



Deep neural networks for parameterized homogenization in concurrent multiscale structural optimization

Nolan Black¹ · Ahmad R. Najafi¹

Received: 22 May 2022 / Revised: 23 November 2022 / Accepted: 1 December 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

Concurrent multiscale structural optimization is concerned with the improvement of macroscale structural performance through the design of microscale architectures. The multiscale design space must consider variables at both scales, so design restrictions are often necessary for feasible optimization. This work targets such design restrictions, aiming to increase microstructure complexity through deep learning models. The deep neural network (DNN) is implemented as a model for both microscale structural properties and material shape derivatives (shape sensitivity). The DNN's profound advantage is its capacity to distill complex, multidimensional functions into explicit, efficient, and differentiable models. When compared to traditional methods for parameterized optimization, the DNN achieves sufficient accuracy and stability in a structural optimization framework. Through comparison with interface-aware finite element methods, it is shown that sufficiently accurate DNNs converge to produce a stable approximation of shape sensitivity through back propagation. A variety of optimization problems are considered to directly compare the DNN-based microscale design with that of the Interface-enriched Generalized Finite Element Method (IGFEM). Using these developments, DNNs are trained to learn numerical homogenization of microstructures in two and three dimensions with up to 30 geometric parameters. The accelerated performance of the DNN affords an increased design complexity that is used to design bio-inspired microarchitectures in 3D structural optimization. With numerous benchmark design examples, the presented framework is shown to be an effective surrogate for numerical homogenization in structural optimization, addressing the gap between pure material design and structural optimization.

Keywords Topology optimization · Deep learning · Multiscale design · Parameterized microstructures

1 Introduction

Multiscale structural design targets two scales to achieve a desirable structural response (e.g., maximum stiffness, minimum stress, or tailored mechanical actuation). The observable or macroscale domain is evaluated based on this performance metric, while the microscale domain is resolved to interpretable material characteristics. The intimate connection between scales creates a prohibitively large design space, and the resulting optimization is generally ill-posed (Allaire et al. 1997). To resolve the prohibitive design space, topology optimization (Bendsøe and Kikuchi 1988) has

implemented numerical homogenization to frame admissible designs and join microscale features to a macroscale space (Guedes and Kikuchi 1990; Sigmund 1994; Hassani and Hinton 1998; Wu et al. 2021a).

Numerical homogenization represents composite media through the local repetition of very small microstructures (Guedes and Kikuchi 1990; Torquato and Haslach 2002; Allaire and Brizzi 2005; Andreassen and Andreassen 2014). Given a microstructure unit cell, the effective material properties of a composite material may be approximated through the asymptotic expansion of its governing equations with respect to the ratio between length scales (Guedes and Kikuchi 1990). Numerical homogenization has been extensively used in heterogeneous cellular material optimization (Torquato 2010), including the design of free material distributions (Sigmund 1994; Guedes et al. 2003; Andreassen and Sigmund 2012), truss geometries (Watts and Tortorelli 2017; Kazemi and Norato 2022), and nonlinear materials (Wang et al. 2014; Najafi et al. 2021). Although numerical

Responsible Editor: Ramin Bostanabad

✉ Ahmad R. Najafi
arn55@drexel.edu

¹ Department of Mechanical Engineering and Mechanics,
Drexel University, Philadelphia, PA 19104, USA

homogenization is an effective method to resolve microscale structures, its implementation in multiscale design optimization presents computational challenges. The homogenization formulation requires that scales must be separated by multiple orders of magnitude (Guedes and Kikuchi 1990; Allaire et al. 1997), and simplified geometries are often necessary to adequately model multiscale structures (Wu et al. 2021a; Cheng et al. 2019; Groen and Sigmund 2018; Garner et al. 2019).

Parameterized representations of the microscale unit cell are used in multiscale optimization to alleviate the computational burdens of the intractable design space. Through parameterization of the microscale material, as originally presented in Bendsøe and Kikuchi (1988), multiscale design optimization has produced stiffness optimal designs that approach the theoretical limit using rank-2 laminates (Sigmund et al. 2016). The de-homogenization approach (Pantz and Trabelsi 2008), a post-processing method for realizing viable structures composed of varying microarchitectures, has been implemented to produce high-resolution monoscale designs optimized through parameterized, homogenization-based topology optimization (Groen and Sigmund 2018; Groen et al. 2019). The de-homogenization approach for stiffness optimal structures typically employs plate-like microscale geometry, but other objectives (e.g., tailorable actuation, increased buckling strength) benefit from increased microscale geometric complexity (Zhu et al. 2017; Wang and Sigmund 2021). Therefore, there is a need for increased parameterization of the microscale unit cell, and multiscale modeling techniques must adapt to accommodate the increasing design complexity.

With increasing parameterization and design complexity, surrogate models of numerical homogenization are used to approximate effective microstructure properties for multiscale design. A surrogate model for homogenization in multiscale optimization must resolve the effective material properties—and their derivative information—from the design parameters. As such, increasing parameterization requires sophisticated regression techniques to adequately approximate numerical homogenization. Low-order polynomial approximations (Watts et al. 2019) and multidimensional surrogates (Imediegwu et al. 2019) have been used to relate microscale features to their respective macroscale properties. Parameterization also offers explicit control of material interfaces, aiding design for manufacturing applications. Parameterized truss-like microstructures have been extensively evaluated in multiscale structures designed for additive manufacturing (Imediegwu et al. 2019; Murphy et al. 2021; Watts et al. 2019; Wu et al. 2021b). Other highly parameterized multiscale optimization methods employ the metamaterial, a strategy for design that uses geometry rather than material constituents to tailor performance (Yu et al. 2018). Metamaterial design generally favors intricate

microscale features, so data-driven methods are necessary for efficient implementation in multiscale design (Wang et al. 2020).

To address the growing complexity of microscale geometry, machine learning (ML) models have been proposed as a new surrogate for multiscale analysis. ML has attracted attention in multiscale modeling with applications to material optimization (Kollmann et al. 2020; Wang et al. 2020) and nonlinear analysis (Logarzo et al. 2021). ML frameworks, specifically the neural network (NN), have also been implemented as surrogate models for numerical homogenization in multiscale optimization. The NN offers an explicit model for multidimensional functions and has shown the capacity to approximate highly nonlinear functions (Hornik et al. 1989; Gallant and White 1992; Nguyen-Thien and Tran-Cong 1999; Goodfellow et al. 2016).

NN techniques are commonly employed in parameterized multiscale optimization to approximate the effective elastic properties of a microstructure. Zhou et al. implemented a NN in the multiscale optimization of parameterized lattice structures (Zhou et al. 2022); White et al. implemented a Sobolev norm NN (a NN that is directly trained on derivative data) in fine-scale multiscale truss optimization (White et al. 2019); Kim et al. used a DNN surrogate model for homogenization in graded composite structural design (Kim et al. 2021). Zheng et al. implemented a NN surrogate for evaluation and sensitivity analysis of spinodoid metamaterials (Zheng et al. 2021). Other non-parameterized techniques may use the NN to navigate a library of microstructure geometries (Wang et al. 2020; Chan et al. 2022). These techniques, however, fail to capture highly complex microstructure features with tens to hundreds of features. Furthermore, it is still unclear how NN architecture and training procedures affect its sensitivity analysis in the context of multiscale optimization.

The development of these NN surrogate models requires pre-computation of a training dataset followed by the iterative optimization of the model's parameters. These two characteristics affect both the model's execution accuracy and the accuracy of its derivative information. This work explores these characteristics as they apply to multiscale design and surrogate models for numerical homogenization. This work is presented to evaluate the DNN as a surrogate model for numerical homogenization in the context of multiscale design. The goal of this evaluation is to characterize the surrogate model's capacity for geometric complexity and increase the design space from lattice-like designs to more complex geometries akin to metamaterials. To this end, we establish the DNN as a viable surrogate for evaluation and sensitivity through comparison with Interface-enriched Generalized Finite Element Methods (IGFEM). The verification process involves experimentation with DNN architecture and training data to establish a framework for training DNNs as surrogates in highly parameterized multiscale optimization.

With this verification, we can characterize DNN performance and generate effective training datasets. Finally, we are able to extend these models to highly parameterized microstructures, approaching the design freedom characteristic of mechanical metamaterials. As an example of this complexity, we introduce a bio-inspired 3D truss design defined by 30 geometric parameters.

Section 2 presents the proposed framework for integrating ML in multiscale analysis with the DNN. Section 3 introduces the formulation for structural optimization, then Sect. 4 examines the DNN performance compared to IGFEM in the context of multiscale optimization. Once the ML framework is sufficiently scrutinized, various optimization examples are presented in Sect. 5 to extend the approach to highly parameterized designs.

2 Numerical model and DNN surrogate

In this study, we aim to perform gradient-based optimization of a multiscale structure. To reduce the computational burden of the microscale design parameters, we integrate the DNN as a surrogate model for numerical homogenization. Such a surrogate model must not only provide an approximation of a microstructure's behavior, but it also must model its sensitivity with respect to multiple design parameters. We are concerned with the accuracy and stability of this implementation in the context of multiscale design.

Numerical homogenization is used to approximate the effective properties of a microstructure. Each microstructure is defined within the bounds of a repeated unit cell geometry consisting of two or more linear elastic materials (Fig. 1). Given sufficient separation of length scales such that the microstructure's geometry is much smaller than its macroscale domain, the composite microstructure can be distilled

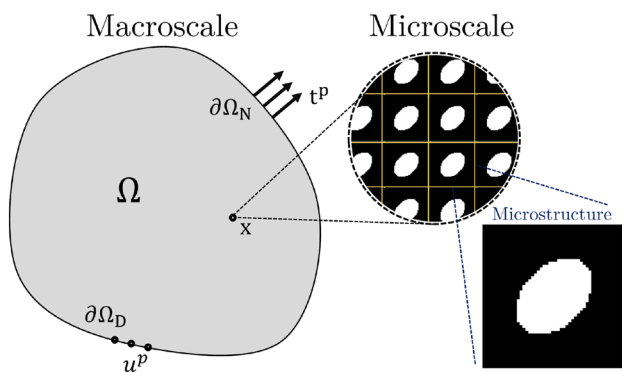


Fig. 1 A macroscale domain subjected to generic boundary conditions (t^p on $\partial\Omega_N$ and u^p on $\partial\Omega_D$) is characterized by microscale geometries at X defined with periodically varying microstructures. This multiscale representation is shown in 2D for simplicity, but the relations are equally applicable to 3D

into a single homogenized representation. As such, numerical homogenization is a powerful tool in multiscale optimization used to link the multiscale features to macroscale performance. We include the relevant results of the energy-based homogenization approach here; a thorough presentation of the homogenization procedure is offered in Guedes and Kikuchi (1990), Hassani and Hinton (1998), Allaire and Brizzi (2005), and Andreassen and Andreassen (2014).

For the constitutive relation $\sigma_{ij} = C_{ijkl}\epsilon_{kl}$ of stress σ_{ij} and strain ϵ_{kl} , the effective constitutive tensor of a periodic composite can be expressed as

$$C_{ijkl}^H = \frac{1}{|Y|} \int_Y C_{pqrs} \left(\epsilon_{pq}^{0(ij)} - \epsilon_{pq}^{*(ij)} \right) \left(\epsilon_{rs}^{0(kl)} - \epsilon_{rs}^{*(kl)} \right) dY, \quad (1)$$

where Y represents the microstructure cell volume, $\epsilon_{pq}^{0(ij)}$ refers to the prescribed unit strain case on the unit cell (there are 3 unique prescribed elastic strains in 2D, 6 in 3D). The local strain field $\epsilon_{pq}^{*(ij)}$ is associated with the Y -periodic solution to the elasticity condition

$$\int_Y C_{ijpq} \epsilon_{pq}^{*(kl)} \frac{\partial v_i}{\partial y_j} dY = \int_Y C_{ijpq} \epsilon_{pq}^{0(kl)} \frac{\partial v_i}{\partial y_j} dY \quad (2)$$

for the Y -periodic admissible displacement field v . The constitutive tensor C_{ijkl} can be expressed in matrix form using Voigt notation; discretized for finite element analysis (FEA), the microstructure is homogenized using (1) in matrix notation appropriate for FEA as

$$C^H = C_{ij}^H = \frac{1}{|Y|} \sum_{e_\mu=1}^{N_\mu} \left(\mathbf{u}_{e_\mu}^{0(i)} - \mathbf{u}_{e_\mu}^{(i)} \right)^T \mathbf{k}_{e_\mu} \left(\mathbf{u}_{e_\mu}^{0(j)} - \mathbf{u}_{e_\mu}^{(j)} \right) \quad (3)$$

for N_μ elements e_μ with element stiffness matrix $\mathbf{k}_{e_\mu}$, displacement field $\mathbf{u}_{e_\mu}^0$ corresponding to the unit strain test case ϵ^0 , and displacement field $\mathbf{u}_{e_\mu}$ corresponding to the strain ϵ^* induced by microstructure geometry. Note that the subscript e_μ indicates an element in the microscale structure. For the most general elastic problems, C^H has 6 unique components in 2D and 21 unique components in 3D. The DNN surrogate models developed here will target the homogenized constitutive tensor represented by C^H .

2.1 Multiscale analysis

In a macroscale domain discretized for FEA as N elements (Fig. 2), we introduce the macroscale design variable $\boldsymbol{\varphi} = \{\varphi_1, \varphi_2, \dots, \varphi_N\} \in \{0, 1\}$ to indicate the presence of material and the microscale design variable $\mathbf{x} = \{x_1^1, x_2^1, \dots, x_N^1, x_1^2, x_2^2, \dots, x_N^2, \dots, x_1^M, x_2^M, \dots, x_N^M\}$ to represent N microarchitectures defined by M parameters.

The solid isotropic microstructure with penalty (SIMP) method (Zhou and Rozvany 1991; Rozvany et al. 1992) is

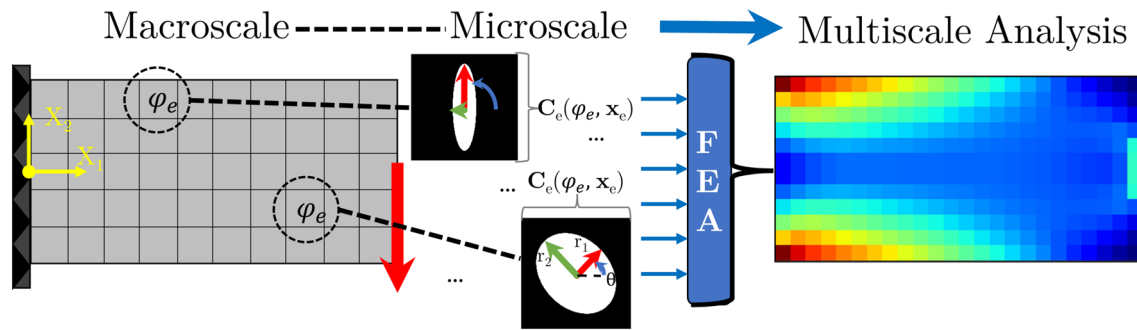


Fig. 2 A macroscale, finite element domain is characterized by the variable φ at each element to indicate the presence of a microstructure geometry. Within each element, a parameterized microstructure is characterized by $\mathbf{x}$. For illustrative purposes, three parameters are

used as a simple example for an elliptical inclusion defined by the major r_1 and minor r_2 axis and an angle of rotation θ . Ultimately, each microstructure is homogenized and implemented into macroscale FEA

applied to penalize intermediate macroscale representations. The SIMP method links the macroscale analysis to the homogenized constitutive behavior [cf. (3)] through the following penalized expression for the constitutive matrix:

$$C_e(\varphi_e, \mathbf{x}_e) = C_{\min} + \varphi_e^p (C^H(\mathbf{x}_e) - C_{\min}). \quad (4)$$

A nearly zero valued matrix $C_{\min}$ is introduced to avoid singularities in numerical analysis, and the penalization power p favors φ_e toward 0 or 1. Note that (4) implements C^H , which may be determined through numerical homogenization and (3) or through an appropriate surrogate model.

With this complete expression for effective macroscale constitutive behavior in (4), implementation into FEA is performed with the element stiffness matrix

$$k_e = \int_{\Omega_e} \mathbf{B}^T C_e \mathbf{B} d\Omega_e, \quad (5)$$

where $\mathbf{B}$ is the strain–displacement matrix and C_e is the constitutive matrix of the element e from (4). Macroscale equilibrium for linear elastic FEA is subsequently expressed as

$$\mathbb{K} \mathbb{U} = \mathbb{F}, \quad (6)$$

where $\mathbb{K}$ is the assembled global stiffness matrix, $\mathbb{U}$ is the finite element nodal displacement, and $\mathbb{F}$ is the finite element nodal force.

Parameterization of the microarchitecture via $\mathbf{x}$ offers explicit control over the microarchitecture’s geometric limits. The added control may be leveraged to enforce a specific engineering quality, unit cell connectivity, or manufacturing constraints.

Implementing numerical homogenization into a numerical framework for concurrent multiscale optimization, however, presents some practical challenges. Namely, the design space quickly becomes intractable considering (i)

the number of M parameters necessary to achieve sufficient design flexibility in each microstructure and (ii) the number of N microstructures necessary to achieve sufficient separation of scales. Facing these challenges, we turn to the DNN as a surrogate model for numerical homogenization. The subsequent section introduces the NN formulation and the developments necessary for implementation into concurrent multiscale optimization.

2.2 Deep learning model for homogenization and design sensitivity

The NN is a common building block in scientific ML frameworks (Baker et al. 2019), and is broadly promoted as a universal function approximator given sufficient parametrization (Hornik et al. 1989). Practically, the NN represents a sequence of matrix operations. Given an input vector $\mathbf{x} = \mathbf{h}^0$, each layer T^i of the NN produces a latent representation $\mathbf{h}^i$ from a variable matrix of weights $\mathbf{W}^i$ and vector of biases $\mathbf{b}^i$ such that $T^{i+1} = \mathbf{W}^i \mathbf{h}^i + \mathbf{b}^i$. Optionally, the layer output is passed element-wise through an activation function $\mathcal{A}$ (e.g., sigmoid, tanh, Rectified Linear Unit) so the layer output becomes $\mathcal{A}^i(T^i) = A^i$. With multiple layers added sequentially, the NN is classified as a deep NN (DNN) whose output representation $\hat{\mathbf{y}}$ becomes a composite function of the weights and biases for L total layers,

$$\text{DNN}(\mathbf{x}) = \hat{\mathbf{y}} = A^L \circ A^{L-1} \circ \dots \circ A^1(T^1(\mathbf{x})). \quad (7)$$

The DNN parameters of $\mathbf{W}$ and $\mathbf{b}$ are trained through the iterative optimization of an objective function $\mathcal{L}$. With numerical homogenization as an example, the microstructure design parameters become the input $\mathbf{x}$, and the components of C^H are arranged as the output vector $\hat{\mathbf{y}}$. Using the mean squared error objective function,

$$\mathcal{L}(\mathbf{W}, \mathbf{b}, \mathbf{y}) = \frac{1}{N_y} (\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}}), \quad (8)$$

the NN is trained or optimized to minimize the difference between output $\hat{\mathbf{y}}$ and known value $\mathbf{y}$. The inclusion of $\mathbf{y}$ in this approach requires the pre-computation of a training dataset. Again with numerical homogenization as an example, the training dataset might consist of sampled microscale geometries and their respective homogenized constitutive properties. A thorough exploration of the computational cost of this approach must include the generation of these training data. Critically, the Jacobian of the DNN model can be recovered efficiently using back propagation through the network, shown as

$$\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{x}} = \frac{\partial \hat{\mathbf{y}}}{\partial A_L} \frac{\partial A_L}{\partial T_L} \frac{\partial T_L}{\partial A_{L-1}} \dots \frac{\partial T_1}{\partial \mathbf{x}}. \quad (9)$$

As the NN model is trained, the model's error is reduced through repeated application of (9) applied to the model's variable parameters or weights. This training process is particularly appealing for design optimization applications, as not only is the DNN trained to mimic some complex analytical function [i.e., numerical homogenization (1)], but the model's sensitivity is also tuned indirectly. Note that the indirect tuning of the sensitivity offers no guarantees of accuracy or stability. This sensitivity behavior is investigated by White et al. (2019) for a single layer NN. Evaluating the DNN's sensitivity limits is notoriously difficult, and characterization of smoothness is typically constrained to bounded Lipschitz constants (Bian and Chen 2012; Fazlyab et al. 2019; Gouk et al. 2021). This work is primarily concerned with (i) how increasing the nonlinearity of the DNN model through the addition of multiple layers can improve the model's sensitivity behavior in an optimization context, (ii) if this nonlinearity can compensate for low-fidelity, computationally efficient training data, and (iii) how such a model functions with highly parameterized microstructure designs.

To answer these questions, the following work investigates the indirect influence of DNN architecture and training procedure on the accuracy of its sensitivity. We evaluate this "accuracy" both as a value relative to an analytical solution and as an operation within multiscale structural optimization. A successful surrogate model may therefore not produced perfect analytical sensitivities; instead the successful surrogate should provide enough information to navigate the multiscale design space.

3 Concurrent multiscale optimization framework

Concurrent multiscale optimization aims to improve structural performance by optimizing material distribution $\boldsymbol{\varphi}$ and microstructure geometry $\mathbf{x}$. This work considers the design methods for a microstructure defined by $\mathbf{x}$, comparing advanced finite element methods (IGFEM) and DNN-based surrogate models. For convenience, the design variables are combined into $\boldsymbol{\alpha} = \{\boldsymbol{\varphi}, \mathbf{x}\}$, so the optimization problem is formulated as

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \Theta(\mathbb{U}(\mathbf{X}(\boldsymbol{\alpha}), \boldsymbol{\alpha}), \mathbb{F}(\boldsymbol{\alpha}), \boldsymbol{\alpha}), \\ \text{such that: } \quad & \alpha_i^{\min} < \alpha_i < \alpha_i^{\max}, \\ \text{and: } \quad & \mathbf{g}(\mathbb{U}(\mathbf{X}(\boldsymbol{\alpha}), \boldsymbol{\alpha}), \mathbb{F}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \leq \mathbf{0}, \\ & \mathbb{K}(\mathbf{X}(\boldsymbol{\alpha})) \mathbb{U}(\mathbf{X}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \mathbb{F}(\mathbf{X}(\boldsymbol{\alpha})), \end{aligned} \quad (10)$$

where Θ is the objective function, $\mathbf{X}$ is the nodal coordinates for the finite element mesh, and $\mathbf{g}$ represents applicable inequality constraints. We define two objective functions for investigating the proposed method for concurrent structural optimization. The first, structural compliance, is expressed as

$$\Theta_c(\mathbb{U}(\mathbf{X}(\boldsymbol{\alpha}), \boldsymbol{\alpha}), \mathbb{F}(\boldsymbol{\alpha})) = \mathbb{F}^T \mathbb{U}. \quad (11)$$

The second objective represents the difference between a target structural deformation $\mathbb{U}_T$ and the actual structural deformation field $\mathbb{U}$ (Wang et al. 2020), formulated as

$$\begin{aligned} \Theta_p(\mathbb{U}(\mathbf{X}(\boldsymbol{\alpha}), \boldsymbol{\alpha}), \mathbb{U}_T) \\ = \frac{1}{N_T} (\boldsymbol{\gamma} \cdot (\mathbb{U}_T - \mathbb{U}))^T (\boldsymbol{\gamma} \cdot (\mathbb{U}_T - \mathbb{U})), \end{aligned} \quad (12)$$

where N_T represents the number of nodes with targeted deformation behavior and $\boldsymbol{\gamma}$ is a binary vector indicating a targeted or untargeted node.

Volume constraints are defined for the macroscale, microscale, and net volume of the optimized structure. Given the microstructure volume v_x , the appropriate volume constraints are

$$\mathbf{g} = \left\{ \frac{1}{N} \sum_{e=1}^N \varphi_e - V_\varphi, \frac{1}{N} \sum_{e=1}^N \varphi_e v_x - V_x \right\}, \quad (13)$$

where V_φ and V_x are the volume fractions for the macroscale distribution of material and volume fraction of the microscale structure. The macroscale volume fraction V_φ controls the fraction of the macroscale domain that contains any material and ranges from 0 (no material in domain) to 1 (material present in entire domain). The microscale volume fraction V_x represents the volume fraction of the microscale features present in the design and ranges from 0 (the

structure is composed of empty cells) to 1 (the structure is composed of solid cells). The minimum volume fraction of each microscale cell is a function of its parameterization and is limited by the geometric restrictions imposed by $\alpha_i^{\min}$ and $\alpha_i^{\max}$. Given this formulation, the net volume fraction within the design domain V is then $V_\phi V_x$.

A mesh-independent filtering technique of the design variables is also employed (Bruns and Tortorelli 2001). For an element i and filter radius r , the elements j within that radius are defined as those whose centroid falls within r to the centroid of i . The centroid–centroid distance d_j is used in Gaussian-weighted kernel for each design variable,

$$\bar{\alpha}_i = \sum_j \frac{\omega_j}{\omega} \alpha_j, \quad (14)$$

$$\omega_j = \max\left(1 - \frac{d_j}{r}, 0\right), \quad (15)$$

$$\omega = \sum_j \omega_j. \quad (16)$$

Filtering is a popular technique in topology optimization to enforce a length-scale on the design variable ϕ (Bourdin 2001); in multiscale design, we propose the same filter as a technique for limiting local variations in the microstructure geometry. The filtered geometry represents a local averaging of the features, penalizing rapid local changes in the microscale geometry.

3.1 Sensitivity analysis for macroscale performance

This subsection begins with a presentation of the macroscale sensitivity analysis formulated from the macroscale objectives (11) and (12). The objective function's sensitivity with respect to the design variable $\frac{\partial \Theta}{\partial \alpha}$ is found through adjoint sensitivity analysis. The full multiscale sensitivity analysis requires a microscale sensitivity (discussed in Sect. 3.2), evaluated from the microarchitecture's homogenized properties, the microarchitecture's parameterization, and a macroscale sensitivity evaluated based on the macroscale structural response.

For FEA on a fixed mesh (i.e., $\frac{\partial X}{\partial \alpha} = \frac{\partial \mathbb{F}}{\partial \alpha} = \mathbf{0}$) and the objective functions (11) and (12), differentiation (in the macroscale) with respect to a single design variable α_i produces

$$\frac{\partial \Theta}{\partial \alpha_i} = \frac{\partial \Theta^T}{\partial \mathbb{U}} \frac{\partial \mathbb{U}}{\partial \alpha_i}, \quad (17)$$

where $\frac{\partial \Theta}{\partial \mathbb{U}}$ can be found analytically, and $\frac{\partial \mathbb{U}}{\partial \alpha_i}$ is annihilated through the addition of an adjoint variable. The adjoint expression is

$$\begin{aligned} \frac{\partial \Theta}{\partial \alpha_i} &= \frac{\partial \Theta^T}{\partial \mathbb{U}} \frac{\partial \mathbb{U}}{\partial \alpha_i} + \lambda^T \left(-\mathbb{K} \frac{\partial \mathbb{U}}{\partial \alpha_i} + \mathbb{F}_{\text{pseudo}} \right) \\ &= \left(\frac{\partial \Theta^T}{\partial \mathbb{U}} - \lambda^T \mathbb{K} \right) \frac{\partial \mathbb{U}}{\partial \alpha_i} + \lambda^T \mathbb{F}_{\text{pseudo}}, \end{aligned} \quad (18)$$

where λ is the arbitrary adjoint vector and $\left(-\mathbb{K} \frac{\partial \mathbb{U}}{\partial \alpha_i} + \mathbb{F}_{\text{pseudo}} \right) = \mathbf{0}$. The pseudo-force $\mathbb{F}_{\text{pseudo}}$ is derived from known quantities:

$$\mathbb{F}_{\text{pseudo}} = -\frac{\partial \mathbb{K}}{\partial \alpha_i} \mathbb{U} + \frac{\partial \mathbb{F}}{\partial \alpha_i}. \quad (19)$$

The following expression for the adjoint variable λ is used to annihilate $\frac{\partial \mathbb{U}}{\partial \alpha_i}$ in (18):

$$\frac{\partial \Theta}{\partial \mathbb{U}} - \lambda^T \mathbb{K} = \mathbf{0}. \quad (20)$$

With the adjoint variable, we obtain the gradient of the objective function in terms of known quantities:

$$\frac{\partial \Theta}{\partial \alpha_i} = \lambda^T (\mathbb{F}_{\text{pseudo}}) = \lambda^T \left(-\frac{\partial \mathbb{K}}{\partial \alpha_i} \mathbb{U} + \frac{\partial \mathbb{F}}{\partial \alpha_i} \right). \quad (21)$$

For both objective functions (11) and (12), adjoint sensitivity analysis eliminates the unknown quantity $\frac{\partial \mathbb{U}}{\partial \alpha_i}$ through the addition of the adjoint variable λ . Solving (20) for λ and evaluating (21) produces the sensitivity of our objective functions with respect to the design variables. In compliance minimization problems (11), $\frac{\partial \Theta}{\partial \mathbb{U}} = \mathbb{F}$ and $\frac{\partial \mathbb{F}}{\partial \alpha_i} = \mathbf{0}$. For prescribed deformation problems (12), $\frac{\partial \Theta}{\partial \mathbb{U}} = \frac{1}{N_T} (\boldsymbol{\gamma} \cdot (\mathbb{U}_T - \mathbb{U}))$ and $\frac{\partial \mathbb{F}}{\partial \alpha_i} = \mathbf{0}$.

The assembly of $\mathbb{F}_{\text{pseudo}}$ is achieved through the element-wise assembly of $\frac{\partial \mathbb{K}}{\partial \alpha_i}$ through

$$\frac{\partial \mathbf{k}_e}{\partial \alpha_i} = \int_{\Omega_e} \mathbf{B}^T \frac{\partial \mathbf{C}_e}{\partial \alpha_i} \mathbf{B} d\Omega_e \quad (22)$$

for

$$\frac{\partial \mathbf{C}_e}{\partial \alpha_e} = \left\{ \frac{\partial \mathbf{C}_e}{\partial \phi_e}, \frac{\partial \mathbf{C}_e}{\partial x_e^1}, \frac{\partial \mathbf{C}_e}{\partial x_e^2}, \dots, \frac{\partial \mathbf{C}_e}{\partial x_e^M} \right\}, \quad (23)$$

where

$$\frac{\partial \mathbf{C}_e}{\partial \phi_e} = \mathbf{C}_{\min} + p \cdot \phi_e^{p-1} (\mathbf{C}_H(\mathbf{x}_e) - \mathbf{C}_{\min}) \quad (24)$$

and

$$\frac{\partial \mathbf{C}_e}{\partial x_e^i} = \mathbf{C}_{\min} + \phi_e^p \left(\frac{\partial \mathbf{C}_H}{\partial x_e^i} - \mathbf{C}_{\min} \right) \quad (25)$$

$i = 1, 2, \dots, M$.

Microscale sensitivity analysis of the homogenized micro-architectures enters the sensitivity analysis through $\frac{\partial C_e}{\partial x_e^i}$. Section 3.2 compares a finite element-based evaluation of $\frac{\partial C_e}{\partial x_e^i}$ with its DNN surrogate, including a discussion of various ML formulations and their implications in multiscale optimization. If filtering via (14) is used, the filtered variable may be expressed in terms of a weight matrix $\mathbf{G}$ as

$$\bar{\alpha} = \mathbf{G}\alpha. \quad (26)$$

Evaluations are then performed using the filtered design variable $\bar{\alpha}$. Sensitivities with respect to the raw design variables are recovered using

$$\frac{\partial \bar{\alpha}}{\partial \alpha} = \mathbf{G}^T. \quad (27)$$

3.2 Sensitivity analysis for microscale geometry

Numerical representations of the microstructure geometry require an expression of material interface. In the formulation for multiscale analysis, the material interface is parameterized in each finite element by x_e . The interface is marked by a solid–void transition, where void material is approximated by an artificially soft material. For example, Fig. 4a shows a unit cell with a uniform mesh and material interface represented by two materials E_1 and E_2 where $E_2 \ll E_1$. Although this representation may appropriately approximate numerical homogenization through (3), the step-wise interface is fundamentally incompatible with the shape derivative $\frac{\partial C^H}{\partial x_e^i}$ used in (25).

The optimization, therefore, requires an explicit expression for C^H in terms of x_e . This section presents a finite element-based model for the material interface and compares performance to a DNN-based surrogate model in the context of multiscale structural optimization. The DNN formulation presented in (7) and Fig. 3 provides such an expression but requires the iterative optimization of the DNN parameters. As discussed in Sect. 2.2, this optimization is

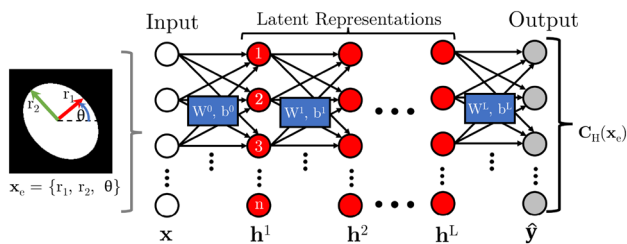


Fig. 3 Implemented as a surrogate model for numerical homogenization, the DNN uses the unit cell parameters (e.g., r_1 , r_2 , θ) as an input vector $\mathbf{x}$, produces a sequence of latent representations $\mathbf{h}^i$ via the optimized variables $\mathbf{W}$ and $\mathbf{b}$, and approximates an output $\hat{\mathbf{y}}$ that is reconstructed into the microstructure's effective properties C^H

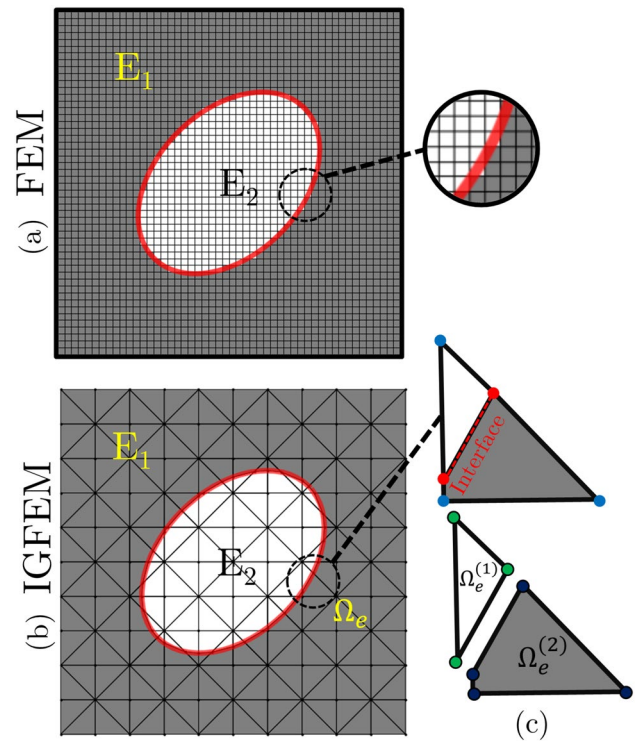


Fig. 4 **a** The finite element method approximates the microstructure geometry using solid E_1 , and void E_2 material but creates a discontinuous material interface. **b** The Interface-enriched Generalized Finite Element Method uses a series of enriched elements to create a smooth approximation of the microstructure's material interface. **c** An enriched element Ω_e is shown, including the enriched or interface nodes that separate the two child elements $\Omega_e^{(1)}$ and $\Omega_e^{(2)}$

formulated to ensure the DNN produces an accurate evaluation $\hat{C}^H$ relative to some set of training data C^H . Additionally, (9) can approximate $\frac{\partial \hat{C}^H}{\partial x_e^i}$ as

$$\frac{\partial \hat{C}^H}{\partial x_e^i} = \frac{\partial \hat{C}^H}{\partial A_L} \frac{\partial A_L}{\partial T_L} \frac{\partial T_L}{\partial A_{L-1}} \dots \frac{\partial T_1}{\partial x_e^i}. \quad (28)$$

In practice, training via (8) directly targets the approximation of C^H which indirectly affects to accuracy of (28). This correlation is explored hereafter as we seek to improve the evaluation of $\frac{\partial C^H}{\partial x_e^i}$ using the DNN trained with a dataset of finite element training data.

4 Sensitivity verification with finite element methods

The Interface-enriched Finite Element Method (IGFEM) (Soghrati et al. 2012; Safdari et al. 2015, 2016) was used as an evaluation metric for the DNN formulation. IGFEM represents the material interface through an element enrichment

scheme that directly evaluates displacement values at the interface (Fig. 4b). The enriched or parent element (Fig. 4c) is split at the interface into two child elements. The interface nodes are a direct function of the microstructure's geometric parameters $\mathbf{x}_e$, so (3) can be expressed in terms of $\mathbf{k}_{e_\mu}(\mathbf{x}_e)$ and an analytical expression for shape sensitivity is formulated as

$$\frac{\partial \mathbf{C}_{\text{IGFEM}}^H}{\partial x_e^i} = \frac{1}{|Y|} \sum_{e_\mu=1}^{N_\mu} \left(\mathbf{u}_{e_\mu}^{0(i)} - \mathbf{u}_{e_\mu}^{(i)} \right)^T \frac{\partial \mathbf{k}_{e_\mu}}{\partial x_e^i} \left(\mathbf{u}_{e_\mu}^{0(j)} - \mathbf{u}_{e_\mu}^{(j)} \right). \quad (29)$$

Equation (29) and the formulation of $\frac{\partial \mathbf{k}_{e_\mu}}{\partial x_e^i}$ are discussed further in “Appendix 1,” with additional formulation available in Najafi et al. (2015, 2017, 2021) and Brandyberry et al. (2020). The formulation has also been verified using finite difference methods to an absolute relative error on the order of 10^{-6} or lower for all material shape derivative terms.

Root mean square error (RMSE) is used as an evaluation metric for the DNN surrogate model. As an example, the error in predicting $\mathbf{C}^H$ is evaluated as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_{Ci}} \sum_{j=1}^{N_{Ci}} (C_{ij} - \hat{C}_{ij})^2 \text{ where } i \leq j}{\sum_{k=1}^{N_{Ci}} k}}, \quad (30)$$

where N_{Ci} represents the number of independent components of $\mathbf{C}^H$, the component C is evaluated via IGFEM and (3), and $\hat{C}$ is evaluated via DNN. Similarly, RMSE is used to evaluate shape sensitivity performance using the independent components of $\frac{\partial \mathbf{C}^H}{\partial x_e^i}$.

To compare the DNN and IGFEM formulation, a simple elliptical inclusion was considered with parameters $\mathbf{x}_e = \{r_1, r_2, \theta\}$ as in Fig. 3. Various DNN training strategies were examined to explore the relationship between (28) and the iterative optimization of (8). Through comparing the shape sensitivity produced via the trained DNN (28) and the analytical shape sensitivity produced via IGFEM (29), the performance of the DNN in an optimization context can be evaluated. Simple geometry is used here for illustrative purposes; homogenization of elliptical architecture may be approximated using simpler methods (e.g., polynomial models). For our purposes, the elliptical inclusion provides sufficient complexity to evaluate the DNN.

The training dataset was generated from randomized samples of 20×20 IGFEM mesh with base material ($E_1 = 1$ Pa, $\nu = 0.3$) and void material ($E_1 = 10^{-6}$ Pa, $\nu = 0.3$). With 667 examples dedicated to training and 333 examples used for testing, model training was performed using the Adam optimizer (Kingma and Ba 2017) with an initial learning

rate of 10^{-4} over 10^5 iterations of full-batch gradient descent. All layers except the final layer are activated by the sigmoid logistic function. The results of (8), reported as an average over the 333 test examples, are shown in Table 1 for different model architectures. As expected, sufficiently large networks ($n \geq 32$) were increasingly accurate, and deeper networks ($L \geq 1$) generally outperformed shallower networks. As the layer count increased beyond $L = 4$, the networks failed to converge due to vanishing gradients. “Appendix 2” offers more information of this phenomenon and presents more practical considerations for training DNNs for concurrent multiscale optimization. Based on these results, all subsequent analysis of DNN architecture will limit the number of hidden layers to $L \leq 3$.

4.1 DNN shape sensitivity performance

A pedagogical compliance minimization problem is considered to illustrate the DNN's performance in concurrent multiscale optimization as an approximator of the homogenized constitutive properties $\mathbf{C}^H$, microscale sensitivity $\frac{\partial \mathbf{C}^H}{\partial \mathbf{x}}$, and resulting macroscale sensitivity $\frac{\partial \Theta_c}{\partial \mathbf{x}}$. It is important to note that the DNN has been trained to approximate the homogenized constitutive properties, and this approximation is independent of the particular objective (i.e., compliance or prescribed deformation). The domain, illustrated in Fig. 5a, features a unit load applied to the first free node along $X_2 = 1$. The domain is defined by $L = 3$, $W = 1$, and $m = \frac{1}{4}$. A randomized selection of microarchitectures was generated to demonstrate the DNN's performance as a surrogate model. For the 48 randomized microstructures in the 12×4 macroscale mesh Fig. 5a, RMSE [cf. (30)] was used to compare the independent components of the DNN's $\mathbf{C}^H$, $\frac{\partial \mathbf{C}^H}{\partial \mathbf{x}}$, and $\frac{\partial \Theta_c}{\partial \mathbf{x}}$ with the equivalent IGFEM analysis.

RMSE for $\mathbf{C}^H$ is plotted for various DNN architectures in Fig. 5b; the DNN's performance in the pedagogical example affirms its evaluation by the test dataset (cf. Table 1), with increasing layer size h and neuron count

Table 1 The Loss values of (8) are reported as $\mathcal{L} \times 10^3$ for different DNN frameworks

Layers L	Neurons n				
	8	16	32	64	128
1	0.0622	0.0424	0.0418	0.0578	0.0646
2	0.0837	0.0579	0.0724	0.0647	0.0481
3	0.3078	0.2541	0.0323	0.0256	0.0230
4	0.2719	0.2899	0.0319	0.0270	0.0171
5	3.6462	0.2703	0.8665	0.0180	0.0313
6	3.6454	3.6111	0.0534	3.7668	3.9569
7	3.6420	3.6428	3.8530	3.9963	4.1985

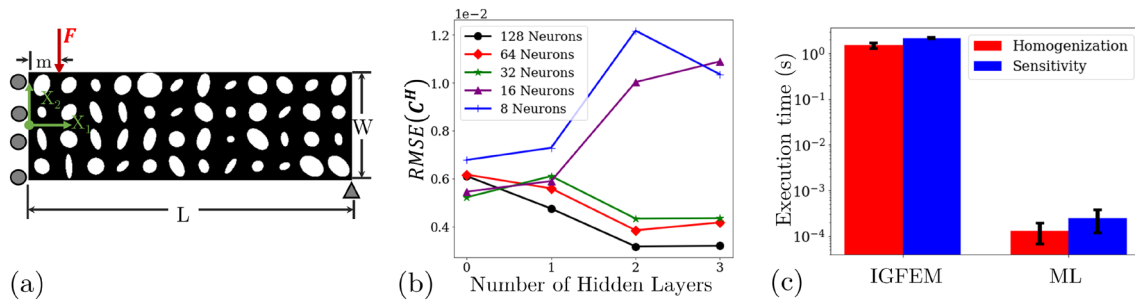


Fig. 5 DNN performance is shown for a pedagogical compliance minimization example. **a** A randomized collection of microarchitectures is assembled into a 2D beam subjected to nodal force F . **b** The root mean squared error is shown for the ML-predicted homogenized

constants compared to the IGFEM-evaluated homogenized constants. **c** The CPU time for the analytical IGFEM method is compared to its ML equivalent

n leading to increased accuracy. The plotted RMSE in Fig. 5b demonstrates sufficient accuracy on the order of 1% relative to IGFEM. Figure 5c also shows the CPU time, measured with a single PC with Intel Core i7 (6x 2.20 GHz) and 32 GB of DDR4RAM over 48 microarchitectures, where the DNN model reduces computation time by a factor of approximately 10^4 . These performance gains are expected, as the DNN is executed as a sequence of matrix multiplications.

The pedagogical compliance minimization problem in Fig. 5a was also used to evaluate the DNN's microscale sensitivity performance. RMSE of the independent components of $\frac{\partial c^H}{\partial x_i}$ was calculated for different DNN architectures and is illustrated in Fig. 6. Using normalized error

norms, the DNN's error in evaluating c^H is strongly correlated to its resulting shape sensitivity $\frac{\partial c^H}{\partial x_i}$ (Fig. 6b).

The macroscale sensitivity $\frac{\partial \theta_c}{\partial x_i}$ was also evaluated using different DNN architectures. Figure 7a compares the macroscale sensitivity performance for IGFEM and DNN models using RMSE. The DNN model was shown to closely approximate shape sensitivity for many architectures. The DNN's RMSE in evaluating macroscale sensitivity was consistently near or less than 10 for all parameters. Because $\frac{\partial \theta_c}{\partial r_1}$ and $\frac{\partial \theta_c}{\partial r_2}$ are significantly larger in magnitude than $\frac{\partial \theta_c}{\partial \theta}$ (Fig. 7b), the relative error induced by the DNN surrogate model is shown to be larger for less-sensitive parameters like θ . Regardless of this error, the DNN surrogate model correctly indicates the direction of

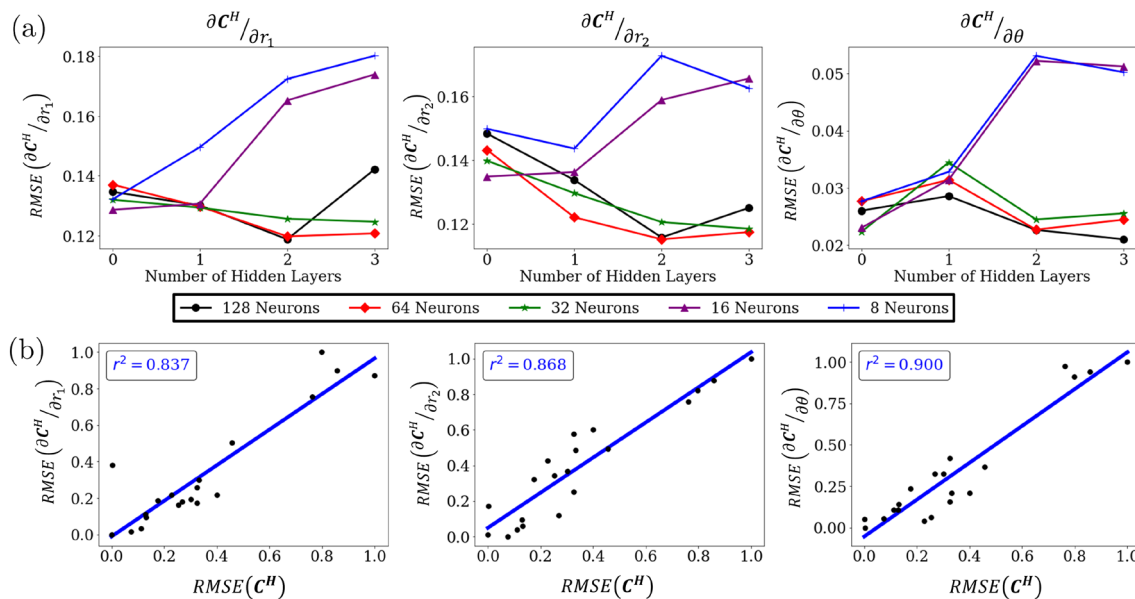


Fig. 6 The DNN's performance in shape optimization is illustrated. **a** Shape sensitivity for the three elliptical parameters is compared using RMSE of the ML approach compared to the analytical IGFEM

approach. **b** Using normalized error metrics, the correlation between training accuracy and the resulting approximation of shape sensitivity is shown

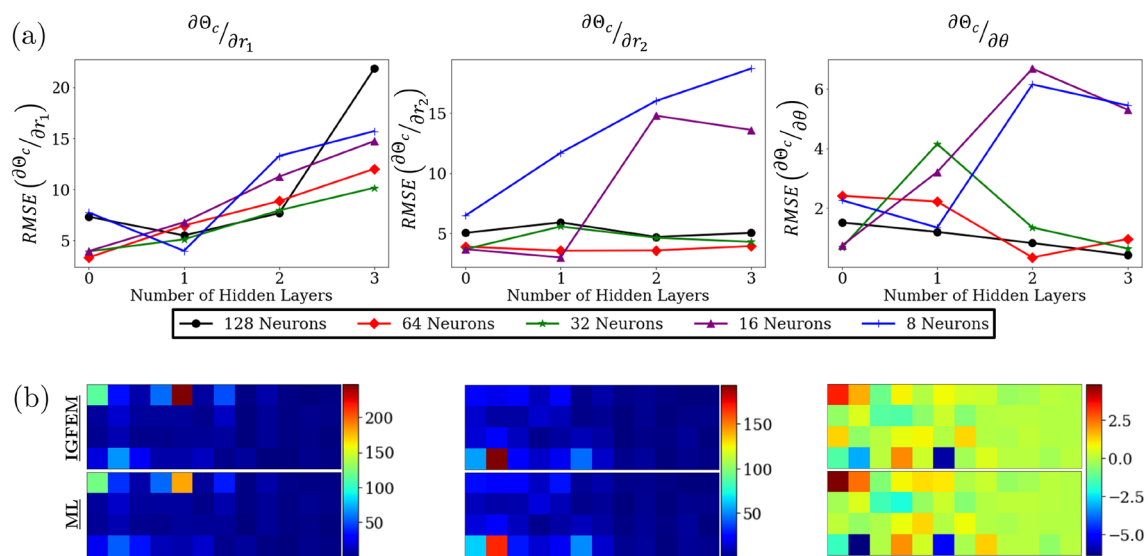


Fig. 7 **a** For the microstructure defined by r_1 , r_2 , and θ , the three components of $\frac{\partial \theta_c}{\partial x_i}$ are shown for different DNN configurations. Root mean square error (RMSE) is used to compare the evaluation of

shape sensitivity via ML and via IGFEM. **b** The predicted sensitivity $\frac{\partial \theta_c}{\partial x_i}$ for the DNN ($h = 3, n = 32$) is compared to its analytical IGFEM equivalent

increasing performance for all microscale parameters and successfully navigates the microscale design space.

For both the test dataset and pedagogical example, the sufficiently parameterized DNNs achieved higher accuracy in predicting C^H from geometric parameters. Instability was also observed for over parameterized models including the 3 layer, 128 neuron model as shown in Fig. 6a, indicating an upper limit to the model's parameterization. Perhaps more interesting is the strong correlation between the accuracy of the DNN's $\hat{C}^H$ and its predicted shape sensitivity evaluated via (28). The strong correlations, illustrated in Fig. 6b, indicate that a sufficiently accurate DNN also produces a sufficiently accurate (but not analytical) approximation of shape sensitivity, provided that the DNN was successfully trained without vanishing/exploding gradients or overfitting. Therefore, we theorize that if a DNN is successfully trained via (8), then the model will still produce shape sensitivity information capable of navigating the multiscale design space. This implies that the DNN—trained under the right conditions without over parameterization—can infer a smooth approximation of physical properties without an explicit representation of the shape geometry. A DNN surrogate for homogenization and shape sensitivity would have significant impact in multiscale design, as more complex parameterized architectures may be introduced to the concurrent design space.

4.2 FEM vs. IGFEM training data

The previous section demonstrated a strong correlation between evaluation accuracy and the resulting shape sensitivity for multiple DNN architectures trained using IGFEM training data. The DNN was shown to infer a smooth and accurate approximation of parameterized homogenization. This section examines the training data necessary to achieve this smooth approximation. If imperfect training data (i.e., data without an analytical representation of shape parameters) can inform a suitable DNN surrogate, then the computational cost of dataset generation may be reduced. We propose FEM training data with an approximate representation of material interface to inform the DNN training. The FEM-trained DNN, IGFEM-trained DNN, and IGFEM models for homogenization are subsequently compared in concurrent multiscale optimization.

From evidence in the previous section, we propose a 3 hidden layer, 32 neuron DNN with all layers except the final activated by the sigmoid logistic function. The FEM-trained DNN uses 1000 examples of 50×50 mesh to approximate an elliptical microstructure, as in Fig. 4a, with linear, 4-node elements. In contrast, the IGFEM-trained DNN uses 1000 examples of 20×20 IGFEM mesh to provide a better approximation of the elliptical microstructure with linear, 3-node elements. The different discretization fidelities (50×50 FEM and 20×20 IGFEM) were chosen as acceptable resolutions to capture the material interface (Fig. 4) while maintaining computational efficiency. The different fidelities and element formulations will produce slightly

varied homogenized parameters and thus slightly different displacement fields. To ensure fair comparison, designs produced via IGFEM, IGFEM-trained DNN, and FEM-trained DNN are all evaluated using IGFEM.

For both FEM and IGFEM training configurations, 667 instances of input ($\mathbf{x}_e = \{r_1, r_2, \theta\}$) and output ($\mathbf{C}_H$) pairs were used during model training, while 333 examples were used for testing and the subsequent model evaluations. Model training was performed using the Adam optimizer (Kingma and Ba 2017) with an initial learning rate of 10^{-4} over 10^5 iterations of full-batch gradient descent. The FEM-trained DNN converged to $\mathcal{L} = 7.611 \times 10^{-5}$, and when compared to the IGFEM performance shown in Table 1, was considered sufficiently accurate to implement in multiscale structural optimization.

Once trained, the FEM-trained DNN and IGFEM-trained DNN were compared to analytical IGFEM in two concurrent multiscale structural optimizations. A 2D, plane-strain structure was considered. Each microstructure was defined by a base material ($E_1 = 1$ Pa, $\nu = 0.3$) and void material ($E_1 = 10^{-6}$ Pa, $\nu = 0.3$) to apply elliptical inclusions. The elliptical inclusion's radii (r_1 and r_2) were limited to $\alpha_{\min} = 10\%$ and $\alpha_{\max} = 80\%$ of the unit cell width, while the volume constraint of the entire structure was defined as $V = V_x = 0.75$. Both optimizations were performed using the MMA algorithm for gradient-based optimization (Svanberg 1987). Although this macroscale mesh is too coarse to adequately model a multiscale structure and the separation between scales is not adequate (cf. Sect. 2), the mesh is appropriate for comparing our models for homogenization, as we can still directly compare the optimization convergence of an analytical IGFEM model against its DNN alternatives.

4.2.1 Compliance minimization verification example

For the first optimization, the domain presented in Sect. 4 was optimized to minimize the compliance of a 12×4 2D structure (Fig. 8a). Over 100 iterations, three methods were compared for the microstructure sensitivity: (i) the IGFEM method and (29), (ii) ML-based differentiation (28) with an FEM-trained DNN, and (iii) ML-based differentiation (28) with an IGFEM-trained DNN. The three methods are compared in Fig. 8c with the optimization results shown in Fig. 8d–f. Evaluated analytically using IGFEM, the converged compliance values Θ_c were 125.4, 124.6, and 125.0 for the IGFEM, IGFEM-trained DNN, and FEM-trained DNN designs, respectively; the converged objective was 66% of the original objective for all three designs. Furthermore, the optimized designs all agree well with the density-based topology optimization result (120×40 mesh, filtered with radius $r = 1.5$, and penalized by $p = 4$). Shown in Fig. 8b the density-based topology optimization result produced similar material concentrations as the multiscale design results. As a reference, on a single PC with Intel Core i7 (6x 2.20 GHz) and 32 GB of DDR4RAM, the optimization in Fig. 8d took roughly 8 h, while the ML optimizations each took roughly 1 min with 47 min of data generation time and 11 min of model training. The data generation and model training times are one-time costs for each training dataset (training is only necessary once for each microstructure geometry), so the same model can be reused for multiple macroscale structures; the model must be trained just once for a given microstructure parameterization and may be reused as a model for homogenization in any context.

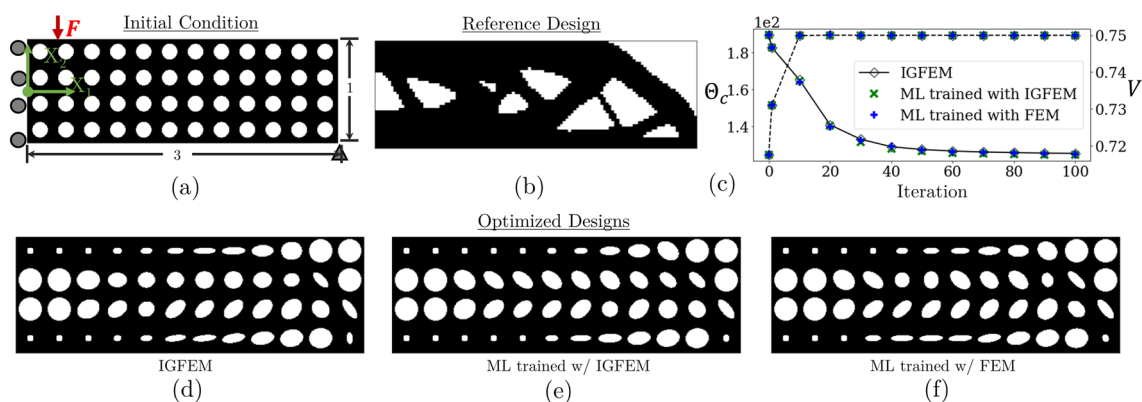


Fig. 8 **a** The boundary conditions for the compliance minimization [cf. (11)] problem used to validate the DNN in multiscale structural optimization. **b** The density-based topology optimization result is included as a reference for the same problem. **c** Convergence behavior

of the IGFEM optimization, ML trained with IGFEM data, and ML trained with FEM data is illustrated over 100 iterations. **d–f** The optimization results are shown for the three models of homogenization

4.2.2 Prescribed deformation verification example

The second optimization example uses the parameterized microstructures to achieve a prescribed deformation. For a 12×6 2D structure compressed 20% of its length, a target displacement was assigned to the structure's free edges (Fig. 9a). The target displacement $\mathbb{U}_T$ is defined by cosine waves for the X_2 component

$$\begin{aligned} \mathbb{U}_{T2}(X_1, X_2) &= c_1 - c_1 \cos(2X_1\pi/c_2), \\ X_2 &= 0.5 \vee X_2 = -0.5 \end{aligned} \quad (31)$$

for an amplitude c_1 and period c_2 . For this example, c_1 is 2.5% of the beam width, and c_2 is 50% of the beam length. A large prescribed strain is used here to sufficiently differentiate optimization results. Over 100 iterations, the three design paradigms—IGFEM, ML trained with IGFEM data, and ML trained with FEM data—converged to similar measured performance (Fig. 9b). Evaluated analytically using IGFEM, the converged objective values Θ_p were 1.46×10^{-5} , 2.44×10^{-5} , and 2.57×10^{-5} for the IGFEM, IGFEM-trained DNN, and FEM-trained DNN designs, respectively; relative to the initial objective of $\Theta_p = 2.72 \times 10^{-4}$, the converged objectives were 5.4%, 9.0%, and 9.4% of the objective for the IGFEM, IGFEM-trained DNN, and FEM-trained DNN designs, respectively. Additionally, converged designs were geometrically similar for all three paradigms (Fig. 9c–e). We do not address buckling at either macro- or microscales, but we do observe design patterns that align with the diagonal, shear buckling behavior observed in Vilardell et al. (2019).

For both compliance and prescribed displacement examples, the FEM-trained DNN, IGFEM-trained DNN, and analytical IGFEM models produced homogenized properties and shape sensitivities for concurrent multiscale structural optimization. All three approaches produced nearly identical designs in both performance and topology. In these applied examples, we note that the previously reported errors in shape sensitivity (cf. Figs. 6, 7) resulted in no adverse effects in the structural optimization process. Instead, we observed computation time decrease by multiple orders of magnitude. Furthermore, the IGFEM-trained and FEM-trained DNNs were shown to produce practically identical optimized structures. This phenomenon, which leads to many local optima that satisfy the design constraints, is further explored in Sect. 5.1. The DNN was able to infer a smooth function of material interface from the imperfect, discontinuous surface representations of FEM, permitting extension to DNN-driven optimization of highly parameterized, geometrically complex microarchitectures represented by computationally efficient FEM models.

5 Examples

5.1 Elliptical inclusion

With increased computational efficiency afforded by the ML model for homogenization, the number of unique microarchitectures in each structure can be increased. The following examples present some key findings of this increased design space using the ML model trained with FEM data.

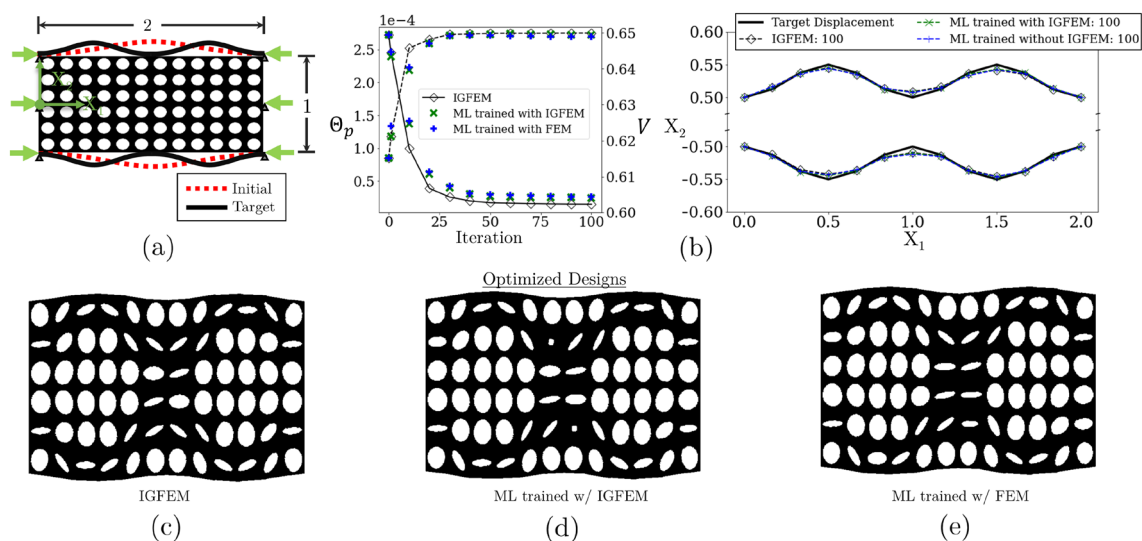


Fig. 9 **a** The initial displacement (red-dotted line) and target displacement (black-solid line) are shown for the verification example formulated as a prescribed displacement optimization [cf. (12)]. **b** The convergence behavior is shown for the IGFEM optimization, ML trained

with IGFEM data, and ML trained with FEM data. **c–e** The results for the three methods of optimization are shown in their deformed conditions. (Color figure online)

Even with this rather simple geometry, there exists some important properties of the concurrent microscale design space: (i) a dependence on initial condition, (ii) a dependence on the macroscale mesh, and (iii) the existence of multiple local optima. The elliptical inclusion of Fig. 4a is again considered with three parameters per element $\mathbf{x}_e = \{r_1, r_2, \theta\}$ with radii (r_1 and r_2) limited to $\alpha_{\min} = 0\%$ and $\alpha_{\max} = 99\%$ of the unit cell width.

The dependence on initial condition is evident in Fig. 10, where an optimization for prescribed deformation was performed using two sets of initial elliptical inclusions. The prescribed deformation is defined similar to (31)

$$\mathbb{U}_{T2}(X_1, X_2) = c_1 - c_1 \cos(2X_1\pi/c_2), \quad (32)$$

$$X_2 = 0$$

for an amplitude c_1 set to 25% of the beam width and period c_2 set to the beam length. In the first group, the initial condition was set to $\mathbf{x}_e^{(\text{init})} = \{0.1, 0.1, 0\}$ for each unit cell, while the second group's initial condition was set to $\mathbf{x}_e^{(\text{init})} = \{0.35, 0.35, 0\}$. Both optimizations feature a prescribed deformation problem for a elastic 2D structure compressed 10% of its length and use the ML model trained with FEM data for microscale analysis and sensitivity. The target displacement is defined by (32) for c_1 equal to 25% of beam width and $c_2 = 10$. Figure 10b shows the results of an unconstrained optimization, where the initial conditions converged to different passable designs with $\Theta_p^{(1)} = 2.093 \times 10^{-4}$ and $\Theta_p^{(2)} = 4.329 \times 10^{-4}$ as evaluated by the FEM-trained DNN. A volume constraint of $V = 0.75$ was then applied to each initial condition (Fig. 10c), and the multiscale optimization converged to nearly identical designs with $\Theta_p^{(1)} = 1.380 \times 10^{-4}$ and $\Theta_p^{(2)} = 1.360 \times 10^{-4}$ as evaluated

by the FEM-trained DNN. For both cases of constrained and unconstrained optimization, the ML model for microscale sensitivity performed as expected and remained stable during multiscale optimization.

Concurrent multiscale design is also mesh-dependent. To illustrate this phenomenon, a prescribed deformation problem was considered for a beam compressed 10% of its length with the ML model trained with FEM data applied for microscale analysis and sensitivity. Two mesh fidelities are introduced in Fig. 11 with identical initial conditions and volume constraints; an additional example is provided that enforces symmetry along the beam's length via a fixed displacement boundary condition in X_2 . Over 100 iterations, Θ_p converged to 1.556×10^{-5} , 4.604×10^{-6} , and 1.941×10^{-6} for the 12×6 , 48×24 , and 48×12 macroscale meshes as evaluated by the FEM-trained DNN. Despite the varying objective values, the three designs follow similar trends with inclusions orienting in similar orientations and locations.

There also are multiple functionally equivalent designs in the multiscale design space. The ML-based model for microscale analysis captures this phenomenon as illustrated in Fig. 12. A reference design was considered as a baseline within the parameter space. As in the previous example, the design was compressed 10% of its length with the FEM-trained DNN applied for microscale analysis and sensitivity. For the multiple initial conditions outlined in Fig. 12, the multiscale structural optimization process produced multiple unique and functionally equivalent designs. For the five scenarios in Fig. 12, Θ_p converged to 4.687×10^{-10} , 4.258×10^{-9} , 1.282×10^{-8} , 1.1404×10^{-9} , and 1.999×10^{-10} as evaluated by the FEM-trained DNN.

Various applications of multiscale design for compliance minimization are shown in Fig. 13 for the 2D domain

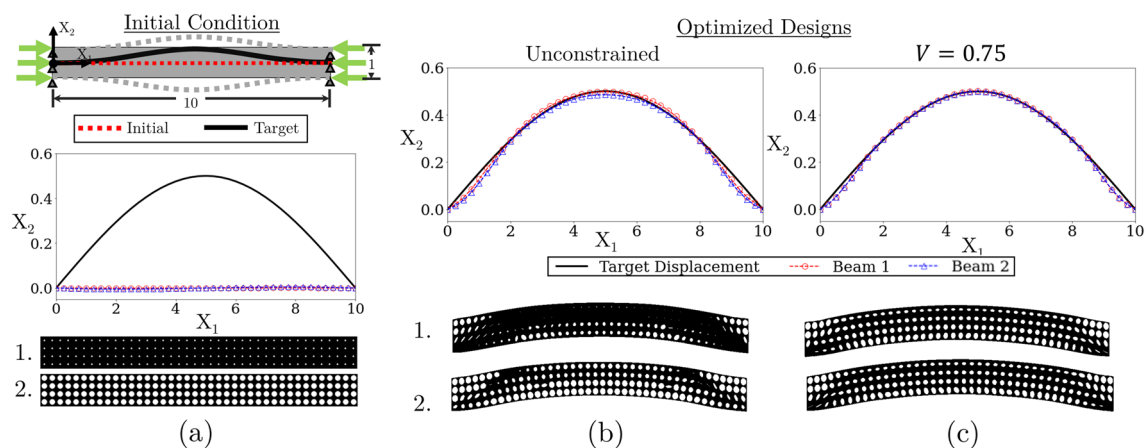


Fig. 10 **a** An optimization for prescribed deformation is formulated for an initial displacement (red-dotted line) and target displacement (black-solid line). Two initial conditions are considered to explore the multiscale design space using a ML model for microarchitecture. **b**

Unconstrained optimization produces different designs with similar performance, while **c** constrained optimization produces nearly identical designs and performance. (Color figure online)

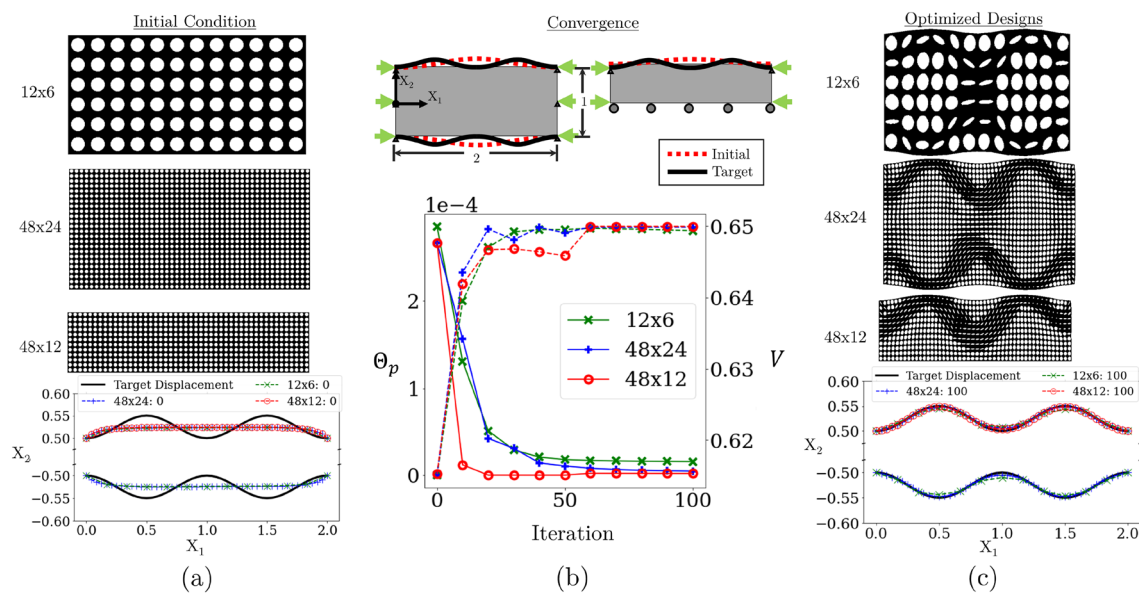


Fig. 11 **a** Three macro-scale domains are considered marked by their 12×6 , 48×24 , and 48×12 macro-scale meshes. **b** A prescribed deformation optimization problem is considered for an initial displacement (red-dotted line) and target displacement (black-solid line).

For illustrative purposes, the 48×12 uses a symmetry condition along $X_2 = 0$. **c** The optimized designs for the three cases are compared after 100 iterations. (Color figure online)

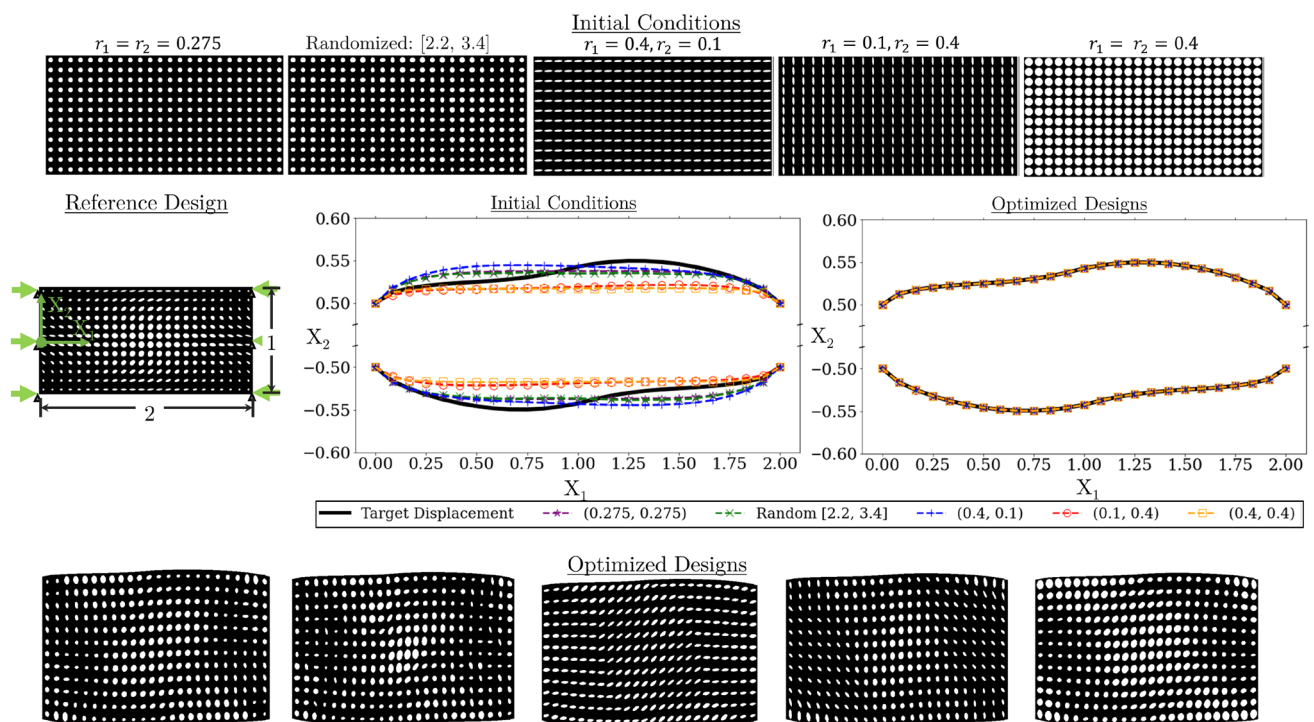
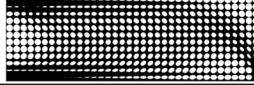


Fig. 12 Given a reference design and a prescribed deformation, the five different initial conditions for multiscale optimization with ML micro-structure analysis produce five unique optimized designs with nearly identical structural performance

presented in Sect. 4. Depending on the selection of volume constraints (V_ϕ, V_x, V), filter radius r , and penalization parameter p , a variety of optimized designs were achieved.

The density-based result (Fig. 13a) was used as a benchmark for Θ_c and different optimization formulations were with the ML model trained with FEM data applied for microscale

Fig. 13 Compliance minimization problems illustrate the use of design constraint to produce unique designs. *Objective values reported via analysis with a FEM-trained DNN

Optimized Designs		Macroscale	Microscale	Net Volume	Compliance
φ		$V_\varphi; r; p$	$V_x; r; p$	V	$\Theta_c/160$
					
(a)		$V_\varphi \leq 0.6$ $r = 1.5$ $p = 4$	-	0.6	1.00
(b)		-	$V_x \leq 0.6$	0.6	1.16*
(c)		$V_\varphi \leq 1.0$ $p = 3$	$V_x \leq 1.0$	0.6	1.12*
(d)		$V_\varphi \leq 0.8$ $r = 1.5$ $p = 3$	$V_x \leq 0.75$	0.6	1.03*
(e)		$V_\varphi \leq 0.8$ $r = 1.5$ $p = 3$	$V_x \leq 0.75$ $r = 1.5$	0.6	1.02*
(f)		$V_\varphi \leq 0.6$ $r = 1.5$ $p = 3$	$V_x \leq 1.0$ $r = 1.5$	0.6	1.00

analysis and sensitivity (Fig. 13b–f). Figure 13b presents the formulation with a macroscale distribution variable (i.e., $\varphi = 1$) and constrained microscale volume fraction V_x . As expected, this formulation performed worse than those with more design freedom and did not improve compliance over the density-based result. Figure 13c initializes both φ and x as design variable, but only constrains the net volume fraction V . The converged result produces a macroscale volume fraction of $V_\varphi = 0.92$ and microscale volume fraction $V_x = 0.65$ and improved performance over the restricted design space of Fig. 13b. Figure 13d–f introduces strict constraints for both macro- and microscale volumes with filtering of φ (Fig. 13d) or both φ and x (Fig. 13e–f). Design variable filtering via (14) did not drastically change the microarchitecture design, but did improve the objective calculation in Fig. 13e. Lastly, as the macroscale volume fraction approached the density-based formulation Fig. 13f, structural compliance converged to the density-based result (Fig. 13a). The relative importance of the macroscale and microscale design variables depends on the objective (Wu et al. 2021a); in this compliance optimization problem, the macroscale distribution of material dominates the microscale effects.

The concurrent multiscale design space is characterized by extensive dependence on initial condition, mesh fidelity, and constraints leading to many local optima. Any error inherent in the ML-based approximation of numerical homogenization is quickly overshadowed by these obstacles in multiscale design. Within this context, the FEM-trained DNN inferred a smooth approximation of a parameterized microstructure from imperfect training data and was integrated unobtrusively into multiscale optimization.

5.2 BioTruss

The previous sections introduced a microscale geometry parameterized by a simple elliptical inclusion. The DNN can also interpret highly parameterized geometry, accurately predicting both the homogenized elastic parameters and shape sensitivities. To illustrate this flexibility, we introduce the BioTruss, a microstructure geometry inspired by the twisted structures in trabecular bone. The BioTruss is defined using quadratic Bézier curves to represent material interfaces. In parametric form, the Bézier curve is defined as

$$\mathbf{B}(t)^{(i)} = (1-t)^2 \mathbf{f}_a^{(i)} + 2t(1-t) \boldsymbol{\beta}^{(i)} + t^2 \mathbf{f}_b^{(i)}, \quad (33)$$

$$0 \leq t \leq 1$$

for the 2D control points $\mathbf{f}_a$, $\mathbf{f}_b$, and $\boldsymbol{\beta}$. For a fixed starting point $\mathbf{f}_a$ and end point $\mathbf{f}_b$, the two components of $\boldsymbol{\beta}$ are tuned to achieve gradual variation in the curve (Fig. 14a). If $\mathbf{f}_a$ and $\mathbf{f}_b$ are defined in terms of the unit cell edge width a and b , the parameters $\{a, b, \boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \boldsymbol{\beta}^{(3)}, \boldsymbol{\beta}^{(4)}\}$ define the 2D BioTruss architecture (Fig. 14b). In the following examples, we set $a = b = 0.5$ to ensure smooth connectivity between unit cells.

Expansion into 3D is achieved by defining twelve Bézier curves for each edge of a unit cube then revolving the curves around each edge (Fig. 15a, b). To ensure uniform connectivity between cells, the term ζ is introduced with six components (one component for each cube face). The components of ζ define a face-thinning parameter that removes material within a cone with base diameter 1.1 and height ζ_i . Once material is removed at each face, the connections between each BioTruss cell are made invariant to changes of $\boldsymbol{\beta}$, so

Fig. 14 **a** A quadratic Bézier curve is shown to demonstrate sufficient geometric freedom, where two control points (i) β_1 and (ii) β_2 control the curve. **b** Four Bézier curves applied to a 2D microstructure produce unique geometry with 10 parameters

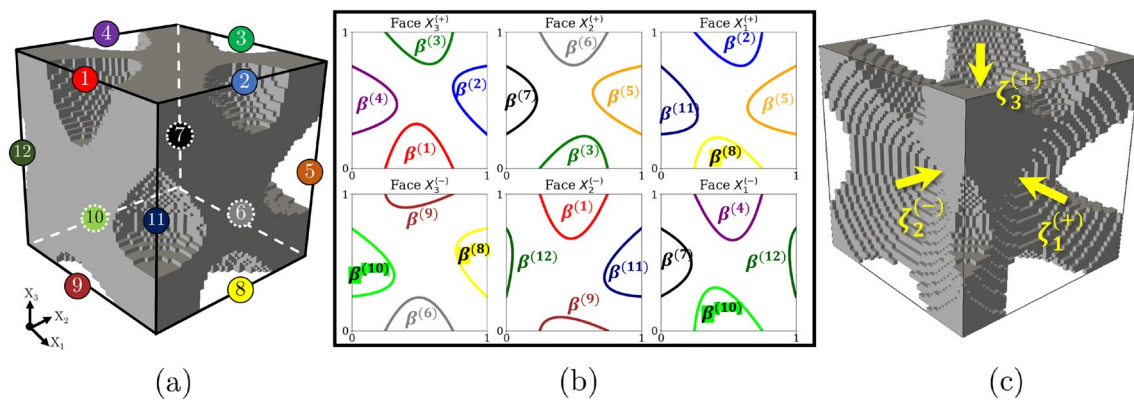
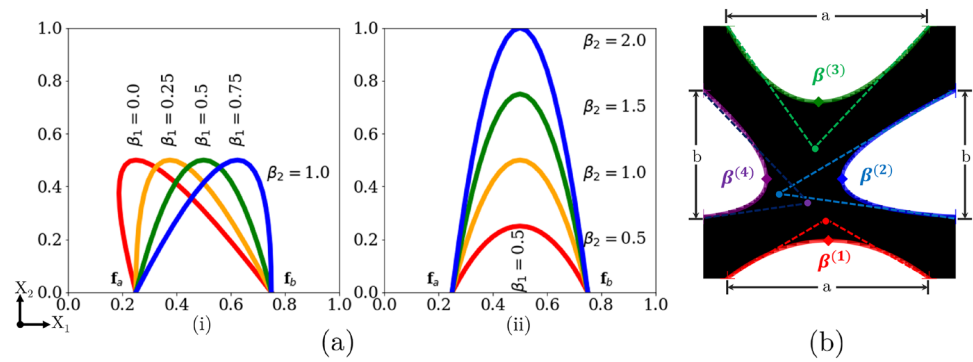


Fig. 15 **a** Twelve unique Bézier curves are revolved around the edges of a unit cube. **b** The twelve Bézier curves are shown as projections onto each of the cube's six faces. **c** The 3D BioTruss shown parameterized by 12 Bézier curves and 6 face-thinning parameters ζ

connectivity is ensured. Given twelve Bézier curves (24 parameters) for each edge and 6 face-thinning parameters, the BioTruss is fully defined in 3D by 30 unique parameters $\{\beta^{(1)}, \dots, \beta^{(12)}, \zeta^{(1)}, \dots, \zeta^{(6)}\}$ (Fig. 15c).

The DNN ($h = 3$, $n = 32$, all layers except the final layer activated by the sigmoid logistic function) was implemented to homogenize the BioTruss geometry. Training was performed as in Sect. 3.2. In 2D, a DNN trained with 667 examples of FEM data (50×50 mesh, linear 4-node elements in plane strain) achieved $\mathcal{L} = 8.23 \times 10^{-5}$ on a 333 example validation dataset. Another DNN trained with 3D data ($24 \times 24 \times 24$ mesh, linear 8 node elements) achieved $\mathcal{L} = 8.38 \times 10^{-6}$ on a 333 example validation dataset. Because both the 2D and 3D DNNs were sufficiently accurate compared to the validation examples (cf. Sect. 3.2), both models were considered appropriate for implementation into multiscale structural optimization.

The results of both 2D and 3D applications of the BioTruss in compliance-based optimizations (Fig. 16) agree with previous designs. The 2D example (Fig. 16) was characterized in the macroscale by $V_\varphi \leq 0.8$, $r = 1.5$, and $p = 3$ and in the microscale by $V_x \leq 0.75$ and $r = 1.5$. After 100 iterations of multiscale optimization, the resulting

compliance was 13% higher than the density-based result (Fig. 13a), likely because filtering effects limited spatial variation of microarchitectures. In 3D (Fig. 16b–d), the DNN-based design of each BioTruss in a $36 \times 6 \times 12$ mesh produced spatially varying architecture that emulate the previously observed behavior in 2D. Although the BioTruss is a sub-optimal parameterization for compliance minimization (Sigmund 1994), the DNN surrogate model was still able to produce sufficiently varying multiscale features within the geometric limits imposed by its parameterization (Fig. 17). This characteristic implies that the DNN can interpret shape sensitivity from highly parameterized microarchitectures.

The BioTruss was also implemented into a prescribed displacement problem in 2D and 3D (Fig. 18). For an 18×6 mesh in 2D, the prescribed displacement was defined using (31) for c_1 set to 2.5% of the beam width, and c_2 set to 50% of the beam length. After 100 iterations of concurrent multiscale optimization, the DNN-based model reduced Θ_p from 2.61×10^{-4} to 8.78×10^{-6} (evaluated via the DNN model for homogenization). The 3D prescribed displacement problem defined the target displacement as

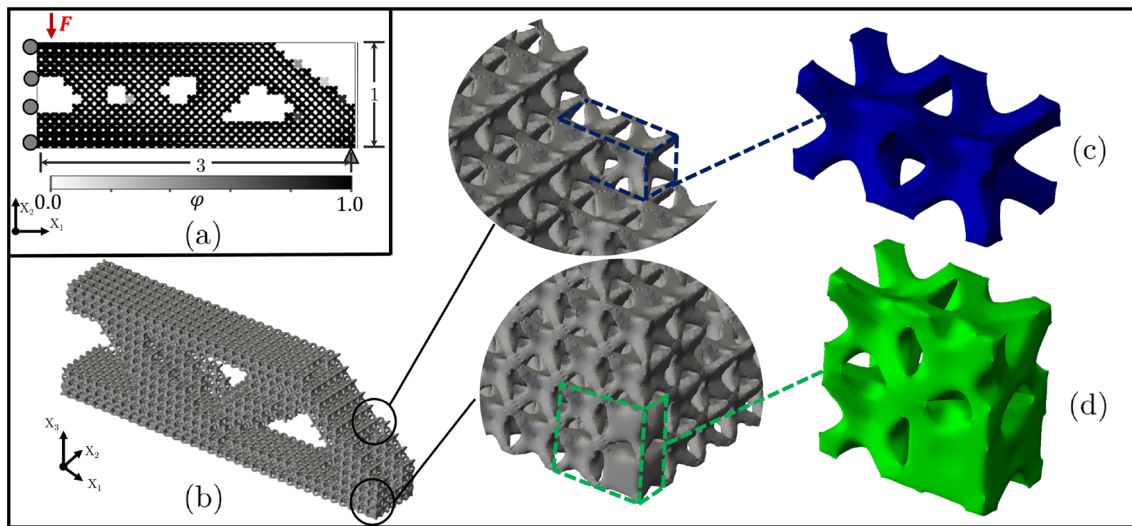


Fig. 16 A compliance optimization is shown for the **a** 2D BioTruss and **b** 3D BioTruss. The DNN surrogate model for homogenization produced spatially varying architecture from the parameterized cell geometry (**c**, **d**)

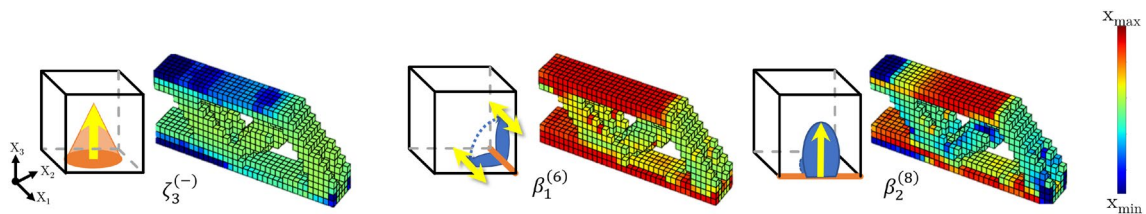


Fig. 17 The distribution of 3 out of 30 microscale parameters is shown for the compliance minimization problem of Fig. 16b

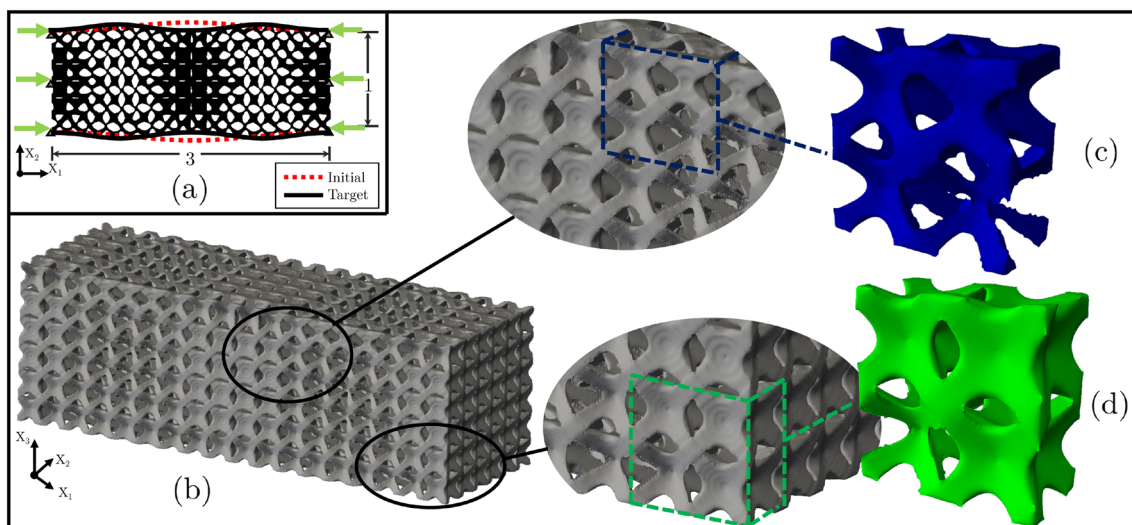


Fig. 18 A prescribed displacement optimization is shown for the **a** 2D BioTruss and **b** 3D BioTruss. **c** and **d** The variance in 3D microarchitecture agrees with previous designs, producing increasingly intricate geometries

$$\begin{aligned}
\mathbb{U}_{T2}(X_1, X_2) &= c_1 - c_1 \cos(2X_1\pi/c_2), \\
X_2 &= 0.5 \vee X_2 = -0.5 \\
\mathbb{U}_{T3}(X_1, X_3) &= c_1 - c_1 \cos(2X_1\pi/c_2), \\
X_3 &= 0.5 \vee X_3 = -0.5
\end{aligned} \tag{34}$$

for c_1 set to 5% of the beam width and c_2 set to 50% of the beam length. The DNN-based model reduced Θ_p from 4.98×10^{-2} to 1.26×10^{-2} (evaluated via the DNN model for homogenization) after 100 iterations of multiscale optimization. The optimized result is approximately 25% of the initial condition; as shown in Fig. 19, this optimization reached the geometric limits of the microarchitecture. As with the compliance minimization application, the prescribed displacement optimization of the 3D BioTruss produced significant variations in microscale geometry for the 30 design parameters (Fig. 19).

6 Conclusion

In this paper, the DNN was used as a surrogate model for numerical homogenization in the context of concurrent multiscale design optimization. Using a parameterized unit cell, we compared the shape sensitivities produced through back propagation of the DNN with the analytical sensitivities produced via IGFEM. These results indicated a clear positive correlation between the DNN's evaluation accuracy and its accuracy in producing shape sensitivities through back propagation. Therefore, the DNN offers an appropriately smooth approximation of parameterized homogenization provided it is successfully trained.

Through numerous optimization examples, we compared IGFEM with a IGFEM-trained DNN and FEM-trained DNN. Although the FEM material interface is discontinuous, the continuous DNN approximation of the geometry provided reliable shape sensitivities in concurrent multiscale optimization. This affirms our conclusion that a sufficiently accurate DNN also provides sufficiently accurate shape sensitivity for navigating the design space. Because FEM training data were sufficient to train an accurate DNN surrogate model, the resulting DNN was effective in providing

sensitivity information during design optimization. With this revelation, we extended the DNN into 3D with the highly parameterized BioTruss. With numerous design optimizations and an experimental validation, we illustrated the DNN's effectiveness in exploring the microscale design space. The DNN surrogate model extends the multiscale design space, allowing more geometric freedom in parameterizing the microarchitecture in spatially varying structures. Future work in parameterized concurrent multiscale optimization may explore the imposition of manufacturing constraints on the parameterization.

We implemented the DNN as a tool for exploring the multiscale design space, which is distinctly different than a tool for multiscale analysis. Accurate analysis of multiscale structures will require a larger computational investment including high-fidelity microscale data and analysis of local, nonlinear structural deformation. Furthermore, the designs presented in this work (particularly the BioTruss) demonstrate the increased geometric freedom afforded by the DNN, but are not necessarily optimal representations. Future work should couple results from material optimization to define the microarchitecture parameterization.

Appendix 1: IGFEM shape sensitivity of the homogenized elasticity tensor

This section introduces the relevant IGFEM sensitivity analysis for the homogenized elasticity tensor in relation to material shape parameters. We begin with the energy-based expression of the homogenized elasticity tensor:

$$C^H = C_{ij}^H = \frac{1}{|Y|} \sum_{e_\mu=1}^{N_\mu} \left(u_{e_\mu}^{0(i)} - u_{e_\mu}^{(i)} \right)^T k_{e_\mu} \left(u_{e_\mu}^{0(j)} - u_{e_\mu}^{(j)} \right) \tag{35}$$

$$= \frac{1}{|Y|} \sum_{e_\mu=1}^{N_\mu} \left(\epsilon_{e_\mu}^{0(i)} - \epsilon_{e_\mu}^{(i)} \right)^T C_{e_\mu} \left(\epsilon_{e_\mu}^{0(j)} - \epsilon_{e_\mu}^{(j)} \right). \tag{36}$$

If we simplify the expression to a single component of the homogenized tensor and omit the subscripts used to indicate

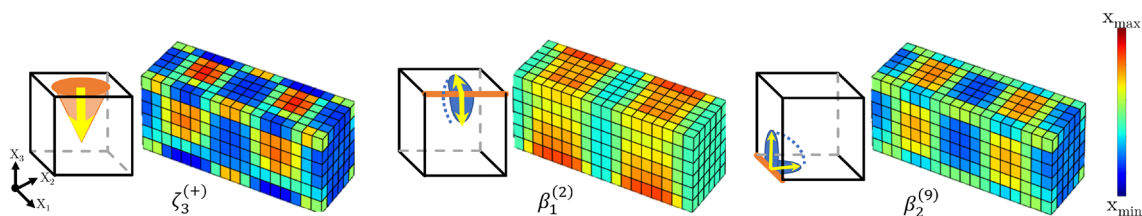


Fig. 19 The distribution of 3 out of 30 microscale geometries is shown for the prescribed displacement problem in Fig. 18b

microscale element quantities, we can write an element's contribution to the homogenized tensor as

$$c = \int_{\Omega_e} (\epsilon^0 - \epsilon)^T \mathbf{C}_{e_\mu} (\epsilon^0 - \epsilon) d\Omega. \quad (37)$$

We remark that in this expression, only the strain ϵ is a function of the shape parameters. The strain ϵ^0 is prescribed, the element-wise constitutive relation $\mathbf{C}_{e_\mu}$ is not a function of the design variables in the IGFEM shape optimization formulation.

The strain ϵ can be represented as the function $\epsilon(\mathbf{X}(\mathbf{x}), \mathbf{x}) = \mathbb{B}(\mathbf{X}(\mathbf{x}), \mathbf{x}) \mathbb{U}(\mathbf{X}(\mathbf{x}), \mathbf{x})$ for the shape parameters $\mathbf{x}$. Hereafter, we consider a single shape parameter x_i . We introduce the simple notation $\frac{\partial \mathbb{B}}{\partial x_i}$ as the expression for the shape derivative of $\mathbb{B}$. The defining feature of IGFEM is $\mathbb{B}(\mathbf{X}(\mathbf{x}), \mathbf{x})$, where the strain–displacement is a function of the shape parameters; for more information on the IGFEM implementation of $\frac{\partial \mathbb{B}}{\partial x_i}$, see Najafi et al. (2015, 2017, 2021) and Brandyberry et al. (2020). The shape material derivative of $\mathbb{U}$ is introduced as $\mathbb{U}_i^*$. Following these definitions, the shape derivative of ϵ is expressed element-wise as

$$\frac{d\epsilon}{dx_i} = \frac{\partial \mathbb{B}}{\partial x_i} \mathbb{U}_e + \mathbb{B} \mathbb{U}_{ei}^*, \quad (38)$$

while the shape derivative $\mathbb{U}_i^*$ is evaluated through the following pseudo-problem:

$$\mathbb{K} \mathbb{U}_i^* = \mathbb{P}_{ps}^i = -\frac{\partial \mathbb{K}}{\partial x_i} \mathbb{U} + \frac{\partial \mathbb{F}}{\partial x_i}. \quad (39)$$

The pseudo-problem is assembled from the element quantities $\frac{\partial \mathbf{K}_e}{\partial x_i}$ and $\frac{\partial \mathbf{F}_e}{\partial x_i}$. We assert that the material derivative $\frac{\partial \mathbf{C}_{e_\mu}}{\partial x_i}$ is zero, so the element stiffness derivative follows:

$$\frac{\partial \mathbf{K}_e}{\partial x_i} = \int_{\Omega_e} \left(\frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \mathbb{B} + \mathbb{B}^T \mathbf{C}_{e_\mu} \frac{\partial \mathbb{B}}{\partial x_i} + \mathbb{B}^T \mathbf{C}_{e_\mu} \mathbb{B} \text{div}(\mathbb{V}_i) \right) d\Omega, \quad (40)$$

where we note that $\frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \mathbb{B}$ is symmetric and $\text{div}(\mathbb{V}_i)$ follows from the shape velocity term (Najafi et al. 2015). For the homogenization case where $\frac{d\epsilon_0}{dx_i} = 0$, the element force derivative is

$$\frac{\partial \mathbf{F}_e}{\partial x_i} = \int_{\Omega_e} \left(\frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \epsilon^0 + \mathbb{B}^T \mathbf{C}_{e_\mu} \epsilon^0 \text{div}(\mathbb{V}_i) \right) d\Omega, \quad (41)$$

where

$$\epsilon^0 = \mathbb{B} \mathbb{U}_e^0. \quad (42)$$

Recalling that only ϵ is a function of the design parameter with its shape sensitivity in (38), then the sensitivity

expression of the homogenized elasticity tensor can be defined similar to (40):

$$\begin{aligned} \frac{dc}{dx_i} = & - \int_{\Omega_e} \left[\left(\frac{\partial \mathbb{B}}{\partial x_i} \mathbb{U}_e + \mathbb{B} \mathbb{U}_{ei}^* \right)^T \mathbf{C}_{e_\mu} (\epsilon^0 - \epsilon) \right. \\ & + (\epsilon^0 - \epsilon)^T \mathbf{C}_{e_\mu} \left(\frac{\partial \mathbb{B}}{\partial x_i} \mathbb{U}_e + \mathbb{B} \mathbb{U}_{ei}^* \right) \\ & \left. + (\epsilon^0 - \epsilon)^T \mathbf{C}_{e_\mu} (\epsilon^0 - \epsilon) \text{div}(\mathbb{V}_i) \right] d\Omega. \end{aligned} \quad (43)$$

Next we target the term $\frac{\partial \mathbb{B}}{\partial x_i} \mathbb{U}_e + \mathbb{B} \mathbb{U}_{ei}^*$. If we combine the expression for the pseudo-element with the relation $\mathbb{K}_e = \mathbb{B}^T \mathbf{C}_{e_\mu} \mathbb{B}$, the pseudo-element can be used to eliminate $\mathbb{B} \mathbb{U}_{ei}^*$ in (43) using the element-wise pseudo-force of (39):

$$\begin{aligned} & \mathbb{B}^T \mathbf{C}_{e_\mu} \left(\frac{\partial \mathbb{B}}{\partial x_i} \mathbb{U}_e + \mathbb{B} \mathbb{U}_{ei}^* \right) \\ &= \mathbb{B}^T \mathbf{C}_{e_\mu} \frac{\partial \mathbb{B}}{\partial x_i} \mathbb{U}_e + \mathbb{P}_{pse}^i \\ &= \mathbb{B}^T \mathbf{C}_{e_\mu} \frac{\partial \mathbb{B}}{\partial x_i} \mathbb{U}_e - \frac{\partial \mathbf{K}_e}{\partial x_i} \mathbb{U} + \frac{\partial \mathbf{F}_e}{\partial x_i} \\ &= \mathbb{B}^T \mathbf{C}_{e_\mu} \frac{\partial \mathbb{B}}{\partial x_i} \mathbb{U}_e \\ &\quad - \left(\frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \mathbb{B} + \mathbb{B}^T \mathbf{C}_{e_\mu} \frac{\partial \mathbb{B}}{\partial x_i} + \mathbb{B}^T \mathbf{C}_{e_\mu} \mathbb{B} \text{div}(\mathbb{V}_i) \right) \mathbb{U}_e \\ &\quad + \left(\frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \epsilon^0 + \mathbb{B}^T \mathbf{C}_{e_\mu} \epsilon^0 \text{div}(\mathbb{V}_i) \right) \\ &= \frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \mathbb{B} \mathbb{U}_e^0 - \frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \mathbb{B} \mathbb{U}_e \\ &\quad + \mathbb{B}^T \mathbf{C}_{e_\mu} \mathbb{B} \text{div}(\mathbb{V}_i) \mathbb{U}_e^0 - \mathbb{B}^T \mathbf{C}_{e_\mu} \mathbb{B} \text{div}(\mathbb{V}_i) \mathbb{U}_e \\ &= \left(\frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \mathbb{B} + \mathbb{B}^T \mathbf{C}_{e_\mu} \mathbb{B} \text{div}(\mathbb{V}_i) \right) (\mathbb{U}_e^0 - \mathbb{U}_e). \end{aligned} \quad (44)$$

Applying the symmetry of $\frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \mathbb{B}$ in (44), we conclude

$$\begin{aligned} & \mathbb{B}^T \mathbf{C}_{e_\mu} \left(\frac{\partial \mathbb{B}}{\partial x_i} \mathbb{U}_e + \mathbb{B} \mathbb{U}_{ei}^* \right) \\ &= \left(\mathbb{B}^T \mathbf{C}_{e_\mu} \frac{\partial \mathbb{B}}{\partial x_i} + \mathbb{B}^T \mathbf{C}_{e_\mu} \mathbb{B} \text{div}(\mathbb{V}_i) \right) (\mathbb{U}_e^0 - \mathbb{U}_e) \\ &= \mathbb{B}^T \mathbf{C}_{e_\mu} \left(\frac{\partial \mathbb{B}}{\partial x_i} + \mathbb{B} \text{div}(\mathbb{V}_i) \right) (\mathbb{U}_e^0 - \mathbb{U}_e) \\ &\Rightarrow \frac{\partial \mathbb{B}}{\partial x_i} \mathbb{U}_e + \mathbb{B} \mathbb{U}_{ei}^* = \left(\frac{\partial \mathbb{B}}{\partial x_i} + \mathbb{B} \text{div}(\mathbb{V}_i) \right) (\mathbb{U}_e^0 - \mathbb{U}_e). \end{aligned} \quad (45)$$

Using (45) in the expression for constitutive sensitivity (43), we produce

$$\begin{aligned}
\frac{dc}{dx_i} &= \int_{\Omega_e} \left[- \left(\left(\frac{\partial \mathbb{B}}{\partial x_i} + \mathbb{B} \operatorname{div}(\mathbb{V}_i) \right) (\mathbb{U}_e^0 - \mathbb{U}_e) \right)^T \mathbf{C}_{e_\mu} (\boldsymbol{\varepsilon}^0 - \boldsymbol{\varepsilon}) \right. \\
&\quad \left. - (\boldsymbol{\varepsilon}^0 - \boldsymbol{\varepsilon})^T \mathbf{C}_{e_\mu} \left(\left(\frac{\partial \mathbb{B}}{\partial x_i} + \mathbb{B} \operatorname{div}(\mathbb{V}_i) \right) (\mathbb{U}_e^0 - \mathbb{U}_e) \right) \right. \\
&\quad \left. + (\boldsymbol{\varepsilon}^0 - \boldsymbol{\varepsilon})^T \mathbf{C}_{e_\mu} (\boldsymbol{\varepsilon}^0 - \boldsymbol{\varepsilon}) \operatorname{div}(\mathbb{V}_i) \right] d\Omega \\
&= - \int_{\Omega_e} \left[- (\mathbb{U}_e^0 - \mathbb{U}_e)^T \frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \mathbb{B} (\mathbb{U}_e^0 - \mathbb{U}_e) \right. \\
&\quad - (\mathbb{U}_e^0 - \mathbb{U}_e)^T \mathbb{B}^T \mathbf{C}_{e_\mu} \mathbb{B} \operatorname{div}(\mathbb{V}_i) (\mathbb{U}_e^0 - \mathbb{U}_e) \\
&\quad - (\mathbb{U}_e^0 - \mathbb{U}_e)^T \mathbb{B}^T \mathbf{C}_{e_\mu} \frac{\partial \mathbb{B}}{\partial x_i} (\mathbb{U}_e^0 - \mathbb{U}_e) \\
&\quad - (\mathbb{U}_e^0 - \mathbb{U}_e)^T \mathbb{B}^T \mathbf{C}_{e_\mu} \mathbb{B} \operatorname{div}(\mathbb{V}_i) (\mathbb{U}_e^0 - \mathbb{U}_e) \\
&\quad \left. + (\boldsymbol{\varepsilon}^0 - \boldsymbol{\varepsilon})^T \mathbf{C}_{e_\mu} (\boldsymbol{\varepsilon}^0 - \boldsymbol{\varepsilon}) \operatorname{div}(\mathbb{V}_i) \right] d\Omega \\
&= \int_{\Omega_e} \left[(\mathbb{U}_e^0 - \mathbb{U}_e)^T \frac{\partial \mathbb{B}^T}{\partial x_i} \mathbf{C}_{e_\mu} \mathbb{B} (\mathbb{U}_e^0 - \mathbb{U}_e) \right. \\
&\quad + (\mathbb{U}_e^0 - \mathbb{U}_e)^T \mathbb{B}^T \mathbf{C}_{e_\mu} \mathbb{B} \operatorname{div}(\mathbb{V}_i) (\mathbb{U}_e^0 - \mathbb{U}_e) \\
&\quad \left. + (\mathbb{U}_e^0 - \mathbb{U}_e)^T \mathbb{B}^T \mathbf{C}_{e_\mu} \frac{\partial \mathbb{B}}{\partial x_i} (\mathbb{U}_e^0 - \mathbb{U}_e) \right] d\Omega \\
&= (\mathbb{U}_e^0 - \mathbb{U}_e)^T \frac{\partial \mathbf{K}_e}{\partial x_i} (\mathbb{U}_e^0 - \mathbb{U}_e).
\end{aligned} \tag{46}$$

Using this element contribution to the shape sensitivity of the constitutive parameters, we recover the form in (29).

Appendix 2: Practical considerations

The appropriate DNN architecture and training procedure heavily depend on the application (that is, it depends on the function space to be emulated). For multiscale optimization problems employing homogenization, including applications in structural, thermal, and acoustic simulations that employ parameterized microstructures, this section may be used to generally guide the DNN training process. This section reviews some of the key issues associated with DNN training including vanishing/exploding gradients, batch size, and training dataset generation.

Gradient propagation

As the DNN trains, its weights are iteratively updated to improve some objective function. The back propagation procedure [cf. (9)] is used to update the weights and biases of the DNN. The convergence of these model parameters is not guaranteed; some combinations of model initialization and training procedures will produce unstable gradients, often referred to as vanishing or exploding gradients (Glorot and Bengio 2010; Goodfellow et al. 2016).

In this work, vanishing gradients were observed and are reported in Table 1 as the number of DNN hidden layers was increased past $L = 3$. Figure 20 illustrates the propagation of the DNN's Jacobian for a collection of architectures all trained

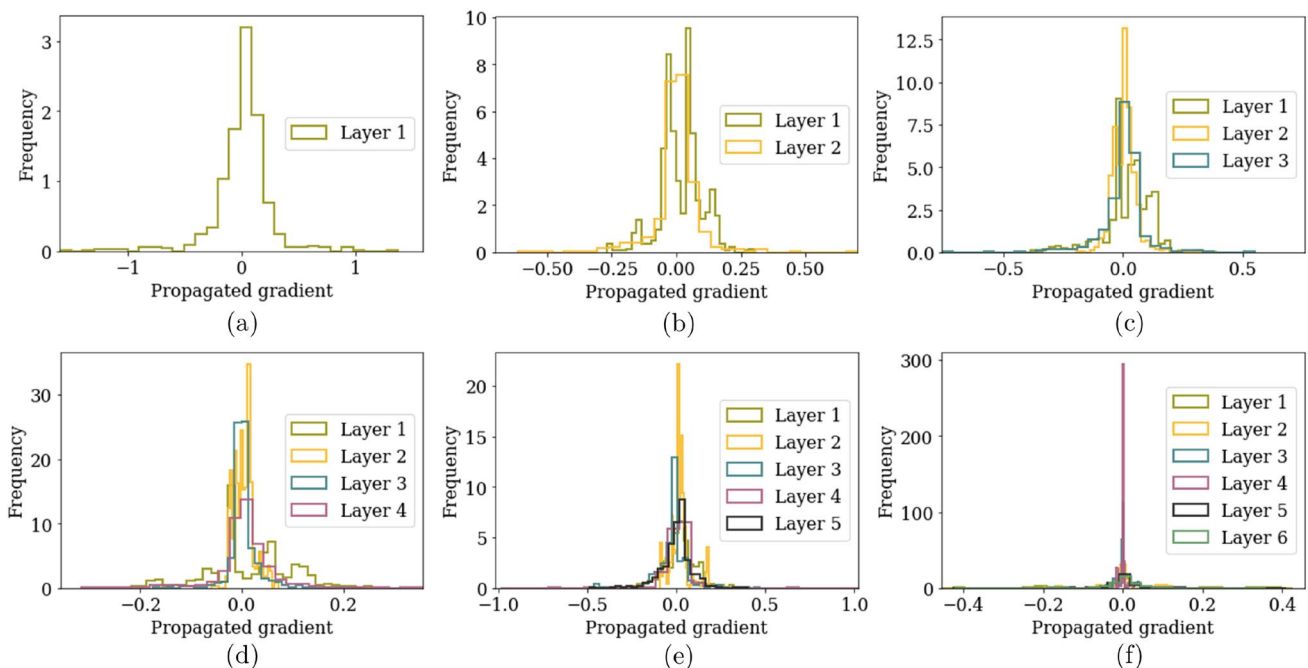


Fig. 20 The gradient produced through back propagation is represented as a histogram for a DNN of size $n = 32$ and $L = 1, 2, 3, 4, 5$ or 6 for (a–f), respectively. Each DNN was trained to homogenize the IGFEM ellipse microstructure using full-batch training

with 667 IGFEM elliptical training examples with an initial learning rate of 10^{-4} over 10^5 iterations of full-batch gradient descent. As the number of hidden layers increases, as in Fig. 20e, f, the Jacobian tends toward zero and caused the training failures reported in Table 1. In all examples, the vanishing gradient phenomena manifested during training and resulted in blatantly poor models. For the relatively small models in this work, if the training process was stable, then the DNN's Jacobian was adequate for applications in multiscale optimization.

Batch size

The batch size in a DNN training procedure refers to the number of training examples used to calculate the model's sensitivity for a given training iteration (Goodfellow et al. 2016). Full-batch training was implemented in this work because the training datasets are relatively small (100s to 1000s of examples) and an efficient training procedure was

desirable. Training in mini batches, generally samples of 4–32 training examples may improve generalization and robustness (Nikolakis et al. 2022; Novak et al. 2018). Table 2 compares the objective values for two identical DNN architectures trained via full-batch gradient descent and small-batch gradient descent (batch size = 32). Small-batch training did improve the DNN's performance as parameterization increased. The sensitivity, shown in Fig. 21, was inconsistently improved. Based on this evidence, the gains achieved through small-batch training do not significantly outweigh the additional training cost. For more complicated systems that require highly parameterized models, however, small-batch training may be necessary to build accurate surrogate models Fig. 22.

Training dataset size

A training dataset is necessary to construct a viable DNN surrogate model for engineering applications. The ideal training dataset captures the depth and complexity of the target function so that the DNN may learn a general and robust map within the function space. Whether due to excessively costly data generation or incalculable complexity, the ideal training dataset is not always feasible.

Parameterized homogenization is apt for building effective training datasets. Input parameters are bounded by geometric limits, and output parameters are bounded by the constitutive limits of the material. Given these conditions, it is possible to create a representative dataset with 100s to 1000s of examples that may be used to create a relatively small yet general surrogate model for homogenization. Figure 23 illustrates correlation between accurate execution and training dataset size for a DNN of $L = 3$ and $n = 32$. For more complicated geometric parameterizations and/or nonlinear physics, it is likely that more data are needed to capture the complexity of the feature space.

Table 2 The Loss values of (8) are reported as $\mathcal{L} \times 10^3$ for different DNN frameworks to compare a full-batch and small-batch training strategies

Layers L	Neurons n				
	8	16	32	64	128
1	0.0480 (0.0622)	0.0367 (0.0424)	0.0311 (0.0418)	0.0297 (0.0578)	0.0330 (0.0646)
2	0.0451 (0.0837)	0.0318 (0.0579)	0.0210 (0.0724)	0.0159 (0.0647)	0.0160 (0.0481)
3	3.0948 (0.3078)	0.0330 (0.2541)	0.0149 (0.0323)	0.0122 (0.0256)	0.0135 (0.0230)
4	3.6491 (0.2719)	0.0538 (0.2899)	0.0256 (0.0319)	0.0163 (0.0270)	0.0163 (0.0171)

Full-batch training results are denoted by parenthesis

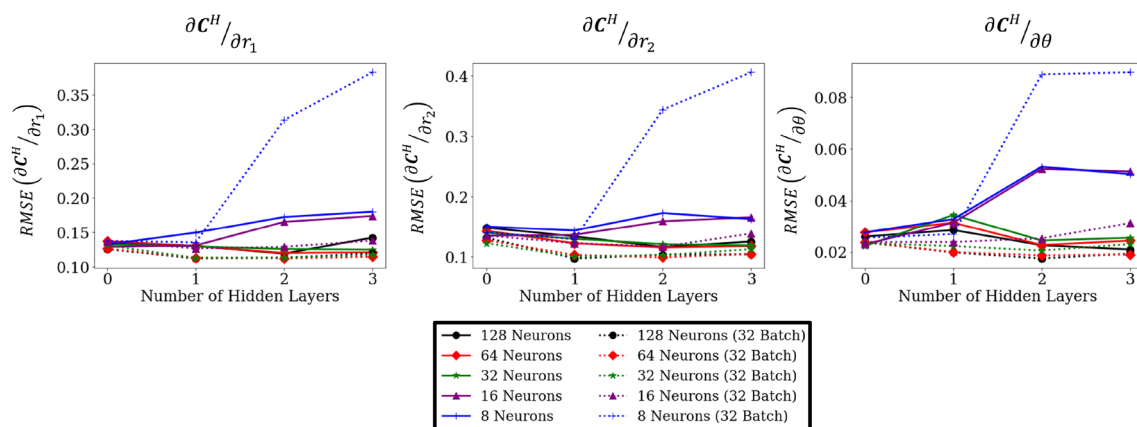


Fig. 21 The shape sensitivities are compared for full-batch and batch size 32 DNNs trained to homogenized the IGFEM ellipse microstructure. RMSE error is reported using Eq. (30)

Fig. 22 The gradient produced through back propagation is represented as a histogram for a DNN of size $n = 32$ and $L = 1, 2, 3$, or 4 for (a–d), respectively. Each DNN was trained to homogenize the IGFEM ellipse microstructure using a batch size of 32

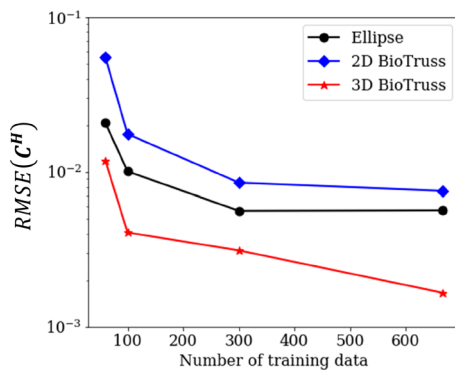
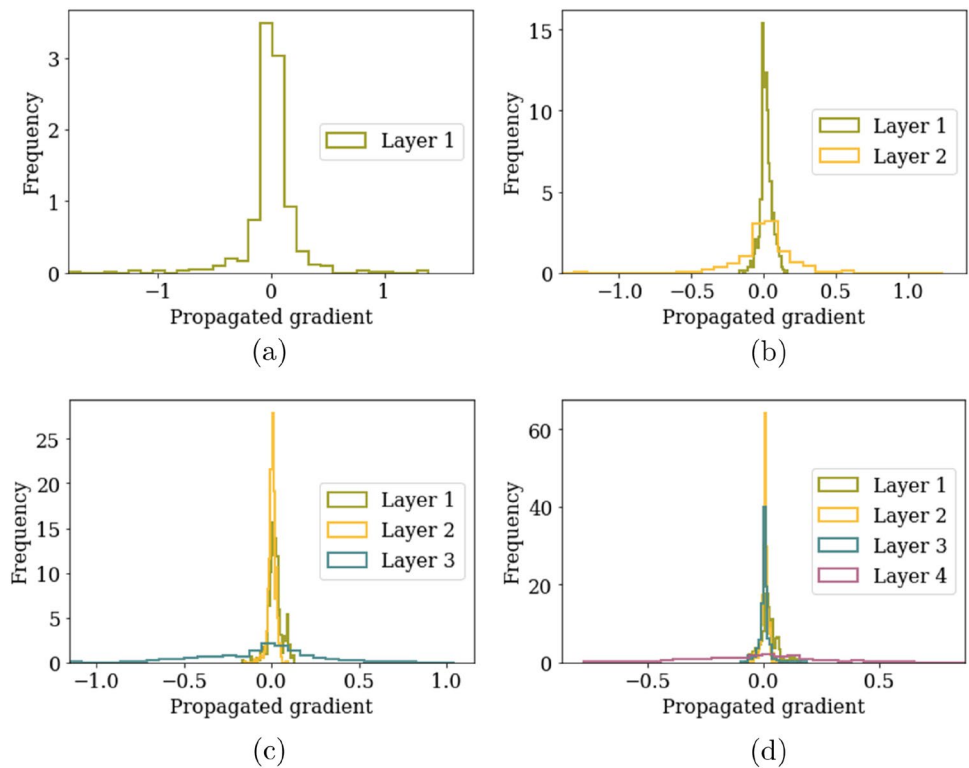


Fig. 23 The Ellipse microstructure (3 parameters per cell), the 2D BioTruss (8 parameters per cell), and the 3D BioTruss (30 parameters per cell) illustrate the correlation between number of training data and the accuracy of a trained DNN ($n = 32$, $L = 3$)

Homogenization in multiscale optimization

Homogenization assumes a significant separation of scales, approximating the local effects of a periodically varying microstructure (cf. Sect. 2). The examples presented in this work have largely focused on the numerical behavior of DNN surrogate models for homogenization in a selection of optimization exercises. Continued work through full-scale simulation and physical experimentation is necessary to judge the effects of homogenization on multiscale structures. This “Appendix” is presented as

a short illustrative study to show the limits of homogenization-based multiscale design.

The test case for experimental validation is derived from a prescribed deformation problem characterized by

$$\mathbb{U}_{T1} = \mathbb{U}_{T2} = \mathbb{U}_{T3} = \mathbf{0} \quad (47)$$

which targets the displacement of a zero Poisson’s ratio structure given the boundary conditions shown in Fig. 24a. Design optimization was performed using the FEM-informed DNN model for the 3D BioTruss, producing the $10 \times 10 \times 1$ structure shown in Fig. 24a after 100 iterations. Designs are compared using measured Poisson’s ratio of the macroscale structure

$$\nu = \frac{-\epsilon_{\text{lat}}}{\epsilon_{\text{long}}}, \quad (48)$$

where the strains ϵ_{lat} and ϵ_{long} are the lateral and longitudinal strains measured along the specimen’s centroidal axes. The initial uniform specimen $[\beta^{(i)} = \{0.5, 0.5\}_{i=1:12}; \zeta^{(i)} = 0.5_{i=1:6}]$ has a Poisson’s ratio of 0.33 as evaluated by FEM-based homogenization. After 200 iterations of design optimization ($V_x = 0.2$), the BioTruss design converged to a Poisson’s ratio of 0.00 (as evaluated by FEM-based homogenization). Because the design space reached the parameter limits imposed by the BioTruss geometry (Fig. 25), this specific microarchitecture formulation is likely unable to produce a negative Poisson’s ratio.

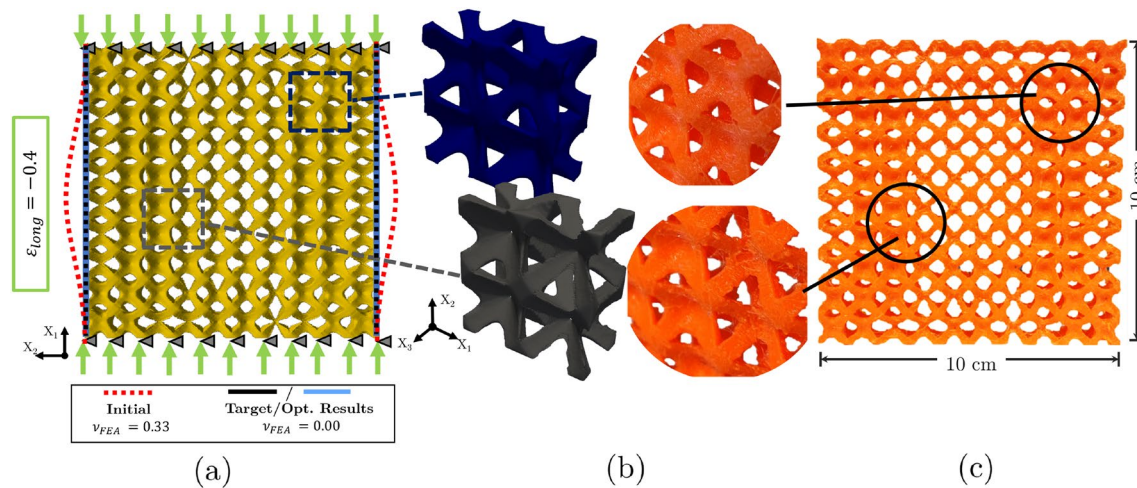


Fig. 24 **a** The boundary conditions and optimization result are shown for the test specimen with spatially varying microstructures **(b)**. **c** The manufactured specimen is shown in its undeformed condition

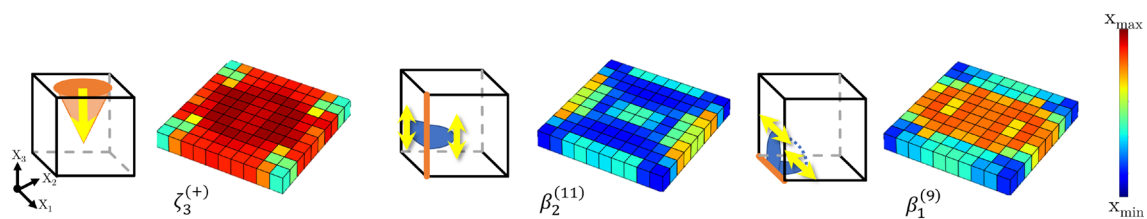


Fig. 25 A selection of the BioTruss features is illustrated to explore the optimization result of Fig. 24a

The design produced via DNN-driven multiscale optimization was manufactured using 3D printing of TPU 95a filament ($E_1 = 49$ MPa, $\nu = 0.32$ Lee et al. 2022) via fused deposition modeling (Fig. 24b, c). The properties of TPU 95a differ from the simulated fictitious material ($E_1 = 1$ Pa, $\nu = 0.30$), but because the structural deformation is displacement controlled, the deformation of both materials are sufficiently similar for comparison. Indeed both the fictitious material and TPU 95a produce an initial Poisson's ratio of 0.33 for the uniform specimen and 0.00 for the optimized structure, as evaluated by FEM-based homogenization.

The optimized design of TPU 95a microarchitectures was analyzed in the displacement controlled compression fixture shown in in Fig. 26. The Poisson's ratio was measured experimentally using ϵ_{lat} and ϵ_{long} measured along the specimen's respective centroidal axes. At $\epsilon_{long} = -0.10$, the calculated Poisson's ratio was -0.06 , and at $\epsilon_{long} = -0.20$, the measured Poisson's ratio was -0.02 . The variation between modeled ($\nu = 0.00$) and experimental Poisson's ratios is attributed to localized buckling near the compression plates. A full exploration of the observed nonlinear behavior is well outside the scope of this work; we simply conclude that the optimized design did indeed approach the targeted displacement within the

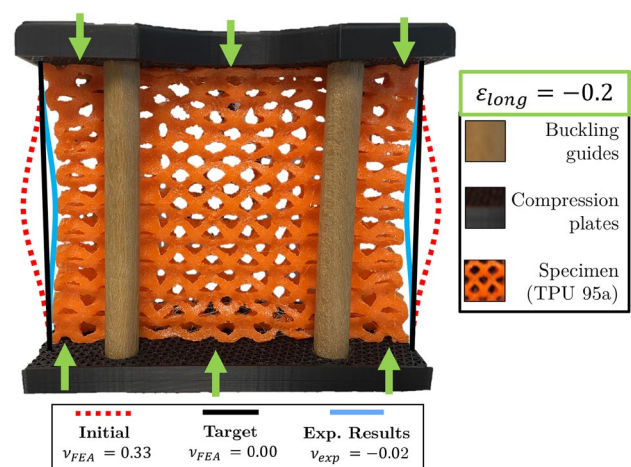


Fig. 26 The compression fixture (buckling guides and compression plates) used to reproduce the boundary conditions for the auxetic design is shown at 20% longitudinal compression

limits of its parameterized geometry provided the DNN's evaluations and shape sensitivities. Beyond navigating the design space, a thorough validation of the final analysis would require full-scale simulation and experimentation as in Cheng et al. (2019).

Acknowledgements The authors would like to acknowledge support from Drexel University. N. Black is grateful for support from the GAANN Grant (Grant Number P200A190036). The work is also supported by the NSF CAREER Award CMMI-2143422. N. Black would also like to acknowledge the fundamental contributions of Dr. Daniel A. Tortorelli to the development of computational algorithms for solid mechanics and Dr. Matthew Burlick to the development of deep learning algorithms. The authors would also like to thank Reza Pejman for insightful discussions related to this work.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results Comprehensive implementation details were provided and the authors are confident that the work is reproducible. For further details and access to the training datasets used in this work, readers are encouraged to contact the authors.

References

- Allaire G, Brizzi R (2005) A multiscale finite element method for numerical homogenization. *Multiscale Model Simul* 4(3):790–812. <https://doi.org/10.1137/040611239>
- Allaire G, Bonnetier E, Francfort G, Jouve F (1997) Shape optimization by the homogenization method. *Numer Math* 76:27–68. <https://doi.org/10.1007/s002110050253>
- Andreassen E, Andreassen CS (2014) How to determine composite material properties using numerical homogenization. *Comput Mater Sci* 83:488–495. <https://doi.org/10.1016/j.commatsci.2013.09.006>
- Andreassen CS, Sigmund O (2012) Multiscale modeling and topology optimization of poroelastic actuators. *Smart Mater Struct* 21(6):065005. <https://doi.org/10.1088/0964-1726/21/6/065005>
- Baker N, Alexander F, Bremer T, Hagberg A, Kevrekidis Y, Najm H, Parashar M, Patra A, Sethian J, Wild S, Willcox K, Lee S (2019) Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence. Technical report. USDOE Office of Science (SC), Washington, DC. <https://doi.org/10.2172/1478744>
- Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. *Comput Methods Appl Mech Eng* 71(2):197–224. [https://doi.org/10.1016/0045-7825\(88\)90086-2](https://doi.org/10.1016/0045-7825(88)90086-2)
- Bian W, Chen X (2012) Smoothing neural network for constrained non-Lipschitz optimization with applications. *IEEE Trans Neural Netw Learn Syst* 23(3):399–411. <https://doi.org/10.1109/TNNLS.2011.2181867>
- Bourdin B (2001) Filters in topology optimization. *Int J Numer Methods Eng* 50(9):2143–2158. <https://doi.org/10.1002/nme.116>
- Brandyberry DR, Najafi AR, Geubelle PH (2020) Multiscale design of three-dimensional nonlinear composites using an interface-enriched generalized finite element method. *Int J Numer Methods Eng* 121(12):2806–2825. <https://doi.org/10.1002/nme.6333>
- Bruns TE, Tortorelli DA (2001) Topology optimization of non-linear elastic structures and compliant mechanisms. *Comput Methods Appl Mech Eng* 190(26):3443–3459. [https://doi.org/10.1016/S0045-7825\(00\)00278-4](https://doi.org/10.1016/S0045-7825(00)00278-4)
- Chan YC, Da D, Wang L, Chen W (2022) Remixing functionally graded structures: data-driven topology optimization with multi-class shape blending. *Struct Multidisc Optim* 65(5):135. <https://doi.org/10.1007/s00158-022-03224-x>
- Cheng L, Bai J, To AC (2019) Functionally graded lattice structure topology optimization for the design of additive manufactured components with stress constraints. *Comput Methods Appl Mech Eng* 344:334–359. <https://doi.org/10.1016/j.cma.2018.10.010>
- Fazlyab M, Robey A, Hassani H, Morari M, Pappas G (2019) Efficient and accurate estimation of Lipschitz constants for deep neural networks. In: *Advances in neural information processing systems*, vol 32. Curran Associates, Inc. Accessed: Dec. 26, 2022. Available: <https://proceedings.neurips.cc/paper/2019/hash/95e1533eb1b20a9777749fb94fdb944-Abstract.html>. Accessed 26 Dec 2022
- Gallant A, White H (1992) On learning the derivatives of an unknown mapping with multilayer feedforward networks. *Neural Netw*. [https://doi.org/10.1016/S0893-6080\(05\)80011-5](https://doi.org/10.1016/S0893-6080(05)80011-5)
- Garner E, Kolken HMA, Wang CCL, Zadpoor AA, Wu J (2019) Compatibility in microstructural optimization for additive manufacturing. *Addit Manuf* 26:65–75. <https://doi.org/10.1016/j.addma.2018.12.007>
- Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR workshop and conference proceedings*, 2010, pp 249–256. ISSN 1938-7228. Accessed: Dec. 26, 2022. <https://proceedings.mlr.press/v9/glorot10a.html>
- Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press. Google-Books-ID omivDQAAQBAJ
- Gouk H, Frank E, Pfahringer B, Cree MJ (2021) Regularisation of neural networks by enforcing Lipschitz continuity. *Mach Learn* 110(2):393–416. <https://doi.org/10.1007/s10994-020-05929-w>
- Groen JP, Sigmund O (2018) Homogenization-based topology optimization for high-resolution manufacturable microstructures. *Int J Numer Methods Eng* 113(8):1148–1163. <https://doi.org/10.1002/nme.5575>
- Groen JP, Wu J, Sigmund O (2019) Homogenization-based stiffness optimization and projection of 2D coated structures with orthotropic infill. *Comput Methods Appl Mech Eng* 349:722–742. <https://doi.org/10.1016/j.cma.2019.02.031>
- Guedes J, Kikuchi N (1990) Preprocessing and postprocessing for materials based on the homogenization method with adaptive finite element methods. *Comput Methods Appl Mech Eng* 83(2):143–198. [https://doi.org/10.1016/0045-7825\(90\)90148-F](https://doi.org/10.1016/0045-7825(90)90148-F)
- Guedes J, Rodrigues H, Bendsøe M (2003) A material optimization model to approximate energy bounds for cellular materials under multiload conditions. *Struct Multidisc Optim* 25(5):446–452. <https://doi.org/10.1007/s00158-003-0305-8>
- Hassani B, Hinton E (1998) A review of homogenization and topology optimization I—homogenization theory for media with periodic structure. *Comput Struct* 69(6):707–717. [https://doi.org/10.1016/S0045-7949\(98\)00131-X](https://doi.org/10.1016/S0045-7949(98)00131-X)
- Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Netw* 2(5):359–366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- Imediegwu C, Murphy R, Hewson R, Santer M (2019) Multiscale structural optimization towards three-dimensional printable structures. *Struct Multidisc Optim* 60(2):513–525. <https://doi.org/10.1007/s00158-019-02220-y>
- Kazemi H, Norato JA (2022) Topology optimization of programmable lattices with geometric primitives. *Struct Multidisc Optim* 65(1):33. <https://doi.org/10.1007/s00158-021-03094-9>
- Kim C, Lee J, Yoo J (2021) Machine learning-combined topology optimization for functionary graded composite structure design. *Comput Methods Appl Mech Eng* 387(114):158. <https://doi.org/10.1016/j.cma.2021.114158>
- Kingma DP, Ba J (2017) Adam: a method for stochastic optimization
- Kollmann HT, Abueidda DW, Koric S, Guleryuz E, Sobh NH (2020) Deep learning for topology optimization of 2D metamaterials. *Mater Des* 196(109):098. <https://doi.org/10.1016/j.matdes.2020.109098>
- Lee YJ, Misra S, Chen WH, Koditschek DE, Sung C, Yang S (2022) Tendon-driven auxetic tubular springs for resilient hopping robots. *Adv Intell Syst* 4(4):2100152. <https://doi.org/10.1002/aisy.202100152>

- Logarzo HJ, Capuano G, Rimoli JJ (2021) Smart constitutive laws: inelastic homogenization through machine learning. *Comput Methods Appl Mech Eng* 373:113482. <https://doi.org/10.1016/j.cma.2020.113482>
- Murphy R, Imediegwu C, Hewson R, Santer M (2021) Multiscale structural optimization with concurrent coupling between scales. *Struct Multidisc Optim* 63(4):1721–1741. <https://doi.org/10.1007/s00158-020-02773-3>
- Najafi AR, Safdari M, Tortorelli DA, Geubelle PH (2015) A gradient-based shape optimization scheme using an interface-enriched generalized FEM. *Comput Methods Appl Mech Eng* 296:1–17. <https://doi.org/10.1016/j.cma.2015.07.024>
- Najafi AR, Safdari M, Tortorelli DA, Geubelle PH (2017) Shape optimization using a NURBS-based interface-enriched generalized FEM. *Int J Numer Methods Eng* 111(10):927–954. <https://doi.org/10.1002/nme.5482>
- Najafi AR, Safdari M, Tortorelli DA, Geubelle PH (2021) Multiscale design of nonlinear materials using a Eulerian shape optimization scheme. *Int J Numer Methods Eng* 122(12):2981–3014. <https://doi.org/10.1002/nme.6650>
- Nguyen-Thien T, Tran-Cong T (1999) Approximation of functions and their derivatives: a neural network implementation with applications. *Appl Math Model* 23(9):687–704. [https://doi.org/10.1016/S0307-904X\(99\)00006-2](https://doi.org/10.1016/S0307-904X(99)00006-2)
- Nikolakakis KE, Haddadpour F, Karbasi A, Kalogieras DS (2022) Beyond Lipschitz: sharp generalization and excess risk bounds for full-batch GD. <http://arxiv.org/abs/2204.12446>, arXiv:2204.12446. [cs, stat]
- Novak R, Bahri Y, Abolafia DA, Pennington J, Sohl-Dickstein J (2018) Sensitivity and generalization in neural networks: an empirical study. <http://arxiv.org/abs/1802.08760>, arXiv:1802.08760, [cs, stat]
- Pantz O, Trabelsi K (2008) A post-treatment of the homogenization method for shape optimization. *SIAM J Control Optim* 47(3):1380–1398. <https://doi.org/10.1137/070688900>
- Rozvany GIN, Zhou M, Birker T (1992) Generalized shape optimization without homogenization. *Struct Optim* 4(3):250–252. <https://doi.org/10.1007/BF01742754>
- Safdari M, Najafi AR, Sottos NR, Geubelle PH (2015) A NURBS-based interface-enriched generalized finite element method for problems with complex discontinuous gradient fields. *Int J Numer Methods Eng* 101(12):950–964. <https://doi.org/10.1002/nme.4852>
- Safdari M, Najafi AR, Sottos NR, Geubelle PH (2016) A NURBS-based generalized finite element scheme for 3D simulation of heterogeneous materials. *J Comput Phys* 318:373–390. <https://doi.org/10.1016/j.jcp.2016.05.004>
- Sigmund O (1994) Materials with prescribed constitutive parameters: an inverse homogenization problem. *Int J Solids Struct* 31(17):2313–2329. [https://doi.org/10.1016/0020-7683\(94\)90154-6](https://doi.org/10.1016/0020-7683(94)90154-6)
- Sigmund O, Aage N, Andreassen E (2016) On the (non-)optimality of Michell structures. *Struct Multidisc Optim* 54(2):361–373. <https://doi.org/10.1007/s00158-016-1420-7>
- Soghrati S, Aragón AM, Armando Duarte C, Geubelle PH (2012) An interface-enriched generalized FEM for problems with discontinuous gradient fields. *Int J Numer Methods Eng* 89(8):991–1008. <https://doi.org/10.1002/nme.3273>
- Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. *Int J Numer Methods Eng* 24(2):359–373. <https://doi.org/10.1002/nme.1620240207>
- Torquato S (2010) Optimal design of heterogeneous materials. *Annu Rev Mater Res* 40(1):101–129. <https://doi.org/10.1146/annurev-matsci-070909-104517>
- Torquato S, Haslach H (2002) Random heterogeneous materials: microstructure and macroscopic properties. *Appl Mech Rev* 55(4):B62–B63. <https://doi.org/10.1115/1.1483342>
- Vilardell AM, Takezawa A, du Plessis A, Takata N, Krakhmalev P, Kobashi M, Yadroitsava I, Yadroitsev I (2019) Topology optimization and characterization of Ti6Al4V ELI cellular lattice structures by laser powder bed fusion for biomedical applications. *Mater Sci Eng A* 766:138330. <https://doi.org/10.1016/j.msea.2019.138330>
- Wang F, Sigmund O (2021) 3D architected isotropic materials with tunable stiffness and buckling strength. *J Mech Phys Solids* 152:104–415. <https://doi.org/10.1016/j.jmps.2021.104415>
- Wang F, Sigmund O, Jensen JS (2014) Design of materials with prescribed nonlinear properties. *J Mech Phys Solids* 69:156–174. <https://doi.org/10.1016/j.jmps.2014.05.003>
- Wang L, Chan YC, Ahmed F, Liu Z, Zhu P, Chen W (2020) Deep generative modeling for mechanistic-based learning and design of metamaterial systems. *Comput Methods Appl Mech Eng* 372:113377. <https://doi.org/10.1016/j.cma.2020.113377>
- Watts S, Tortorelli DA (2017) A geometric projection method for designing three-dimensional open lattices with inverse homogenization. *Int J Numer Methods Eng* 112(11):1564–1588. <https://doi.org/10.1002/nme.5569>
- Watts S, Arrighi W, Kudo J, Tortorelli DA, White DA (2019) Simple, accurate surrogate models of the elastic response of three-dimensional open truss micro-architectures with applications to multiscale topology design. *Struct Multidisc Optim* 60(5):1887–1920. <https://doi.org/10.1007/s00158-019-02297-5>
- White DA, Arrighi WJ, Kudo J, Watts SE (2019) Multiscale topology optimization using neural network surrogate models. *Comput Methods Appl Mech Eng* 346:1118–1135. <https://doi.org/10.1016/j.cma.2018.09.007>
- Wu J, Sigmund O, Groen JP (2021a) Topology optimization of multiscale structures: a review. *Struct Multidisc Optim* 63(3):1455–1480. <https://doi.org/10.1007/s00158-021-02881-8>
- Wu J, Wang W, Gao X (2021b) Design and optimization of conforming lattice structures. *IEEE Trans Vis Comput Graph* 27(1):43–56. <https://doi.org/10.1109/TVCG.2019.2938946>
- Yu X, Zhou J, Liang H, Jiang Z, Wu L (2018) Mechanical metamaterials associated with stiffness, rigidity and compressibility: a brief review. *Prog Mater Sci* 94:114–173. <https://doi.org/10.1016/j.pmatsci.2017.12.003>
- Zheng L, Kumar S, Kochmann DM (2021) Data-driven topology optimization of spinodoid metamaterials with seamlessly tunable anisotropy. *Comput Methods Appl Mech Eng* 383:113894. <https://doi.org/10.1016/j.cma.2021.113894>
- Zhou M, Rozvany GIN (1991) The COC algorithm, Part II: topological, geometrical and generalized shape optimization. *Comput Methods Appl Mech Eng* 89(1):309–336. [https://doi.org/10.1016/0045-7825\(91\)90046-9](https://doi.org/10.1016/0045-7825(91)90046-9)
- Zhou H, Zhu J, Wang C, Zhang Y, Wang J, Zhang W (2022) Hierarchical structure optimization with parameterized lattice and multiscale finite element method. *Struct Multidisc Optim* 65(1):39. <https://doi.org/10.1007/s00158-021-03149-x>
- Zhu B, Skouras M, Chen D, Matusik W (2017) Two-scale topology optimization with microstructures. *ACM Trans Graph* 36(5):164:1–164:16. <https://doi.org/10.1145/3095815>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.