoded in the covariance function of the GP. Ultimately, we want a unit cell design  $d(\mathbf{x}^j)$  that only comprises voxels of material or void, so we apply a threshold (e.g., for each  $\mathbf{x}^j$  in the grid, if  $f(\mathbf{x}^j) > \text{thresh}$ , assign  $d(\mathbf{x}_j) \leftarrow 1$ , else  $d(\mathbf{x}_j) \leftarrow 0$ ) to obtain a binary material distribution. In other words, each voxel of the unit cell either contains material ( $d(\mathbf{x}^j) = 1$ ) or is void ( $d(\mathbf{x}^j) = 0$ ).

### 2.2. Connectivity check

Because of the void voxels in the geometries, we found it essential to establish an efficient connectivity-check algorithm to ensure manufacturability. This algorithm enables us to quickly and inexpensively check whether a design has any ‘floating’ (disconnected) portions that would disrupt the structural integrity of the design and prevent fabrication. Designs generated by the method described in Section 2.1 are often connected, because of the periodicity constraints and spatial correlation between voxel values. However,



**Fig. 1.** Examples of unit cell designs generated using the periodic Gaussian process method. The length scale vectors for designs (a), (b), and (c) are (1.0, 1.0, 1.0), (0.5, 0.2, 1.0), and (1.0, 0.5, 0.2), respectively. Dark blue voxels indicate the largest connected component identified by the `find_all_connected_groups` algorithm (Algorithm 1), while lighter green-blue voxels represent regions not connected to the main structure through face-to-face contact. In (b), the long vertical features result from a larger length scale in the vertical direction. In (c), elongated in-plane features are visible due to increased in-plane length scales. Note that some lighter voxels may appear clustered (as in (c)) or isolated (as in (b)), but they are excluded from the main structure as they do not meet the face-connectivity criterion required for structural stability. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 2.** Step-by-step procedure of the connectivity-check algorithm using voxel growth. (a) Iteration step 1: The input voxel design with a randomly selected starting point. (b) Iteration step 2: Neighboring voxels are recursively added (newly found neighbors in each iteration are shown in red). Note that the isolated red cube in (b) is on the back face and connected to the dark blue cube on the front face via periodicity. (c) Iteration step 22: Additional voxels have been added to the connected group. (d) Iteration step 49: Final result showing the fully connected voxel group in dark blue and disconnected regions in light green-blue. The number of iteration steps depends on the voxel count and connectivity pattern of the design. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

connectivity is not a guarantee, and must be checked. Additionally, in later Sections 2.4 and 2.5, objective-based design methods are introduced. The connectivity check described in this section is also helpful for screening designs proposed by optimization steps.

The connectivity check algorithm we implement works by starting in a random voxel and iteratively expanding outwards (in a connected neighbor fashion). In this context, we define a connection strictly as a face-to-face contact between voxels; edge or corner contacts are disregarded, as they do not provide sufficient structural stability during 3D printing. This process continues until no new neighbors can be found. The resulting connected group is then compared against the full set of design voxels to determine whether any voxel remains unaccounted for. If so, the process repeats, starting from these voxels, to identify additional connected groups. After all groups are identified, the algorithm retains only the largest connected component and removes all smaller, isolated clusters. The overall connectivity-check procedure is described in Algorithm 1, which iteratively identifies all connected components in a design by repeatedly growing connected regions from randomly selected starting voxels. Each connected group is identified using Algorithm 2, which progressively expands the region by adding connected neighboring voxels layer by layer from a seed voxel. This expansion relies on Algorithm 6 (given in the Appendix), which returns connected neighboring voxels while accounting for periodicity in the unit cell. Because the initial seed voxel is selected at random, the first group found is not necessarily the largest, which necessitates a complete search over all connected clusters. Fig. 1 showcases examples of the generated designs and Fig. 2 shows each step in the connectivity check algorithm. It is worth mentioning that this procedure follows the same principle as standard graph-search algorithms such as breadth-first search (BFS) or depth-first search (DFS) on the voxel adjacency graph. In our implementation, however, the search is specialized to enforce periodic boundary conditions and face-to-face connectivity (excluding edge or corner contacts) to ensure manufacturability. This tailored formulation allows efficient integration with our property-computation pipeline and large-scale optimization framework.

---

**Algorithm 1:** Find all connected groups Algorithm to find all connected groups of material voxels in a unit cell design.

---

**Input:**  $D$  - a 3-dimensional boolean array representing the unit cell design, containing 1 where there is material, and 0 where there is void

**Output:**  $G$  - a set of sets, where each set  $G_c \in G$  contains the voxel indices of connected group  $c$

```

1  $M_r \leftarrow \{\text{indices where } D == 1\}$  % Initialize the set of remaining material voxels
2  $c \leftarrow 0$  % Initialize group index
3 while  $M_r$  not empty do
4   increment  $c$ 
5    $i \leftarrow$  random voxel index from  $M_r$ 
6    $G_c \leftarrow \text{find\_connected\_group}(M_r, i, D)$  % See Alg. 2.
7    $M_r \leftarrow M_r \setminus G_c$  % Remove the most recently found connected group from the remaining voxel set

```

---



---

**Algorithm 2:** Find connected group Function to find all voxels connected to a starting voxel.

---

**Input:**  $M_r$  - set of remaining material voxel indices  $i$  - starting voxel index,  $D$  - 3-dimensional boolean design array

**Output:**  $G_c$  - set of voxel indices that are connected to  $i$ , and contain material

```

1 Initialize  $G_c \leftarrow \{i\}$ 
2 Initialize  $G_{c\text{-new}} \leftarrow \{i\}$ 
3 while  $G_{c\text{-new}}$  is not empty do
4   % Note that get_neighbors accounts for periodicity.
5    $G_{c\text{-new}} \leftarrow \text{get\_neighbors}(G_c, D)$  % See Alg. 6 in Appendix.
6    $G_c \leftarrow G_c \cup G_{c\text{-new}}$ 

```

---

### 2.3. Impedance matching and stiffness calculation

The acoustic impedance ( $Z$ ) of a medium is defined with respect to a propagating elastic wave as the ratio between stress and particle velocity in the direction of propagation. In anisotropic solids, this generally depends on the wave vector  $\mathbf{k}$  and the polarization of the mode. In this work, we restrict attention to normal-incidence longitudinal waves propagating along the  $z$ -axis, for which the displacement is parallel to  $\mathbf{k} = k_z \hat{\mathbf{z}}$ . Under this condition, the effective impedance reduces to

$$Z = \rho c_z, \quad (5)$$

where  $\rho$  is the effective density of the unit cell and  $c_z$  is the longitudinal wave speed along  $z$ . This simplification is consistent with prior homogenized formulations of impedance for axis-aligned elastic media.

The longitudinal wave speed  $c_z$  is obtained from the slope of the longitudinal mode branch of the dispersion relation [67],

$$c_z = \frac{d\omega}{dk_z}, \quad (6)$$

evaluated near the  $\Gamma$  point (small  $k_z a$ ) where the branch is linear and group and phase velocities coincide. At the operating frequency of interest, the criterion  $k_z a \ll 1$  is satisfied, so this definition remains valid. The effective impedance of the metamaterial is then matched to that of water ( $Z_w$ ) by enforcing [5]

$$Z_{\text{eff}} = Z_w, \quad (7)$$

where  $Z_{\text{eff}}$  represents the homogenized impedance of the periodic structure.

For our second objective of gradient stiffness, we aim to calculate the homogenized stiffness of each unit cell. This is achieved by equalizing the strain energy of the unit cell ( $U$ ) with that of an equivalent uniform medium under identical boundary conditions [68–70]:

$$U = \frac{1}{2} \int_V \sigma : \epsilon \, dV, \quad (8)$$

where  $\sigma$  and  $\epsilon$  are the stress and strain tensors.

In the most general case, the effective constitutive relation of a heterogeneous unit cell can be expressed through the full anisotropic stiffness matrix:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{23} \\ 2\epsilon_{13} \\ 2\epsilon_{12} \end{bmatrix}. \quad (9)$$

Each component  $C_{ij}$  can be extracted by imposing specific boundary conditions on the unit cell and applying Eq. (8), as detailed in homogenization studies such as [69]. For instance, diagonal terms like  $C_{11}$ ,  $C_{22}$ ,  $C_{33}$  are obtained by applying uniaxial strains in  $x$ ,  $y$ , and  $z$  directions, respectively. Off-diagonal terms (e.g.,  $C_{12}$ ,  $C_{13}$ ,  $C_{23}$ ) require coupled loading conditions, while shear terms (e.g.,  $C_{44}$ ,  $C_{55}$ ,  $C_{66}$ ) are obtained from pure shear loading.

In our case, since the wave propagation problem of interest is strictly along the  $z$ -direction, only the effective stiffness component  $C_{33}$  is directly relevant. The other coefficients could in principle be computed, but are not required for the present application. Accordingly, we compute  $C_{33}$  by constraining all other deformations and applying  $\epsilon_{33} \neq 0$ :

$$U = \frac{V}{2} C_{33} \epsilon_{33}^2, \quad (10)$$

from which  $C_{33}$  is obtained given the strain energy  $U$  from FE simulations and the applied strain  $\epsilon_{33}$ . Periodic boundary conditions were applied to maintain continuity across opposite unit-cell faces [71].

It should be noted that the density is not fixed during optimization. As voxels are added or removed to achieve the target impedance and stiffness gradient, the density of each candidate design is recalculated, and the effective stiffness  $C_{33}$  is recomputed for that updated geometry using Eq. (8). This ensures that both density and stiffness consistently reflect the actual unit cell configuration at each optimization step.

It is also noted that in a general anisotropic solid, elastic wave propagation along a given direction can result in one quasi-longitudinal and two quasi-shear eigenmodes, all of which may be coupled through the acoustic tensor and associated stiffness matrix elements (e.g.,  $C_{33}$ ,  $C_{44}$ ,  $C_{55}$ ,  $C_{34}$ , and  $C_{35}$ ). Even for waves normally incident along a principal axis, nonzero off-diagonal terms can produce mode conversion, so a longitudinal excitation may also generate shear displacement. For a rigorous treatment of wave propagation and mode coupling in anisotropic media, see [72]. In this work, we deliberately restrict our analysis to axis-aligned, normal-incidence longitudinal waves  $\mathbf{k} = k_z \hat{\mathbf{z}}$  and disregard coupling to quasi-shear modes, as a proof-of-concept simplification consistent with established impedance homogenization frameworks. This assumption allows us to focus on the effective longitudinal mode, for which the impedance reduces to Eq. (5), with the effective stiffness determined by  $C_{33}$ .

Finite Element simulations are conducted using COMSOL Multiphysics to compute both the effective acoustic impedance and the homogenized stiffness of the unit cell. Also, Periodic boundary conditions are imposed to ensure consistent deformation behavior across opposite faces. The full simulation setup (unit-cell dimensions, voxel discretization, material parameters, meshing strategy, solver settings, and boundary conditions) is described in Section 3.1. We also performed a mesh sensitivity study (reported in Appendix A) to confirm the accuracy and reliability of the simulation results.

#### 2.4. Genunit optimization algorithm

We implement a customized multi-objective optimization algorithm, named GenUnit, based on multiobjective NSGA-II, specifically tailored for generating periodic, connected voxel-based unit cell designs. In addition to the standard NSGA-II components, GenUnit integrates voxel-based add/remove modification functions and an explicit connectivity constraint, which ensures manufacturability and structural feasibility of each candidate design. We adopt an NSGA-II framework due to its proven effectiveness in multi-criteria optimization and its ability to achieve meaningful exploration of the design space with relatively modest population sizes, making it a strong fit for our cost-aware computational framework.

A multi-objective optimization problem comprises a set of decision variables, multiple objective functions, and a series of constraints. The problem can be mathematically expressed as follows [73,74]:

$$\begin{aligned} \text{Min/Max } y &= F(x) = (F_1(x), \dots, F_k(x)), \quad k \geq 2 \\ \text{subject to } g_i(x) &\leq 0, \quad i = 1, 2, \dots, m \\ h_j(x) &= 0, \quad j = 1, 2, \dots, p. \end{aligned} \quad (11)$$

where  $x = (x_1, x_2, \dots, x_n) \in \mathbb{D}$  represents an  $n$ -dimensional vector in decision space  $\mathbb{D} = \mathbb{D}_1 \times \mathbb{D}_2 \times \dots \times \mathbb{D}_n$ , with  $\mathbb{D}_n$  being the domain of  $x_n$ . The vector  $y$  is a  $k$ -dimensional objective vector within  $\mathbb{R}^k$ ,  $F$  is the mapping function,  $g_i$  denotes the  $i^{\text{th}}$  inequality constraint, and  $h_j$  indicates the  $j^{\text{th}}$  equality constraint. Here,  $k$ ,  $m$ , and  $p$  signify the number of objective functions, inequality constraints, and equality constraints, respectively.

The Non-dominated Sorting Genetic Algorithm 2 (NSGA-II) is a widely-used multi-objective optimization algorithm [75] based on four key components that support the optimization process: Non-dominated Sorting, Elite Preserving Operator, Crowding Distance, and the Selection Operator. Non-dominated Sorting categorizes solutions based on Pareto dominance, creating distinct ‘fronts’ from the highest to the lowest dominance. The Elite Preserving Operator ensures that the best (non-dominated) solutions are retained across generations. Crowding Distance promotes diversity by estimating the density of solutions in the vicinity, favoring those from less-crowded regions for selection. The Selection Operator employs a crowded tournament method, selecting solutions based on their rank and crowding distance to form a diverse next generation [74]. These components work in concert to guide the optimization process toward a diverse and well-distributed approximation of the Pareto front.

In our customized GenUnit algorithm,  $x$  represents the design variables, which are voxel configurations. The objective function  $F(x)$  consists of two components: the effective acoustic impedance  $Z_{\text{eff}}$  and the stiffness in the  $z$  direction,  $C_{33}$ . For each optimization run, we target a fixed value for impedance, specifically the impedance of water ( $Z_{\text{target}} = 1.48 \text{ MRayl}$ ), and vary the target value of  $C_{33}$  within an applicable range. Rather than using a fixed scalarized objective function, GenUnit simultaneously evaluates multiple

objectives: effective acoustic impedance and stiffness. The algorithm constructs a Pareto front that captures the trade-offs between these two objectives, effectively exploring multiple implicit weightings across the design space.

The algorithm seeks to minimize the deviation from each target independently, and the Pareto front is constructed to reflect the trade-off between these objectives. The constraints  $g$  includes physical conditions related to the target property ranges, while  $h$  encodes structural feasibility constraints, such as connectivity and periodicity of the designs. We initiate the optimization with an initial population of voxelated unit cell designs. Here, the term population refers to a set of candidate solutions (designs) that are considered in each generation of the optimization process.

To generate new designs for subsequent generations, we utilize the Add and Remove functions instead of traditional crossover and mutation operations. These functions are tailored to preserve periodicity and connectivity that ensure each design remains manufacturable. For efficient design generation, the algorithm randomly selects an integer between 1 and the grid size, adding or removing multiple voxels to the design. In the Add function, the placement ensures the new voxels are connected to the existing voxels while supporting the periodicity of the design. Specifically, additions to a corner replicate voxels to all corners, those placed on edges are added to locations on four parallel edges, boundary face additions are placed twice on the opposite faces, and interior voxels are added once. Similarly, the Remove function randomly selects voxels for deletion, applying equivalent periodicity constraints. For example, removal of a corner voxel requires removing all eight voxels located in the vertices. For the connectivity check, after removing the voxels, the new design is sent to the connectivity check function to ensure it remains connected. If removing the selected voxels violates connectivity, we reconsider which voxels should be removed by reselecting the voxels. More details of the Add and Remove functions are provided in [Algorithm 3](#) and [Algorithm 4](#), respectively.

---

**Algorithm 3:** Arithmetic Add Function to make new designs from the previous ones by adding cubes.

---

**Input:**  $p1$  - initial structure, *Grid Number*  
**Output:**  $popc1$  - updated structure

```

1 Initialize  $a1 \leftarrow$  random integer between 1 and Grid Number;
2 Initialize  $AFlag \leftarrow$  true;
3 Initialize  $popc1 \leftarrow p1$ ;
4 while  $a1 > 0$  and  $AFlag$  do
5     if  $group\_data.all\_empty\_neighbors$  is empty then
6          $AFlag \leftarrow$  false % The design is full; there is no empty spot to add voxels;
7         return;
8      $A\_index \leftarrow$  random index from  $group\_data.all\_empty\_neighbors$ ;
9     if  $A\_index$  refers to a voxel inside then
10         Add that voxel only to the new design;
11     else if  $A\_index$  is on the corners then
12         Add that voxel to all 8 corners;
13     else if  $A\_index$  is at the boundary edges then
14         Add that voxel to the 4 opposite edges;
15     else
16         Add that voxel to both opposite faces;
17     Update  $design$  and  $group\_data$  on neighbor list;
18     Update  $popc1$  properties with new values;
19      $a1 \leftarrow a1 - 1$ ;

```

---

After generating offspring designs, using the Add and Remove modification functions, the parent and offspring populations are merged and sorted into Pareto fronts using non-dominated sorting. The next generation is composed of solutions from these fronts, prioritizing the highest-ranked individuals, with additional selection pressure applied to maintain diversity using crowding distance. If a front exceeds the population size limit, designs from less crowded regions are preferentially retained to maintain diversity. This process is guided by the four core components of NSGA-II non-dominated sorting, elite preservation, crowding distance, and tournament selection—which collectively balance exploration and exploitation in the design space. These components ensure that a diverse set of high-performing candidates is preserved across generations and enable the algorithm to explore trade-offs between impedance and stiffness, thereby progressively shaping a well-distributed Pareto front. Each design is evaluated based on its impedance and stiffness, which serve as the optimization objectives. The algorithm iterates for several generations, progressively refining the Pareto-optimal set to achieve the target impedance and stiffness values. The optimization is repeated over a range of target stiffness values combined with a target impedance to produce a series of Pareto-optimal sets. A flowchart of the algorithm is shown in [Fig. 3](#).

Note that GenUnit relies on the exact solution for the stiffness and impedance obtained for each chosen design configuration by direct numerical simulation in COMSOL. In the next section, we introduce an approach that uses a surrogate model to replace the full Finite Element simulation.

**Algorithm 4:** Arithmetic Remove Function to make new designs from the previous ones by removing cubes.

---

**Input:**  $p2$  - initial structure, *Grid Number*  
**Output:**  $popc2$  - updated structure

```

1 Initialize  $a2 \leftarrow$  random integer between 1 and Grid Number;
2 Initialize  $RFlag \leftarrow$  true;
3 Initialize  $popc2 \leftarrow p2$ ;
4 while  $a2 > 0$  and  $RFlag$  do
5   Initialize  $CFlag \leftarrow$  true % Checks connectivity;
6   Initialize  $trial \leftarrow 0$ ,  $trial_{max} \leftarrow 100$ ;
7   while  $trial < trial_{max}$  do
8      $R\_index \leftarrow$  random index from non_empty_neighbors;
9     if  $R\_index$  refers to a voxel inside then
10       Remove that voxel only;
11     else if  $R\_index$  is at the corners then
12       Remove the voxel from all 8 corners;
13     else if  $R\_index$  is at the boundary edges then
14       Remove the voxel from the 4 parallel edges (periodic counterparts);
15     else
16       Remove the voxel from both opposite faces;
17     Update design and group_data;
18     if design is connected then
19        $CFlag \leftarrow$  true;
20       return;
21     else
22        $CFlag \leftarrow$  false;
23        $trial \leftarrow trial + 1$ ;
24   if  $CFlag$  then
25     Update  $popc2$  properties with new values;
26      $a2 \leftarrow a2 - 1$ ;
27   else
28     Break from the loop; no feasible removal found. Continue with Add function.;
29    $a2 \leftarrow a2 - 1$ ;

```

---

## 2.5. MLMatch Inverse design algorithm

While GenUnit (Section 2.4) provides a reliable and cost-effective approach to multi-objective optimization, it requires iterative population-based search and multiple full-order evaluations, which may limit scalability when rapid generation of designs across a wide range of targets is needed. To overcome these limitations, we propose a differentiable machine learning-based inverse design framework, called MLMatch, which enables fast, on-demand generation of voxel configurations given target impedance and stiffness.

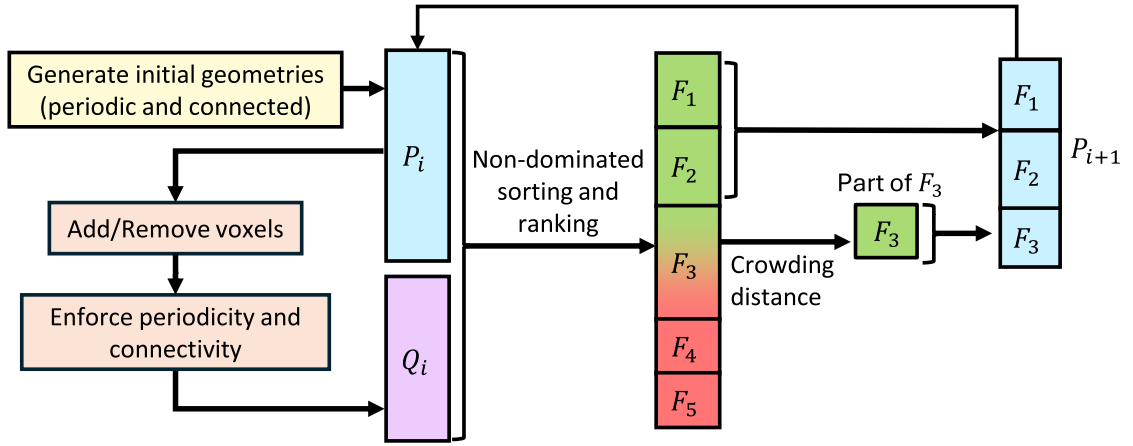
The goal of this framework is to complement GenUnit by accelerating the inverse design process, particularly in applications requiring functional grading over continuous property ranges. Specifically, the ML-based approach allows the creation of large design datasets without running full optimization for each target condition, enabling the downstream application discussed in Section 3.4. We construct two key tools: (1) an ML regressor that offers on-the-fly forward evaluation from a voxel configuration to the two effective properties (Section 2.5.1) and (2) a connectivity classifier that predicts whether the configuration is internally connected or disconnected (Section 2.5.2). Given the ML predictors, Automatic Differentiation [76] is harnessed to solve the constrained inverse problem through gradient-based optimization (Section 2.5.3). Fig. 4 gives a conceptual illustration of the proposed framework.

### 2.5.1. Regressor for impedance and stiffness

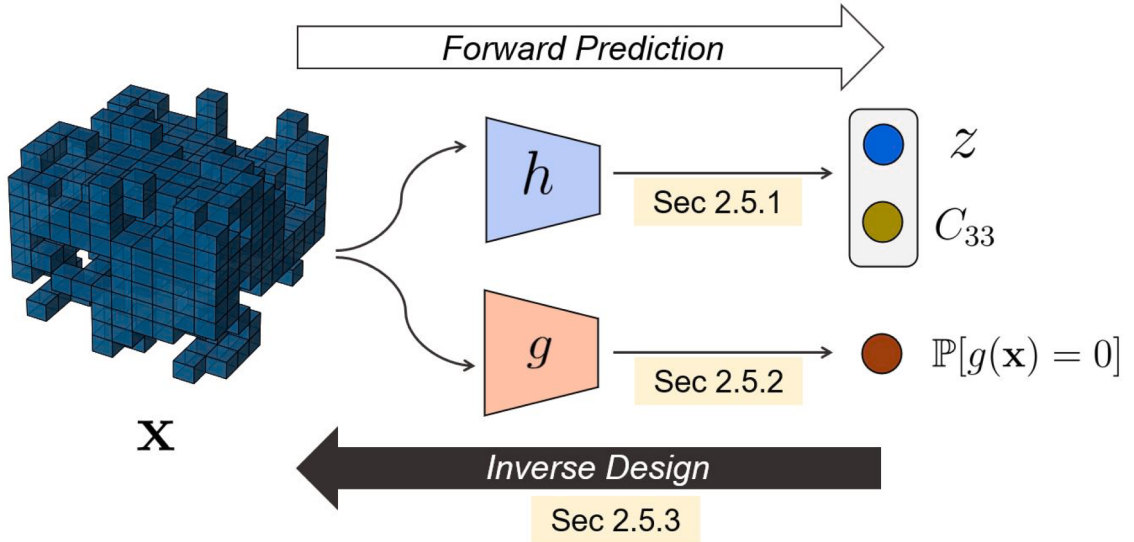
As the first step of this section, we construct an ML regressor that predicts the impedance and stiffness given a voxel configuration. The model is parameterized as a 3D convolutional neural network (CNN) that learns the underlying mapping between a voxel configuration and the corresponding effective property. Formally, a voxel configuration  $\mathbf{x}$  is represented as a collection of spatially distributed voxels, where the grid size  $N \in \mathbb{N}$  is set  $N = 10$  throughout this work. The learnable map of interest entails  $h : \{0, 1\}^{N^3} \rightarrow \mathbb{R}^2$ , where  $h$  denotes the impedance regressor that predicts two scalar values based on the input voxel configuration.

The model consists of  $d_{conv}$  3D convolutional layers for feature extraction, with ReLU activations for each layer except the final one, followed by a multi-layer perceptron (MLP) of depth  $d_{fc}$  for regression. The forward pass can be described by the recursive maps:

$$v_{l+1} = \sigma(W_l * v_l), \quad l = 1, \dots, d_{conv},$$



**Fig. 3.** Schematic illustration of the GenUnit multiobjective optimization framework. Here,  $P_i$  represents the population at the beginning of generation  $i$ . New designs  $Q_i$  are created by adding or removing voxels from  $P_i$ . The combined set is evaluated and sorted into fronts  $F_i$  using non-dominated sorting based on two objectives: impedance and stiffness. To maintain a fixed population size, some entire fronts (e.g.,  $F_4$  and  $F_5$ ) or individual solutions within a front (e.g.,  $F_3$ ) are removed. The resulting set forms  $P_{i+1}$ , the population at the beginning of the next generation.



**Fig. 4.** A visual illustration of the proposed framework.  $h$  and  $g$  denote the property regressor and the connectivity classifier, respectively.

$$\begin{aligned}
 f_{d_{\text{conv}}+1} &= \text{Flatten}(v_{d_{\text{conv}}}), \\
 f_{l+1} &= \sigma(W_l f_l + b_l), \quad l = d_{\text{conv}} + 1, \dots, d_{\text{conv}} + d_{\text{fc}} - 1, \\
 h(\mathbf{x}) &= W_{d_{\text{conv}}+d_{\text{fc}}} f_{d_{\text{conv}}+d_{\text{fc}}-1} + b_{d_{\text{conv}}+d_{\text{fc}}},
 \end{aligned} \tag{12}$$

where  $v_1 = \mathbf{x} \in \{0, 1\}^{N \times N \times N}$ ,  $\sigma$  denotes the ReLU activation function,  $*$  represents the 3D convolution operation, and  $W_l$  and  $b_l$  are the trainable weights and biases. The final output  $h(\mathbf{x}) \in \mathbb{R}^2$  represents the predicted impedance and stiffness values. Each hidden layer in the MLP has width  $w_{fc}$ , which is set to 64. The shape dataset  $S$  containing 18,917 unit cells is generated by the periodic unit cell generation method delineated in Section 2.1.1. Given a unit cell, the corresponding effective impedance  $Z$  and stiffness  $C_{33}$  are computed following the finite element simulation described in Section 2.3. Additional training details are provided under Details of Model Training Protocols in the Appendix.

### 2.5.2. Connectivity classifier

The connectivity check in this work, in principle, can be formulated as standard graph-search algorithms (e.g., breadth-first search [77], depth-first search [78]) on the voxel adjacency graph under the periodic boundary condition. However, in our setting that involves both property computation and connectivity check, every design evaluation requires a 2–3 minute ground-truth simulation

to obtain impedance and stiffness. Running a graph-search for every candidate design would, critically, be non-differentiable and thus incompatible with gradient-based updates used for property optimization. In addition, while GPU implementations reported in the literature [79,80] would reduce per-instance runtime, they provide no natural mechanism for batch-level early rejection, so the cumulative overhead still becomes substantial when screening millions of candidates.

For gradient-based inverse design, we conceive a connectivity classifier that predicts whether a given voxel configuration is connected or not. Although the connectivity algorithm presented in Section 2.2 serves an on-the-fly checker for voxel connectivity, it does not inform what specific voxels of given a disconnected configuration should be updated to become connected. To enable gradient-based design update with a target impedance given the connectivity requirement, we develop an ML surrogate of the connectivity check algorithm, viewing Algorithm 1 as the ground-truth procedure to be approximated. The connectivity classifier is modeled as a 3D CNN. It performs binary classification over a voxel configuration as connected or disconnected, assigned to 0 and 1, respectively. The model consists of  $d_{conv}$  3D convolutional layers with ReLU activations for feature extraction, followed by an MLP of depth  $d_{fc}$ . Similar with the formal description on the regressors above, the classifier can involve a map  $g : \{0, 1\}^{N \times N \times N} \rightarrow [0, 1]$ , where  $g$  represents the feasibility classifier that predicts (dis)connectivity. The forward pass given an arbitrary configuration  $\mathbf{x} \in \mathcal{X}$  is recursively defined as follows:

$$\begin{aligned} v_{l+1}(\mathbf{x}) &= \sigma(W_l * v_l(\mathbf{x})), \quad l = 1, \dots, d_{conv}, \\ v_{d_{conv}+1} &= \text{Flatten}(v_{d_{conv}}(\mathbf{x})), \\ f_{l+1}(v) &= \sigma(W_l f_l(v) + b_l), \quad l = d_{conv} + 1, \dots, d_{conv} + d_{fc} - 1, \\ g(\mathbf{x}) &= \text{sigmoid}(f_{d_{conv}+d_{fc}}(v)), \end{aligned} \quad (13)$$

where  $v_l(\mathbf{x})$  represents the output of the  $l$ -th layer, each hidden layer in the MLP has width  $w_{fc} = 8$ , and  $*$  denotes the 3D convolution. The hyperparameters are set as  $(d_{conv}, d_{fc}) = (2, 2)$ , which amounts to 88,833 trainable parameters. Details of the training protocol and results are stated under Details of Model Training Protocols in Appendix.

In preparing the training data for the connectivity classifier, we need to prepare both a set containing connected instances, and another containing disconnected ones. In doing so, given a shapset  $S$  containing only connected voxel configurations, we conceive a synthetic data generation heuristic that generates a set of perturbed instances  $\tilde{S} = \{\tilde{\mathbf{x}}\}$ , each of which is close to connected solution  $\mathbf{x}$  while remaining disconnected. A pseudo-algorithm for numerical implementations of this procedure is presented in Algorithm 7 in Appendix. Details of the training data, protocol, and results are stated under Details of Model Training Protocols in Appendix.

### 2.5.3. Inverse design with automatic differentiation

With access to the trained ML regressor and classifier stated above, we present a formal description on the gradient-based inverse design that identifies voxel configurations with a targeted impedance. Given a target impedance  $Z^*$  and the regressor  $h(\mathbf{x})$  and the classifier  $g(\mathbf{x})$  trained following the procedure in Section 2.5, the inverse optimization is written:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathcal{L}(Z^*, \hat{Z}) \\ \text{s.t.} \quad & g(\mathbf{x}) = 0, \end{aligned} \quad (14)$$

where  $\hat{Z}$  is the impedance predicted by  $h(\mathbf{x})$  and  $\mathcal{L}(\cdot, \cdot)$  is the regression loss involving  $\mathcal{L} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ . In specifying the inverse problem further, it is required to relax the equality-constrained problem for its numerical implementation. Given a configuration  $\mathbf{x}$ , we build on the predicted probability of it being connected, i.e.,  $\mathbb{P}[g(\mathbf{x}) = 0]$  based on Section 2.5.2, and introduce a soft probability constraint with the threshold of violation  $p_{cut} \in [0, 1]$  with the penalty coefficient  $\lambda \in \mathbb{R}^+$ . The optimization problem reads:

$$\min_{\mathbf{x}} \quad \|Z^* - \hat{Z}\|_1 + \lambda_{\max}(0, p_{cut} - \mathbb{P}[g(\mathbf{x}) = 0]), \quad (15)$$

where  $\|Z^* - \hat{Z}\|_1 = |Z^* - \hat{Z}|$  is the L1 loss of impedance and  $p_{cut}$  is set as 0.9 throughout this work unless stated otherwise. Automatic Differentiation [76] gives access to the numerical gradients that solving Eq. (15) involves, namely  $\nabla_{\mathbf{x}} \hat{Z}$  and  $\nabla_{\mathbf{x}} \mathbb{P}[g(\mathbf{x}) = 0]$ . With both, the design update is conducted iteratively until termination criteria are met. This procedure is summarized as pseudocode in Algorithm 5.

## 3. Results and discussion

In this section, we first present the results obtained using GenUnit to find the multiobjective designs with target impedance and stiffness. Then, we proceed to the results achieved by MLMatch, an ML-driven framework whose learned feature space enables quick stiffness grading via Pairwise Controlled Manifold Approximation [81] (PaCMA) embedding without post-optimization, and also supports on-demand identification of new voxel configurations in sparsely represented target regions through gradient-based inverse design. Finally, we present demonstrative examples that are achievable using this framework. A key distinction between the two methods lies in their computational requirements. The GenUnit approach involves running a limited number of high-fidelity simulations (COMSOL), making it less computationally expensive up front. In contrast, MLMatch method requires generating and simulating a substantially larger dataset to train the surrogate model, incurring a much higher initial computational cost. However, once the ML model is trained, it offers greater flexibility and scalability in exploring the design space efficiently without further expensive simulations.

---

**Algorithm 5:** Optimize impedance Algorithm for inverse optimization given a target impedance subject to the connectivity constraint.

---

**Input:**  $Z^*$  - target impedance  $\mathbf{x}$  - initial design  $p_{cut}$  - probability threshold of connectivity  $\lambda$  - penalty parameter

**Output:**  $\mathbf{x}^*$  - optimized design

---

```

1 for  $i = 1$  to  $i_{\max}$  do
2    $\hat{Z} \leftarrow h(\mathbf{x})$ ;
3    $\hat{p} \leftarrow \mathbb{P}[g(\mathbf{x}) = 0]$ ;
4    $\nabla_z \leftarrow \nabla_{\mathbf{x}} |Z^* - \hat{Z}|$ ;
5    $\nabla_p \leftarrow -\lambda \nabla_{\mathbf{x}} (\max(0, p_{cut} - \hat{p}))$ ;
6    $\nabla_{\text{total}} \leftarrow \nabla_z + \nabla_p$ ;
7    $\mathbf{x} \leftarrow \mathbf{x} - \nabla_{\text{total}}$ ;
8 return  $\mathbf{x}^*$ ;

```

---

### 3.1. Finite element simulation framework

Each unit cell is modeled as a  $1 \text{ mm} \times 1 \text{ mm} \times 1 \text{ mm}$  cube, discretized into a  $10 \times 10 \times 10$  voxel grid. Occupied voxels (0.1 mm edge) are converted to solid subdomains meshed with structural elements ( $\sim 16^3$  elements per voxel, element size  $\approx 0.00625 \text{ mm}$ ). The base material is a metallic alloy ( $E = 100 \text{ GPa}$ ,  $\nu = 0.39$ ,  $\rho = 7850 \text{ kg/m}^3$ ), chosen to illustrate the methodology rather than a specific alloy. Effective density is recalculated for each design as the total mass over unit-cell volume.

Periodic boundary conditions are enforced on opposite faces to impose Floquet–Bloch periodicity. For dispersion calculations, eigenfrequency studies are run along  $\mathbf{k} = k_z \hat{z}$  with  $k_z a \in [0, \pi]$ . The longitudinal velocity  $c_z$  was extracted from the dispersion relation along  $k_z$  at the operating frequency of 30 kHz, by fitting the slope  $\partial\omega/\partial k_z$  of the branch. Because  $k_z a \ll 1$  for  $a = 1 \text{ mm}$  at this frequency, the branch remains nearly linear, and the extracted velocity agrees with the long-wavelength homogenized limit. Thereafter, we obtain impedance  $Z = \rho_{\text{eff}} c_z$ . For homogenization, linear static analysis is done under  $\epsilon_{33} = 0.1$ , with other average strains set to zero, allowing  $C_{33}$  to be extracted via strain-energy equivalence. Mesh sensitivity analysis is reported in [Appendix A](#).

### 3.2. Genunit optimization results

Using the GenUnit algorithm, we aim to generate designs with widely varying target stiffness values and a target impedance close to that of water (1.48 MRayl). Towards this objective, we consider an initial population of size 120 and 10 generations to progressively refine the designs to achieve the target impedance and stiffness values. [Fig. 5](#) illustrates the optimization outcomes, with the first objective always being to achieve a target impedance of 1.48 MRayl. The second objectives shown here include maximizing stiffness, minimizing stiffness, and achieving a stiffness value near specific target values such as 1 GPa and 3 GPa. The gray symbols represent the initial population and the red symbols represent the first Pareto front of the tenth generation. The progression shows significant improvement, with designs converging closer to the desired objectives.

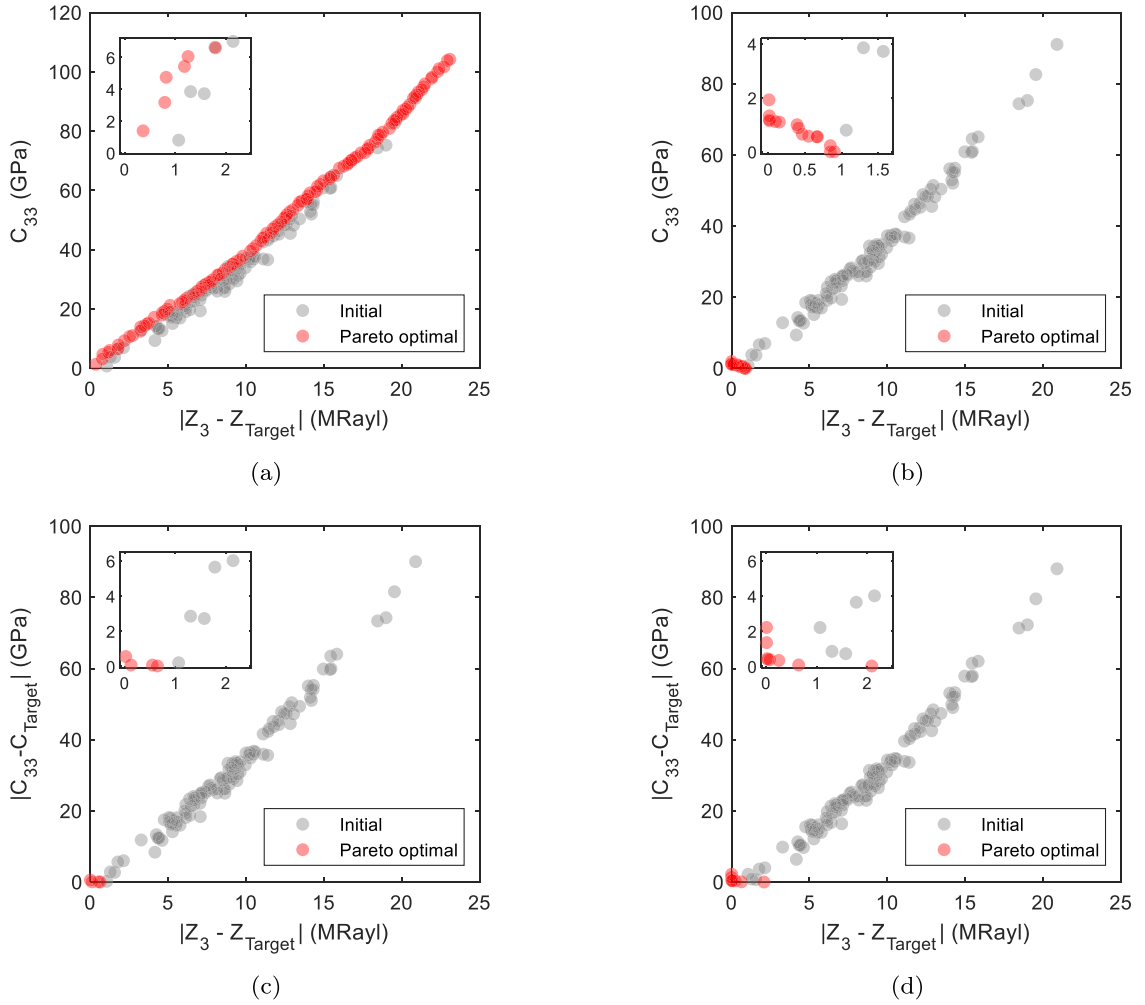
In [Fig. 5\(a\)](#), the objective is to maximize stiffness while maintaining the target impedance. Some designs match the impedance target closely (1.12 MRayl) but have lower stiffness (1.38 GPa), which is ideal for the impedance objective but not for stiffness. Other designs exhibit high stiffness values (up to 104.25 GPa) but also show elevated impedance values (24.56 MRayl). [Fig. 5\(b\)](#) presents the results of minimizing stiffness. The first Pareto front of the tenth generation includes designs with very low stiffness values (1.18 GPa) and an impedance closely matching water (1.49 MRayl), demonstrating the algorithm’s effectiveness in achieving this objective. In [Fig. 5\(c\)](#) (and [Fig. 5\(d\)](#)), the objective is to achieve a target stiffness of around 1 GPa (and 3 GPa) while maintaining the impedance near 1.48 MRayl. The first Pareto front of the tenth generation (red) shows designs that successfully meet these criteria, with one design achieving an impedance of 1.49 MRayl and a stiffness of 1.26 GPa and another design achieving an impedance of 1.46 MRayl and a stiffness of 2.73 GPa. These results indicate that the GenUnit algorithm effectively improves design optimization, converging towards the specified objectives and producing diverse solutions that balance stiffness and impedance. In [Section 3.3](#), we will use this method on a demonstration example for a gradient stiffness material with constant impedance.

### 3.3. Machine learning-based design

We demonstrate the efficacy of MLMatch through two results: (1) rapid stiffness grading via PaCMAP embedding without post-optimization; (2) discovery of new voxel configurations in target regions that are sparsely covered by the training data gradient-based inverse design.

#### 3.3.1. PaCMAP-based stiffness grading

To demonstrate the efficacy of MLMatch for stiffness grading purposes, we first use PaCMAP [\[81\]](#), a nonlinear dimensionality reduction method that balances local and global structure preservation. In our implementation, PaCMAP offers a 2D embedding of the supervised feature space, which serves as a navigable “map” for identifying stiffness grading without post-optimization. We apply PaCMAP to the intermediate 64-D feature representations  $f_{d_{\text{conv}}+2} \in \mathbb{R}^{f_c}$  extracted from the first hidden layer of the MLP in the



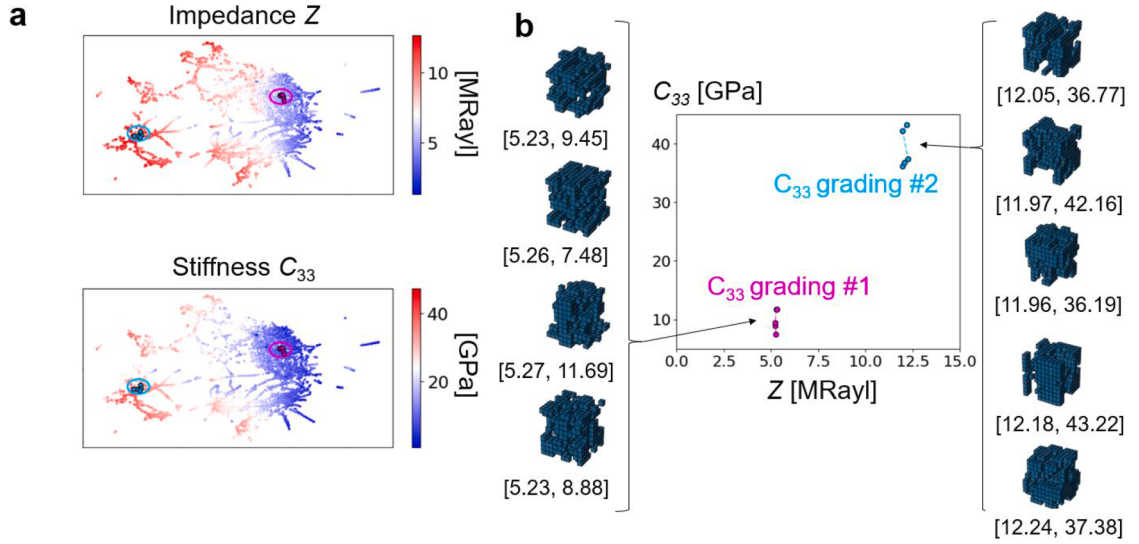
**Fig. 5.** Property space achieved by the GenUnit optimization for multiple objectives: (a) maximizing stiffness while maintaining the impedance near 1.48 MRayl, (b) minimizing stiffness while maintaining the impedance near 1.48 MRayl, (c) achieving stiffness near 1 GPa while maintaining the impedance near 1.48 MRayl, and (d) achieving stiffness near 3 GPa while maintaining the impedance near 1.48 MRayl. Gray symbols represent the initial population and red symbols represent the first Pareto front of the last generation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

trained property regressor (Section 2.5.1). The supervised features of the regressor are encoded colored by the relevant 2D material properties—impedance  $Z$  and stiffness  $C_{33}$ . Default hyperparameters were used for PaCMAP [81].

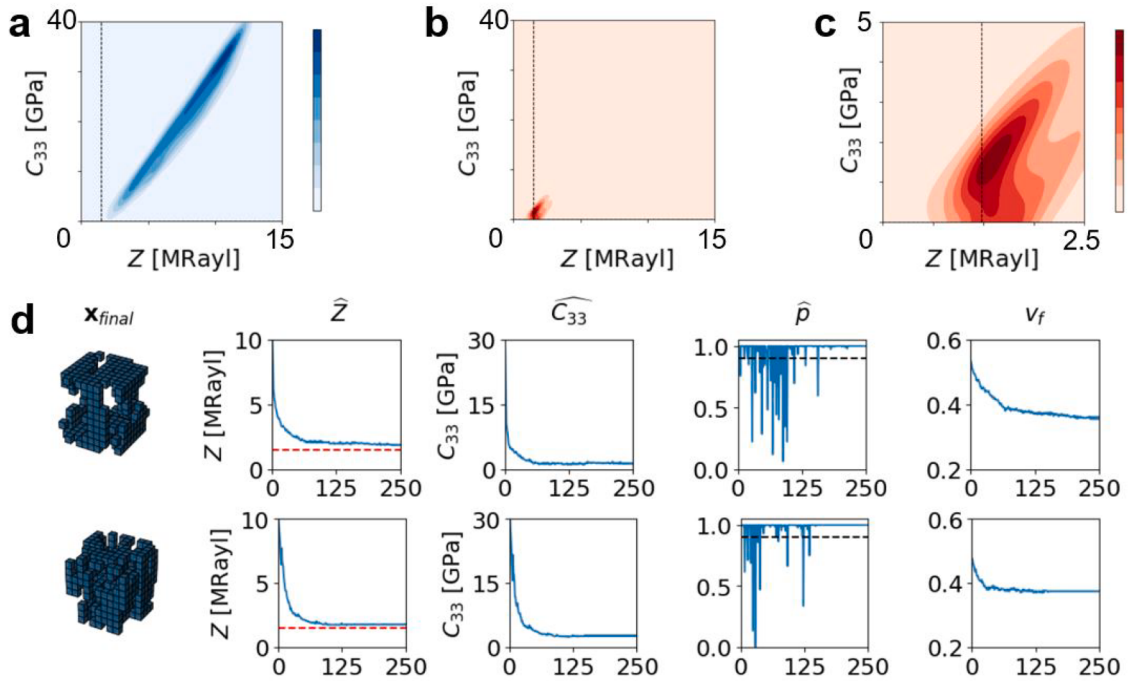
As shown in Fig. 6(a), PaCMAP produces a 2D embedding distilled from the 64-D regression features of the 18,917 training points. Overall, the two property distributions show strong distributional similarity with respect to the embeddings of unit cells, indicating a strong positive correlation between impedance and stiffness, as is physically expected [82]. Unit cells with high stiffness tend to have high impedance (red regions in the top and bottom of Fig. 6(a)), and vice versa. Interestingly, the horizontal component of the PaCMAP embedding tends to exhibit a negative correlation with both properties. Selecting a segment in the 2D PaCMAP plot enables fast exploration of stiffness grading across the dataset. In Fig. 6(b), we show two examples of stiffness variations chosen from the red and the blue regions of the PaCMAP visualization, where the target impedance  $Z$  is set to 5.23 MRayl (left column) and 12 MRayl (right column), respectively. Additional investigation could be done using PaCMAP to explore the meaning of the clusters observed in the high impedance, high stiffness regime. Here we have shown that marrying MLMatch with PaCMAP embedding enables optimization-free identification of a collection of unit cells stiffness grading for some selected target impedance values.

### 3.3.2. Gradient-based inverse design

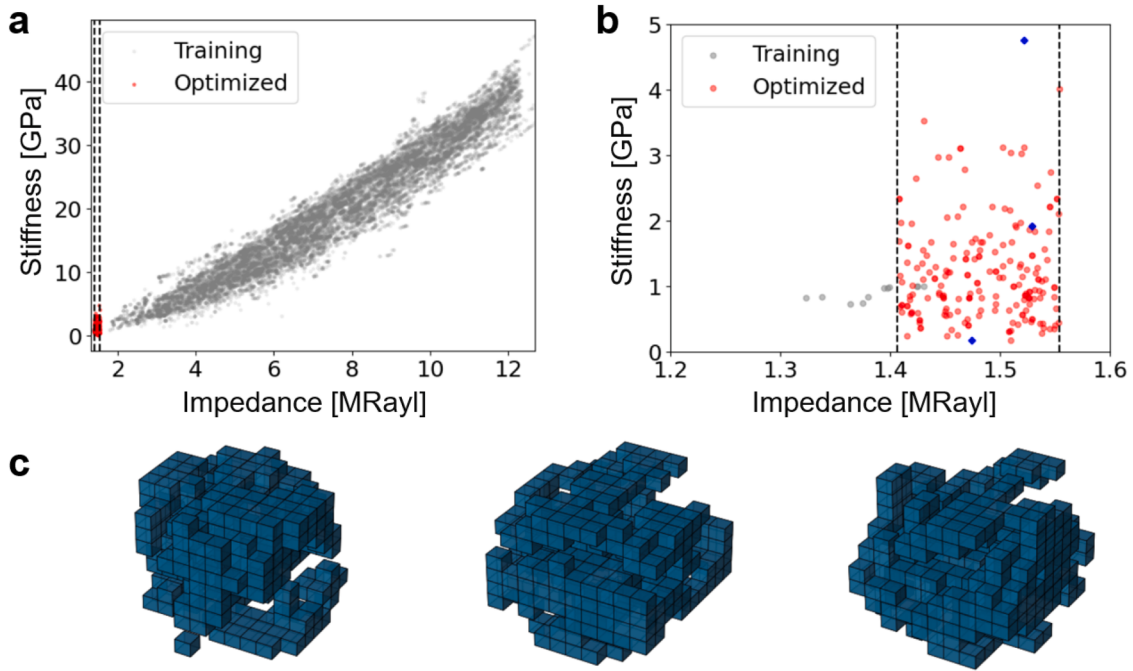
We show the inverse design results of the proposed design framework from Section 2.5 in Fig. 7. For these results, the target impedance and threshold of violation are set to  $Z^* = Z_w$  (1.48 MRayl) and  $p_{cut} = 0.9$ , respectively. The data density distribution with random initialization in Fig. 7(a) shows that a very small portion of the samples are close enough to the target (vertical black dotted line). Fig. 7(b) and (c) illustrate a distributional shift towards the target impedance after the inverse design conducted with the two



**Fig. 6.** Latent vectors from stiffness neural network embeddings displayed by PaCMAP. Axes are abstract, arising from PaCMAP. (a) The 2D PaCMAP feature distributions applied to the 64D feature of the trained property regressor: (top) colored by impedance and (bottom) colored by stiffness. (b) Illustration of the stiffness grading achievable from the two clusters, each marked as a circle in (a). For each stiffness grading, the target impedance  $Z$  is set to 5.23 MRayl (left) and 12 MRayl (right), respectively. Randomly selected unit cells within each circle are displayed with the corresponding ground-truth property  $Z$  and  $C_{33}$ , with units of [MRayl] and [GPa], respectively.



**Fig. 7.** MLMatch inverse design results. (a)-(c) Illustration of a shift of property distribution before and after the inverse design. The optimization is tested for 300 different starting designs randomly sampled from the training dataset. Black dotted lines denote the target impedance  $Z^*$ . (a) The density distribution of the random initial set, (b) optimized set, and (c) its zoom-in view, all of which are estimated through kernel density estimation. (d) Optimization history of two runs. From left to right, each column displays optimized voxel configurations, impedance, stiffness, probability of connectivity, and volume fraction. The red dotted line denotes the target impedance, while the black one denotes the probability threshold  $p_{cut} = 0.9$ . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



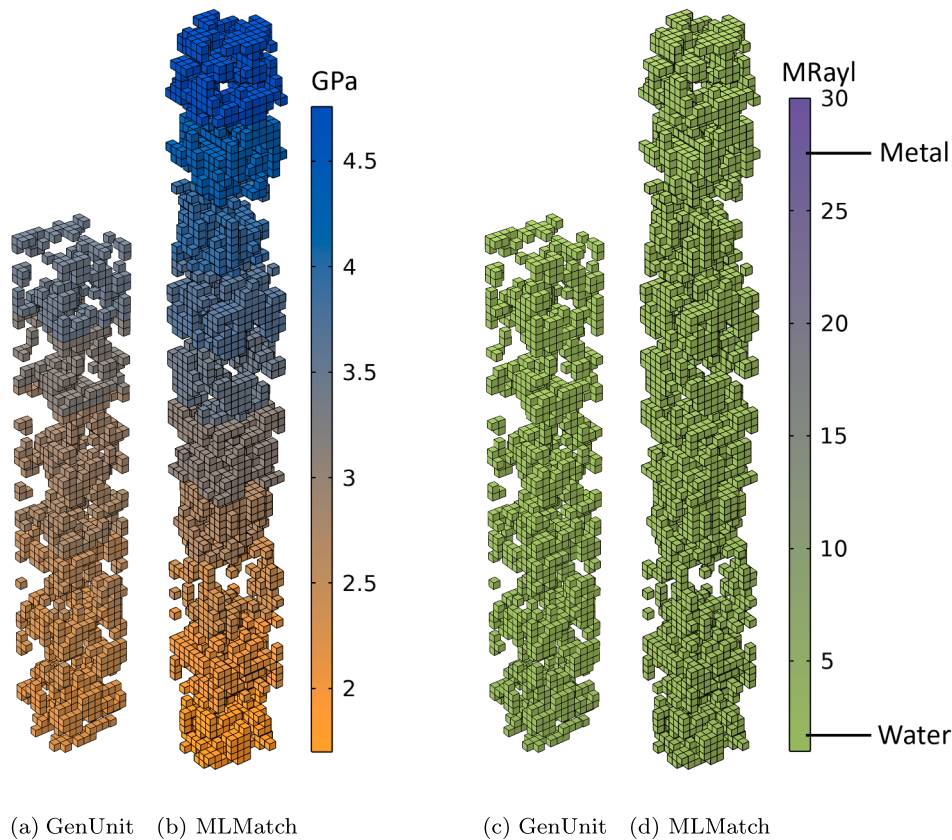
**Fig. 8.** The resulting  $Z - C_{33}$  data distribution after inverse design. The subregion formed between the black dotted lines in (a) and (b) is the target region  $\Omega^* = Z_{\text{water}}[0.95, 1.05] \times [0, \infty)$ . (a) The data distribution after the inverse design (red), taking the training data as a background (gray with 0.05 transparency). (b) A zoomed-in view centered on the target region with the maximum relative error of impedance set as 5% in the raw scale. Note the 3 original samples that meet the impedance error threshold in gray are very close together with nearly identical stiffness (gray in  $\Omega^*$ ). (c) Three selected examples of optimized design (blue dots in (b)). The resulting property ( $Z, C_{33}$ ) from left to right based the wave analysis in COMSOL reads (1.474 MRayl, 0.175 GPa), (1.529 MRayl, 1.921 GPa), (1.522 MRayl, 4.762 GPa), respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ML surrogates in Section 2.5.1, i.e., the property regressor  $h(\mathbf{x})$  and the connectivity classifier  $g(\mathbf{x})$  and the gradient-based solution procedure in Algorithm 5 that runs 250 iterations for each optimization run. Fig. 7(d) shows two optimization runs. At early stages of each run, the connectivity constraint tends to be active, while it becomes less so as the optimization proceeds. As the inverse design is driven by the gradients offered by Automatic Differentiation, all the results are subject to local optimality, divergence, and infeasibility. The intrinsic limitations of gradient-based optimization are acceptable in the proposed inverse design; it builds on the ML surrogates that provide on-the-fly predictions, thus gracefully scales with the number of initial starting points and the maximum number of iterations per optimization run.

The distribution of each property after optimization is reported in Fig. 8. The target subdomain in the  $Z - C_{33}$  property space is specified as  $\Omega^* = Z_{\text{water}}[0.95, 1.05] \times [0, \infty)$ , whose boundary with respect to  $Z$  is shown as the black lines in Fig. 8(a) and (b). Within the original 18,917-size dataset including both training/test splits, only three samples (0.016%) meet the 5% relative error threshold of impedance and these three samples (gray within the target region Fig. 8(b)) are nearly identical in stiffness with a value of around 1 GPa. Our design framework identifies 196 new designs within  $\Omega^*$  while addressing the connectivity requirement. The resulting stiffness  $C_{33}$  coverage spans a large range of [0.175, 4.762] GPa, which facilitates the goal of building an acoustic metamaterial array with functional grading for tailored stiffness distributions  $C_{33}$ , while maintaining impedance within 5% of  $Z_w$  and geometric connectivity. The newly discovered designs show a rich spread across the target region, including examples of high stiffness ( $C_{33} > 3$  GPa). The distribution of the results tends to be denser in the low elasticity region, reflecting the strong positive correlation between elasticity and impedance in our unit cell representation [82]. This expanded coverage facilitates the construction of stiffness grading with impedance matching; see Section 3.4 for a demonstrative case study.

### 3.4. Demonstrative example

To demonstrate the efficacy of our framework, we present two examples of optimized designs both featuring a gradient stiffness profile and constant impedance matching that of water. The chosen design is a column with stiffness varying linearly along its length, with the highest stiffness at the top and the lowest stiffness at the bottom. To achieve this design, we choose an array of unit cells from each of the two optimization methods to meet the objectives. We then simulate the response of each design in COMSOL and to account for periodicity in all three directions, we repeat each unit cell four times in the vertical direction and for the other two directions we set the periodic boundary condition. We systematically select unit cells for the first example design from the Pareto-optimal solutions generated by GenUnit, while for the second example design, we use the MLMatch algorithm. We choose the stiffness gradient profile



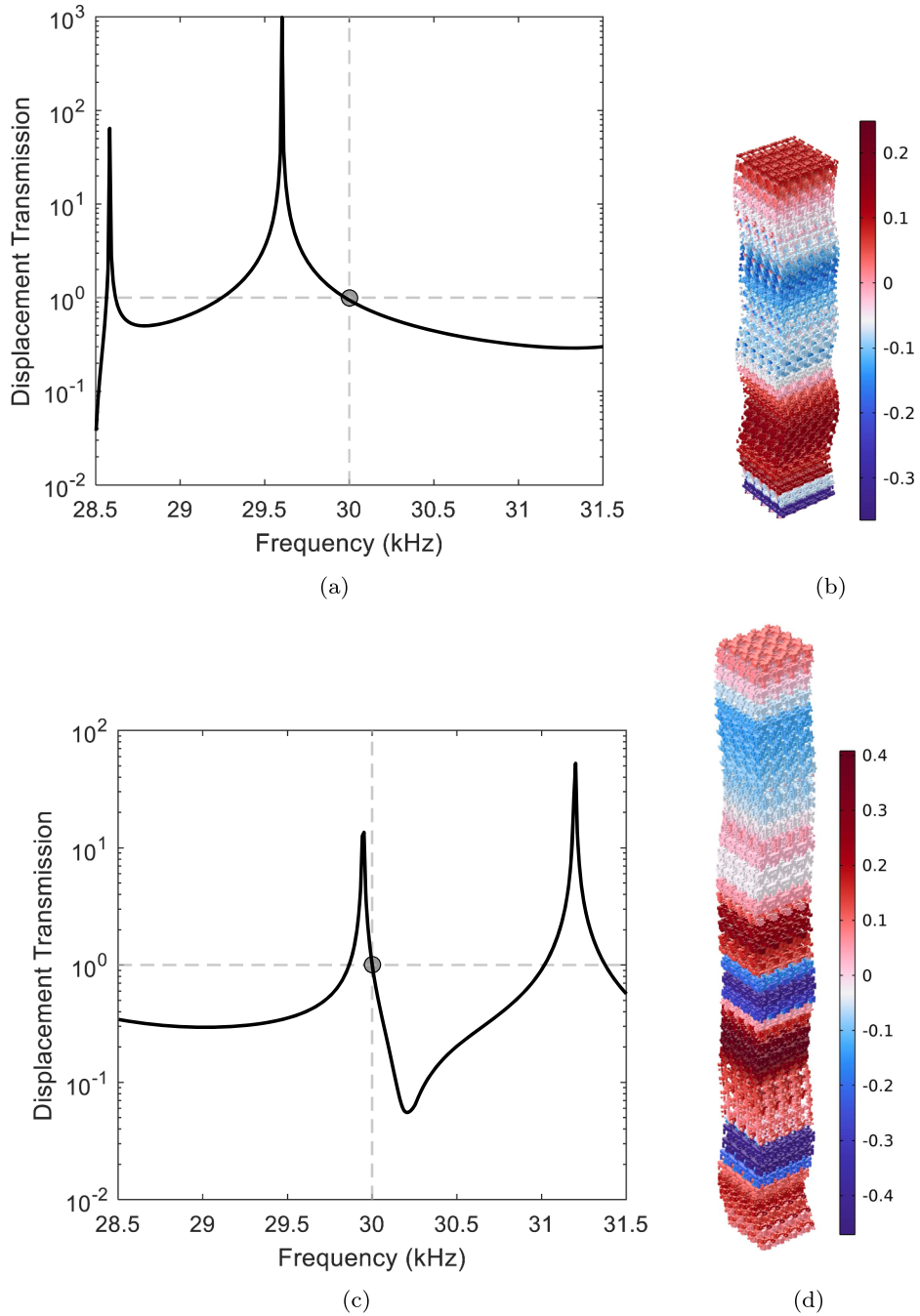
**Fig. 9.** Illustration of design properties generated by GenUnit and MLMatch. (a) and (b) show the stiffness gradients for GenUnit and MLMatch designs, respectively. (c) and (d) show the impedance distributions for the same designs. Notably, both designs exhibit impedance values that closely resemble that of water, despite being composed of metallic alloy. The GenUnit structure is more compact but requires longer inference time per design due to its iterative nature, whereas MLMatch demands substantial upfront training but enables rapid retrieval of design values thereafter.

based on the maximum and minimum values achieved in our optimal solutions in the previous section, ensuring the closest match for impedance.

Using GenUnit, we set various target values for stiffness and select the optimal designs from the Pareto front. In contrast, MLMatch provides a large library of designs from which we select those that closest meet our target impedance and stiffness criteria. The larger computational overhead of the MLMatch approach results in greater flexibility to replicate the gradient stiffness. Consequently, the range of stiffness values achieved using MLMatch is almost double that obtained with GenUnit and the gradient increments are finer. Fig. 9 illustrates two representative designs generated using GenUnit and MLMatch. Details of the properties are listed in Table 1. Both methods achieve close impedance matching with water and a significant stiffness gradient.

In Fig. 10, we illustrate the displacement color map at 30 kHz and the transmission ratio (the ratio of the applied displacement at the top boundary to the displacement received at the bottom surface) for designs generated using GenUnit and MLMatch. The transmission plots demonstrate effective displacement transmission from the top to the bottom of the designs, with values approximating 1 for both GenUnit and MLMatch designs, approaching near perfect wave transmission, in spite of the large stiffness variation. This effective transmission further confirms the structural connectivity between adjacent unit cells, allowing seamless propagation of displacement through the entire design. The observed peaks in the transmission plots are attributed to structural resonances that naturally occur due to the finite size of the design. To the best of our knowledge, this is the first demonstration of a metallic metamaterial with both a large stiffness gradient and near-perfect acoustic wave transmission—an achievement that opens new avenues for medical device design and other precision acoustic applications.

MLMatch clearly enables broader exploration and finer resolution of the stiffness gradient while both methods achieve tight impedance control. The resulting stiffness gradient is 50 % larger for the MLMatch approach. At the same time, the MLMatch approach requires a large up-front computational investment for model training. Specifically, training the MLMatch surrogate model required nearly 20,000 unit cell designs simulated in COMSOL. In contrast, the GenUnit method started with ~100 initial designs with offspring totaling approximately 5000 simulations. The four-fold decrease of computation costs highlights an advantage of GenUnit in scenarios with limited computational resources, as it avoids the up-front cost of large-scale dataset generation while still producing viable designs.



**Fig. 10.** Displacement transmission ratio ( $|w_{\text{bottom face}}|/|w_{\text{top face}}|$ ) versus frequency for the demonstrator designed using (a) GenUnit and (c) ML-Match, along with the color map of displacement along the z-axis for the structure designed using (b) GenUnit and (d) MLMatch at 30 kHz. With matched impedance between the unit cell designs, the ideal displacement transmission ratio is 1, with values exceeding 1 at resonance frequencies. The gray circle highlights the specific operating condition used as the optimization target.

**Table 1**

Detail results of the demonstrative examples designed by GenUnit and ML-Match with a target impedance of 1.48 throughout and as large of a gradient in stiffness as possible.

| GenUnit           |                 | Machine Learning  |                 |
|-------------------|-----------------|-------------------|-----------------|
| Impedance (MRayl) | Stiffness (GPa) | Impedance (MRayl) | Stiffness (GPa) |
| 1.54              | 0.97            | 1.51              | 0.17            |
| 1.48              | 1.20            | 1.41              | 0.68            |
| 1.55              | 1.46            | 1.53              | 1.19            |
| 1.44              | 1.80            | 1.41              | 1.69            |
| 1.49              | 1.92            | 1.49              | 2.22            |
| 1.42              | 2.33            | 1.54              | 2.74            |
| 1.46              | 2.74            | 1.50              | 3.12            |
|                   |                 | 1.43              | 3.52            |
|                   |                 | 1.55              | 4.02            |
|                   |                 | 1.52              | 4.76            |

These results highlight a fundamental trade-off between training time, inference efficiency, and design flexibility. MLMatch requires a significant upfront computational investment for training but enables rapid evaluations and precise control over property gradients during inference. In contrast, GenUnit eliminates the need for surrogate model training and directly interfaces with finite element simulations, offering higher fidelity results and lower implementation overhead—albeit at the cost of slower, design-by-design inference. Importantly, the strengths of both approaches are complementary and can be leveraged in tandem. MLMatch is well-suited for global exploration of the design space, efficiently identifying promising regions across a wide domain. Once such candidates are located, GenUnit can be employed to perform high-accuracy local refinement, benefiting from its direct use of physics-based simulations. This hybrid strategy enables scalable, yet precise design synthesis by combining the broad search capability of MLMatch with the accuracy and simulation grounding of GenUnit.

#### 4. Conclusion

We presented a design framework of metamaterials with tailored acoustic impedance and stiffness gradients. The framework builds on the generation of unit cells through a periodic covariance function. The approach demonstrates optimization of the designs for target properties using two different multiobjective optimization methodologies. Specifically, we propose two algorithms: GenUnit, based on a non-dominated sorting genetic algorithm II (NSGA-II), and MLMatch, which employs differentiable machine learning techniques. These algorithms are designed to work independently or in tandem. The overall framework benefits from the embedded algorithm that ensures inter-connectivity of the voxels in each design, thereby guaranteeing manufacturability. The efficacy of the approach was demonstrated for a demonstration case study in which we generated an on-demand dataset of manufacturable metallic metamaterial designs exhibiting the impedance of water while following a gradient stiffness pattern.

The proposed framework is a novel contribution to the literature on acoustic impedance matching. Unlike previous approaches, which relied heavily on human intuition and were subject to trial-and-error and limited design freedom, this framework does not require human intuition or intervention in the design process. Additionally, it is capable of achieving impedance matching with various impedance values. In our case study, we showcased a unique and interesting example: impedance matching with water using graded stiffness metallic metamaterials. MLMatch involves an upfront training cost but enables rapid inference across a wide design domain, making it ideal for global exploration of candidate designs. In contrast, GenUnit requires no training and performs direct physics-based simulations, offering higher accuracy at the expense of longer inference time per design. When used in tandem, MLMatch can efficiently identify promising regions of the design space, which can then be locally refined by GenUnit to generate high-fidelity designs that simultaneously match target impedance and stiffness. Given the authenticity of our framework's design generation process, it can be applied to the exploration of other wave-based metamaterials for various mechanical and acoustic properties.

#### CRedit authorship contribution statement

**Rayehe Karimi Mahabadi:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Doksoo Lee:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Alexander C. Ogren:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Data curation, Conceptualization; **Chiara Daraio:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization; **Wei Chen:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization; **Cynthia Rudin:** Writing – review & editing, Methodology, Funding acquisition, Conceptualization; **L. Catherine Brinson:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

## Data availability

The code and data can be found at this GitHub repository: <https://github.com/RayeheKM/Acoustic-Metamaterial-Impedance-Matching>.

## Declaration of competing interest

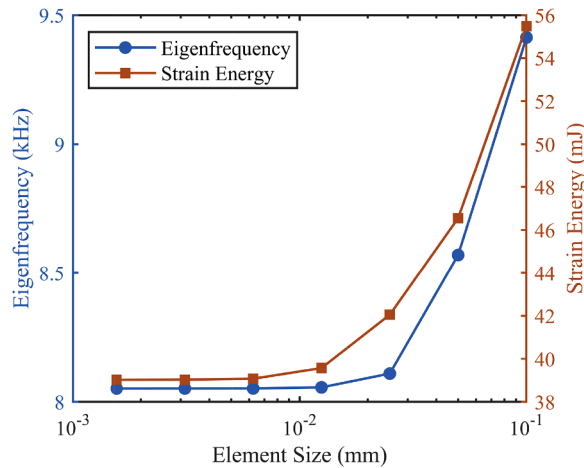
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

Rayehe Karimi Mahabadi, Alexander C. Ogren, Chiara Daraio, Cynthia Rudin, and L. Catherine Brinson gratefully acknowledge support from the U.S. Department of Energy under Grant No. DE-SC0021358. Wei Chen and Doksoo Lee acknowledge support from the NSF Boosting Research Ideas for Transformative and Equitable Advances in Engineering (BRITE) Fellow Program (CMMI 2227641). L. Catherine Brinson and Cynthia Rudin acknowledge support from the US National Science Foundation under Grant No. DGE-2022040.

## Appendix A. Mesh sensitivity of the FEM model

To verify the adequacy of the discretization, we conducted a mesh sensitivity study for both the static analysis (used to compute stiffness) and the eigenfrequency analysis (used to compute impedance). The initial mesh corresponded to one element per voxel, and successive refinements were obtained by halving the element edge size at each step. In the static analysis, the total strain energy  $U$  under a prescribed strain  $\bar{\epsilon}_{33} = 0.1$  was computed and used to evaluate  $C_{33}$ . As shown in Fig. A.11,  $U$  converges rapidly as the mesh is refined, with changes below a few percent between the finest levels. In the dispersion analysis, we tracked the eigenfrequency of the longitudinal branch at a small Bloch wavenumber ( $k_z a = \pi/100$ ). Fig. A.11 shows the eigenfrequency versus element size, again demonstrating convergence with differences below a few percent at the finest levels. These results confirm that the adopted mesh (element size  $\approx 0.00625$  mm) is sufficient for both analyses.



**Fig. A.11.** Mesh sensitivity analysis showing convergence of the static (strain energy  $U$ ) and eigenfrequency analyses with mesh refinement. Results demonstrate stable predictions for element sizes smaller than 0.00625 mm.

## Appendix B. Connectivity Check: Neighbor-Finding Algorithm

This section provides the auxiliary algorithm used within the connectivity check. The routine identifies all new face-to-face neighboring voxels of a given connected group, accounting for periodic boundary conditions at the unit cell boundaries. It is called iteratively to grow connected components.

---

**Algorithm 6:** Get neighbors Function to get all *new* neighboring material voxels of a given voxel group.

---

**Input:**  $G_c$  - set of voxel indices,  $D$  - 3-dimensional boolean design array

**Output:**  $G_{c-new}$  - voxel indices that contain material, are neighbors of  $G_c$ , but are not in  $G_c$

```

1  $N = \text{empty}$  % Initialize set of neighbors
2 foreach  $i \in G_c$  do
3    $N = N \cup \{\text{all 6 neighboring indices of } i\}$  % Periodicity is considered for boundary indices, i.e., along each dimension
   the  $N^{th}$  voxel and 1st voxel are neighbors
4  $N = N \setminus \{N \text{ where } D(N) == 0\}$  % Filter out neighbors that do not contain material
5  $N = N \setminus G_c$  % Filter out neighbors that were part of the input group
6  $G_{c-new} = \text{neighbors}$ 

```

---

## Appendix C. Synthetic Shapeset Generation for the Classifier

We start by creating a 3D Gaussian random field  $\bar{G}$  with a radial frequency grid  $\kappa$  and computing the corresponding power spectrum  $P(\kappa)$  specified in Fourier space. We then apply random phases  $\Phi$  to introduce variability, and subsequently perform an inverse FFT to transform the data into real space. Normalization follows to ensure that the resulting field  $\bar{G}$  achieves the desired standard deviation  $\sigma$ . This field is used to perturb the initial state  $\mathbf{x}$ , yielding a new field that approximates connected solutions while maintaining a degree of infeasibility.

---

**Algorithm 7:** Perturb voxels GRF Pseudo-algorithm to produce a disconnected set of perturbed voxel configurations from a connected set.

---

**Input:**  $S$  - connected set of voxel configurations  $\alpha = 0.2$  - scale parameter  $\sigma = 0.25$  - standard deviation  $N$  - predefined number of instances per configuration

**Output:**  $\tilde{S}$  - disconnected set of perturbed periodic states

```

1  $\tilde{S} \leftarrow \emptyset$ ;
2 foreach  $\mathbf{x} \in S$  do
3   while  $|\tilde{S}| < N$  do
4      $k_x, k_y, k_z \in \mathbb{R}^{n \times n \times n}$ ;
5      $\kappa \leftarrow \sqrt{k_x^2 + k_y^2 + k_z^2}$  with  $\kappa(0, 0, 0) = 1$ ;
6      $P(\kappa) \leftarrow \exp\left(-\frac{\alpha \cdot \kappa^2}{2}\right)$ ;
7      $\Phi \leftarrow \exp(2\pi i \cdot \mathbb{U})$ , where  $\mathbb{U} \sim U[0, 1]$ ;
8      $\mathbb{F} \leftarrow P(\kappa) \cdot \Phi$ ;
9      $\mathcal{G} \leftarrow \mathcal{F}^{-1}(\mathbb{F})$ ;
10     $\bar{G} \leftarrow \frac{\mathcal{G} - \mathbb{E}[\mathcal{G}]}{\sigma(\mathcal{G})} \cdot \sigma$ ;
11     $\bar{\mathbf{x}} \leftarrow \begin{cases} 1 & \text{if } \mathbf{x} + \bar{G} \geq 0.5; \\ 0 & \text{otherwise} \end{cases}$ ;
12    Enforce face-to-face periodicity at the boundary faces along  $x$ ,  $y$ , and  $z$ ;
13    if  $\bar{\mathbf{x}}$  is disconnected (Algs. 1 and 2) then
14       $\tilde{S} \leftarrow \tilde{S} \cup \{\bar{\mathbf{x}}\}$ ;

```

---

## Appendix D. Details of Model Training Protocols

**Impedance Regressors.** The regressor learns a map  $h : \{0, 1\}^{N^3} \rightarrow \mathbb{R}^2$  through a 3D CNN. For the regressor training the L1 loss is employed as follows:

$$\mathcal{L}(\theta) = \frac{1}{|B|} \sum_{i \in B} \|h(\mathbf{x}_i; \theta) - \mathbf{y}_i\|_{L1}, \quad (\text{D.1})$$

where  $\mathbf{x}_i \in \{0, 1\}^{N \times N \times N}$  is the input voxel configuration,  $\mathbf{y}_i \in \mathbb{R}^2$  is the normalized property being either impedance or stiffness, and  $B$  is the mini-batch of size 128. The model is trained through Adam [83] with a learning rate of  $\eta_0 = 10^{-3}$  over 250 epochs. Based on a grid search on model complexity, we set  $(d_{conv}, d_{fc}) = (2, 8)$ , while each hidden layer in the MLP has width  $w_{fc} = 64$ . This amounts to number of trainable parameters is 4,177,410. Fig. D.12(a) and (b) illustrates the training history. As the regression data we generated 18,917-size labeled data following Section 2.1, with the train-test split ratio being 0.8:0.2. With the stated configuration and the data, the training results in  $\mathcal{L}_{train}(\theta^*) = 0.0144$  and  $\mathcal{L}_{test}(\theta^*) = 0.0290$ .

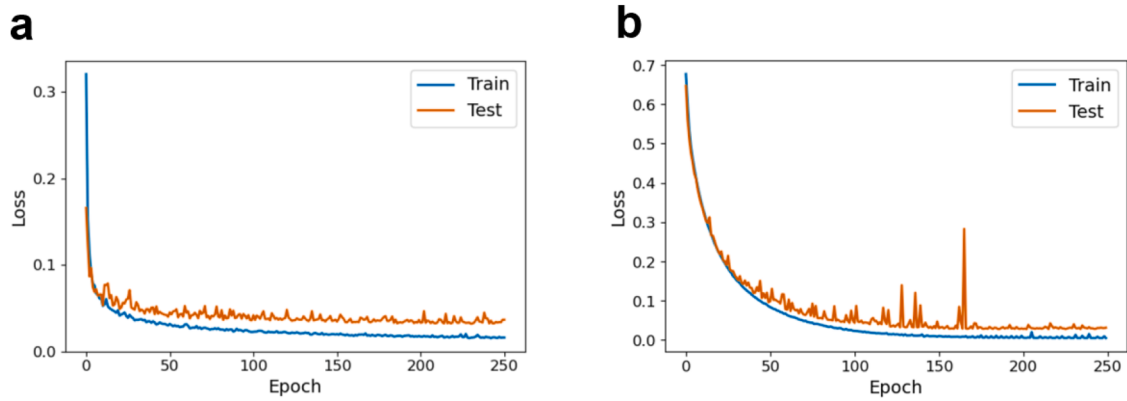


Fig. D.12. Training history of ML models in Section 2.5. (a) The impedance regressor  $h$ . (b) The connectivity classifier  $g$ .

**Connectivity Classifier.** The connectivity classifier  $g : \{0, 1\}^{N^3} \rightarrow [0, 1]$  is trained on a dataset consisting of the connected voxel configurations  $S$  where  $y = 0$  and the disconnected set  $\tilde{S}$  where  $y = 1$ , synthesized following Section 2.1 and Algorithm 7, respectively. Both sets include 86,410 designs, forming  $S \cup \tilde{S}$  of data size 172,820. Although this is a fairly large amount of data, note that the forward call of the connectivity checker algorithm (Algorithm 1) is near-instantaneous, so the resources incurred for the data synthesis are orders of magnitude lower than those for the regression data. Given the two synthetic datasets, the binary cross-entropy training loss reads:

$$\mathcal{L}(\theta) = -\frac{1}{|B|} \sum_{i \in B} [y_i \log \sigma(g(\mathbf{x}_i)) + (1 - y_i) \log(1 - \sigma(g(\mathbf{x}_i)))], \quad (\text{D.2})$$

where  $\mathbf{x}_i \in \{0, 1\}^{N \times N \times N}$  is the voxelized input state,  $B$  is a mini-batch of size  $|B| = 128$ ,  $y_i \in \{0, 1\}$  is the classification label (connected  $\rightarrow 0$ ; disconnected  $\rightarrow 1$ ), and  $\sigma(\cdot)$  is the sigmoid activation. The trainable model parameters are updated using the Adam optimizer [83]. Initial learning rate is set as  $\eta_0 = 10^{-4}$ , which is scheduled at epoch  $t$  as  $\eta_t = \eta_0 \gamma^{[t/T]}$  with decay rate  $\gamma = 0.9$  and step size  $T = 100$ . The training is conducted for 250 epochs. Each hidden layer in the MLP has width  $w_{fc} = 8$ . A grid search over model complexity was conducted for  $(d_{conv}, d_{fc})$ , empirically identifying  $(d_{conv}, d_{fc}) = (2, 2)$  with the number of trainable parameters being 131,713 as the best given the synthesized dataset  $S \cup \tilde{S}$  with train-test split of ratio 0.8:0.2. Fig. D.12(c) illustrates the training history. The training results in  $\mathcal{L}_{train}(\theta^*) = 0.0077$  and  $\mathcal{L}_{test}(\theta^*) = 0.0210$ .

## References

- [1] K. Zhu, J. Ma, X. Qi, B. Shen, Y. Liu, E. Sun, R. Zhang, Enhancement of ultrasonic transducer bandwidth by acoustic impedance gradient matching layer, *Sensors* 22 (20) (2022) 8025.
- [2] B. Altun, I. Demirkan, E.O. Isik, O. Kocaturk, M.B. Unlu, B. Garipcan, Acoustic impedance measurement of tissue mimicking materials by using scanning acoustic microscopy, *Ultrasonics* 110 (2021) 106274.
- [3] N. Hiremath, V. Kumar, N. Motahari, D. Shukla, An overview of acoustic impedance measurement techniques and future prospects, *Metrology* 1 (1) (2021) 17–38.
- [4] C.M. Park, S.H. Lee, Zero-reflection acoustic metamaterial with a negative refractive index, *Sci. Rep.* 9 (1) (2019) 3372.
- [5] C. Liu, J. Luo, Y. Lai, Acoustic metamaterials with broadband and wide-angle impedance matching, *Physical Review Materials* 2 (4) (2018) 045201.
- [6] J.P. Maxfield, H.C. Harrison, Methods of high quality recording and reproducing of music and speech based on telephone research, *Trans. Am. Inst. Electr. Eng.* 45 (1926) 334–348.
- [7] E. Dong, Z. Song, Y. Zhang, S. Ghaffari Mosanenzadeh, Q. He, X. Zhao, N.X. Fang, Bioinspired metagel with broadband tunable impedance matching, *Sci. Adv.* 6 (44) (2020) eabb3641.
- [8] Z. Li, D.-Q. Yang, S.-L. Liu, S.-Y. Yu, M.-H. Lu, J. Zhu, S.-T. Zhang, M.-W. Zhu, X.-S. Guo, H.-D. Wu, et al., Broadband gradient impedance matching using an acoustic metamaterial for ultrasonic transducers, *Sci. Rep.* 7 (1) (2017) 42863.
- [9] S. Saffar, A. Abdullah, Determination of acoustic impedances of multi matching layers for narrowband ultrasonic airborne transducers at frequencies < 2.5 MHz—Application of a genetic algorithm, *Ultrasonics* 52 (1) (2012) 169–185.
- [10] S. Qu, N. Gao, A. Tinel, B. Morvan, V. Romero-García, J.-P. Groby, P. Sheng, Underwater metamaterial absorber with impedance-matched composite, *Sci. Adv.* 8 (20) (2022) eabm4206.
- [11] L. Bai, H.Y. Dong, G.Y. Song, Q. Cheng, B. Huang, W.X. Jiang, T.J. Cui, Impedance-Matching wavefront-Transformation lens based on acoustic metamaterials, *Adv. Mater. Technol.* 3 (11) (2018) 1800064.
- [12] R. Al Jahdali, Y. Wu, High transmission acoustic focusing by impedance-matched acoustic meta-surfaces, *Appl. Phys. Lett.* 108 (3) (2016).
- [13] H. Tanaka, S. Machida, Super broadband acoustic impedance matching by MUT-type acoustic metamaterial, *Electron. Commun. Jpn.* 103 (1–4) (2020) 38–47.
- [14] S.A. Cummer, J. Christensen, A. Alù, Controlling sound with acoustic metamaterials, *Nat. Rev. Mater.* 1 (3) (2016) 1–13.
- [15] G. Ma, P. Sheng, Acoustic metamaterials: from local resonances to broad horizons, *Sci Adv* 2 (2) (2016) e1501595.
- [16] G. Yan, Y. Li, W. Li, J. Yan, S. Yao, X. Huang, Floating projection topology optimization framework for efficient design of bi-connected 3D acoustic metamaterials, *Comput. Methods Appl. Mech. Eng.* 441 (2025) 118020.
- [17] R. Karimi Mahabadi, T. Goudarzi, R. Fleury, R. Naghdabadi, Multifunctional hyperelastic structured surface for tunable and switchable transparency, *Appl. Sci.* 11 (5) (2021) 2255.
- [18] H. Zhang, R.K. Mahabadi, C. Rudin, J. Guilleminot, L.C. Brinson, Uncertainty quantification of acoustic metamaterial bandgaps with stochastic material properties and geometric defects, *Comput. Struct.* 305 (2024) 107511.
- [19] R.K. Mahabadi, T. Goudarzi, R. Fleury, S. Sohrabpour, R. Naghdabadi, Effects of resonator geometry and substrate stiffness on the tunability of a deformable microwave metasurface, *AEU-Int. J. Electron. Commun.* 146 (2022) 154123.

- [20] X. Yu, J. Zhou, H. Liang, Z. Jiang, L. Wu, Mechanical metamaterials associated with stiffness, rigidity and compressibility: a brief review, *Prog. Mater. Sci.* 94 (2018) 114–173.
- [21] M.V. Bastawrous, Z. Chen, A.C. Ogren, C. Daraio, C. Rudin, L.C. Brinson, A multiscale design method using interpretable machine learning for phononic materials with closely interacting scales, *Comput. Methods Appl. Mech. Eng.* 440 (2025) 117833.
- [22] J. Mei, Y. Wu, Controllable transmission and total reflection through an impedance-matched acoustic metasurface, *New J. Phys.* 16 (12) (2014) 123007.
- [23] M. Oudich, N.J. Gerard, Y. Deng, Y. Jing, Tailoring structure-borne sound through bandgap engineering in phononic crystals and metamaterials: a comprehensive review, *Adv. Funct. Mater.* 33 (2) (2023) 2206309.
- [24] N. Kaina, F. Lemoult, M. Fink, G. Lerosey, Negative refractive index and acoustic superlens from multiple scattering in single negative metamaterials, *Nature* 525 (7567) (2015) 77–81.
- [25] B.-I. Popa, S.A. Cummer, Non-reciprocal and highly nonlinear active acoustic metamaterials, *Nat. Commun.* 5 (1) (2014) 3398.
- [26] H. Chen, C.T. Chan, Acoustic cloaking and transformation acoustics, *J. Phys. D Appl. Phys.* 43 (11) (2010) 113001.
- [27] D. Li, L. Zigoneanu, B.-I. Popa, S.A. Cummer, Design of an acoustic metamaterial lens using genetic algorithms, *J. Acoust. Soc. Am.* 132 (4) (2012) 2823–2833.
- [28] Z. Li, Y. Sun, G. Wu, M. Tao, Inverse design of broadband acoustic metasurfaces for reflective wavefront modulation through the topology optimization method, *Appl. Acoust.* 204 (2023) 109247.
- [29] Y. Lu, Y. Wang, Structural optimization of metamaterials based on periodic surface modeling, *Comput. Methods Appl. Mech. Eng.* 395 (2022) 115057.
- [30] C. Sharpe, C.C. Seepersad, S. Watts, D. Tortorelli, Design of mechanical metamaterials via constrained bayesian optimization, in: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 51753, American Society of Mechanical Engineers, 2018, p. V02AT03A029.
- [31] A. Sakurai, K. Yada, T. Simomura, S. Ju, M. Kashiwagi, H. Okada, T. Nagao, K. Tsuda, J. Shiomi, Ultranarrow-band wavelength-selective thermal emission with aperiodic multilayered metamaterials designed by bayesian optimization, *ACS Cent. Sci.* 5 (2) (2019) 319–326.
- [32] M.A. Bessa, P. Glowacki, M. Houlder, Bayesian machine learning in metamaterial design: fragile becomes supercompressible, *Adv. Mater.* 31 (48) (2019) 1904845.
- [33] A. Raßloff, P. Seibert, K.A. Kalina, M. Kästner, Inverse design of spinodoid structures using bayesian optimization, *Comput. Mech.* (2025) 1–22.
- [34] O. Sigmund, J. Søndergaard Jensen, Systematic design of phononic band-gap materials and structures by topology optimization, *Philos. Trans. R. Soc. Lond. Ser. A* 361 (1806) (2003) 1001–1019.
- [35] F. Hadizadeh, W. Mallik, R.K. Jaiman, A Graph Neural Network Surrogate Model for Multi-Objective Fluid-Acoustic Shape Optimization, (2024). [arXiv:2412.16817](https://arxiv.org/abs/2412.16817)
- [36] W. Xu, J. Ning, Z. Lin, W. Qi, H. Liu, W. Wang, Multi-objective topology optimization of two-dimensional multi-phase microstructure phononic crystals, *Mater. Today Commun.* 22 (2020) 100801.
- [37] Y. Noguchi, K. Matsushima, T. Yamada, Level set-based topology optimization for the design of labyrinthine acoustic metamaterials, *Mater. Des.* 219 (2022) 110832.
- [38] Y. Noguchi, T. Yamada, K. Izui, S. Nishiwaki, Topology optimization for hyperbolic acoustic metamaterials using a high-frequency homogenization method, *Comput. Methods Appl. Mech. Eng.* 335 (2018) 419–471.
- [39] H.-W. Dong, S.-D. Zhao, P. Wei, L. Cheng, Y.-S. Wang, C. Zhang, Systematic design and realization of double-negative acoustic metamaterials by topology optimization, *Acta Mater.* 172 (2019) 102–120.
- [40] S.D. Campbell, D. Sell, R.P. Jenkins, E.B. Whiting, J.A. Fan, D.H. Werner, Review of numerical optimization techniques for meta-device design, *Opt. Mater. Express* 9 (4) (2019) 1842–1863.
- [41] M.M.R. Elsayy, S. Lanteri, R. Duvinneau, J.A. Fan, P. Genevet, Numerical optimization methods for metasurfaces, *Laser Photon. Rev.* 14 (10) (2020) 1900445.
- [42] P.-I. Schneider, X. Garcia Santiago, V. Soltwisch, M. Hammerschmidt, S. Burger, C. Rockstuhl, Benchmarking five global optimization approaches for nano-optical shape optimization and parameter reconstruction, *ACS Photon.* 6 (11) (2019) 2726–2733.
- [43] J. Kennedy, C.W. Lim, et al., Machine learning and deep learning in phononic crystals and metamaterials—A review, *Mater. Today Commun.* 33 (2022) 104606.
- [44] A. Bacigalupo, G. Gnecco, M. Lepidi, L. Giambarotta, Machine-learning techniques for the optimal design of acoustic metamaterials, *J. Optim. Theory Appl.* 187 (2020) 630–653.
- [45] Y. Jin, L. He, Z. Wen, B. Mortazavi, H. Guo, D. Torrent, B. Djafari-Rouhani, T. Rabczuk, X. Zhuang, Y. Li, Intelligent on-demand design of phononic metamaterials, *Nanophotonics* 11 (3) (2022) 439–460.
- [46] X. Zheng, X. Zhang, T.-T. Chen, I. Watanabe, Deep learning in mechanical metamaterials: from prediction and generation to inverse design, *Adv. Mater.* 35 (45) (2023) 2302530.
- [47] D. Lee, W. Chen, L. Wang, Y.-C. Chan, W. Chen, Data-Driven design for metamaterials and multiscale systems: a review, *Adv. Mater.* 36 (8) (2024) 2305254.
- [48] C.-X. Liu, G.-L. Yu, Z. Liu, Machine learning models in phononic metamaterials, *Curr. Opin. Solid State Mater. Sci.* 28 (2024) 101133.
- [49] X. Li, S. Ning, Z. Liu, Z. Yan, C. Luo, Z. Zhuang, Designing phononic crystal with anticipated band gap through a deep learning based data-driven method, *Comput. Methods Appl. Mech. Eng.* 361 (2020) 112737.
- [50] C. Gurbuz, F. Kronowetter, C. Dietz, M. Eser, J. Schmid, S. Marburg, Generative adversarial networks for the design of acoustic metamaterials, *J. Acoust. Soc. Am.* 149 (2) (2021) 1162–1174.
- [51] Z. Wang, W. Xian, M.R. Baccouche, H. Lanzerath, Y. Li, H. Xu, Design of phononic bandgap metamaterials based on gaussian mixture beta variational autoencoder and iterative model updating, *J. Mech. Des.* 144 (4) (2022) 041705.
- [52] W. Chen, R. Sun, D. Lee, C.M. Portela, W. Chen, Generative inverse design of metamaterials with functional responses by interpretable learning, *Adv. Intell. Syst.* (2023) 2400611.
- [53] S. Han, Q. Han, T. Jiang, C. Li, Inverse design of phononic crystals for anticipated wave propagation by integrating deep learning and semi-analytical approach, *Acta Mech.* 234 (10) (2023) 4879–4897.
- [54] C.-X. Liu, G.-L. Yu, Deep learning for the design of phononic crystals and elastic metamaterials, *J. Comput. Des. Eng.* 10 (2) (2023) 602–614.
- [55] M. Yan, R. Wang, K. Liu, Populating cellular metamaterials on the extrema of attainable elasticity through neuroevolution, *Comput. Methods Appl. Mech. Eng.* 440 (2025) 117950.
- [56] D. Lee, L. Zhang, Y. Yu, W. Chen, Deep neural operator enabled concurrent multitask design for multifunctional metamaterials under heterogeneous fields, *Adv. Opt. Mater.* (2024) 2303087.
- [57] G. Bordiga, E. Medina, S. Jafarzadeh, C. Bösch, R.P. Adams, V. Tournat, K. Bertoldi, Automated discovery of reprogrammable nonlinear dynamic metamaterials, *Nat. Mater.* (2024) 1–9.
- [58] C. Luo, S. Ning, Z. Liu, Z. Zhuang, Interactive inverse design of layered phononic crystals based on reinforcement learning, *Extreme Mech. Lett.* 36 (2020) 100651.
- [59] L. He, H. Guo, Y. Jin, X. Zhuang, T. Rabczuk, Y. Li, Machine-learning-driven on-demand design of phononic beams, *Sci. China Phys., Mech. Astronomy* 65 (1) (2022) 214612.
- [60] Y. Ding, E.C. Statharas, K. Yao, M. Hong, A broadband acoustic metamaterial with impedance matching layer of gradient index, *Appl. Phys. Lett.* 110 (24) (2017).
- [61] W. Tang, C. Ren, Total transmission of airborne sound by impedance-matched ultra-thin metasurfaces, *J. Phys. D Appl. Phys.* 50 (10) (2017) 105102.
- [62] Y. Xie, A. Konneker, B.-I. Popa, S.A. Cummer, Tapered labyrinthine acoustic metamaterials for broadband impedance matching, *Appl. Phys. Lett.* 103 (20) (2013).
- [63] S.B. Gebrekidan, Y.-I. Hwang, H.-J. Kim, S.-J. Song, Broadband impedance matching using acoustic metamaterial with a helical hole, *Appl. Phys. Lett.* 115 (15) (2019).
- [64] X. Jia, M. Yan, M. Hong, Sound energy enhancement via impedance-matched anisotropic metamaterial, *Mater. Des.* 197 (2021) 109254.
- [65] P. Hao, D. Liu, H. Liu, S. Feng, B. Wang, G. Li, Intelligent optimum design of large-scale gradual-stiffness stiffened panels via multi-level dimension reduction, *Comput. Methods Appl. Mech. Eng.* 421 (2024) 116759.

- [66] A.C. Ogren, B.T. Feng, K.L. Bouman, C. Daraio, Gaussian process regression as a surrogate model for the computation of dispersion relations, *Comput. Methods Appl. Mech. Eng.* 420 (2024) 116661.
- [67] J. Wang, D.-H. Chen, H.-L. Zhang, X.-M. Zhang, X. He, X.-M. Wang, Studies on phase and group velocities from acoustic logging, *Appl. Geophys.* 9 (1) (2012) 108–113.
- [68] Y. Wang, J. Liu, On the yield surface of a typical bending-dominant periodic lattice metamaterial, *J. Theor. Appl. Mech.* 61 (2023).
- [69] H. Reda, S.E. Alavi, M. Nasimsobhan, J.-F. Ganghoffer, Homogenization towards chiral cosserat continua and applications to enhanced timoshenko beam theories, *Mech. Mater.* 155 (2021) 103728.
- [70] S.E. Alavi, J.F. Ganghoffer, M. Sadighi, M. Nasimsobhan, A.H. Akbarzadeh, Continualization method of lattice materials and analysis of size effects revisited based on cosserat models, *Int. J. Solids Struct.* 254 (2022) 111894.
- [71] R.K. Mahabadi, T. Goudarzi, R. Fleury, B. Orazbayev, R. Naghdabadi, Effect of mechanical nonlinearity on the electromagnetic response of a microwave tunable metamaterial, *J. Phys. D Appl. Phys.* 55 (20) (2022) 205102.
- [72] B.A. Auld, *Acoustic fields and waves in solids*, Wiley, 1973.
- [73] M. Elarbi, S. Bechikh, A. Gupta, L.B. Said, Y.-S. Ong, A new decomposition-based NSGA-II for many-objective optimization, *IEEE Trans. Syst., Man, Cybern.: Systems* 48 (7) (2017) 1191–1210.
- [74] S. Verma, M. Pant, V. Snasel, A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems, *IEEE Access* 9 (2021) 57757–57791.
- [75] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, MOEA/d and NSGA-II, *IEEE Trans. Evol. Comput.* 13 (2) (2008) 284–302.
- [76] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch (2017).
- [77] A. Bundy, L. Wallen, Breadth-first search, in: *Catalogue of Artificial Intelligence Tools*, Springer, 1984, pp. 13.
- [78] R. Tarjan, Depth-first search and linear graph algorithms, *SIAM J. Comput.* 1 (2) (1972) 146–160.
- [79] L. Luo, M. Wong, W.-m. Hwu, An effective GPU implementation of breadth-first search, in: *Proceedings of the 47Th Design Automation Conference*, 2010, pp. 52–55.
- [80] A. Gaihre, Z. Wu, F. Yao, H. Liu, XBFS: Exploring runtime optimizations for breadth-first search on GPUs, in: *Proceedings of the 28Th International Symposium on High-performance Parallel and Distributed Computing*, 2019, pp. 121–131.
- [81] Y. Wang, H. Huang, C. Rudin, Y. Shaposhnik, Understanding how dimension reduction tools work: an empirical approach to deciphering t-SNE, UMAP, triMAP, and paCMAP for data visualization, *J. Mach. Learn. Res.* 22 (2021) (2021) 1–73.
- [82] R.K. Mahabadi, Z. Chen, A.C. Ogren, H. Zhang, C. Daraio, C. Rudin, L.C. Brinson, Graph-based design of irregular metamaterials, *Int. J. Mech. Sci.* 295 (2025) 110203.
- [83] D.P. Kingma, Adam: A method for stochastic optimization, (2014). [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)