

Multi-fidelity design and optimization framework of damage-tolerant fiber-reinforced composite structures via CNN-assisted homogenization

Fulong Hu

School of Civil Engineering, The University of Queensland, Brisbane, Australia

Yangpan Fu

*School of Civil Engineering and Transportation, Guangzhou University,
Guangzhou, China and*

School of Civil Engineering, The University of Queensland, Brisbane, Australia

Shujian Chen

School of Civil Engineering, The University of Queensland, Brisbane, Australia

Da Chen

*School of Civil and Environmental Engineering, University of New South Wales,
Sydney, Australia, and*

Weiqliang Chen

*School of Civil and Environmental Engineering, Rice University,
Houston, Texas, USA*

Abstract

Purpose – Neighbor interactions among representative volume elements induce systematic perturbations in their effective stiffness and Poisson's ratio when these units are embedded in larger composite structures. However, classic homogenization treats each unit's effective properties as invariant, regardless of the surrounding microstructure. To bridge this gap at minimal cost, we introduce a multi-fidelity homogenization framework augmented by a convolutional neural network correction.

Design/methodology/approach – First, 2D unit cells are homogenized and then assembled into large structures via layout matrices. For each unit, its neighborhood is encoded into a two-channel input and passed to high-capacity neural network models, which predict the local deviations from baseline effective properties. Then, the homogenization results of each unit cell within an assembled structure are corrected correspondingly.

Findings – This approach eliminates more than 92.42% of expensive full-fidelity simulations, yielding a 22-fold speed-up. To enable deployment in resource-constrained or high-throughput settings, we then systematically prune and reorganize the neural network architectures into a lightweight model that trains in under 30 min on a single graphics processing unit and achieves five times faster inference with comparable accuracy (over 80% of the high-fidelity accuracy can be recovered). Convergence and generalization studies confirm that the model remains robust and scalable across unseen configurations, paving the way for rapid structural design and optimization campaigns.

Originality/value – This study introduces a convolutional neural network-enhanced multi-fidelity framework for designing and optimizing fiber-reinforced composite structures intended for segmental failure. Unlike previous works focused on material behavior and fabrication, this research preliminarily addresses the unexplored gap of integrating design, performance prediction, and geometry-driven optimization.



The framework enables fast, accurate and scalable exploration of local geometry effects, delivering a damage-tolerant, energy-absorbing composite structure that enhances resilience.

Keywords Fiber-reinforced composites, Design and structural assessment, Finite element method, Convolutional neural networks, Multi-fidelity simulation, Structural Performance evaluation

Paper type Research article

1. Introduction

Lightweight fiber-reinforced materials (FRM) have emerged as essential materials in various engineering domains due to their exceptional strength-to-weight ratios and customizable behaviors (Talreja and Waas, 2022; Duan *et al.*, 2025; Dou *et al.*, 2025), including aerospace, automotive, and civil engineering. Their structural composition heavily influences their mechanical performance, particularly the arrangement of fibers and the type of filler materials used within the matrix (Nguyen-Thanh *et al.*, 2022). By integrating multiple representative units, each featuring distinct fiber trajectories and filler compositions, engineers can design advanced composite structures with properties precisely tailored to meet the demands of specific applications (Zhu *et al.*, 2025; Niu *et al.*, 2024; Hyde *et al.*, 2020).

However, identifying the optimal combination and spatial arrangement of these unit cells can become computationally prohibitive as more unit cells are introduced, limiting design innovation and efficiency. To overcome this, various advanced computational optimization techniques have been explored, including evolutionary algorithms (Wang and Sobey, 2020; Cai and Aref, 2015), reinforcement learning (Shen *et al.*, 2025; Li *et al.*, 2024d), Bayesian optimization (Coelho *et al.*, 2025), swarm intelligence methods (Martínez-Muñoz *et al.*, 2023), and neural network-based methods (Li *et al.*, 2024d; Zhan *et al.*, 2025). These approaches can explore the extensive design space within an affordable computational cost by strategically selecting and evaluating promising unit cell configurations. Nevertheless, full simulations on the mechanical behaviors of massive combinations are required for each of these methods (Shih *et al.*, 2023; Barbero and Barbero, 2023). When dealing with large amounts of design variations, these methods show less efficiency. Current advancements on minimizing the number of full simulations while still identifying high-performance designs are neural network (NN) models (Li *et al.*, 2024c; Kim *et al.*, 2023), multi-fidelity simulations (Ahn *et al.*, 2022; Pan *et al.*, 2025), transfer learning from small simulations (Kim *et al.*, 2021; Wang *et al.*, 2025), and active learning coupled with Bayesian optimization (Zhao *et al.*, 2024), etc.

Among these advanced methodologies, the multi-fidelity (MF) simulation method has recently gained considerable attention due to its ability to balance computational efficiency with predictive accuracy (Wang *et al.*, 2024; An *et al.*, 2023; Liu *et al.*, 2024b). Traditional MF simulations run fast, low-fidelity simulations (homogenized unit cells) first to search for high-performance candidates (Li *et al.*, 2024b, 2025). Then, high-fidelity finite element analyses (FEA) will be conducted on these high-performance candidates to identify the optimal combination further. Although homogenized models can save excessive computational resources, they often sacrifice accuracy (interactions between different local units not thoroughly considered), potentially posing threats of error that propagate into optimization outcomes. Moreover, the process of calibrating and validating low-fidelity models against high-fidelity simulations can be computationally expensive and labor-intensive. These considerations are the focus of this paper.

Introducing machine learning into the MF workflow can improve the predictive accuracy. Recent machine learning (ML)-based homogenization methods learn a mapping from micro-structural descriptors to global effective properties of a single representative volume element (RVE), and most MF pipelines use such predictions to screen candidates prior to high-fidelity analyses (Lee *et al.*, 2023). Deep Material Networks approximate RVEs via hierarchical building blocks with guaranteed kinematic/traction compatibility, offering strong accuracy but nontrivial training cost and limited handling of spatial context across interacting units

(Su *et al.*, 2022). U-Net-style NN learn pixel-to-property mappings directly from image RVEs (Croom *et al.*, 2022); variants include encoder–decoder NNs for elasticity/thermal fields and attention/U-Net hybrids that improve spatial feature reuse. Graph neural networks (GNNs) (Storm *et al.*, 2024) and message-passing models operate on phase/voxel graphs, capturing long-range dependencies and irregular topologies; extensions include graph U-Nets and spectral GNNs for field prediction. Earlier surrogate-based homogenization combined handcrafted descriptors (e.g. 2-point correlations) with regressors (Bishara *et al.*, 2023). These methods typically return global RVE properties for an isolated cell and do not model unit-level property shifts induced by local neighbors in an assembled structure.

In contrast, this study introduces a novel convolutional neural network (CNN)-based optimization approach into the multi-fidelity simulation framework to improve accuracy and efficiently identify the optimal configurations of RVE, each characterized by unique fiber arrangements and fillers (numerically encoded). CNNs were chosen because the spatial configuration of local RVEs in the composite structure can be naturally represented as two-dimensional matrices, analogous to image data. This makes CNNs particularly effective for capturing RVE interactions and spatial correlations that directly influence the effective material properties. We conducted FEAs separately on RVEs with all phases explicitly modeled to complete homogenization. Spatial arrangements of RVEs were encoded through a series of $n \times m$ layout matrices. High fidelity simulations on the RVE assemblies were conducted to identify the property deviations of the same RVE under different surrounding geometries. For each RVE in the assemblies, the local influence is quantified and stored as Pytorch tensors for subsequent training of convolutional neural networks (CNN) (Liu *et al.*, 2024a). The trained models achieve a significant improvement in error compensation and reduction in computational cost without repeated high-fidelity simulation. By systematically removing some layers, a lightweight CNN model that shows comparable ability in capturing the nonlinear local influences with standard CNN models is proposed.

To be more specific, this work contributes: (1) a unit-level, local-neighbor correction that predicts geometry-dependent ratios deviations for each embedded unit cell inside an assembly, rather than only estimating bulk properties of an isolated RVE; (2) a two-channel spatial encoding that preserves both macro-scale layout (cell identifiers) and meso-scale phase content (feature matrices), enabling the CNN to learn short-range interactions; (3) an NN-informed MF selector that applies these corrections to re-rank candidates at low cost, recovering HF quality while promoting a smaller fraction of designs; and (4) a lightweight CNN with block-wise down-sampling that achieves accuracy comparable to deep benchmarks with substantially lower training time and inference latency. We further provide a computational efficiency analysis demonstrating HF savings under identical hardware and solver settings.

2. Finite element simulation and data collection

2.1 Unit cell homogenization

The principle of homogenization (Li *et al.*, 2025; Jain, 2019) is replacing the original elements within a specific volume/area with homogenized global material properties, ensuring the homogenized elements store approximately the same amount of strain energy as the original elements. In numerical homogenization, the properties of FRMs are commonly determined by modeling RVEs, which capture the key characteristics of the underlying structure (Jain, 2019).

2.1.1 Generation of representative unit cells. To form the fundamental RVEs, five 2-dimensional, square unit cells with a length of L , each featuring unique internal fiber paths and filler material distributions, were modeled using the Abaqus FEA package as illustrated in Figure 1.

These unit cells serve as the basic components for assembling larger composite structures with tailored mechanical performance. Each unit consists of three different phases: a Polypropylene (PP) matrix (Hossain *et al.*, 2024), which performs as a stable base; Several

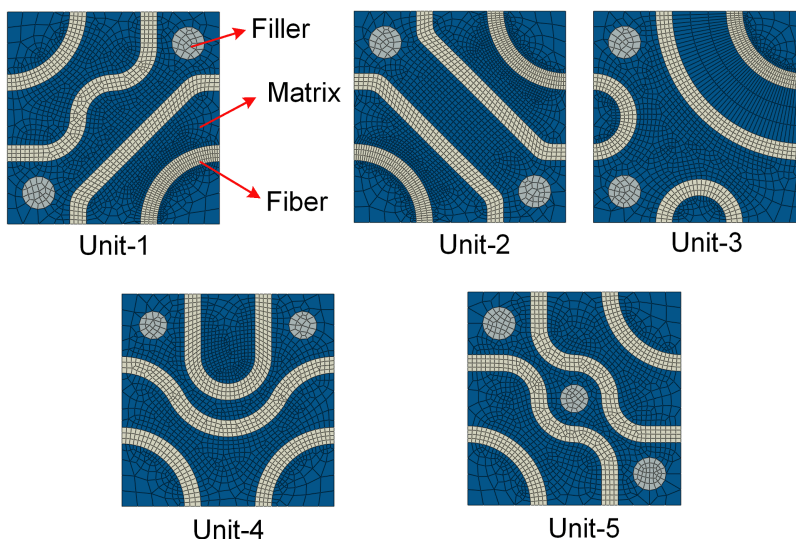


Figure 1. Overview of all five 2D unit-cell geometries with distinct fiber paths and filler layouts, and corresponding finite-element meshes, showing element sizing (matrix: 0.0045 mm, fiber: 0.0015 mm)

embedded carbon fibers, which offer strength; and fillers, strategically positioned to boost energy absorption and vibration control (Ramesh *et al.*, 2022). The specific arrangement and interaction of these three phases within each unit cell are critical in determining the overall mechanical performance of the formed structure.

The start and end points of inner fibers are set at every $1/3L$ points on the edges of each square block to ensure the continuity of fibers in adjacent blocks when assembled. The PP matrix was treated as an isotropic elastic-plastic material defined by its Young's modulus and Poisson's ratio. Plasticity of PP was specified in Abaqus via a table of yield stress and plastic strain pairs (no kinematic hardening or rate dependence was included). Fillers were treated as a linear elastic material. For the carbon fibers, an orthotropic elastic model was used, implemented in Abaqus via engineering constants. Detailed material properties are listed in Table 1.

Mesh convergence was first established at the RVE level by refining matrix/filler and fiber seeds until changes in E_{11} , E_{22} , ν_{12} were less than 1%. Specifically, three refinement levels were considered: (1) Matrix/filler seed 0.0060 mm, fiber seed 0.0020 mm. (2) Matrix/filler seed 0.0045 mm, fiber seed 0.0015 mm. (3) Matrix/filler seed 0.0030 mm, fiber seed 0.0010 mm.

For each unit cell, the effective stiffness and Poisson's ratio were computed at each refinement level under the same macroscopic loading. Across all cases, differences between

Table 1. Material properties used in simulations

Properties	PP matrix	Rubber filler	Carbon fiber
Constitutive model	Small-strain isotropic elastic-plastic	Isotropic linear elastic	Engineering constants
E (MPa)	1,700	30	—
ν	0.4	0.47	—
E_1 (MPa)	—	—	135,000
E_2 (MPa)	—	—	7,000
G_{12} (MPa)	—	—	4,000
ν_{12}	—	—	0.3
Yield stress (Mpa)	[30, 33, 37]	—	—
Plastic strain	[0, 0.02, 0.05]	—	—

level 2 and level 3 satisfied the precision criteria (less than 1%). Eventually, the meshing seed size of the PP matrix was set as 0.0045 mm, while the carbon fibers, which experience higher stress gradients and require higher accuracy, are refined to a seed size of 0.0015 mm.

2.1.2 Implementation of periodic boundary conditions and macroscopic strain. Applying periodic boundary conditions (Tian and Qi, 2023) (PBC) to each unit cell is critical to accurately simulate the unit cells' mechanical response and ensure continuity across adjacent units. This approach allows the models to mimic infinite composite material by ensuring that the displacements and tractions on opposing boundaries remain consistent, thereby preserving the representativeness of the micro mechanics in larger-scale assemblies. Figure 2 shows the principle of PBC application on the boundaries of each unit cell.

To apply the PBCs rigorously, we begin with the deformation gradient tensor $\mathbf{F}$:

$$\mathbf{F} = \mathbf{I} + \nabla \mathbf{u}, \tag{1}$$

where $\mathbf{I}$ is the identity tensor and $\nabla \mathbf{u}$ is the displacement gradient tensor. The general periodicity condition for a unit cell of length L in each direction is:

$$\mathbf{u}(\mathbf{x} + \mathbf{L}) - \mathbf{u}(\mathbf{x}) = (\mathbf{F} - \mathbf{I}) \cdot \mathbf{L} = \boldsymbol{\varepsilon} \cdot \mathbf{L}, \tag{2}$$

where $\boldsymbol{\varepsilon}$ is the symmpetric macroscopic strain tensor under small deformation assumptions. Expressed in component form for square unit cells:

$$u_x(R, y) - u_x(L, y) = \varepsilon_{11}L_x, \tag{3}$$

$$u_y(R, y) - u_y(L, y) = \varepsilon_{12}L_x, \tag{4}$$

$$u_x(x, T) - u_x(x, B) = \varepsilon_{12}L_y, \tag{5}$$

$$u_y(x, T) - u_y(x, B) = \varepsilon_{22}L_y. \tag{6}$$

These relations enforce a displacement field that is periodic while allowing a uniform average strain to develop across the RVE.

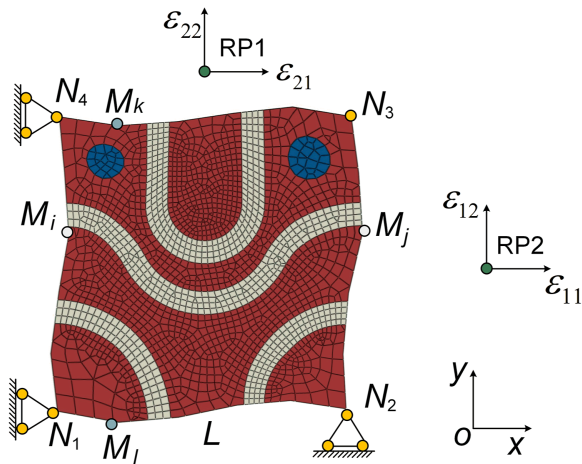


Figure 2. Periodic boundary conditions on unit cells

Following the implementation of PBCs, multiple reference points (RP) were introduced and kinematically coupled to the boundary nodes of the unit cells with the multi-point constraints (MPC) method (Tian and Qi, 2023). The periodic constraints were implemented in Abaqus using the *EQUATION command. Nodes on opposite boundaries were paired according to their coordinates, and their displacements were coupled through linear constraint equations (Wang et al., 2025; Qi et al., 2015). The macroscopic strain components were represented by reference points (RPs) whose prescribed displacements correspond to $\varepsilon_{11}L_x$, $\varepsilon_{22}L_y$, and $\varepsilon_{12}L_x$, $\varepsilon_{12}L_y$. For example, the constraint between the left and right boundaries was defined as.

*EQUATION

3, Rnode,1,1.0, Lnode,1,-1.0, RP_EXX,1,-1.0

*EQUATION

3, Rnode,2,1.0, Lnode,2,-1.0, RP_GXYx,1,-1.0

Analogous equations were applied for the top and bottom boundaries.

2.1.3 Homogenized properties. For each unit cell, three independent loading conditions were applied: uniaxial tension in the x and y direction, and shear in the x - y plane. The effective stiffness components were calculated using the volume-averaged form of Hooke's law (Jain, 2019):

$$\bar{\sigma}_{ij} = E_{ijkl}^{\text{eff}} \bar{\varepsilon}_{kl} \quad (7)$$

Where $\bar{\sigma}_{ij}$ is the average stress tensor components over the unit, $\bar{\varepsilon}_{kl}$ is the applied average strain tensor components, E_{ijkl}^{eff} is the effective stiffness tensor of the unit cell. For 2D plane stress or plane strain, the fourth-order stiffness tensor (Bauer and Böhlke, 2022) reduces to a symmetric 3×3 matrix using Voigt notation:

$$\begin{bmatrix} \bar{\sigma}_{11} \\ \bar{\sigma}_{22} \\ \bar{\sigma}_{12} \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} & E_{16} \\ E_{12} & E_{22} & E_{26} \\ E_{16} & E_{26} & E_{66} \end{bmatrix} \begin{bmatrix} \bar{\varepsilon}_{11} \\ \bar{\varepsilon}_{22} \\ 2\bar{\varepsilon}_{12} \end{bmatrix} \quad (8)$$

where E_{11} and E_{22} are the effective axial stiffness in x and y direction, E_{12} is the coupling between ε_{11} and σ_{22} , E_{66} is in-plane shear modulus. These were computed by solving the system of equations generated from the three loading cases. The macroscopic stress $\bar{\sigma}_{ij}$ can be calculated by volume/area averaging:

$$\bar{\sigma}_{ij} = \frac{1}{V} \int_V \sigma_{ij}(x, y) dV \quad (9)$$

This integration is implemented in Abaqus using field output averaging over the entire unit cell area. Through defining $\varepsilon_{11} \neq 0$, $\varepsilon_{22} = \varepsilon_{12} = 0$, E_{11} and E_{12} can be extracted. Similarly, E_{22} and E_{12} can be extracted through defining $\varepsilon_{22} \neq 0$, $\varepsilon_{11} = \varepsilon_{12} = 0$. The corresponding effective properties were derived as:

$$E_{11} = \frac{\bar{\sigma}_{11}}{\bar{\varepsilon}_{11}}, \quad (10)$$

$$E_{22} = \frac{\bar{\sigma}_{22}}{\bar{\varepsilon}_{22}}, \quad (11)$$

$$\nu_{12} = -\frac{\bar{\epsilon}_{22}}{\bar{\epsilon}_{11}} \tag{12}$$

All simulations were run in Abaqus (standard) under a 2-D plane-stress assumption with nominal thickness 1.0 mm and all models were built using CPS4R elements with reduced integration and hourglass control enabled. Frictionless contact was assumed for the elastic simulations, while continuity across unit boundaries and interfaces was enforced using tie constraints.

From the resulting stiffness tensors, the effective Young’s modulus E and Poisson’s ratios ν were extracted for each unit cell. These homogenized properties were later used as key parameters to construct representative, low-fidelity models and served as baseline properties (denoted as E^{eff} and ν^{eff}) for comparison in the multi-fidelity framework.

2.2 Multi-unit structure

2.2.1 Assembly rules. Before assembly, the unit cells were encoded using a unique integer identifier ranging from 1 to 5. Through randomly positioning the integers into a series of two-dimensional arrangement matrices of size $n \times m$, diverse spatial configurations of the larger composite structures with size $n \cdot L \times m \cdot L$ were generated. The arrangement matrices serve as a spatial encoding of specific composite structures, where the position of each integer reflects the placement of a particular unit cell within the global structure (Figure 3). Such encoding methods allow for efficient and scalable input generation for both high-fidelity simulations and NN models. Each matrix encodes not only the constituent types but also the geometry information of unit cells, which is essential for capturing local-neighbor influences.

After translating the unit cells into the grid, dedicated node sets were created for every unit. These node sets include all relevant phases within each unit and serve as spatial references for post-processing the stress and strain fields at the unit-cell level using Python scripts. Figure 4 shows a representative of the composite structures built with 30 randomly generated 20×20 arrangement matrices. Only simulation results of structures 1 to 25 were collected for NN training.

Unlike the typical homogenization process that extracts bulk properties of the entire structure (Li et al., 2024c), the focus here is on the local homogenized properties of individual unit cells within the structure. The effective properties of a specific unit vary not only depending on its composition but also on the geometrical arrangements of its neighboring units (Ramesh et al., 2022; De et al., 2024). For instance, unit 1 may exhibit different effective behavior when surrounded by different geometrically distributed unit cells, as shown in Figure 5.

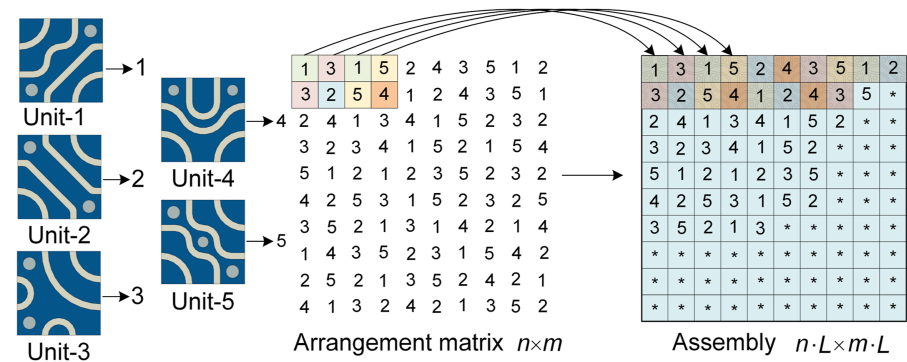


Figure 3. Overview of all five 2D unit-cell geometries with distinct fiber paths and filler layouts, and corresponding finite-element meshes, showing element sizing (matrix: 0.0045 mm, fiber: 0.0015 mm)

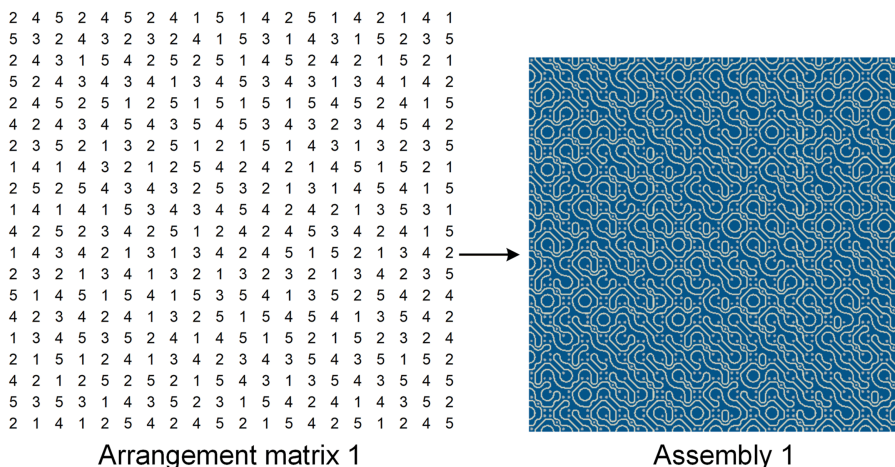


Figure 4. Arrangement matrix and the corresponding assembly

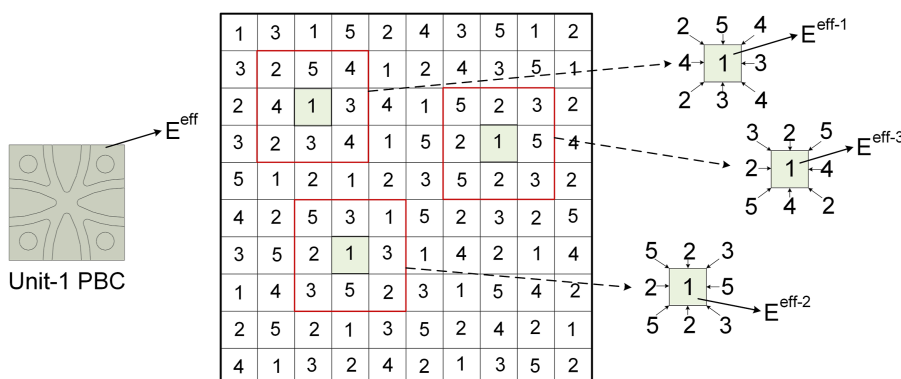


Figure 5. Illustration of the neighborhood influence on the unit cell. A target unit cell (shaded) is embedded in several distinct neighboring layouts composed of the same five canonical unit types, but arranged in different spatial patterns

By comparing these context-sensitive effective properties against the baseline properties obtained from single-unit-cell simulations, the influence of spatial interaction can be quantified.

2.2.2 Simulation on the assembled structures. Parallel simulations were conducted on the assembled structures. The ground truth responses of the structures were captured through high-fidelity simulation, where the fibers, fillers, and matrix phases within the assembled structures remain explicitly resolved (Cheung and Mirkhalaf, 2024; Li *et al.*, 2024a). Meanwhile, the homogenized surrogate behaviors of the structures were captured through standard low-fidelity simulation, where the three phases within the units were replaced with equivalent properties.

Global loading conditions were applied using the same RP-driven approach to simulate the mechanical behavior of the structures. Specifically, PBC was applied to ensure continuity and compatibility with the behavior of the unit cells, thereby capturing the effective macroscopic response. To ensure mechanical continuity throughout the structure, the adjacent boundaries of units are connected through tie constraints. Similarly, fibers were also tied across unit boundaries to preserve the intended load transfer pathways and maintain microstructural realism.

Prescribed vertical displacements corresponding to a 1% global strain in the y direction were then applied to the RPs, inducing a uniform tensile load across the height of the structure. This setup ensures consistent global deformation, enabling stress responses to be computed across all embedded units.

2.2.3 Local geometry influences observation. The homogenized properties of each unit cell within its local environment were calculated through the volume-averaging method. This enabled a direct comparison between the baseline effective property E^{eff} (obtained from isolated simulation) and the local-neighbor-influenced effective property $E^{\text{ln,eff}}$ when the same unit cell is embedded in different spatial geometries.

To quantify the effect, E^{eff} to $E^{\text{ln,eff}}$ ratio R_e and ν^{eff} to $\nu^{\text{ln,eff}}$ ratio R_ν were introduced. We analyzed the R_e and R_ν of all unit cells across a series of geometrical configurations. Figure 6 illustrates the R_e and R_ν scatter plots of each unit cell across 30 randomly selected geometries extracted from the 25 structures.

The results reveal that the stiffness component (particularly in the loading direction) exhibits the most significant variation, suggesting that the load transfer paths and constraint effects attributed to neighboring units play critical roles in modulating local behavior (Liu *et al.*, 2022; Bauer and Böhlke, 2022). This observation supports the need to incorporate local-neighbor effects into the modeling framework. The use of an NN model to capture these nonlinear interactions becomes essential for boosting the predictive accuracy of low-fidelity models without incurring the computational cost of full high-fidelity evaluations.

2.3 Generation of raw training data

2.3.1 Encoding method. For each arrangement matrix, a series of 5×5 submatrices was extracted to represent the local geometry surrounding a target unit cell, as illustrated in Figure 7. The center of a submatrix corresponds to the target unit x , whose homogenized property variations under local-neighbor interaction are to be predicted. The first surrounding layer, which directly contacts unit cell x , is assumed to impose local influence on unit x 's mechanical behavior. The second surrounding layer is considered to introduce neighboring influences, affecting how stress is transferred and distributed through the structure.

By systematically sliding this window across an arrangement matrix (padding the edge units that lack a full neighborhood with zeros), a set of local geometry sub-matrices can be extracted from a single assembled structure. Specifically, from each $n \times m$ arrangement matrix, up to $n \cdot m$ valid sub-matrices of size 5×5 can be generated (duplicated sub-matrices

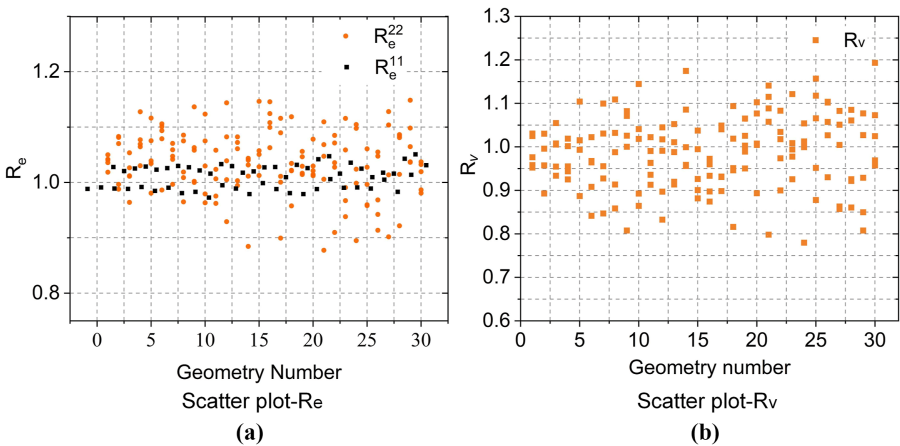


Figure 6. Scatter plot of local-neighbor effect ratios across randomly selected unit-cell samples

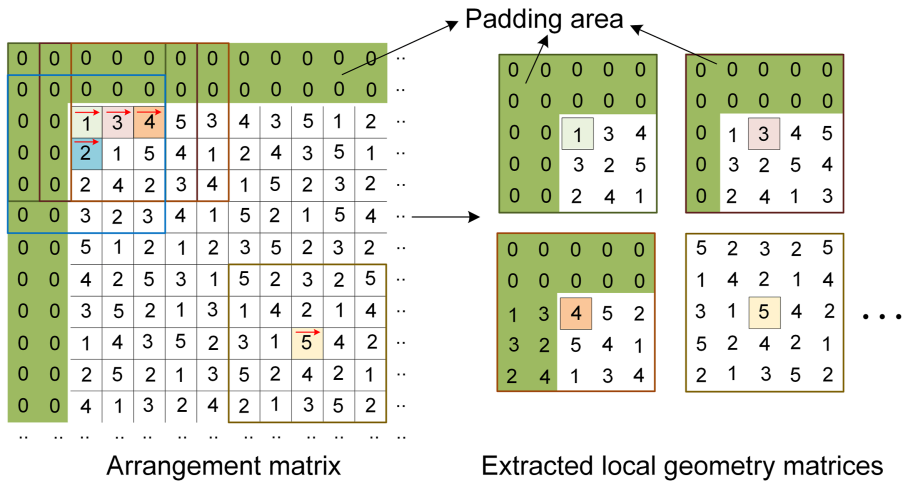


Figure 7. Illustration of the sliding-window extraction process used to generate local 5×5 geometry matrices from a larger arrangement. A 5×5 window is moved across the full arrangement matrix (padding boundary cases), and at each position, the centered unit cell and its first- and second-layer neighbors are captured as a distinct input sample for the neural network

were removed). Each extracted 5×5 sub-matrix is then transformed into a two-channel 500×500 input array for the CNN. For the first channel, each integer in the 5×5 matrix, representing unit cell types and padding, is expanded into a corresponding 100×100 block using nearest-neighbor upsampling. This creates a 500×500 matrix preserving the original spatial arrangement at an enhanced resolution suitable for CNN processing.

The second channel encodes internal structural information of each unit cell type through predefined feature matrices. Six 100×100 feature matrices (including the value 0 in the padding area) were generated based on finite-element analysis images of the internal microstructure, representing the spatial distribution of matrix, fiber, and filler phases, and numerically encoded with distinct mechanical property values. Initially, the internal phase information of each unit cell was identified by segmenting RGB images obtained from FE simulations. As shown in Figure 8, each color region corresponded uniquely to one of the internal phases (matrix, fiber, or filler). Subsequently, these RGB-segmented images were translated into numerical feature values representing the distinct mechanical properties assigned to each phase.

The resulting numerical arrays form feature matrices that explicitly encode the heterogeneous internal composition of each unit cell. Each element in the original 5×5 arrangement matrix is then mapped to its corresponding feature matrix according to the unit type number. The mapped 100×100 feature matrices are subsequently assembled into a second 500×500 matrix, spatially matching the first channel. Each two-channel input effectively captures both the macro-scale geometric configuration and the meso-scale physical composition, enabling the neural network to learn local neighbor influences accurately.

2.3.2 Dataset generation. After assembling all two-channel 500×500 input arrays, the complete dataset was compiled into a single NumPy file for efficient loading and access during model training. Each input sample in this dataset corresponds to a unique local neighborhood configuration extracted from a larger composite structure, paired with its respective target output values R_e and R_L . The generated input dataset has a shape of $(N, 2, 500, 500)$, where N denotes the total number of samples ($N = 10,000$ in this study), 2 represents the two input channels, and 500×500 corresponds to the spatial resolution of each sample derived from assembling 5×5 unit cells, each mapped to a 100×100 feature region.

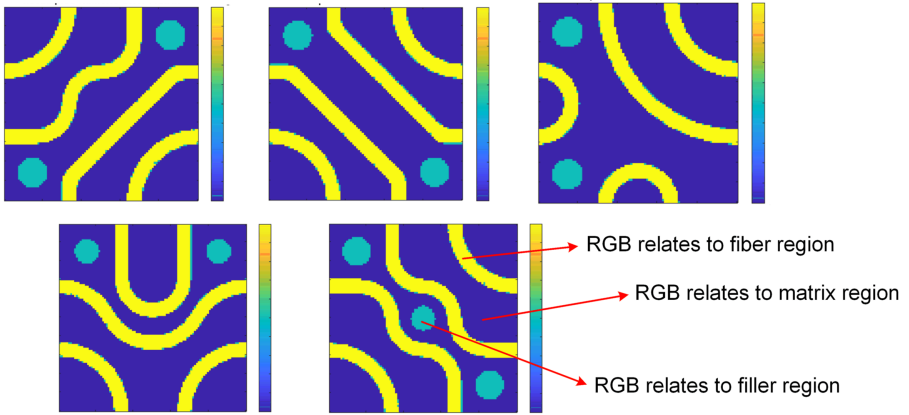


Figure 8. Segmentation of finite-element simulation images into phase-specific regions. Each color in the RGB image uniquely represents one of the internal material phases: matrix, fiber, or filler. These segmented regions were then translated into numerical values based on the mechanical properties of each phase, forming the foundation of the unit cell's feature matrix

Based on previous observations, the stiffness component aligned with the primary loading direction exhibits the greatest sensitivity to local-neighbor influences (Ramesh *et al.*, 2022; Wang *et al.*, 2025). Therefore, only the stiffness component E_{22} and Poisson's ratio ν_{21} are considered for inclusion in the output vector during dataset preparation. For each of the 5×5 sub-matrices, the corresponding output vector is calculated as:

$$R_e = \frac{E^{\text{ln,eff}}}{E^{\text{eff}}}, \quad R_\nu = \frac{\nu^{\text{ln,eff}}}{\nu^{\text{eff}}} \quad (13)$$

where:

$$\mathbf{E}^{\text{ln,eff}} = \begin{bmatrix} E_{11}^{\text{ln,eff}} & E_{22}^{\text{ln,eff}} & \nu_{12}^{\text{ln,eff}} & \nu_{21}^{\text{ln,eff}} \end{bmatrix}, \quad \mathbf{E}^{\text{eff}} = \begin{bmatrix} E_{11}^{\text{eff}} & E_{22}^{\text{eff}} & \nu_{12}^{\text{eff}} & \nu_{21}^{\text{eff}} \end{bmatrix} \quad (14)$$

Considering the models were loaded in y direction, each output sample is a 2-dimensional vector:

$$\mathbf{R} = \begin{bmatrix} \frac{E_{22}^{\text{ln,eff}}}{E_{22}^{\text{eff}}} & \frac{\nu_{21}^{\text{ln,eff}}}{\nu_{21}^{\text{eff}}} \end{bmatrix} = [R_e \quad R_\nu] \quad (15)$$

When generated for an entire $n \times m$ structure, these outputs form a 3D tensor of shape:

$$\text{Output Tensor} \in \mathbb{R}^{n \times m \times 2} \quad (16)$$

The output tensors are spatially aligned with the corresponding position matrix of the assembled structure, ensuring a direct mapping between the encoded geometry (input) and the predicted property variations (output). To facilitate neural network training, all input–output pairs are consolidated into structured datasets: each assembly is assigned to exactly one of the sets (train, validation, test), and no structure contributes patches to more than one split. The assignment is implemented with a group-aware splitter keyed by the structure ID (Novac *et al.*, 2022). The input/output datasets were then stored as Pytorch tensors and then split into three parts randomly according to the assignment. 70% of the data sets are used as training sets, 15% for validation, and 15% for evaluation.

3. Neural network integration and training

3.1 Preparation of training dataset

Data normalization (Fan *et al.*, 2023) is crucial for achieving stable and efficient training of convolutional neural networks, particularly when input channels exhibit drastically different numerical ranges. Without appropriate normalization, gradient-based optimization methods may struggle to converge, due to numerical instability and uneven scaling between input channels.

In this study, the input data consists of two channels with distinctly different numerical characteristics. The first channel, representing the unit cell type (integers ranging from 0 to 5), was normalized using simple min–max scaling:

$$x' = \frac{x}{5} \quad (17)$$

where x represents the original integer value, ensuring the channel's numerical range is constrained within $[0, 1]$.

The second channel, containing the feature matrices with values of 30 (filler), 1700 (matrix), and 135,000 (fiber), exhibited significant numerical disparities. To address this, a logarithmic transformation followed by min–max scaling was applied to compress and standardize the data:

$$x' = \frac{\log(1+x) - \min(\log(1+x))}{\max(\log(1+x)) - \min(\log(1+x))} \quad (18)$$

This normalization technique significantly reduced numerical variance, placing all values within a stable numerical interval of $[0, 1]$. The effectiveness of the normalization was confirmed by examining the resulting data ranges (Table 2), verifying that both input channels were now suitably scaled for efficient network training.

In addition to input normalization, min–max scaling was also applied to the output dataset, specifically targeting the effective property ratios R_e and R_v . The normalization for output data followed a similar linear rescaling:

$$y' = \frac{y - \min(y)}{\max(y) - \min(y)} \quad (19)$$

where y denotes each property difference or ratio in the original dataset. This step ensured that the network outputs also fall within a bounded range of $[0, 1]$, significantly improving the stability and efficiency of gradient computations during network training.

Data loading was conducted with no parallel workers to maintain reproducibility and stability across platforms. Empirical results demonstrated significantly improved convergence speeds and predictive performance following normalization, underscoring its importance in the neural network training pipeline. After normalization, both inputs and outputs exhibited consistent numerical ranges, leading to enhanced training stability and predictive robustness.

Table 2. Value ranges before and after normalization

Channel/Data	Original range	Normalized range
Input Channel 1 (Cell Types)	{0, 1, 2, 3, 4, 5}	[0,1]
Input Channel 2 (Feature Matrices)	{30, 1700, 135,000}	[0,1]
Output Data (R_e and R_v , Ratios)	[0,2]	[0,1]

3.2 Model architectures and setups

In this study, we selected CNN for predicting R_e and R_r due to their inherent capability to efficiently capture and leverage spatial dependencies and local geometric patterns embedded within image-based input data. Unlike fully connected or recurrent neural networks (Sathish et al., 2023; Ghane et al., 2024), CNNs excel at extracting hierarchical spatial features through convolutional kernels, which makes them particularly suitable for processing structured two-dimensional input arrays representing macro-scale geometry and meso-scale internal features of composite materials (Li et al., 2023; Novac et al., 2022). However, standard CNN typically requires extensive computational resources, substantial training durations, and large memory capacities to process high-resolution input data (Li et al., 2024c). These constraints can limit their practical utility, especially when rapid inference or resource-constrained deployment environments are required. Hence, a lightweight CNN architecture employing specialized strategies, such as block-wise down-sampling, was developed. The lightweight model significantly reduces computational complexity and memory footprint while maintaining comparable prediction accuracy to conventional CNNs.

3.2.1 Standard CNN models. We implemented two widely adopted, high-capacity CNN architectures as benchmarks: a baseline CNN model (model-1) and a ResNet-18 network (model-2). Each network was modified (first layer and output layer) to processes the full $2 \times 500 \times 500$ input tensor through its respective stack of convolutional, pooling (or residual) layers and final fully-connected layers to predict the values of R_e and R_r . Model-1 (approximately 11M parameters) uses cascades of 3×3 convolutions interleaved with max-pooling to compress spatial resolution, whereas Model-2 (randomly initialized, approximately 12.5 M parameters) employs eight residual blocks to ensure stable gradient propagation in a moderately deep topology (Azad et al., 2025). Specific architectures and tunable hyperparameters for these two baseline models are summarized in Table 3.

The models were trained for 100 and 110 epochs on identical data-augmentation pipelines and optimizer settings. We systematically recorded key performance metrics—training duration, memory footprint, inference speed, and predictive accuracy to establish a rigorous baseline against which we compare our lightweight CNN variants.

3.2.2 Lightweight model. The lightweight CNN model was developed by systematically reducing the complexity of model-1 and model-2, as shown in Figure 9. Unlike the standard models, this lightweight model utilizes a block-wise down-sampling strategy, compressing spatial dimensions into a manageable feature representation without sacrificing the critical spatial and structural information required for accurate prediction. Specifically, given the large

Table 3. Training and performance hyperparameters for benchmark CNN models

Parameter	ResNet-18	Deep Baseline CNN
Input size	$2 \times 500 \times 500$	$2 \times 500 \times 500$
Parameter quantity	12.5 m	11m
Batch size	64	64
Learning rate	1×10^{-3}	1×10^{-3}
Optimizer	Adam	Adam
Weight decay	1×10^{-5}	1×10^{-5}
Data augmentation	Random flips, crops	Random flips, crops
Epochs	100	110
Training time (wall-clock)	6 h	6.5 h
GPU memory peak	8 GB	8 GB
Inference latency per sample	45 ms	50 ms
List of channels in conv layers	[64, 128, 256, 512]	[64, 128, 256, 512]
Sizes of FC hidden layers	[512, 1,024, 256]	[512, 1,024, 256]
Dropout rate	0.3	0.3

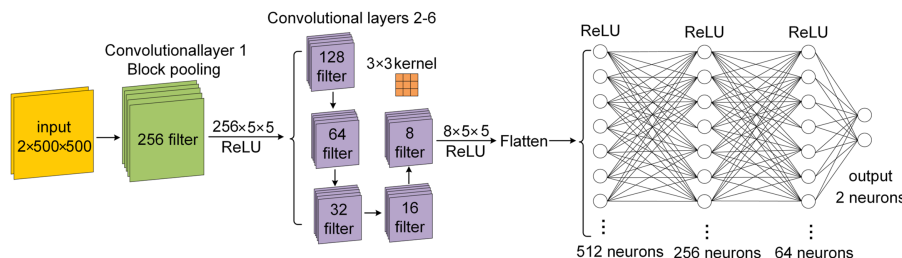


Figure 9. Architecture of the lightweight CNN model used for predicting local property ratios

input size ($N, 2, 500, 500$), the input is partitioned into smaller, non-overlapping blocks (block pooling layer). In this context, the CNN employs a convolutional layer with a kernel size and stride equal to the block size (100×100). Each block is processed individually by this convolutional kernel, effectively summarizing and compressing localized spatial and structural information into a smaller, lower-resolution feature map (in this case, from 500×500 down to 5×5). This block-wise pooling mechanism slashes both FLOPs and peak memory usage. Moreover, the entire model comprises approximately 3 million trainable parameters, roughly one-quarter of model-1, enabling training times on the order of 20–30 min and inference latencies below 5 ms per sample, all while matching the predictive accuracy of the larger architectures. This down-sampling strategy significantly reduces computational complexity and memory requirements, while retaining essential spatial patterns and features critical for accurately predicting local effective property ratios.

The block pooling layer transforms the input of size ($N, 2, 500, 500$) into a representation, where 256 learnable filters extract abstract local spatial features across the downsampled grid. The number of filters in this block-pool layer was increased from the baseline to 256 to improve feature richness and capacity. The output from the block-wise layer is processed through a sequential stack of six convolutional layers with decreasing filter widths. Each convolution uses a kernel size of 3×3 with padding of 1, followed by batch normalization and ReLU activation (Song et al., 2023) to promote gradient stability and non-linearity. This deep stack enhances local pattern recognition and supports robust spatially distributed structural effects learning. Following convolutional processing, the $5 \times 5 \times 8$ output is flattened and passed through a FC (Li et al., 2023) head consisting of three hidden layers with dimensions [512, 256, 64], each followed by ReLU activation and dropout (0.4) to reduce overfitting. The final FC layer maps the learned features to the two output targets, representing the normalized R_e and R_v values, respectively. All parameters used in the training process of the lightweight model are listed in Table 4.

3.3 Training and validation

The CNN models are set up using the Pytorch framework (Novac et al., 2022) according to the above architectures and then trained with the Adam Optimizer (Li et al., 2023; Song et al., 2023; Ghane et al., 2024; Li et al., 2024c). Given the dataset size, the architecture is compact, and an early stopping criterion (Yang et al., 2019) based on validation loss is implemented to minimize the risk of overfitting while still maintaining enough expressive ability to mimic local interactions. At each epoch, training and validation losses, along with scores, were computed and visualized in real-time to monitor convergence behavior. The best model was selected based on validation performance and further evaluated on a held-out test set. All experiments were run on CUDA-enabled GPUs (Nvidia GeForce RTX 3070).

Figure 10 depicts the training and validation loss curves that reflect the training stability by indicating the convergence behavior of both training and validation losses. In the early stage,

Table 4. Tunable hyperparameters in the CNN architecture and training pipeline

Parameter	Description	Default value
block_ch	Number of filters in block pooling layer	256
block_size	Size/stride of block pooling kernel	100
conv_chs	List of channels in conv layers	[256, 128, 64, 32, 16, 8]
fc_dims	Sizes of FC hidden layers	[512, 256, 64]
dropout_rate	Dropout probability in FC layers	0.4
learning_rate	Learning rate for Adam optimizer	1×10^{-3}
batch_size	Number of samples per batch	128
num_epochs	Total number of training epochs	120
val_split	Validation set fraction	0.15
test_split	Test set fraction	0.15
output_dim	Dimension of output (R_e , R_v)	2
in_channels	Input channels (geometry and features)	2

the training and validation losses of all models both decrease rapidly, indicating stable and effective model learning (Li *et al.*, 2023). As the training epoch grows, the improvement rate (dropping velocity of the loss curves) gradually slows down. The validation losses of model 1 and model 2 plateau after approximately 90th and 70th epoch, respectively, while the training loss continues to decline slightly. Such divergence marks the point where the model begins overfitting. The training processes are halted at 90th and 70th epochs, respectively, to preserve the best generalization performance according to the preset early-stopping criterion (Yang *et al.*, 2019). The stable gap between the training and validation loss curves, along with their smooth convergence behavior, indicates that the CNN models have achieved a good balance between underfitting and overfitting.

For the lightweight model, the dropping rate of its training loss slows down after the 85th epoch. Although it continues to decline modestly through to the 120th epoch, the rate of improvement falls below 0.1% per epoch, indicating diminishing returns. Meanwhile, validation loss reaches its minimum around the 90th epoch, after which it remains effectively flat, suggesting the network has captured the dominant feature mappings. Consequently, we adopt an early-stopping criterion at epoch 100, when validation improvement drops below our 0.05% threshold to prevent overfitting and minimize total training time to under 30 min. This strategy yields a final lightweight model whose test-set RMSE matches the standard models within 2%, while inference latency is reduced by more than 95%.

Figure 11 presents a set of overlaid scatter plots, each comparing the CNN-predicted values of the normalized effective stiffness ratio R_e (left panel) and the normalized Poisson’s ratio R_v (right panel) against the corresponding ground-truth data from the validation set. Points are color-coded by model type (model-1, model-2, and lightweight Model), with the solid black line denoting the ideal $y = x$ relationship.

On both axes, values span the interval [0.83, 1.18], capturing the full range of local property perturbations induced by varying 5×5 neighborhoods (Hyde *et al.*, 2020). All three models produce a tight cluster of points around the identity line, with negligible systematic bias or spread even in regions of extreme neighbor effects. Model-1 and Model-2 achieve root-mean-square errors of approximately 0.015, while the lightweight Model, despite its drastically reduced parameter count, attains an RMSE of just 0.017 for R_e and 0.013 for R_v .

This visualization confirms that each network, and in particular our optimized lightweight architecture, robustly approximates the nonlinear mapping from local geometry to effective property corrections. The near-perfect alignment with $y = x$ further demonstrates the models’ capacity to generalize across hundreds of assembly configurations, enabling rapid, high-fidelity microstructure predictions without resorting to costly full-scale finite-element simulations (Cheung and Mirkhalaf, 2024; Li *et al.*, 2024a).

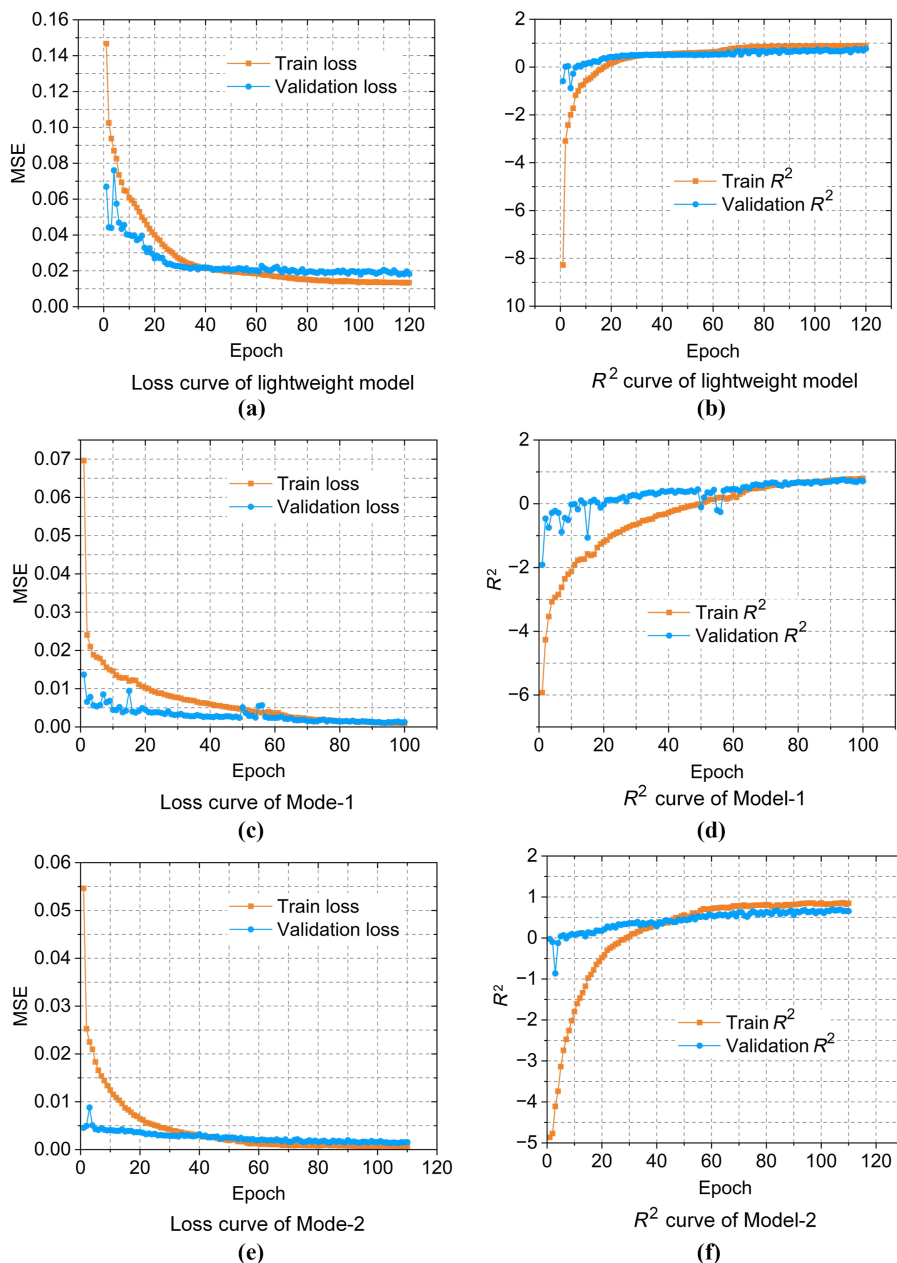


Figure 10. Architecture of the enhanced CNN model used for predicting local property ratios

4. Results and discussion

4.1 Evaluation of homogenization quality

The homogenization quality of unit cells 1–5 was further validated by direct comparison with full-fidelity FE models of the same microstructures. Figure 12 displays the equivalent uniaxial

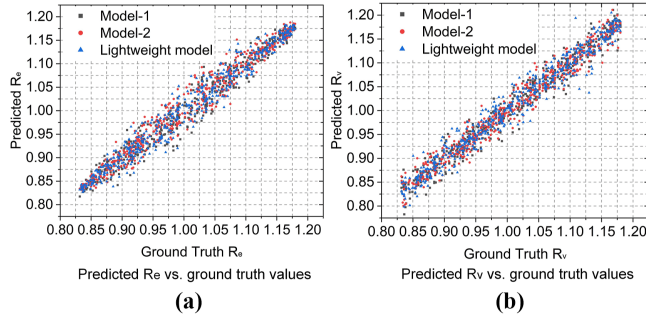


Figure 11. Scatter plots comparing predicted values of R_e and R_v against their respective ground truth values from the validation dataset

stress–strain responses for each homogenized unit cell overlaid with the corresponding high-resolution simulation results. In every case, the two curves exhibit closely initial slopes (effective Young’s moduli) and yield points, with maximum deviations below 5% in stress.

These comparisons confirm that our homogenized constitutive models faithfully reproduce the key mechanical behaviors—linear stiffness, onset of nonlinearity, and post-yield softening of the detailed micro-structural simulations (Wen *et al.*, 2024; Barbero and Barbero, 2023). Such close alignment not only validates the baseline effective properties but also demonstrates that the reduced, homogenized representation captures the essential physics needed for accurate downstream predictions in both the standard and lightweight CNN frameworks.

4.2 Accuracy of the CNN models

Traditional homogenization treats each unit cell as possessing a fixed, context-independent set of effective properties, thereby overlooking the perturbations introduced by the neighboring environment (Li *et al.*, 2024c). In contrast, our CNN-augmented frameworks explicitly account for these local interactions and predict the resulting property discrepancies. To evaluate its accuracy, we compare three sets of predictions: standard homogenization, NN corrections, and full-fidelity simulation results, using the deviation metrics as follows:

$$\delta_{\text{hom}} = |E_{22}^{\text{full}} - E_{22}^{\text{eff}}| \quad (20)$$

$$\gamma_{\text{hom}} = |\nu_{21}^{\text{full}} - \nu_{21}^{\text{eff}}| \quad (21)$$

where E_{22}^{full} and ν_{21}^{full} stand for ground truth value of E_{22} and ν_{21} respectively. Similarly, the prediction errors of NN model can be expressed as:

$$\delta_{\text{NN}} = |E_{22}^{\text{full}} - R_e \cdot E_{22}^{\text{eff}}| \quad (22)$$

$$\gamma_{\text{NN}} = |\nu_{21}^{\text{full}} - R_v \cdot \nu_{21}^{\text{eff}}| \quad (23)$$

where δ_{NN} and γ_{NN} are errors of NN predicted effective stiffness and Poisson’s ratio, respectively. For a cell within its local geometry, a higher value of δ_{hom} reflects strong neighbor influence, whereas a lower value of δ_{NN} indicates the NN correction effectively compensates for the deviations. Conversely, a smaller δ_{hom} implies negligible neighbor influences.

Based on the quantified errors, two indicators, I_e and I_v , are introduced to assess both the extent of deviations of homogenized properties from the ground truth value and the accuracy of NN correction within a cell’s local geometrical context, as follows:

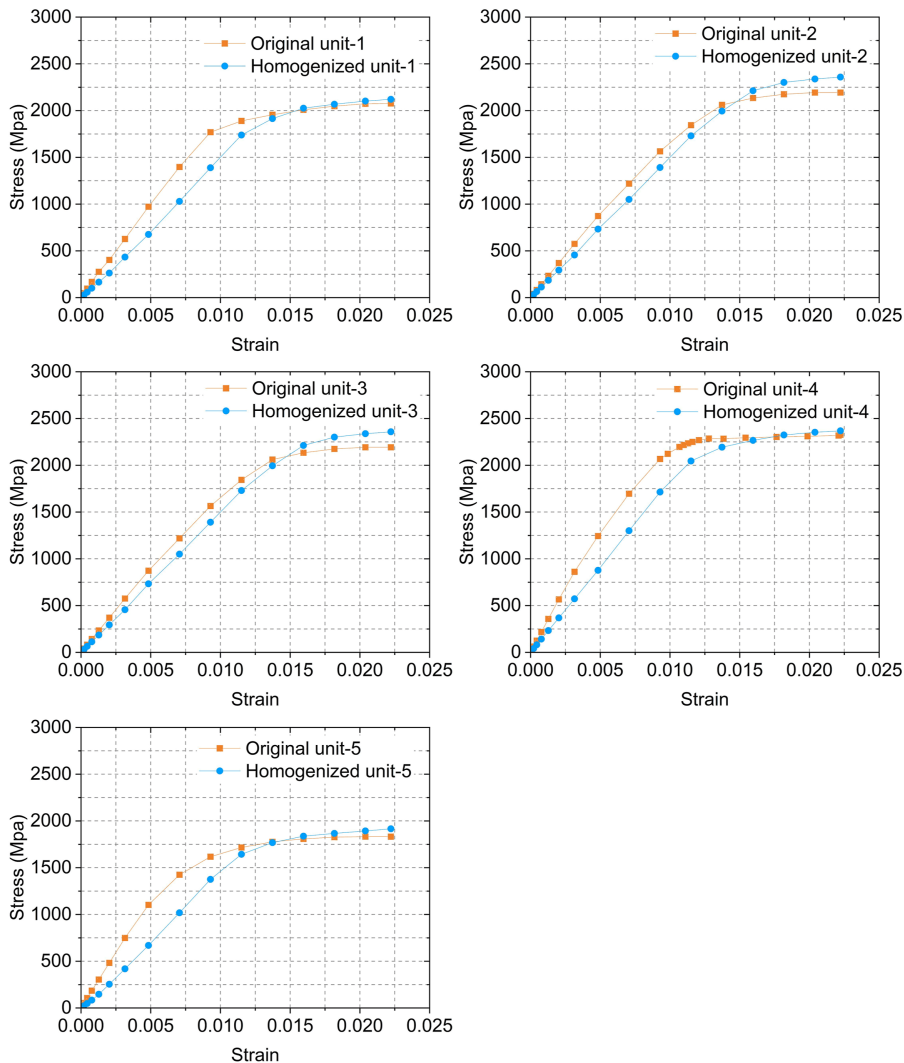


Figure 12. Comparison of equivalent stress–strain curves obtained from the homogenized unit cell responses and those derived from full-fidelity finite element simulations

$$I_e = \frac{\delta_{\text{hom}} - \delta_{\text{NN}}}{\delta_{\text{hom}}} \times 100\%, \quad (24)$$

$$I_v = \frac{\gamma_{\text{hom}} - \gamma_{\text{NN}}}{\gamma_{\text{hom}}} \times 100\%, \quad (25)$$

A higher value of I indicates higher homogenization error and greater error reduction through NN correction. In Figure 13, the heatmap shows the improvement ratios I of the three CNN models across unseen structure (specifically, structure No.27). The color gradations in the heatmap indicate how good the NN correction works. Regions with darker colors reflect lower improvements (lower values of I), while lighter areas indicate larger improvements.

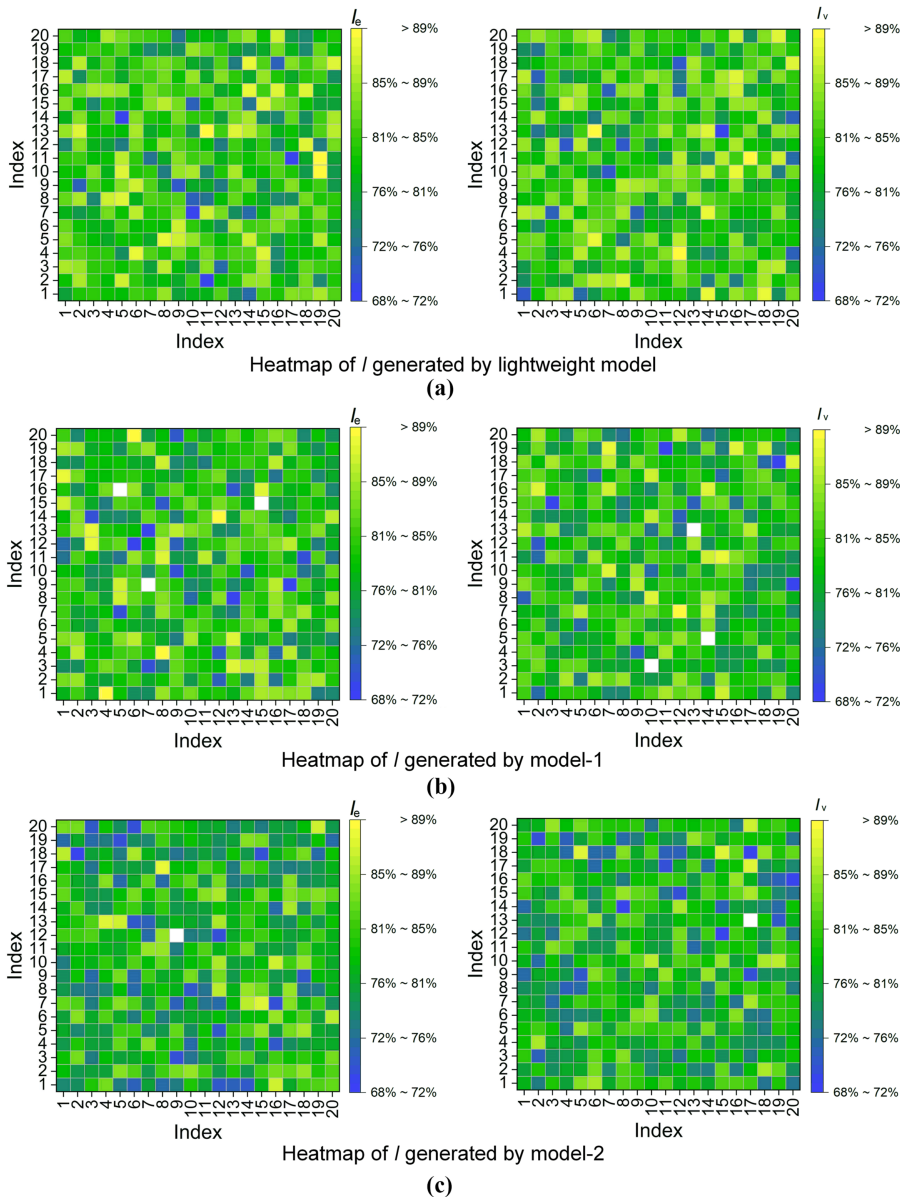


Figure 13. Heatmap visualization of the improvement ratio, i.e. and I_v , representing the reduction in effective stiffness homogenization error due to neural network correction

Across all three models, a large majority of locations recover at least 80% of the high-fidelity correction. Furthermore, the models exhibit similar patterns, with the highest improvement ratios generally observed in regions where local influences are more pronounced. Indeed, the intrinsic error of the homogenization can substantially bias the initial candidate ranking. The designs with target performance that barely clear the threshold may be underestimated and

discarded, creating false negatives. This spatial analysis supports the assertion that the incorporation of local geometry information significantly boosts prediction performance.

Furthermore, three simulation strategies were implemented on the unseen Structures: full-fidelity simulation, homogenized simulation, and homogenized simulation with NN corrections, enabling a rigorous assessment of the models' ability to extrapolate beyond known data. Results show that the model exhibited stable performance when predicting R_e and R_v for unit cells embedded in unfamiliar environments.

The comparison between the full fidelity simulation and the homogenized simulation informed by the lightweight NN model, as shown in Figure 14, reveals the effectiveness of homogenization in simplifying complex composite behavior (Bauer and Böhlke, 2022). In the full model (left), stress and strain distributions show significant local variations due to fiber-matrix interactions and geometric complexities, reflecting microscale heterogeneity. In contrast, the homogenized model (right) exhibits smoother, more uniform fields, indicating that local effects have been averaged into effective material properties. The lightweight model informed result (center), however, exhibits a much closer match to the full-fidelity result. Notably, the red dashed circle highlights regions where the NN-informed model successfully recovers stress intensification patterns absent in the conventional homogenized model, thereby demonstrating its ability to reconstruct local stress effects without explicitly modeling the fine-scale geometry. The strain field shows similar trends. The lightweight model informed simulation captures both magnitude and spatial variation better than the standard model, which oversimplifies deformation. Overall, the lightweight CNN effectively reconstructs local effects, offering high-fidelity mechanical response with reduced computational cost, validating the proposed multi-fidelity framework.

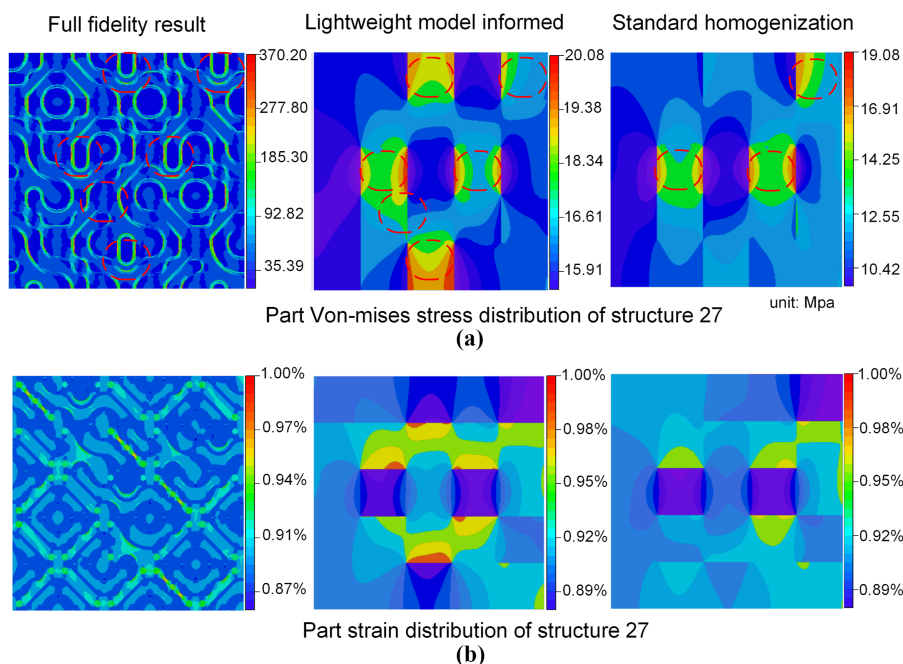


Figure 14. Comparison of stress and strain distributions in (left panel) the detailed heterogeneous model and (right panel) the homogenized model. The heterogeneous model exhibits localized variations due to fiber-matrix interfaces, while the homogenized model displays a smooth distribution consistent with the effective material properties

Figure 15 depicts the equivalent stress–strain curves obtained from the simulation results. The standard homogenized model significantly underestimates the stress response, particularly beyond the elastic regime, due to its inability to account for local stiffness variations and stress intensification. In contrast, the CNN-informed and lightweight models provide stress–strain predictions that closely track the full-fidelity curve across the entire deformation range.

Importantly, despite a dramatic reduction in parameters and compute, the lightweight model maintains accuracy on par with the standard CNN models. By systematically pruning convolutional layers and adopting a block-wise, down-sampling strategy, the lightweight model achieves comparable R^2 and error statistics (within 5% of the standard model), while reducing training time to under 30 min and inference latency by over 10-fold. Such a level of predictive accuracy and robustness supports the model’s suitability for accelerating exploration of targeted composite design without the computational burden of full-scale simulations (Cheung and Mirkhalaf, 2024).

4.3 Computational efficiency and resource analysis

An essential motivation for integrating NN corrections into the multi-fidelity framework is to reduce the overall computational cost with enhanced accuracy (Ghane *et al.*, 2024). To quantify the NN models’ efficiency, we compared three strategies on a representative set of 500 combinations using the three CNN models.

- (1) Full fidelity (FF) simulation for baseline: Structures are simulated by high fidelity FEA.
- (2) Standard multi-fidelity simulation (MF): all 500 structures undergo fast, low-fidelity homogenization; the top 10% by low-fidelity ranking are then re-evaluated with high-fidelity FEA.
- (3) NN informed multi-fidelity (NN-MF): as in the MF approach, but the NN correction is used to adjust low fidelity estimates and re-rank candidates. Only the top 4% are promoted to high fidelity FEA (NN-MF at 4% matched the top objectives achieved by MF at 10%).

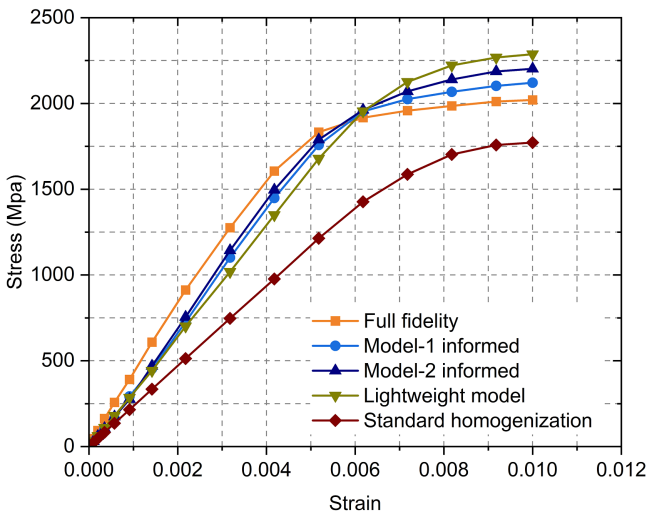


Figure 15. Comparison of equivalent stress–strain curves obtained from the neural network-predicted homogenized unit cell responses and those derived from full-fidelity finite element simulations

All methods were applied to the same 500 assembled combinations, generated once with a fixed random seed. No candidates were added or removed between methods. MF ranks by the baseline homogenized objective; NN-MF ranks by the same objective after applying the CNN ratios to the effective properties. If multiple designs tie at the boundary score, selection is made deterministically by: (1) secondary key equals lower predicted variance, then (2) lexicographic index of the layout (row-major). No FF warm starts were used (to avoid bias). Geometry assembly, node or face sets, and meshing templates were cached and reused across methods; solver tolerances, element types, and hardware settings were identical for MF and NN-MF to ensure fair wall-clock comparisons. All running times were recorded on a 12-core AMD Ryzen9 5900X (3.7 GHz, 32 GB RAM) for CPU-bound homogenizations, and a single NVIDIA RTX 3070 GPU (8 GB) for NN inference. Detailed computational costs on training and inference per structure are listed in Table 5.

Table 6 shows the comparison of computational time cost for the 500 configurations. The standard MF strategy already cuts the simulation time by nearly 90% compared to FF evaluation. Incorporating the NN correction further halves this cost, yielding a total speed-up of over 22-fold relative to the baseline FF strategy, with negligible overhead of NN inference for all 500 samples.

The cost of low-fidelity simulation scales linearly with the number of candidates and remains small per sample (Cheung and Mirkhalaf, 2024); full-fidelity analyses are surely the dominant term in the entire optimization workflow. By dynamically adjusting that term with NN-based re-ranking, the NN-informed multi-fidelity workflow scales very favorably: doubling the design space induces only a proportional increase in low-fidelity cost and NN inference, while the number of full-fidelity simulations can be held roughly constant to maintain a given accuracy level. These results demonstrate that our Lightweight CNN strikes an optimal balance, maintaining parity in accuracy with deeper benchmarks while drastically reducing computational cost and resource demands.

5. Conclusions

This work has introduced and validated a neighbor-aware, NN correction within a multi-fidelity homogenization framework to predict effective property deviations of composite structures assembled from 2-dimensional RVE unit cells. The main findings include.

Table 5. Training and inference resource metrics for CNN models

Model	Epoch time	Converging epoch	GPU mem	Inference
Model-1	180s	100	8 GB	55 ms
Model-2	115s	110	8 GB	61 ms
Lightweight model	35s	80	6 GB	3.5 ms

Table 6. Training and inference resource metrics for benchmark models

Model	Time/structure	Time cost total	Cost reduction
FF simulation	33 ± 1.3 min	16,500 min	/
Homogenized simulation	2.35 ± 0.15 min	1,175 min	93.81%
Model-1 informed	5.85 ± 0.27 min	2,925 min	82.27%
Model-2 informed	5.99 ± 0.31 min	2,995 min	81.84%
Lightweight model	2.5 ± 0.12 min	1,250 min	92.42%

- (1) Local-neighbor influence captured by NN model. By encoding each unit's local context into a 5×5 matrix and training the NN model to predict the deviation, over 80% of the high-fidelity accuracy can be recovered while only requiring a small portion of full FE solves.
- (2) Models' generalization to unseen assemblies. When evaluated on previously unseen structures, the NN predictions maintain sub-10% error on effective property deviations.
- (3) Substantial Computational Savings. By systematically pruning the architecture of conventional CNN models and implementing a down-sampling strategy, our lightweight model achieves a 92.42% reduction in total simulation cost while holding error below 10%.

Despite these promising outcomes, the approach has clear limitations. The training dataset is obtained from FEA on planar 2-dimensional unit cells with minor plastic behavior (Xu and An, 2022); extensions to 3-dimensional units or to inelastic and damage-evolution models would require retraining on higher-dimensional data (multi-channel inputs) and potentially more complex NN architectures. Likewise, while a fixed 5×5 local window proved sufficient for the materials and layouts studied here, highly anisotropic fiber trajectories or graded-volume architectures may demand adaptive neighborhood sizes or trainable attention weights. Given the localized, normalized targets and strong cross-structure heterogeneity, single scalars may obscure systematic, context-dependent effects. Accuracy is instead summarized via by-structure splits and steady-state validation R^2 , future work may add complementary error metrics. Finally, the model has so far been demonstrated on a regular three-phase system (Tian and Qi, 2023; Cheung and Mirkhalaf, 2024); material systems with vastly different phase contrasts or interface behaviors will require further transfer-learning experiments to verify data efficiency and accuracy.

Future developments will extend the framework to include full plasticity and damage evolution to capture nonlinear and dissipative behaviors of composites. The approach will also be scaled to larger assemblies to examine size convergence of effective properties. Methodologically, adaptive neighborhood learning using attention or graph-based models, uncertainty-guided active sampling, and physics-informed loss functions enforcing mechanical symmetry and bounded corrections will be explored to enhance robustness and extrapolation to highly heterogeneous configurations (Ghane *et al.*, 2024). Additionally, the methodology will be extended to three-dimensional simulations to capture out-of-plane interactions and validate its robustness for realistic composite architectures.

References

- Ahn, J.-G., Yang, H.-Ik and Kim, J.-G. (2022), "Multi-fidelity meta modeling using composite neural network with online adaptive basis technique", *Computer Methods in Applied Mechanics and Engineering*, Vol. 388, 114258, doi: [10.1016/j.cma.2021.114258](https://doi.org/10.1016/j.cma.2021.114258).
- An, H., Youn, B.D. and Kim, H.S. (2023), "Variable-stiffness composite optimization using dynamic and exponential multi-fidelity surrogate models", *International Journal of Mechanical Sciences*, Vol. 257, 108547, doi: [10.1016/j.ijmecsci.2023.108547](https://doi.org/10.1016/j.ijmecsci.2023.108547).
- Azad, M.M., Shah, A.u.R., Prabhakar, M. and Kim, H.S. (2025), "Deep learning-based autonomous morphological fracture analysis of fiber-reinforced composites", *Engineering Failure Analysis*, Vol. 170, 109292, doi: [10.1016/j.engfailanal.2025.109292](https://doi.org/10.1016/j.engfailanal.2025.109292).
- Barbero, E.J. and Barbero, E.J. (2023), *Finite Element Analysis of Composite Materials Using Abaqus®*, 2nd ed., Boca, Raton, p. 571, ISBN: 978-1-00-310815-3.
- Bauer, J.K. and Böhlke, T. (2022), "On the dependence of orientation averaging mean field homogenization on planar fourth-order fiber orientation tensors", *Mechanics of Materials*, Vol. 170, 104307, doi: [10.1016/j.mechmat.2022.104307](https://doi.org/10.1016/j.mechmat.2022.104307).

- Bishara, D., Xie, Y., Liu, W.K. and Li, S. (2023), "A state-of-the-art review on machine learning-based multiscale modeling, simulation, homogenization and design of materials", *Archives of Computational Methods in Engineering*, Vol. 30 No. 1, pp. 191-222, doi: [10.1007/s11831-022-09795-8](https://doi.org/10.1007/s11831-022-09795-8).
- Cai, H. and Aref, A.J. (2015), "A genetic algorithm-based multi-objective optimization for hybrid fiber reinforced polymeric deck and cable system of cable-stayed bridges", *Structural and Multidisciplinary Optimization*, Vol. 52 No. 3, pp. 583-594, doi: [10.1007/s00158-015-1266-4](https://doi.org/10.1007/s00158-015-1266-4).
- Cheung, H.L. and Mirkhalaf, M. (2024), "A multi-fidelity data-driven model for highly accurate and computationally efficient modeling of short fiber composites", *Composites Science and Technology*, Vol. 246, 110359, doi: [10.1016/j.compscitech.2023.110359](https://doi.org/10.1016/j.compscitech.2023.110359).
- Coelho, R.P.C., Carvalho Alves, A.F., Nogueira Pires, T. and Andrade Pires, F. (2025), "A composite Bayesian optimisation framework for material and structural design", *Computer Methods in Applied Mechanics and Engineering*, Vol. 434, 117516, doi: [10.1016/j.cma.2024.117516](https://doi.org/10.1016/j.cma.2024.117516).
- Croom, B.P., Berkson, M., Mueller, R.K., Presley, M. and Storck, S. (2022), "Deep learning prediction of stress fields in additively manufactured metals with intricate defect networks", *Mechanics of Materials*, Vol. 165, 104191, doi: [10.1016/j.mechmat.2021.104191](https://doi.org/10.1016/j.mechmat.2021.104191).
- De, B., Bera, M., Bhattacharjee, D., Ray, B.C. and Mukherjee, S. (2024), "A comprehensive review on fiber-reinforced Polymer Composites : raw materials to applications, recycling, and waste management", *Progress in Materials Science*, Vol. 146, 101326, doi: [10.1016/j.pmatsci.2024.101326](https://doi.org/10.1016/j.pmatsci.2024.101326).
- Dou, G., Zhao, L., Xia, W., Jiang, H., Piao, Z. and Peng, G. (2025), "Measurement of interfacial bonding strength between the micro-spherical filler and the matrix in microcapsule/epoxy composites", *Composites Science and Technology*, Vol. 265, 111134, doi: [10.1016/j.compscitech.2025.111134](https://doi.org/10.1016/j.compscitech.2025.111134).
- Duan, Z., Liu, Y., Jin, H., Yan, J. and Zhu, J. (2025), "Concurrent multi-scale design optimization of fiber-reinforced composite material based on an adaptive normal distribution fiber optimization scheme for minimum structural compliance and additive manufacturing", *Computer Methods in Applied Mechanics and Engineering*, Vol. 434, 117596, doi: [10.1016/j.cma.2024.117596](https://doi.org/10.1016/j.cma.2024.117596).
- Fan, S., Zhang, J., Wang, B., Chen, J., Yang, W., Liu, W. and Li, Y. (2023), "A deep learning method for fast predicting curing process-induced deformation of aeronautical Composite Structures", *Composites Science and Technology*, Vol. 232, 109844, doi: [10.1016/j.compscitech.2022.109844](https://doi.org/10.1016/j.compscitech.2022.109844).
- Ghane, E., Fagerström, M. and Mirkhalaf, M. (2024), "Recurrent neural networks and transfer learning for predicting elasto-plasticity in woven composites", *European Journal of Mechanics – A: Solids*, Vol. 107, 105378, doi: [10.1016/j.euromechsol.2024.105378](https://doi.org/10.1016/j.euromechsol.2024.105378).
- Hossain, M.T., Shahid, M.A., Mahmud, N., Habib, A., Rana, M.M., Khan, S.A. and Hossain, M.D. (2024), "Research and application of polypropylene: a review", *Discover Nano*, Vol. 19 No. 1, p. 2, doi: [10.1186/s11671-023-03952-z](https://doi.org/10.1186/s11671-023-03952-z).
- Hyde, A., He, J., Cui, X., Lua, J. and Liu, L. (2020), "Effects of microvoids on strength of unidirectional fiber-reinforced composite materials", *Composites Part B: Engineering*, Vol. 187, 107844, doi: [10.1016/j.compositesb.2020.107844](https://doi.org/10.1016/j.compositesb.2020.107844).
- Jain, A. (2019), "Micro and mesomechanics of fibre reinforced composites using mean field homogenization formulations: a review", *Materials Today Communications*, Vol. 21, 100552, ISSN 2352-4928, doi: [10.1016/j.mtcomm.2019.100552](https://doi.org/10.1016/j.mtcomm.2019.100552).
- Kim, C., Lee, J. and Yoo, J. (2021), "Machine learning-combined topology optimization for functionary graded composite structure design", *Computer Methods in Applied Mechanics and Engineering*, Vol. 387, 114158, doi: [10.1016/j.cma.2021.114158](https://doi.org/10.1016/j.cma.2021.114158).
- Kim, D.-J., Kim, G.W., Baek, J.h., Nam, B. and Kim, H.S. (2023), "Prediction of stress-strain behavior of carbon fabric woven composites by deep neural network", *Composite Structures*, Vol. 318, 117073, doi: [10.1016/j.compstruct.2023.117073](https://doi.org/10.1016/j.compstruct.2023.117073).
- Lee, H., Lee, S. and Ryu, S. (2023), "Advancements and challenges of micromechanics based homogenization for the short fiber reinforced composites", *Multiscale Science and Engineering*, Vol. 5 No. 3, pp. 133-146, doi: [10.1007/s42493-024-00100-2](https://doi.org/10.1007/s42493-024-00100-2).

- Li, M., Li, S., Tian, Y., Fu, Y., Pei, Y., Zhu, W. and Ke, Y. (2023), "A deep learning convolutional neural network and multi-layer perceptron hybrid fusion model for predicting the mechanical properties of carbon fiber", *Materials and Design*, Vol. 227, 111760, doi: [10.1016/j.matdes.2023.111760](https://doi.org/10.1016/j.matdes.2023.111760).
- Li, J., Yao, W., Lu, Y., Chen, J., Sun, Y. and Hu, X. (2024a), "High fidelity FEM based on deep learning for arbitrary composite material structure", *Composite Structures*, Vol. 340, 118176, doi: [10.1016/j.compstruct.2024.118176](https://doi.org/10.1016/j.compstruct.2024.118176).
- Li, J., Xin, H., Zhang, Y., Gao, Q. and Zhang, H. (2024b), "Ultimate resistance and fatigue performance predictions of woven-based fiber reinforced polymers using a computational homogenization method", *International Journal of Structural Integrity*, Vol. 15 No. 4, pp. 795-810, doi: [10.1108/ijsi-03-2024-0049](https://doi.org/10.1108/ijsi-03-2024-0049).
- Li, M., Wang, B., Hu, J., Li, G., Ding, P., Ji, C. and Wang, B. (2024c), "Artificial neural network-based homogenization model for predicting multiscale thermo-mechanical properties of woven composites", *International Journal of Solids and Structures*, Vol. 301, 112965, doi: [10.1016/j.ijsolstr.2024.112965](https://doi.org/10.1016/j.ijsolstr.2024.112965).
- Li, Y., Qin, H., Jia, L., Tay, T.E., Tan, V.B.C. and Liu, Y. (2024d), "Microstructure dependent transverse strength criterion for UD-CFRP composites via computational micromechanics and machine learning", *Composites Science and Technology*, Vol. 251, 110551, doi: [10.1016/j.compscitech.2024.110551](https://doi.org/10.1016/j.compscitech.2024.110551).
- Li, X., Li, M., Zhang, F., Kong, F., Yang, D., Qu, W. and Ke, Y. (2025), "A novel method for constructing 3D void RVE elements and rapid homogenization of composite materials", *Composite Structures*, Vol. 360, 119040, doi: [10.1016/j.compstruct.2025.119040](https://doi.org/10.1016/j.compstruct.2025.119040).
- Liu, J., Fu, Y., Chen, Y., Jiang, J., Zhang, Y.L., Xu, Y. and Lin, S. (2022), "A novel algorithm in stochastic chopped carbon fiber composite structure-A study of RVE size effect and homogenization response of directional modulus", *en. Applied Composite Materials*, Vol. 29 No. 2, pp. 565-595, doi: [10.1007/s10443-021-09947-2](https://doi.org/10.1007/s10443-021-09947-2).
- Liu, M., Li, H., Zhou, H., Zhang, H. and Huang, G. (2024a), "Development of machine learning methods for mechanical problems associated with fibre composite materials: a review", *Composites Communications*, Vol. 49, 101988, doi: [10.1016/j.coco.2024.101988](https://doi.org/10.1016/j.coco.2024.101988).
- Liu, Z., Ge, J., Li, H., Liu, X., Wang, B. and Liang, J. (2024b), "Mesoscale modeling of woven composite twisted structures combining digital element embedded model and affine transform", *Composites Science and Technology*, Vol. 249, 110504, doi: [10.1016/j.compscitech.2024.110504](https://doi.org/10.1016/j.compscitech.2024.110504).
- Martínez-Muñoz, D., García, J., Martí, J.V. and Yepes, V. (2023), "Hybrid swarm intelligence optimization methods for low-embodied energy steel-concrete composite bridges", *Mathematics*, Vol. 11 No. 1, p. 140, doi: [10.3390/math11010140](https://doi.org/10.3390/math11010140).
- Nguyen-Thanh, N., Li, W., Huang, J. and Zhou, K. (2022), "Multi phase-field modeling of anisotropic crack propagation in 3D fiber-reinforced composites based on an adaptive isogeometric meshfree collocation method", *Computer Methods in Applied Mechanics and Engineering*, Vol. 393, 114794, doi: [10.1016/j.cma.2022.114794](https://doi.org/10.1016/j.cma.2022.114794).
- Niu, C., Liu, B., Bai, C., Guo, L., Chen, L. and Tang, J. (2024), "Research on composite material riveting simulation method for engineering applications", *International Journal of Structural Integrity*, Vol. 15 No. 5, pp. 836-850, doi: [10.1108/ijsi-06-2024-0091](https://doi.org/10.1108/ijsi-06-2024-0091).
- Novac, O.-C., Chirodea, M.C., Novac, C.M., Bizon, N., Oproescu, M., Stan, O.P. and Gordan, C.E. (2022), "Analysis of the application efficiency of TensorFlow and PyTorch in convolutional neural network", *en. Sensors*, Vol. 22 No. 22, 8872, doi: [10.3390/s22228872](https://doi.org/10.3390/s22228872).
- Pan, W., Sun, L., Yang, X., Zhang, Y., Sun, J., Shang, J., Yang, Z. and Xu, C. (2025), "Multi-scale impact of geometric uncertainty on the interface bonding reliability of metal/polymer-based composites hybrid (MPH) structures", *Composite Structures*, Vol. 351, 118640, doi: [10.1016/j.compstruct.2024.118640](https://doi.org/10.1016/j.compstruct.2024.118640).
- Qi, L., Tian, W. and Zhou, J. (2015), "Numerical evaluation of effective elastic properties of composites reinforced by spatially randomly distributed short fibers with certain aspect ratio", *Composite Structures*, Vol. 131, pp. 843-851, doi: [10.1016/j.compstruct.2015.06.045](https://doi.org/10.1016/j.compstruct.2015.06.045).

- Ramesh, M., Rajeshkumar, L.N., Srinivasan, N., Kumar, D.V. and Balaji, D. (2022), "Influence of filler material on properties of fiber-reinforced Polymer composites: a review", *E-polymers*, Vol. 22 No. 1, pp. 898-916, doi: [10.1515/epoly-2022-0080](https://doi.org/10.1515/epoly-2022-0080).
- Sathish, T., Sunagar, P., Singh, V., Boopathi, S., sathyamurthy, R., Al-Enizi, A.M., Pandit, B., Gupta, M. and Sehgal, S.S. (2023), "Characteristics estimation of natural fibre reinforced plastic composites using deep multi-layer perceptron (MLP) technique", *Chemosphere*, Vol. 337, 139346, doi: [10.1016/j.chemosphere.2023.139346](https://doi.org/10.1016/j.chemosphere.2023.139346).
- Shen, W., Dong, J. and Huan, X. (2025), "Variational sequential optimal experimental design using reinforcement learning", *Computer Methods in Applied Mechanics and Engineering*, Vol. 444, 118068, doi: [10.1016/j.cma.2025.118068](https://doi.org/10.1016/j.cma.2025.118068).
- Shih, Y.-C., Mohamed, M., Ge, J., Sen, S., Pillay, S. and Ning, H. (2023), "Application of interlaminar shear strength and finite element modeling for failure analysis of 3D printed continuous fiber-reinforced composites", *Prog Addit Manuf*, Vol. 8 No. 6, pp. 1501-1516, doi: [10.1007/s40964-023-00417-8](https://doi.org/10.1007/s40964-023-00417-8).
- Song, Y., Qu, Z., Liao, H. and Ai, S. (2023), "Material twins generation of woven polymer composites based on ResL-U-Net convolutional neural networks", *Composite Structures*, Vol. 307, 116672, doi: [10.1016/j.compstruct.2023.116672](https://doi.org/10.1016/j.compstruct.2023.116672).
- Storm, J., Rocha, I.B.C.M. and van der Meer, F.P. (2024), "A microstructure-based graph neural network for accelerating multiscale simulations", *Computer Methods in Applied Mechanics and Engineering*, Vol. 427, 117001, doi: [10.1016/j.cma.2024.117001](https://doi.org/10.1016/j.cma.2024.117001).
- Su, T.-H., Huang, S.J., Jean, J.G. and Chen, C.S. (2022), "Multiscale computational solid mechanics: data and machine learning", *Journal of Mechanics*, Vol. 38, pp. 568-585, doi: [10.1093/jom/ufac037](https://doi.org/10.1093/jom/ufac037).
- Talreja, R. and Waas, A.M. (2022), "Concepts and definitions related to mechanical behavior of fiber reinforced composite materials", *Composites Science and Technology*, Vol. 217, 109081, doi: [10.1016/j.compscitech.2021.109081](https://doi.org/10.1016/j.compscitech.2021.109081).
- Tian, W. and Qi, L. (2023), "Unified periodic boundary condition for homogenizing the thermo-mechanical properties of composites", *Applied Mathematical Modelling*, Vol. 121, pp. 252-269, doi: [10.1016/j.apm.2023.04.024](https://doi.org/10.1016/j.apm.2023.04.024).
- Wang, Z.Z. and Sobey, A. (2020), "A comparative review between Genetic Algorithm use in composite optimisation and the state-of-the-art in evolutionary computation", *Composite Structures*, Vol. 233, 111739, doi: [10.1016/j.compstruct.2019.111739](https://doi.org/10.1016/j.compstruct.2019.111739).
- Wang, T., Cao, J., Guo, T., Tu, Y., Wang, C., Sas, G. and Elfgren, L. (2024), "The role of deep learning in reducing computational cost when simulating chloride ion attack on hydrated calcium silicate with molecular dynamics", *Construction and Building Materials*, Vol. 417, 135257, doi: [10.1016/j.conbuildmat.2024.135257](https://doi.org/10.1016/j.conbuildmat.2024.135257).
- Wang, Z., Wang, S., Ma, C. and Yang, Z. (2025), "The prediction of homogenized effective properties of continuous fiber composites based on a deep transfer learning approach", *Composites Science and Technology*, Vol. 262, 111050, doi: [10.1016/j.compscitech.2025.111050](https://doi.org/10.1016/j.compscitech.2025.111050).
- Wen, Z., Zhao, H., Yan, X., Li, P. and Ding, H. (2024), "Grinding performance investigation of multidirectional CF/PEEK composites—a comparative with unidirectional CF/PEEK", *Polymer Composites*, Vol. 45 No. 10, pp. 8889-8904, doi: [10.1002/pc.28383](https://doi.org/10.1002/pc.28383).
- Xu, M. and An, B. (2022), "Plastic deformation of polymer matrix composites reinforced with beaded fibers", *International Journal of Solids and Structures*, Vol. 256, 111985, doi: [10.1016/j.ijsolstr.2022.111985](https://doi.org/10.1016/j.ijsolstr.2022.111985).
- Yang, C., Kim, Y., Ryu, S. and Gu, G.X. (2019), "Using convolutional neural networks to predict composite properties beyond the elastic limit", *MRS Communications*, Vol. 9 No. 2, pp. 609-617, doi: [10.1557/mrc.2019.49](https://doi.org/10.1557/mrc.2019.49).
- Zhan, Z., Zhang, M., He, X., Li, X., Wang, Z., Chen, X., Han, B., Hu, W., Meng, Q. and Li, H. (2025), "Advances in machine learning for predicting fatigue behavior: from material properties to fatigue life and fatigue crack growth", *International Journal of Structural Integrity*, Vol. 16 No. 6, pp. 1-62, doi: [10.1108/ijsi-12-2024-0209](https://doi.org/10.1108/ijsi-12-2024-0209).

- Zhao, W., Fu, X., Xu, X., Zhang, L., Wang, L., Lin, J., Hu, Y., Gao, L., Du, L. and Tian, X. (2024), "Design of multicomponent thermosetting polymers with enhanced tensile properties through active learning", *Composites Science and Technology*, Vol. 256, 110779, doi: [10.1016/j.compscitech.2024.110779](https://doi.org/10.1016/j.compscitech.2024.110779).
- Zhu, F., Zhao, J., Duan, L. and Zhuang, B. (2025), "Microscopic finite element modeling and fracture properties of particulate filled Polymer Composites", *Polymer Composites*, Vol. 46 No. 8, pp. 7012-7026, doi: [10.1002/pc.29410](https://doi.org/10.1002/pc.29410).

Corresponding author

Shujian Chen can be contacted at: shujian.chen@uq.edu.au