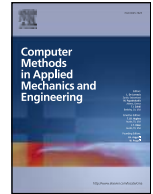


Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

Micrometer: Micromechanics transformer for predicting full field mechanical responses of heterogeneous materials

Sifan Wang^{a,1,*}, Tong-Rui Liu^{b,1}, Shyam Sankaran^c, Paris Perdikaris^c

^a Institution for Foundation of Data Science, Yale University, New Haven, CT, 06520, USA

^b Department of Aeronautics, Imperial college London, London, SW7 2AZ, UK

^c Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, 19104, USA

ARTICLE INFO

Keywords:

Deep learning
Computational solid mechanics
Operator learning
Composite materials
Lippmann-Schwinger equation

ABSTRACT

Predicting mechanical responses of heterogeneous materials across scales remains a significant challenge. Traditional computational methods often struggle with complex and multiscale nature of these materials, limiting their effectiveness in real-world applications. In this paper, we introduce *Micrometer*, a vision transformer based deep learning model designed to predict full field mechanical responses of heterogeneous materials, bridging the gap between computer vision and solid mechanics problems. We show that *Micrometer*, trained on a large-scale high-resolution dataset of 2D fiber-reinforced composites, can achieve state-of-the-art performance in predicting microscale strain fields across a wide range of material properties and loading conditions. Our model demonstrates accuracy and computational efficiency in applications such as computational homogenization and multiscale modeling, reducing computational time by up to two orders of magnitude compared to conventional numerical solvers while maintaining less than 1% errors in predicting macroscale stress fields. Furthermore, we showcase *Micrometer*'s adaptability through transfer learning experiments on new materials with limited data, highlighting its potential to tackle diverse scenarios in computational solid mechanics. These results represent a significant step towards AI-driven innovation in materials science, addressing the limitations of traditional numerical methods and paving the way for more efficient simulations of heterogeneous materials across various industrial applications.

1. Introduction

Heterogeneous materials, such as composites, concretes, and metals, are the unsung heroes of modern engineering. Their complex microstructures grant them exceptional properties, making them indispensable in industries ranging from aerospace to bioengineering [1,2]. However, predicting how these materials behave under stress is a formidable challenge. Their intrinsic multiscale nature, with properties varying across different length scales, makes traditional analytical and experimental methods often fall short [3].

Over the past few decades, computational methods have emerged as powerful tools to tackle this challenge. Techniques like finite element analysis (FEA) [4,5] and fast Fourier transform (FFT) methods [6,7] have enabled engineers to simulate material behavior with increasing accuracy. More recently, several studies have leveraged physics-informed machine learning [8] in various

* Corresponding author.

E-mail addresses: sifan.wang@yale.edu (S. Wang), tongrui.liu18@alumni.imperial.ac.uk (T.-R. Liu), shyamss@seas.upenn.edu (S. Sankaran), pqp@seas.upenn.edu (P. Perdikaris).

¹ These authors have contributed equally to this work.

<https://doi.org/10.1016/j.cma.2025.118373>

Received 1 March 2025; Received in revised form 19 July 2025; Accepted 3 September 2025

Available online 12 October 2025

0045-7825/© 2025 Published by Elsevier B.V.

micromechanical settings. For instance, ensemble wavelet-learning has been applied to predict effective mechanical properties of concrete composites [9], while higher-order multiscale PINNs have been proposed for elastic analysis of authentic composite materials [10]. Other works developed physics-informed homogenization methods for nanoporous composites with energetic interfaces [11], and tackled thermochemical changes in composite materials [12].

Yet, these methods are not without limitations. They often require high computational resources, are constrained to solving one specific instance of a PDE at a time, and lack flexibility when handling variations in geometry, material parameters, or boundary conditions. In contrast to operator learning frameworks that can solve families of parametrized PDEs, traditional computational approaches such as FEM or spectral methods must recompute solutions from scratch for each new configuration. These constraints lead to inefficiencies in practical applications, including material and structural design under uncertainty [13,14], topology optimization [15,16], and concurrent multiscale modeling [17–19].

Recent advances in artificial intelligence (AI), particularly operator learning, offer new possibilities to predict material behavior. This approach, which learns mappings between function spaces [20], has shown promise in fields like fluid dynamics through techniques such as Fourier Neural Operators [21–25] and Deep Operator Networks [26–32]. However, the field of computational solid mechanics has begun to explore the potential of modern AI advancements, such as transformers and diffusion models [33–36], their application in solid mechanics faces significant challenges. The scarcity of high-quality, large-scale datasets in the field has limited progress. Current AI-driven approaches in mechanics often rely on simple, small-scale models to predict either macro-mechanical responses or micro-stress distributions, and are typically limited to specific loading conditions or a narrow range of materials [37–43]. These limitations hinder the development of robust, generalizable AI solutions for complex solid mechanics problems, particularly those involving heterogeneous materials and multiscale phenomena.

To overcome these limitations, our work leverages the Lippmann-Schwinger integral equation—a widely adopted formulation in computational micromechanics—to define an operator learning problem. This equation expresses the mechanical equilibrium condition in terms of a convolution operator acting on the Green’s function of a reference medium. Its integral form allows efficient numerical treatment via FFT-based solvers and naturally accommodates heterogeneous material coefficients. Importantly, it lends itself to operator learning because the mapping from material distribution (elastic tensor) to strain or stress fields can be framed as a parametric solution operator.

In this context, we present CSMBench (Computational Solid Mechanics Benchmark), an open-source dataset collection for operator learning in micromechanics. The CMME subset focuses on linear elasticity and contains 40,000 training and 2000 test samples. Inputs are periodic RVEs with fiber volume fractions between 40% and 60%, generated using the RAND_uSTRU_GEN algorithm and material parameters sampled via Latin hypercube sampling. Outputs are solutions to the Lippmann-Schwinger equation computed with an FFT-based homogenization method. RVEs are discretized at 512×512 pixels with fiber radius around $3.5 \mu\text{m}$. The dataset, with full documentation and tools, is publicly available at <https://github.com/sifanexisted/micrometer>, providing a robust benchmark for AI in solid mechanics.

Building on this foundation, we propose **Micrometer**, a transformer-based operator learning framework for predicting the mechanical response of heterogeneous media. Our key contributions are:

Our key and original contributions are summarized as follows:

- We formulate a novel operator learning problem of mapping the fourth-order elastic tensor to the strain concentration tensor governed by the parametric Lippmann-Schwinger equation, enabling efficient evaluation of mechanical responses across varying microstructures and material properties, under any external loading conditions.
- To overcome the data scarcity, we generate and release CSMBench/CMME; the first large-scale, high-fidelity, high-resolution dataset in micromechanics, which covers a vast range of volume fractions, microstructural morphologies and material properties.
- We develop a transformer-based AI model to learn the target operator, outperforming several popular PDE surrogates and exhibiting strong scalability with respect to data size, model parameters, and computational resources.
- We demonstrate the accuracy and computational efficiency of the proposed model by applying it to fundamental problems in micromechanics, including computational homogenization and concurrent multiscale modeling.
- We showcase the adaptability of our model through transfer learning experiments on new microstructures with limited data, highlighting its potential to tackle diverse scenarios in computational solid mechanics.

Taken all together, our work represents a significant step towards AI-driven innovation and computer vision in computational mechanics, providing a crucial resource for future research in this field.

The structure of this paper is as follows Section 2 introduces the essential mathematical preliminaries, establishing the theoretical foundation for our work. In Section 3, we formulate the problem of interest, identifying the appropriate target solution operator and defining the corresponding input and output function spaces. We then details the pipeline for generating the CSMBench/CMME dataset, crucial for training and evaluating our model. Section 4 elucidates the general framework of operator learning and introduces our proposed model, *Micrometer*. In Section 5, we validate *Micrometer*’s performance through comprehensive comparisons against other state-of-the-art AI baselines. Next, we showcase its robustness and computational efficiency in addressing fundamental problems in computational solid mechanics, including homogenization and concurrent multiscale modeling. Furthermore, we illustrate the adaptability of our model through fine-tuning experiments on out-of-distribution scenarios with limited data. Finally, in Section 6, we conclude by summarizing our key findings, discussing the current limitations of our approach, and outlining promising directions for future research.

2. Mathematical preliminaries

This section provides an overview of micro-elasticity in computational micromechanics, which involves solving the Lippmann-Schwinger equation in a 2D representative volume element (RVE). Let Ω be a computational domain of a RVE in 2D, the quasi-static linear elasticity problem in the absence of body forces can be formulated as

$$\begin{cases} \frac{\partial \sigma_{ij}(\mathbf{x})}{\partial x_j} = 0, \\ \sigma_{ij}(\mathbf{x}) = C_{ijkl}(\mathbf{x}) : \varepsilon_{kl}(\mathbf{x}), \\ \varepsilon_{ij}(\mathbf{x}) = \frac{1}{2} \left(\frac{\partial s_i(\mathbf{x})}{\partial x_j} + \frac{\partial s_j(\mathbf{x})}{\partial x_i} \right). \end{cases} \quad (2.1)$$

The above relations represent equilibrium, constitutive, and infinitesimal strain-displacement equations, respectively. Herein, σ_{ij} , ε_{ij} , C_{ijkl} and s_i denote stress, strain, fourth-order elastic tensor and displacement, respectively. Since the RVE is intrinsically heterogeneous, local constitutive relations and physical fields are denoted as functions of position $\mathbf{x} \in \Omega$. The local elastic tensor $C_{ijkl}(\mathbf{x})$ can be expressed as a pair of Lamé constants as

$$C_{ijkl}(\mathbf{x}) = \lambda(\mathbf{x})\delta_{ij}\delta_{kl} + \mu(\mathbf{x})(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \quad (2.2)$$

where $\lambda(\mathbf{x})$ and $\mu(\mathbf{x})$ can be computed through Young's modulus $E(\mathbf{x})$ and Poisson ratio $\nu(\mathbf{x})$ as

$$\lambda(\mathbf{x}) = \frac{E(\mathbf{x})\nu(\mathbf{x})}{(1+\nu(\mathbf{x}))(1-2\nu(\mathbf{x}))}, \quad \mu(\mathbf{x}) = \frac{E(\mathbf{x})}{2(1+\nu(\mathbf{x}))}. \quad (2.3)$$

To close the above system, we impose periodic boundary conditions (PBC) for the RVE. Moreover, the point-wise displacement field can be decomposed as

$$s_i(\mathbf{x}) = \tilde{s}_i(\mathbf{x}) + \bar{s}_i. \quad (2.4)$$

Without considering strain gradient effects and taking the symmetric gradient of the above equation, the corresponding strain field can be split as

$$\varepsilon_{ij}(\mathbf{x}) = \tilde{\varepsilon}_{ij}(\mathbf{x}) + \bar{\varepsilon}_{ij}, \quad (2.5)$$

where $\tilde{s}_i(\mathbf{x})$, $\tilde{\varepsilon}_{ij}(\mathbf{x})$, \bar{s}_i and $\bar{\varepsilon}_{ij}$ denote the displacement fluctuation, strain field fluctuation, average displacement, and average displacement gradient, respectively. Due to the nature of periodic boundary conditions, $\tilde{s}_i(\mathbf{x})$ and $\tilde{\varepsilon}_{ij}(\mathbf{x})$ are periodic, while the traction vector $\boldsymbol{\sigma} \cdot \mathbf{n}$ behaves anti-periodically on the boundary between adjacent RVEs. Here \mathbf{n} denotes the outward normal vector between RVE boundaries.

Next, we introduce a linear elastic homogeneous reference medium C_{ijkl}^0 [44,45] and express $C_{ijkl}(\mathbf{x})$ as

$$C_{ijkl}(\mathbf{x}) = C_{ijkl}^0 + [C_{ijkl}(\mathbf{x}) - C_{ijkl}^0], \quad (2.6)$$

$$C_{ijkl}^0 = \lambda^0 \delta_{ij} \delta_{kl} + \mu^0 (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}), \quad (2.7)$$

where λ^0 and μ^0 are the Lamé constants of the elastic homogeneous reference material. Note that λ^0 and μ^0 can be computed by the Young's modulus E^0 and Poisson ratio ν^0 through Eq. (2.3). Then, the stress σ_{ij} can be divided into two parts

$$\sigma_{ij}(\mathbf{x}) = \tau_{ij}(\mathbf{x}) + C_{ijkl}^0 : \varepsilon_{kl}(\mathbf{x}), \quad (2.8)$$

$$\tau_{ij}(\mathbf{x}) = (C_{ijkl}(\mathbf{x}) - C_{ijkl}^0) : \varepsilon_{kl}(\mathbf{x}), \quad (2.9)$$

where $\tau_{ij}(\mathbf{x})$ is a polarization stress that quantifies the difference between the stress in the real material and the stress in the reference material under the same strain. Combining Eq. (2.5) and substituting Eq. (2.8) into the first equation in Eq. (2.1), we obtain

$$C_{ijkl}^0 \frac{\partial \tilde{\varepsilon}_{ij}(\mathbf{x})}{\partial x_i} = - \frac{\partial \tau_{ij}(\mathbf{x})}{\partial x_i}. \quad (2.10)$$

Because the second term in the right-hand side of Eq. (2.8) is linear, we can obtain the Lippmann-Schwinger equation [46] for strain field fluctuation in a convolution form by virtue of the Green's function $G_{ijkl}^{(0)}(\mathbf{x}, \mathbf{x}')$,

$$\varepsilon_{ij}(\mathbf{x}) + G_{ijkl}^{(0)}(\mathbf{x}, \mathbf{x}') * \tau_{kl}(\mathbf{x}') - \bar{\varepsilon}_{ij} = 0. \quad (2.11)$$

The above equation establishes the relationship between the strain field fluctuation $\tilde{\varepsilon}_{ij}(\mathbf{x})$ and polarization stress $\tau_{ij}(\mathbf{x})$ in Euclidean space. It can also be written in an integral form as below

$$\varepsilon_{ij}(\mathbf{x}) + \int_{\Omega} G_{ijkl}^{(0)}(\mathbf{x}, \mathbf{x}') : \tau_{kl}(\mathbf{x}') d\mathbf{x}' - \bar{\varepsilon}_{ij} = 0. \quad (2.12)$$

3. Problem formulation

Our primary objective is to develop a deep learning model for accurately predicting mechanical responses of heterogeneous materials. As a representative and practically relevant example, we specifically focus on fiber-reinforced composites (e.g., carbon fiber, glass fiber), which are widely used in applications ranging from high-performance sport cars and aircrafts, to industrial structures like cooling towers. These materials play a crucial role in both everyday products and advanced engineering systems due to their exceptional strength-to-weight ratio and customizable properties.

While fiber-reinforced composites serve as the initial test case, our methodology is designed to be broadly applicable. In Section 5.4, we demonstrate this generality by applying the same framework, via transfer learning, to microstructures with markedly different geometries and mechanical behaviors—including spinodal decomposition patterns and square inclusions with sharp corners and singularities. These cases present significant departures from the original training distribution, both in morphology and in local field behavior. Nonetheless, the model achieves high predictive accuracy with minimal fine-tuning, providing strong quantitative evidence of its robustness and versatility across diverse classes of heterogeneous materials.

We are interested in learning the solution operator of the parameterized Lippmann-Schwinger equation Eq. (2.12), which maps various parameters, including material properties and microstructural configurations, to the mechanical responses of the material under any external loading conditions. These parameters are represented by a fourth order elasticity tensor $\mathbb{C} \in L_{\text{per}}^{\infty}(\Omega, \mathbb{R}^{3 \times 3 \times 3})$, while the mechanical responses are characterized by a strain concentration tensor $\mathbb{A} \in L_{\text{per}}^2(\Omega, \mathbb{R}^{3 \times 3})$. To this end, the operator mapping can be defined as:

$$\Phi : \mathbb{C} \in L_{\text{per}}^{\infty}(\Omega, \mathbb{R}^{3 \times 3 \times 3}) \longrightarrow \mathbb{A} \in L_{\text{per}}^2(\Omega, \mathbb{R}^{3 \times 3}) \quad (3.1)$$

Here Ω represents the computational domain of RVE, and are a set of material properties including \square_{per} is used to describe periodic functions. The rationale for selecting these input and output feature representations will be given as follows.

Output feature representation. The mechanical responses of composite materials are typically characterized by the local strain field $\varepsilon(\mathbf{x})$, which effectively captures the strain distribution pattern inside a RVE. However, selecting the strain field as a model target is not ideal due to its explicit dependence on the macroscopic strain boundary condition $\bar{\varepsilon}$. While we consider periodic boundary conditions, the magnitude and orientation of $\bar{\varepsilon}$ can still vary arbitrarily, resulting in an unbounded output space for $\varepsilon(\mathbf{x})$. This variability makes it impractical to generate sufficient data to cover the full distribution of possible microstrain responses.

To address this difficulty, we choose the point-wise strain concentration tensor $\mathbb{A}(\mathbf{x})$ as the *output* of our model. Based on micromechanics theory (see Hill [47], Michel and Suquet [48], Yvonnet [49], Zohdi and Wriggers [50]), $\mathbb{A}(\mathbf{x})$ establishes a linear transformation from the macrostrain boundary condition $\bar{\varepsilon}$ to the local microstrain $\varepsilon(\mathbf{x})$ as

$$\varepsilon(\mathbf{x}) = \mathbb{A}(\mathbf{x}) : \bar{\varepsilon} = \begin{bmatrix} A_{11}(\mathbf{x}) & A_{12}(\mathbf{x}) & A_{13}(\mathbf{x}) \\ A_{21}(\mathbf{x}) & A_{22}(\mathbf{x}) & A_{23}(\mathbf{x}) \\ A_{31}(\mathbf{x}) & A_{32}(\mathbf{x}) & A_{33}(\mathbf{x}) \end{bmatrix} : \begin{bmatrix} \bar{\varepsilon}_{11} \\ \bar{\varepsilon}_{22} \\ \bar{\varepsilon}_{12} \end{bmatrix}. \quad (3.2)$$

Here, the point-wise strain concentration tensor $\mathbb{A}(\mathbf{x})$ is a 3×3 matrix independent of loading conditions for a RVE exhibiting linear elastic behavior. It can be obtained by solving the Lippmann-Schwinger equation subject to three orthogonal unit macroscopic strains $\bar{\varepsilon}_{ij}$ in each loading case

$$\bar{\varepsilon} = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \quad (3.3)$$

Importantly, once the point-wise strain concentration tensor $\mathbb{A}(\mathbf{x})$ is determined, the local microstrain $\varepsilon(\mathbf{x})$ can be directly computed by using the aforementioned formula for any given macrostrain boundary conditions $\bar{\varepsilon}$.

Input feature representation.

When the RVE exhibits linear elastic behavior, the strain concentration tensor $\mathbb{A}(\mathbf{x})$ induces a unique mapping from macrostrain $\bar{\varepsilon}$ to microstrain $\varepsilon(\mathbf{x})$, independent of the loading conditions. Therefore, the selection of $\mathbb{A}(\mathbf{x})$ not only reduces the effort in generating the output datasets, but also eliminates the requirements for taking loading conditions as *inputs* of our model. Accordingly, we choose the fourth-order elastic tensor $\mathbb{C}(\mathbf{x})$ as the input, which is the coefficient of Eq. (2.12) and is determined by the material properties and microstructural configuration of the RVE.

As illustrated in Section 2, the domain Ω is discretized with pixels, and each pixel \mathbf{x} is given by a set of material properties, including Young's modulus $E(\mathbf{x})$ and Poisson ratio $\nu(\mathbf{x})$. As the RVE is considered as a fiber reinforced composite (FRP), the domain Ω can be decomposed into disjoint fiber and matrix domains, denoted by Ω_f and Ω_m , respectively, such that $\Omega_m \cap \Omega_f = \emptyset$ and $\bar{\Omega}_m \cup \bar{\Omega}_f = \Omega$. We then introduce a characteristic function $\chi(\mathbf{x})$ to denote the underlying microstructural configuration of the RVE as

$$\chi(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \Omega_f, \\ 0, & \text{if } \mathbf{x} \in \Omega_m. \end{cases} \quad (3.4)$$

The function $\chi(\mathbf{x})$ determines the microstructural configuration of the RVE and is generated by a random fiber generation algorithm (see Section 3.1). Then, the point-wise Young's modulus and Poisson ratio can be expressed as

$$\begin{cases} E(\mathbf{x}) = \chi(\mathbf{x})E_f + (1 - \chi(\mathbf{x}))E_m, \\ \nu(\mathbf{x}) = \chi(\mathbf{x})\nu_f + (1 - \chi(\mathbf{x}))\nu_m. \end{cases} \quad (3.5)$$

Table 1

Summary of the CSMBench/CMME dataset parameters used in the numerical simulation of fiber-reinforced composite materials.

Dataset Size	Value
# Training samples	40,000
# Test samples	2000
RVE Characteristics	
RVE size (μm)	50×50
Discretization resolution	512×512
Fiber radius R_d (μm)	3.5
Fiber radius standard deviation Std	1 %
Fiber volume fraction Vof	40-60 %
Material Properties	
Fiber Young's modulus E_f (GPa)	5-85
Fiber Poisson's ratio ν_f	0.05-0.45
Matrix Young's modulus E_m (GPa)	2.5-5
Matrix Poisson's ratio ν_m	0.3-0.4

Here, $E_{\square}, \nu_{\square} \in \mathbb{R}$ are a set of material properties including Young's modulus and Poisson ratio for domain Ω_{\square} , where either the fiber domain Ω_f or the matrix domain Ω_m can be considered. E_{\square} and ν_{\square} are unique for a given RVE, and the selections of these values will be discussed in Section 3.1. In addition, given $E(\mathbf{x})$ and $\nu(\mathbf{x})$, we can compute the fourth order elastic tensor $\mathbb{C}(\mathbf{x})$ using the relations in Eqs. (2.2) and (2.3).

3.1. Data generation

Our data generation process is part of a broader effort coined as CSMBench (Computational Solid Mechanics Benchmark), which we envision as a comprehensive collection of datasets at the intersection of machine learning and computational solid mechanics. CSMBench aims to accelerate research and foster innovation by providing diverse, high-quality datasets for training and evaluating machine learning models in this rapidly evolving field.

Within CSMBench, here we focus on developing CMME (Computational Micromechanics for Elasticity) dataset, specifically designed to address linear elasticity problems in micromechanics. The CMME data generation process comprises two main components: input data generation and output data generation. Each of these components plays a crucial role in creating a robust and representative dataset for our machine learning models. The main metadata of the CMME training and test datasets are summarized in Table 1.

The CSMBench/CMME dataset consists of 40,000 training samples and 2000 test samples, each representing an input-output pair. The input consists of a periodic RVE domain with associated material parameters $E_f, \nu_f, E_m,$ and ν_m , while the output represents corresponding solutions of the Lippmann-Schwinger equation subjected to macrostrain boundary conditions stated in Eq. (3.3).

For the process of RVE generation, we employ the RAND_uSTRU_GEN algorithm proposed by Melro et al. [51], because of its ability to generate RVEs with high fiber volume fractions up to 65%. We implement this algorithm using the open-source project *rvesimulator* [52]. To ensure a diverse and representative dataset, we stratify both the training and test samples into 20 groups, with fiber volume fractions uniformly distributed between 40% and 60%. The radius of fiber is set as $3.5\mu\text{m}$ with 1% standard deviation. Each RVE domain size is set to be $50\mu\text{m} \times 50\mu\text{m}$, discretized at a high resolution of 512×512 pixels. For all training and test samples, the material properties $E_f, \nu_f, E_m,$ and ν_m are generated via Latin hypercube sampling method subject to upper and lower bounds provided in Table 1. Consequently, each microstructure has its own geometric configuration and is paired with a distinct material property set, resulting in a highly diverse and representative dataset for training and evaluation.

To compute the corresponding outputs, we solve Eq. (2.11) using a Fast Fourier Transform (FFT) based homogenization method [6]. This FFT-based homogenization is implemented in MATLAB[®] R2024b and tested on a Dell[®] Precision[™] T7920 Workstation, equipped with dual Intel[®] Xeon[®] Platinum 8268 Processor (2.9 GHz, 48-cores, Ubuntu 24.04 System). The total data generation time for this process is 36 hours for microstructure generation and an additional 39.12 hours for the FFT calculations. The details of the algorithm and the full pipeline of data generation are presented in Appendix C.1.

4. Methods

4.1. Operator learning

In this section, we present the operator learning framework [20,53] used to model the mechanical responses of heterogeneous materials, particularly the mapping from local material properties to mechanical responses under external loading. This framework provides the theoretical foundation for our proposed model.

As described in Section 3, our goal is to learn the solution operator Φ associated with the Lippmann-Schwinger equation, which maps a spatially varying elasticity tensor $\mathbb{C}(\mathbf{x})$ to the resulting strain concentration tensor $\mathbb{A}(\mathbf{x})$:

$$\Phi : \mathbb{C}(\mathbf{x}) \mapsto \mathbb{A}(\mathbf{x}). \tag{4.1}$$

Here, $\mathbb{C}(\mathbf{x})$ encodes the heterogeneous microstructure (via material properties such as Young’s modulus and Poisson’s ratio), while $\mathbb{A}(\mathbf{x})$ captures the local response of the material to arbitrary macroscopic loading through Eq. (3.2).

Specifically, we consider a general nonlinear operator $\Phi : \mathcal{X} \rightarrow \mathcal{Y}$ mapping between separable Banach spaces \mathcal{X} and \mathcal{Y} . In the context of our mechanical problem, the input space \mathcal{X} corresponds to the space of possible elasticity tensor fields, i.e., $\mathcal{X} = L^\infty_{\text{per}}(\Omega, \mathbb{R}^{3 \times 3}_{\text{sym}})$, and the output space \mathcal{Y} corresponds to the space of possible strain concentration tensor fields, i.e., $\mathcal{Y} = L^2_{\text{per}}(\Omega, \mathbb{R}^{3 \times 3})$. Our objective is to learn this operator Φ from a set of samples:

$$\{(\mathbb{C}_n(\mathbf{x}), \mathbb{A}_n(\mathbf{x})) : \mathbb{A}_n = \Phi(\mathbb{C}_n)\}_{n=1}^N. \tag{4.2}$$

In practical implementations, since neural networks operate on finite-dimensional spaces, the foundation of many neural operators lies in the extraction of latent finite-dimensional structures, as illustrated below.

Here, we introduce two encoder/decoder pairs on \mathcal{X} and \mathcal{Y} . We denote $\mathcal{E}_\mathcal{X}$ as the encoder mapping from the Banach space \mathcal{X} to a Euclidean space $\mathbb{R}^{d_\mathcal{X}}$, where $d_\mathcal{X}$ denotes the encoding dimension. The corresponding decoder for \mathcal{X} is defined as $D_\mathcal{X} : \mathbb{R}^{d_\mathcal{X}} \rightarrow \mathcal{X}$. Similarly, for the output space \mathcal{Y} , we define $\mathcal{E}_\mathcal{Y} : \mathcal{Y} \rightarrow \mathbb{R}^{d_\mathcal{Y}}$ as the encoder mapping from \mathcal{Y} to a Euclidean space $\mathbb{R}^{d_\mathcal{Y}}$, with $d_\mathcal{Y}$ representing the encoding dimension for \mathcal{Y} . The corresponding decoder is defined as $D_\mathcal{Y} : \mathbb{R}^{d_\mathcal{Y}} \rightarrow \mathcal{Y}$.

These encoder-decoder pairs approximately satisfy

$$\mathcal{E}_\mathcal{X} \circ D_\mathcal{X} \approx I_\mathcal{X}, \quad \mathcal{E}_\mathcal{Y} \circ D_\mathcal{Y} \approx I_\mathcal{Y}, \tag{4.3}$$

where $I_\mathcal{X}$ and $I_\mathcal{Y}$ are the identity maps on \mathcal{X} and \mathcal{Y} , respectively.

We aim to determine an approximation to $\Phi : \mathcal{X} \rightarrow \mathcal{Y}$ by a family of parameterized functions

$$\Psi : \mathbb{R}^{d_\mathcal{X}} \times \Theta \mapsto \mathbb{R}^{d_\mathcal{Y}}, \tag{4.4}$$

where $\Theta \subseteq \mathbb{R}^p$ denotes the parameter space from which we seek the optimal choice of parameter, denoted θ^* . The approximation of Φ is achieved through the composition

$$D_\mathcal{Y} \circ \Psi \circ \mathcal{E}_\mathcal{X} \approx \Phi. \tag{4.5}$$

Given our dataset $\{(\mathbb{C}_n, \mathbb{A}_n)\}_{n=1}^N$ where each pair represents a material configuration and its corresponding mechanical response, we optimize the parameters θ by solving:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N \left\| \Psi \circ \mathcal{E}_\mathcal{X}(\mathbb{C}_i) - \mathcal{E}_\mathcal{Y}(\mathbb{A}_i) \right\|_2^2 \tag{4.6}$$

This formulation provides a general framework for learning complex nonlinear operators between Banach spaces, with applications spanning various domains in scientific computing and machine learning.

4.2. The micromechanics transformer (micrometer)

Now we describe the network architecture of our proposed model, the Micromechanics Transformer (Micrometer). As illustrated in Fig. 1, this architecture consists of an encoder-decoder structure. The details of both the encoder and decoder components are described below.

Encoder. As discussed in Section 3, we are interested in learning the solution operator of Lippmann-Schwinger equation where its input fourth order elasticity tensor \mathbb{C} is parameterized by a microstructural configuration χ and material properties, specifically the Young’s modulus E_f, E_m and Poisson’s ratios ν_f, ν_m for the fiber and matrix phases, respectively. Consequently, these parameters collectively serve as the inputs to the encoder component of our model.

Let $\mathbf{a} \in \mathbb{R}^{H \times W \times 1}$ represent a discretization of a microstructure of interest, where 0 denotes the matrix and 1 denotes fiber. We remark that here H, W denotes the resolution of the pixel grid, which is consistent with Eq. (B.3). We begin by embedding the material properties, namely the Young’s modulus and Poisson ratio of the matrix and fiber, into our inputs. These scalar properties are repeated for each pixel and concatenated with the original input, resulting in a tensor of shape $\mathbb{R}^{H \times W \times 5}$. Unlike conventional Vision Transformers (ViT), we first employ resolution-invariant Fourier Neural Operator (FNO) [54] layers (see Appendix E) to encode the inputs into a latent space:

$$\mathbf{a}_f = \text{FNO}(\mathbf{a}),$$

with $\mathbf{a}_f \in \mathbb{R}^{H \times W \times D}$, where D is the embedding dimension. This choice preserves the model’s ability to handle varying input resolutions.

Next, we patchify our latent inputs \mathbf{a}_f into a sequence of tokens $\mathbf{a}_p \in \mathbb{R}^{\left(\frac{H}{P} \times \frac{W}{P}\right) \times D}$ using the same patch embedding process as in standard ViTs [55], where P denotes the patch size. To provide the model with information about the spatial arrangement of the patches, we add trainable spatial positional embeddings to each token as

$$\mathbf{a}_{pe} = \mathbf{a}_p + \text{PE}, \quad \text{PE} \in \mathbb{R}^{\left(\frac{H}{P} \times \frac{W}{P}\right) \times D}.$$

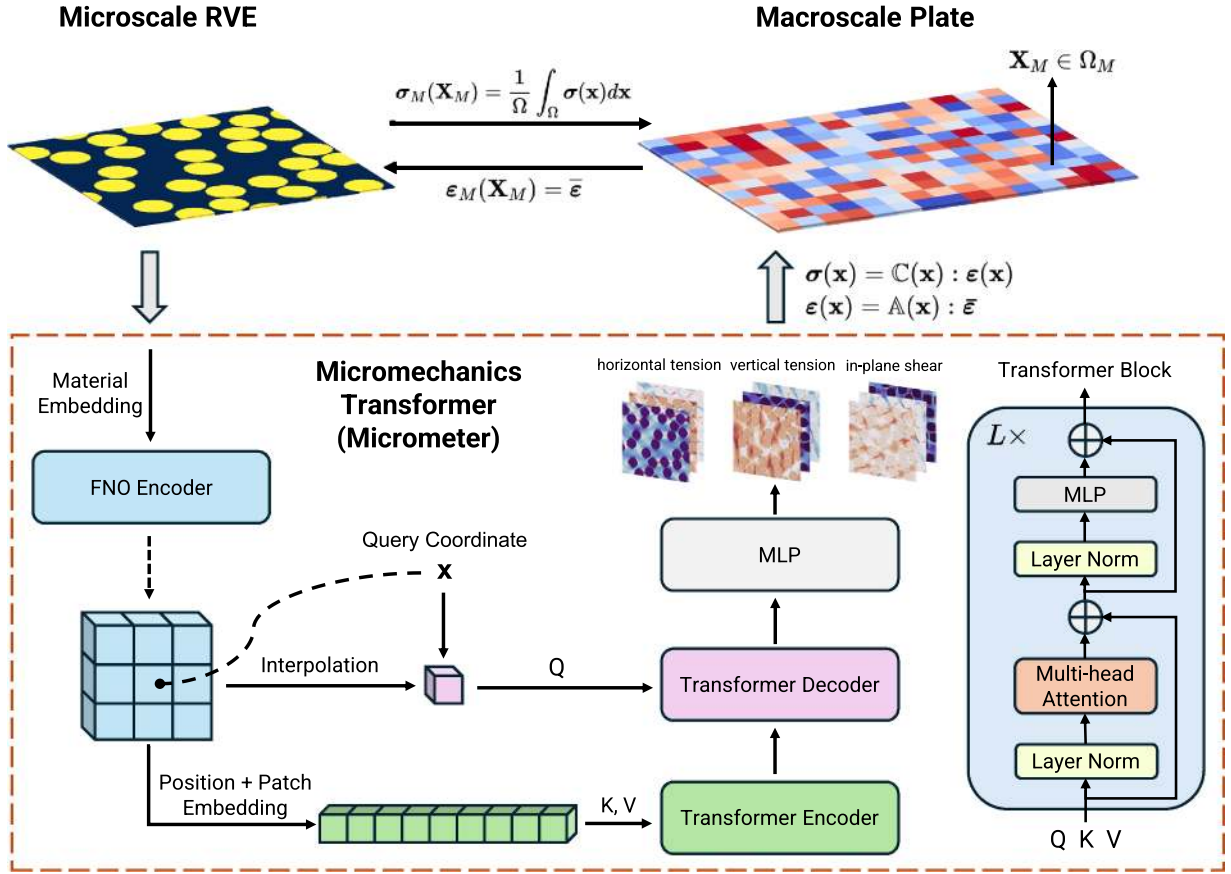


Fig. 1. Pipeline of Micrometer: Micrometer is a transformer-based deep learning model used to predict the mechanical responses of fiber-reinforced composite materials. It takes representative volume elements (RVEs) and material properties at the microscale (e.g., Poisson’s ratio, Young’s modulus) as inputs, and outputs the corresponding point-wise strain concentration tensor governed by Lippmann-Schwinger equation. Micrometer employs an encoder-decoder architecture. The input is first embedded using a resolution-invariant Fourier neural operator encoder. The resulting latent outputs are then patchified into a sequence of tokens, added with positional embeddings, and processed through a standard transformer backbone. A key feature of Micrometer is its ability to continuously evaluate the outputs at any query coordinate. This is achieved by interpolating the FNO encoder outputs using Nadaraya-Watson kernel interpolation to obtain latent query-specific features. These features are then decoded by a standard transformer decoder with cross-attention between the query features and the outputs of the transformer encoder. Here, the queries (Q) represent the interpolated features at the query points, the keys (K) and values (V) are derived from the encoder outputs. Finally, a multilayer perceptron (MLP) is used to decode the output into the physical space. The pre-trained Micrometer model can be easily integrated with computational micromechanics frameworks, facilitating fast and accurate homogenization and multiscale modeling of composite materials. Furthermore, Micrometer can be further fine-tuned for a variety of downstream tasks with limited data.

We then process the tokens \mathbf{a}_{pe} using a sequence of L pre-norm Transformer blocks [56,57],

$$\begin{aligned}
 \mathbf{z}_0 &= \text{LN}(\mathbf{a}_{pe}), \\
 \mathbf{z}'_{\ell} &= \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, & \ell = 1 \dots L, \\
 \mathbf{z}_{\ell} &= \text{MLP}(\text{LN}(\mathbf{z}'_{\ell})) + \mathbf{z}'_{\ell}, & \ell = 1 \dots L.
 \end{aligned}$$

Decoder. The decoder module draws inspiration from the Continuous Vision Transformer (CViT) [58] that enables the continuous evaluation of the model’s outputs at any arbitrary query coordinates. To this end, we start by creating a uniform grid $\{\mathbf{x}_{ij}\} \subset [0, 1]^2$, for $i = 1, \dots, N_x$ and $j = 1, \dots, N_y$, where the hyperparameters N_x, N_y are typically set to match the resolution of the target output functions [59–61]. For a single query point $\mathbf{x} \in \mathbb{R}^2$, we then perform a Nadaraya-Watson interpolation [62,63] over the outputs of the FNO encoder as

$$\mathbf{x}' = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} w_{ij} \mathbf{a}_{ij}^f, \quad w_{ij} = \frac{\exp(-\beta \|\mathbf{x} - \mathbf{x}_{ij}\|^2)}{\sum_{ij} \exp(-\beta \|\mathbf{x} - \mathbf{x}_{ij}\|^2)}, \tag{4.7}$$

where $\beta > 0$ is a hyperparameter that determines the locality of the interpolated features. It is important for determining the smoothness of the interpolation function. Specifically, larger values of β yield more localized weight distributions w_{ij} , resulting in a higher-frequency

Table 2
Different Micrometer configurations considered in this work.

Model	FNO width/modes	Encoder layers	Embedding dim	MLP width	Heads	# Params
Micrometer-S	32/32	4	256	256	8	20 M
Micrometer-B	64/32	6	512	512	16	70 M
Micrometer-L	64/64	8	512	1024	16	292 M

interpolant that captures finer-scale variations. Conversely, smaller β values produce a smoother interpolant by averaging over a broader neighborhood of points.

We then treat the interpolated latent feature $\mathbf{x}_0 = \mathbf{x}' \in \mathbb{R}^{1 \times D}$ as queries and the encoder output \mathbf{z}_L as keys and values, applying K cross-attention Transformer blocks as

$$\begin{aligned} \mathbf{x}'_k &= \mathbf{x}_{k-1} + \text{MHA}(\text{LN}(\mathbf{x}_{k-1}), \text{LN}(\mathbf{z}_L), \text{LN}(\mathbf{z}_L)), & k = 1 \dots K, \\ \mathbf{x}_k &= \mathbf{x}'_k + \text{MLP}(\text{LN}(\mathbf{x}'_k)), & k = 1 \dots K. \end{aligned}$$

The final output of our Micrometer model represents the strain concentration tensor $\mathbb{A} \in \mathbb{R}^{3 \times 3}$, flattened into a 1D vector in \mathbb{R}^9 . To obtain this, we project the high-dimensional latent outputs of the model onto the physical space of the desired dimension through a small MLP.

It is worth noting that the choice of FNO encoder and continuous cross-attention decoder is both theoretically and empirically justified. We empirically find that incorporating an FNO encoder not only enhances training convergence but also better preserves the information of fiber-matrix interfaces. Moreover, the conditioning mechanism on continuous query coordinates is motivated by Wang et al. [58]. By employing grid-based interpolation, it effectively controls the decoder's Lipschitz constant, which helps mitigating spectral bias [64] by adjusting the interpolation hyperparameter β . Consequently, Micrometer achieves state-of-the-art accuracy in predicting stress concentration fields, outperforming several popular PDE surrogate models, as shown in Table 3. More importantly, combined with our problem formulation in Section 3, Micrometer can be directly applied to multi-scale modeling problems without encountering generalization issues in loading conditions—an aspect that poses challenges in other related works [38–41,65].

5. Results

In this section, we demonstrate the effectiveness of the proposed Micrometer model. Our evaluation consists of three main components: (a) a comparative analysis of Micrometer's performance against established neural operators and PDE surrogates; (b) an application of Micrometer to two fundamental problems in micromechanics – computational homogenization and concurrent multiscale modeling; (c) an examination of Micrometer's transfer learning capabilities in out-of-distribution scenarios with scarce data.

5.1. Prediction of microscale strain fields

Here we validate Micrometer's accuracy in predicting microscale strain fields across a wide range of composite material microstructures and material properties. We also compare these results against other popular neural operator surrogates using the extensive CSMBech/CMME dataset described in Section 3.1.

Micrometer configurations. We consider models with various configurations, as summarized in Table 2. We mostly follow the recommended hyperparameter settings in Wang et al. [58], as these choices have been validated by extensive ablation studies. Specifically, unless otherwise stated, for all configurations, we employ an FNO encoder with four layers and a cross-attention decoder with 2 layers, and set the resolution of the grid to match the dataset resolution while using $\beta = 10^5$ to ensure sufficient locality of the interpolated features, thereby capturing sharp transitions in solutions.

Baselines. We select the following methods as our baselines for comparison.

- **Fourier Neural Operator (FNO)** [21]: An efficient framework for operator learning in the frequency domain.
- **U-Net** [66]: A popular network backbone widely used for image segmentation and generative modeling.
- **Vision Transformer (ViT)** [67]: A powerful transformer-based architecture adapted for image-related tasks.

For each model, we conduct comprehensive ablation studies to optimize its respective hyperparameters, adhering to the guidelines provided in the original publications. Detailed information on the implementation of baseline models is presented in Appendix F.

We employ a unified training recipe for all experiments. We employ the AdamW optimizer [68,69] with weight decay set to 10^{-5} . Our learning rate schedule includes an initial linear warm-up phase of 5,000 steps, starting from zero and gradually increasing to 10^{-3} , followed by an exponential decay at a rate of 0.9 for every 5,000 steps. To stabilize training process, we clip all gradients at a maximum norm of 1. If a loss blowup occurs, we restart training from the last saved checkpoint.

The loss function is a mean squared error (MSE) between the model predictions and the corresponding targets evaluated at randomly sampled query coordinates,

$$\text{MSE} = \frac{1}{BQ} \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^B \sum_{l=1}^Q \left| \hat{A}_{ij}^{(k)}(\mathbf{x}_l) - A_{ij}^{(k)}(\mathbf{x}_l) \right|_2^2, \quad (5.1)$$

Table 3

Micrometer vs. State-of-the-Art PDE Surrogate Models: Comparison of the Micrometer model (Small, Base, and Large variants) with established PDE surrogates (U-Net, Fourier Neural Operator (FNO), and Vision Transformer (ViT)) in predicting strain concentration tensors on the test dataset. Reported metrics include relative L^1 and L^2 errors, root-mean-square error (RMSE), training time (in hours), and evaluation time (in seconds). All baseline models are tuned via hyperparameter optimization, with their best results shown. All experiments are conducted on a single NVIDIA A6000 GPU with 48 GB of memory. Evaluation time is measured over 1000 test samples used to compute the reported metrics. Detailed experimental settings and additional results are provided in Tables G.8 and 2. The Micrometer-L variant achieves the highest overall accuracy, though its inference time is slightly slower than the other models.

Model	# Params	Rel. L^1 error (↓)	Rel. L^2 error (↓)	RMSE (↓)	Train time (h)	Eval time (s)
UNet	124 M	14.29 %	14.75 %	0.0752	20.5	96
FNO	268 M	4.79 %	7.19 %	0.0373	14.5	94
ViT	87 M	4.95 %	6.86 %	0.0346	15.2	94
Micrometer-S	20 M	5.32 %	7.24 %	0.0359	9.2	55
Micrometer-B	70 M	4.39 %	6.42 %	0.0337	17.0	99
Micrometer-L	292 M	3.61 %	5.95 %	0.0303	18.2	136

where $A_{ij}^{(k)}(\mathbf{x}_i)$ denotes the ij -th variable of the k -th sample in the training dataset, evaluated at a query coordinate \mathbf{x}_i , and \hat{A} denotes the corresponding model prediction. All models are trained for 10^5 iterations with a batch size $B = 16$. Within each batch, we randomly sample $Q = 4,096$ query coordinates from the grid and corresponding output labels.

Evaluation. After training, we evaluate model accuracy on the test dataset using several commonly used metrics: relative L^1 norm, relative L^2 norm, and root of mean square error (RMSE),

$$\text{Rel. } L^1 = \frac{1}{9N_{\text{test}}} \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^{N_{\text{test}}} \frac{\|\hat{A}_{ij}^{(k)} - A_{ij}^{(k)}\|_1}{\|A_{ij}^{(k)}\|_1}, \quad (5.2)$$

$$\text{Rel. } L^2 = \frac{1}{9N_{\text{test}}} \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^{N_{\text{test}}} \frac{\|\hat{A}_{ij}^{(k)} - A_{ij}^{(k)}\|_2}{\|A_{ij}^{(k)}\|_2}, \quad (5.3)$$

$$\text{RMSE} = \sqrt{\frac{1}{9N_{\text{test}}} \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^{N_{\text{test}}} \|\hat{A}_{ij}^{(k)} - A_{ij}^{(k)}\|_2^2}, \quad (5.4)$$

where the norm is computed over the prediction at all pixel grid points, averaged over each variable of interest.

Table 3 presents a comparison of Micrometer against several competitive baseline models, including Fourier Neural Operator (FNO) [21], U-Net [66], and Vision Transformer (ViT) [55]. Notably, our largest Micrometer model (Micrometer-L) achieves the lowest error across all metrics, with a relative L^1 error of 3.61 %, relative L^2 error of 5.95 %, and RMSE of 3.03 %. This performance is achieved with a parameter count (292M) comparable to or less than that of the baselines, highlighting the efficiency of our architecture. More detailed comparisons are provided in Appendix Table G.8.

To better understand the performance of Micrometer, we visualize the test relative L^2 error with respect to varying volume fractions and Young's modulus ratios E_f/E_m . One can see that the error increases as the volume fractions or Young's modulus ratios increase. This trend is potentially attributed to the increased complexity of the solution under these conditions, as higher volume fractions lead to greater strain concentration in the matrix, while higher ratios result in sharper discontinuities. These observations align with the visualizations of representative predictions for low to medium volume fractions and Young's modulus ratios, as illustrated in Fig. 4.

A key strength of Micrometer lies in its scalability. Our analysis reveals that model performance improves consistently with larger architectures, increased training data, and extended training times (Fig. 3). This scalability suggests that Micrometer can leverage additional computational resources to achieve even higher accuracy, following trends observed in other domains such as natural language processing and computer vision [70–73].

While our primary focus is on 2D problems, the same methodology naturally extends to 3D. As a proof of concept, we include a three-dimensional numerical experiment in Appendix H, which demonstrates the model's effectiveness in handling transversely isotropic 3D fiber reinforced composite microstructures.

5.2. Computational homogenization

Computational homogenization is a fundamental approach in solid mechanics for determining the effective properties of heterogeneous materials based on their microstructural characteristics [74]. We demonstrate that Micrometer not only achieves high accuracy in this task but also offers significant computational advantages over traditional methods.

Our goal is to determine the effective material properties including Young's modulus \bar{E} and Poisson ratio $\bar{\nu}$ for heterogeneous RVE by using the model prediction \mathbb{A} and the fourth order elastic tensor \mathbb{C} . Specifically, for a single RVE, we aim to obtain the homogenized

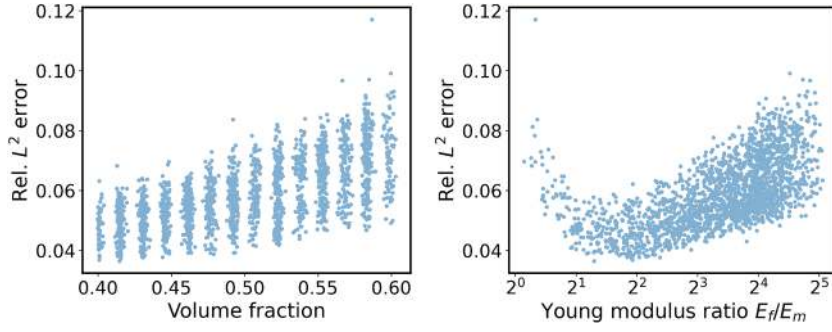


Fig. 2. Performance analysis of Micrometer across volume fractions and material properties: Left: Distribution of relative L^2 errors with respect to volume fraction across the test dataset. Right: Distribution of relative L^2 errors with respect to Young's modulus ratio between fiber and matrix, evaluated over all test samples.

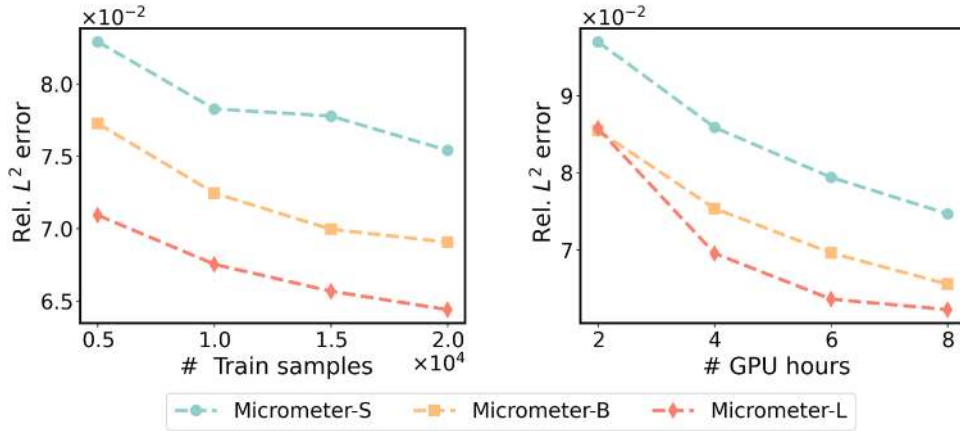


Fig. 3. Scaling Analysis of Micrometer: Left: Relative L^2 errors for different model configurations as a function of training dataset size, with all models trained to convergence. Right: Relative L^2 errors as a function of GPU hours, with fixed training data but varying training time and model capacity. Both plots demonstrate the scalability of Micrometer in the machine learning sense, showing improved accuracy with increased data, training time, and model size.

Table 4

Mechanical properties for four types of industrial composite materials. The unit of E_m and E_f is GPa.

Material type	Material property			
	E_f	ν_f	E_m	ν_m
Silenka E-Glass 1200tex / Bisphenol-A epoxy [75]	74.00	0.2000	3.76	0.39
AS4 [76] / 3501-6 epoxy [77]	15.00	0.0714	4.60	0.34
HTA / 6376 [78]	28.00	0.3300	3.63	0.34
T300 / TDE 86 epoxy [79]	40.00	0.3986	4.35	0.39

fourth order elastic tensor $\bar{\mathbb{C}}$ defined by

$$\bar{\mathbb{C}} = \frac{1}{T_1 T_2} \sum_{j=1}^{T_2} \sum_{i=1}^{T_1} \mathbb{C}(\mathbf{x}_{ij}) :: \mathbb{A}(\mathbf{x}_{ij}), \quad (5.5)$$

where $\bar{\mathbb{C}}$ is a symmetric 3×3 matrix in Voigt notation. Using the relation in Eqs. (2.2) and (2.3), homogenized Young's modulus \bar{E} and Poisson ration $\bar{\nu}$ can be expressed as:

$$\bar{\nu} = \frac{\bar{C}_{1111} - 2\bar{C}_{1212}}{2(\bar{C}_{1111} + \bar{C}_{1212})}, \quad \bar{E} = \bar{C}_{1212} \left(\frac{3\bar{C}_{1111} - 4\bar{C}_{1212}}{\bar{C}_{1111} + \bar{C}_{1212}} \right). \quad (5.6)$$

To demonstrate the effectiveness of Micrometer in computational homogenization for a wide range of materials, we select four representative industrial fiber reinforced composites, with their material properties detailed in Table 4. To evaluate the influence of the fiber volume fraction on effective properties, we set five groups of fiber volume fraction evenly distributed between 40 % and 60 %

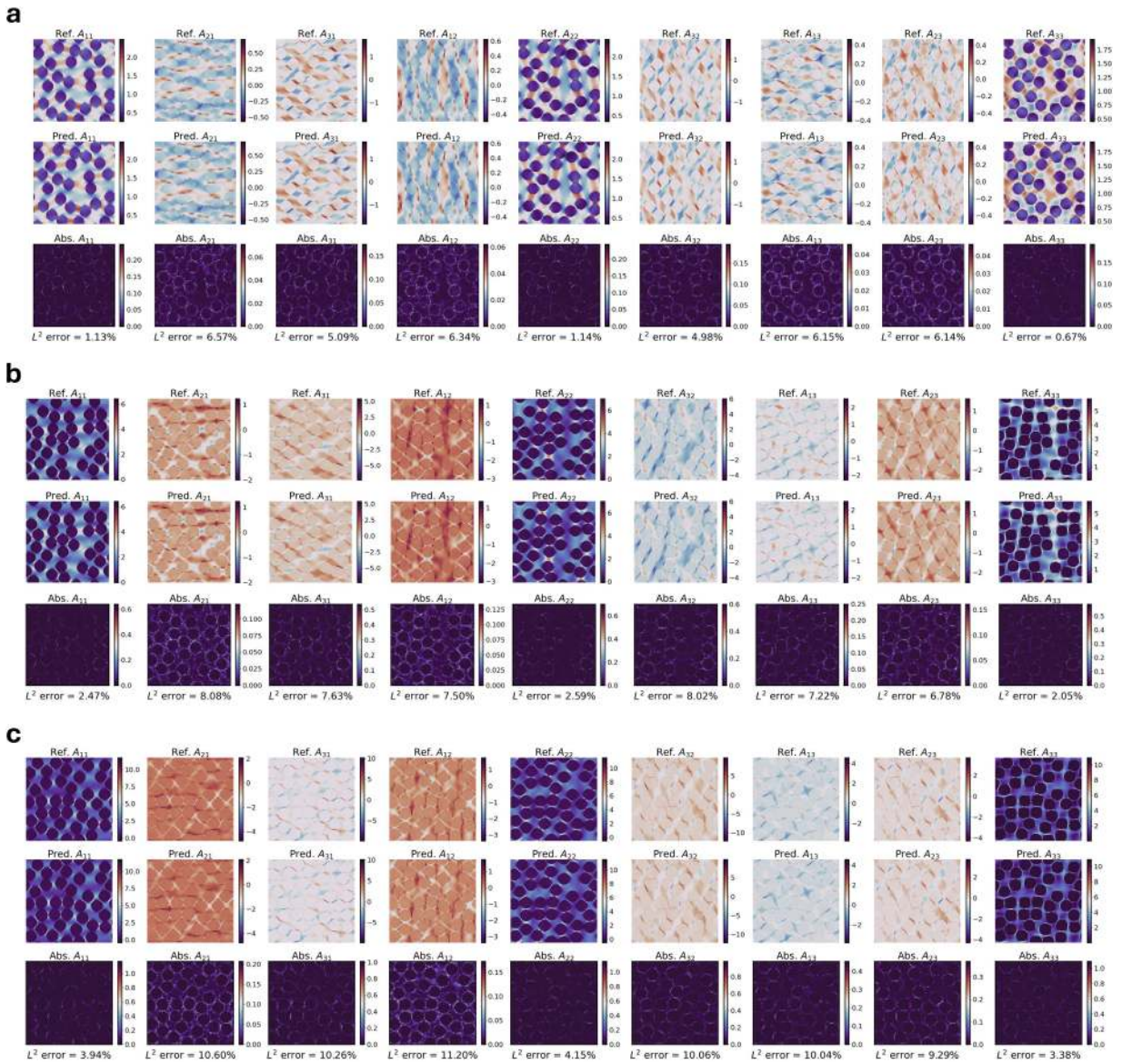


Fig. 4. Representative predictions by Micrometer for microstructures with varying volume fractions and Young's modulus ratios between fiber and matrix. (a) Low volume fraction and low Young's modulus ratio. (b) Medium volume fraction and medium Young's modulus ratio. (c) High volume fraction and high Young's modulus ratio. We observe that our model yields higher accuracy for leading terms (e.g., A_{11} , A_{22} , A_{33}) compared to non-leading terms. Model performance slightly decreases for higher modulus ratios and volume fractions, potentially due to sharper discontinuities and more concentrated fields near interfaces. A more detailed evaluation is presented in Fig. 2.

and each group contains 250 RVE samples with different microstructural configurations. The homogenized properties for each group are computed by averaging Young's modulus \bar{E} and the Poisson ratio $\bar{\nu}$ over these 250 samples.

Fig. 5(c) presents a comparison between Micrometer's predictions and reference values obtained from conventional FFT-based homogenization¹ [6] regarding the effective Young's modulus and Poisson's ratio. The results demonstrate excellent agreement across all material types and volume fractions, with Micrometer accurately capturing the relationships between volume fraction and effective property of RVE.

¹ For the sake of clarity, we distinguish the differences between "computational homogenization" and "FFT based homogenization" in the context of micromechanics. Broadly speaking, "computational homogenization" represents a numerical strategy to predict effective/average properties of heterogeneous materials, while "FFT based homogenization" represents a numerical method for solving the Lippmann-Schwinger equation to determine the local and overall responses of heterogeneous materials based on Fast Fourier Transformation.

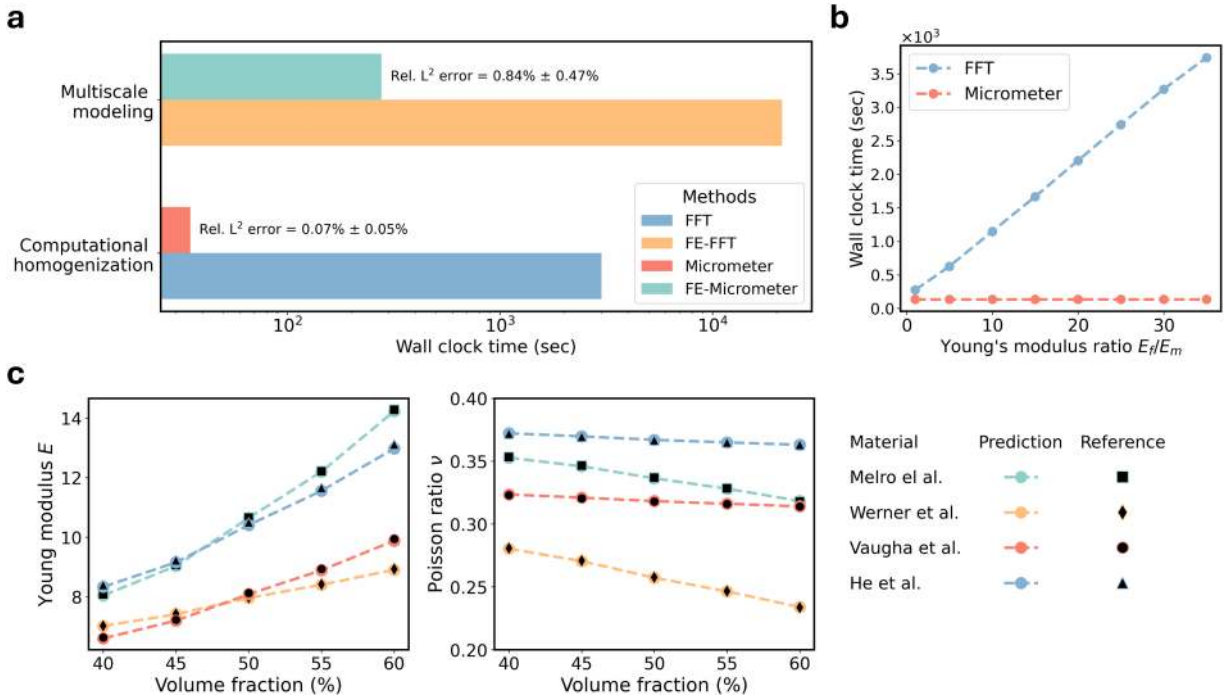


Fig. 5. Computational cost for FFT based homogenization vs. Micrometer: (a) Wall-clock time comparison of traditional methods (FFT, FE-FFT) and Micrometer-based approaches for homogenization and multiscale modeling. (b) Wall-clock time comparison between conventional FFT-based homogenization and Micrometer for predicting full-field strain concentration tensors across 3000 microstructures (volume fractions: 40 %-60 %). This comparison aims to provide a general sense of relative runtime performance using common computational resources. (c) Comparison of homogenized material properties obtained from conventional FFT based homogenization and Micrometer. Results are shown for four representative industrial composite materials with varying volume fractions in their representative volume elements (RVEs). The FFT based homogenization was implemented by MATLAB® R2024b and tested on a Dell® Precision™ T7920 Working Station (CPU: Dual Intel® Xeon® Platinum 8268 Processor 2.9GHz 48-cores, Ubuntu 24.04 system), while the Micrometer model was evaluated on NVIDIA® RTX A6000 GPUs.

Quantitatively, Micrometer achieves an average relative error of less than 0.1 % compared to the reference values (Fig. 5(a)). This level of accuracy is worth highlighting, considering the diverse range of material properties and microstructure configurations analyzed using a single surrogate model. Moreover, as shown in Fig. 5(a), Micrometer reduces the computational time by approximately two orders of magnitude compared to the conventional FFT based homogenization. This speedup is achieved while maintaining high accuracy, offering a powerful new tool for materials scientists and engineers.

The computational advantage of Micrometer becomes particularly evident when considering materials with high contrast in properties between constituents within the range of dataset. Fig. 5(b) illustrates how the computational time for FFT-based homogenization scales linearly with increasing Young’s modulus ratio between fiber and matrix, while Micrometer’s performance remains constant. This characteristic makes Micrometer particularly effective for simulating advanced composites and architected materials with high property variations.

5.3. Multiscale modeling

The ability to bridge microscale material behavior with macroscale structural response is a critical challenge in solid mechanics [17–19]. Here we demonstrate that Micrometer can be seamlessly integrated into multi-scale modeling frameworks, offering high accuracy and computational efficiency. To this end, we introduce FE-Micrometer, a novel concurrent multi-scale framework that combines finite element analysis (FEA) at the macroscale with Micrometer’s predictions at the microscale (see Algorithm 3 in Appendix). This approach enables evaluating the mechanical responses of complex heterogeneous structures while capturing detailed microstructural information, a task that has traditionally been computationally prohibitive.

To evaluate FE-Micrometer, we model a macroscale composite plate in 2D, comprising 40 × 75 RVEs in the horizontal and vertical directions, respectively, resulting in 3000 RVEs in total. These RVEs are divided into 20 groups with fiber volume fraction uniformly distributed ranging from 40 % to 60 % and randomly distributed in spatial coordinates within the plate. To consider the uncertainty of material properties across macroscale during the manufacture process, we introduce a spatially correlated Gaussian random field (GRF) for the Young’s modulus of both matrix and fiber in each RVE, by using the Karhunen-Loève (K-L) expansion:

$$E_{\square}(\mathbf{X}_M, \eta) = \hat{E}_{\square} + \sum_{k=1}^{N_{kl}} \sqrt{\zeta_k} \gamma_k(\mathbf{X}_M) \rho_k(\eta), \tag{5.7}$$

where \mathbf{X}_M denotes a macroscale material point corresponding to a single RVE, and \square represents either the fiber or matrix material domain. In particular, $E_{\square}(\mathbf{X}_M, \eta)$ and \hat{E}_{\square} represent Young's modulus at \mathbf{X}_M and its global average, respectively. N_{kl} represents the number of K-L expansion modes or truncated level. ζ_k and $\gamma_k(\mathbf{X}_M)$ denote the k -th largest eigenvalue and associated normalized eigenfunction of the covariance function, respectively. $\rho_k(\eta)$ denotes the independent standard uncorrelated random variables.

To model a high Young's modulus ratio between fiber and matrix (\hat{E}_f / \hat{E}_m), we set \hat{E}_f and \hat{E}_m to 74 GPa and 3.35 GPa, respectively, with standard deviations of 2.0 GPa and 0.1 GPa. The correlation length scale is set as 0.1 mm. For macroscale loading conditions, as depicted in Panel A of Fig. 6, we impose Dirichlet boundary conditions on the composite plate: a vertical displacement ($s^* = 0.1875\text{mm}$) is applied to the top edge and the bottom edge is fixed in both horizontal and vertical directions. All other edges remain stress-free. We consider a quasi-static loading scenario without inertial effects, with the total loading history s divided linearly into five steps.

Fig. 6 presents a comprehensive summary of our results. The FE-Micrometer predictions show excellent agreement with the reference FE-FFT (Finite element for macroscale calculation and FFT based homogenization for microscale calculation) method across all aspects of the simulation. Specifically, the predicted reaction forces closely match the reference data, with a relative error of just 1.03% (Fig. 6(e)). Both macroscale and microscale stress fields show good agreement with the reference solution, as shown in Fig. 6(f). As such, FE-Micrometer accurately captures the complex stress patterns within individual RVEs, demonstrating its ability to preserve microstructural details in macroscale simulations.

The accuracy of FE-Micrometer extends across a wide range of Young's modulus ratios between fiber and matrix, as demonstrated in additional numerical experiments (see Figs. G.12 and G.13 for details). A key aspect of FE-Micrometer is its computational efficiency. As shown in Fig. 5(a), FE-Micrometer reduces the computational time by nearly two orders of magnitude compared to the conventional FE-FFT approach. This speedup is achieved while maintaining a relative error of less than 1%, demonstrating an excellent balance between accuracy and efficiency.

5.4. Transfer learning and adaptability

The ability of a model to adapt to new scenarios with minimal additional training is crucial for its practical utility in materials science, where novel material systems are constantly emerging. We demonstrate that Micrometer possesses strong transfer learning capabilities, allowing it to generalize to new material configurations and even entirely different microstructural morphologies with limited additional data.

We evaluate Micrometer's adaptability through three challenging case studies – Case I: Fiber-reinforced composites with microstructural parameters outside the original training distribution; Case II: Cellular materials with spinodal topologies, representing a fundamentally different class of microstructures; and Case III: Microstructures with rectangular fiber, serving as a basic benchmark with geometric singularities at the corners of fiber. The sample microstructures of each case are presented in Fig. 7 for illustrative purposes. For each case, we generate small datasets of 1000 training and 100 test samples. In Cases I and II, material properties are sampled from ranges comparable to those used in the original dataset, whereas in Case III, the material properties are held fixed. The detailed problem setups for each case are described as follows.

Case I. This case validates our model's adaptability to fiber-reinforced composite RVEs with different microstructure parameter settings distinct from those used in training. Specifically, we set the fiber radius R_d to $5\mu\text{m}$ and the volume fraction range Vof to 62.5% - 65%. The discretization resolution, RVE size, standard deviation of fiber radius, and material properties remain the same as those in Table 1. Following Algorithm 2, we generate 1000 training and 100 test RVE samples with diverse microstructural configurations, along with corresponding input and output datasets governed by these new settings.

Case II. This case examines our model's adaptability to RVEs with microstructural morphologies fundamentally different from fiber-reinforced composites. We focus on RVEs featuring microscale stochastic architected materials, specifically cellular materials shaped in spinodal topologies. Such materials find wide application in both industrial and biomedical fields, including energy absorption [80], human bone implants [81], and impact-resilient structures [82], among others. The spinodal RVEs are generated by solving the time-dependent Cahn-Hilliard (CH) equation using a spectral FFT method, with mathematical and computational foundations described by Chen and Shen [83]. We set the RVE size to $50\mu\text{m} \times 50\mu\text{m}$, with a 256×256 discretization resolution. The RVE domain is divided into hard and soft phases based on the solution field of the Cahn-Hilliard equation at each pixel, with each phase occupying 50% of the volume. Pixels are labeled as 0 (soft phase) if their concentration exceeds a threshold value of 0.6, and 1 (hard phase) otherwise. We generate 1000 training and 100 test spinodal RVE samples with varying configurations. Material properties are assigned following Step 1 in Algorithm 2 (see Appendix C.1), with the matrix and fiber domains replaced by soft and hard phases, respectively. The calculation of the fourth-order elastic tensor and strain concentration tensor for all spinodal RVEs follows Steps 2 and 3 in Algorithm 2 (see Appendix C.1), respectively.

Case III. This case validates our model's adaptability to the RVE with corner singularities. We focus on rectangular-like fiber inside the microstructure, which is named as the "Obnosov problem" [84]. This problem has been studied using FFT based homogenization for convergence analysis and error estimation [85]. Regarding the geometrical configuration, the length and width of the interior fiber are set to range from 40% to 80% of the RVE length. We set the RVE size to $50\mu\text{m} \times 50\mu\text{m}$, with a 256×256 discretization resolution. The material properties of fiber and matrix are fixed, which are provided in Table 5. We generate 1000 training and 100 test RVE samples with varying size of interior fiber, where the size parameters are sampled using a two-dimensional Latin hypercube sampling (LHS) scheme.

Fig. 8 presents the performance of Micrometer in these transfer learning tasks, comparing fine-tuning of the pre-trained model against training from scratch. The results reveal several key insights. First, fine-tuning the pre-trained Micrometer model consistently yields significantly lower errors than training from scratch, especially when the amount of new data is limited. With just 100 training

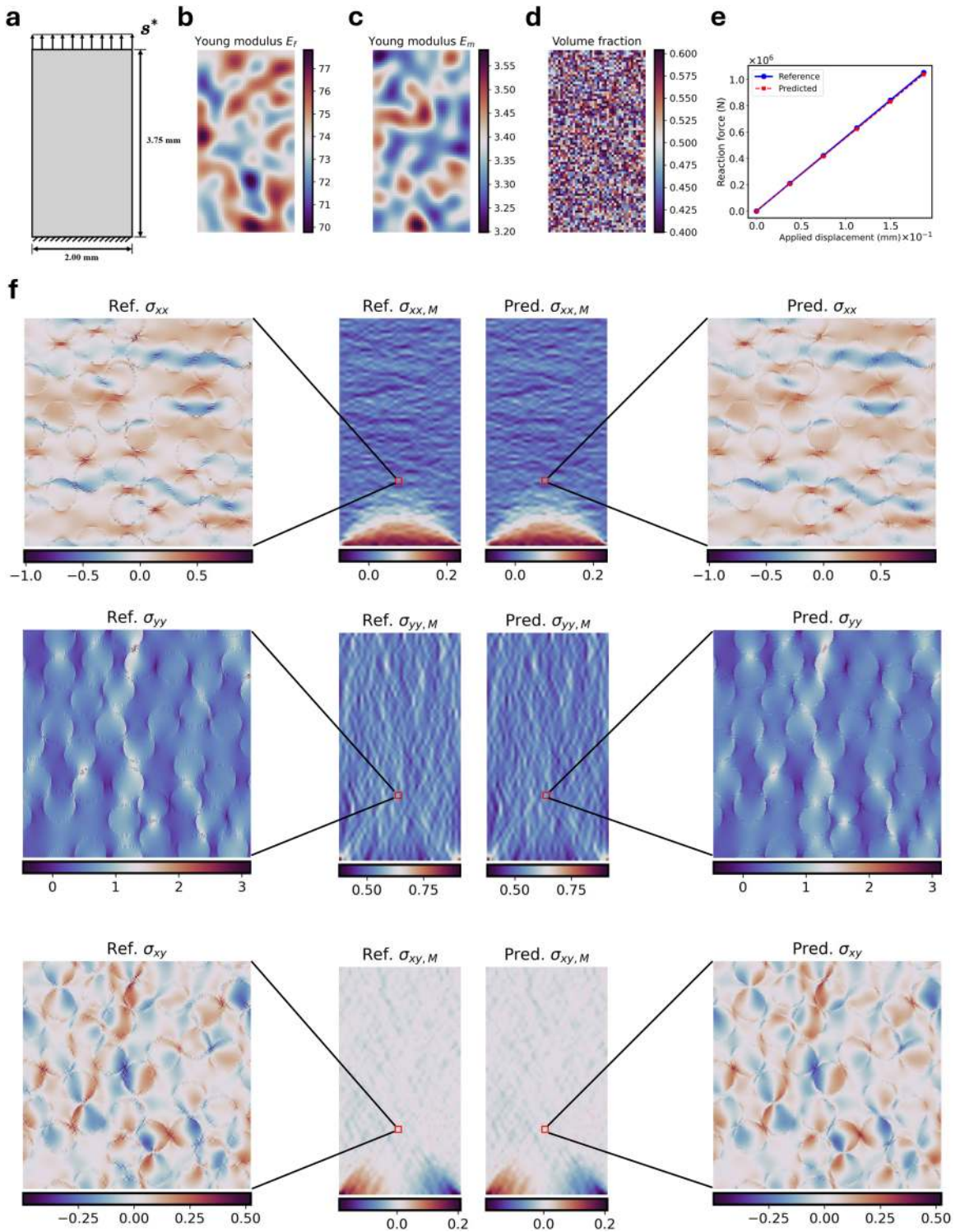


Fig. 6. Concurrent Multiscale Modeling: (a) 2D composite plate composed of 3000 RVEs arranged in a 40×75 grid. A vertical displacement ($s^* = 0.1875\text{mm}$) is applied to the top edge and the bottom edge is fixed in both horizontal and vertical directions. All other edges remain stress-free. (b)–(d) Distributions of material properties and volume fractions. Young’s modulus of fiber and matrix are sampled from Gaussian random fields. (e) Comparison of reaction forces predicted by FE-FFT and FE-Micrometer across different displacements. (f) Comparison of macroscale and microscale stress fields.

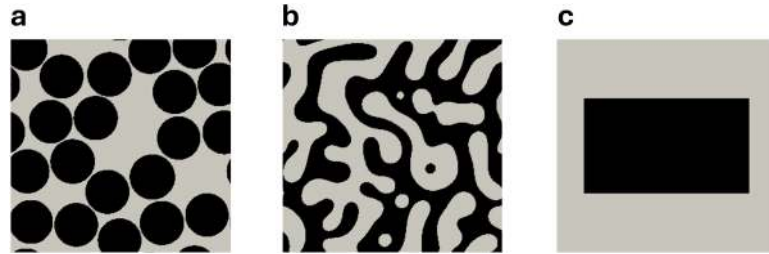


Fig. 7. Representative microstructures for the three transfer learning case studies. (a) Case I: Fiber-reinforced composite with microstructural parameters outside the original training distribution, (b) Case II: Cellular materials with spinodal topologies representing a fundamentally different class of microstructures, and (c) Case III: Microstructures with rectangular inclusions with geometric singularities at corners. Each case includes 1000 training and 100 test samples at spatial resolution 256×256 to evaluate Micrometer’s adaptability to new microstructural configurations.

Table 5
Summary of the dataset parameters for transfer learning tasks.

Dataset Size	Case I	Case II	Case III
# Training Samples		1,000	
# Test Samples		100	
RVE Characteristics			
Discretization resolution		256×256	
Fiber volume Fraction V_{of}	62.5 %	N/A	N/A
Fiber radius R_f (μm)	5	N/A	N/A
Fiber radius standard deviation Std	1 %	N/A	N/A
Material Properties			
Fiber Young’s modulus E_f (GPa)		5-85	15
Matrix Young’s modulus E_m (GPa)		2.5-5	3.76
Fiber Poisson’s ratio ν_f		0.05-0.45	0.2
Matrix Poisson’s ratio ν_m		0.3-0.4	0.39

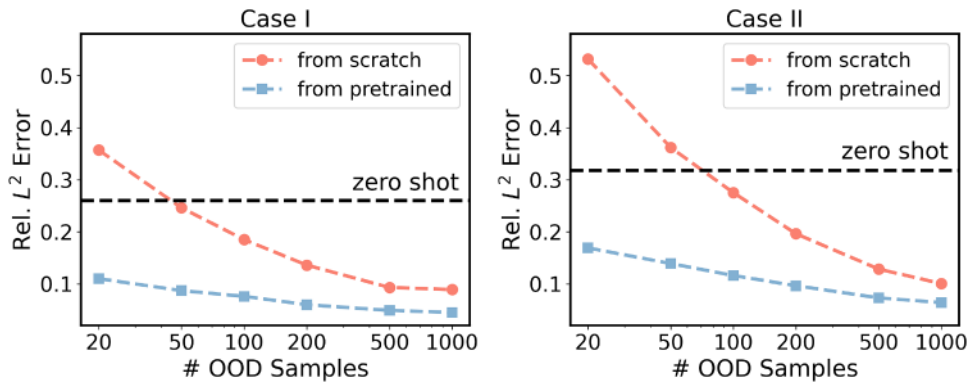


Fig. 8. Finetuning performance of Micrometer: Relative L^2 errors for models trained from scratch versus finetuned from pretrained weights, across varying numbers of training samples. The black dashed line represents zero-shot evaluation. Results highlight Micrometer’s efficient adaptation to new tasks, achieving high accuracy (below 10 % error) with fewer than 100 samples through finetuning.

samples, the fine-tuned model achieves relative L^2 errors below 10 % for both cases, demonstrating strong data efficiency. Even without any fine-tuning (zero-shot evaluation), Micrometer shows reasonable performance on Case I, indicating strong generalization to nearby distributions.

Finally, the successful adaptation to spinodal microstructures (Case II) and microstructures with corner singularities (Case III) highlights Micrometer’s ability to generalize across fundamentally different microstructural morphologies. For Case III in particular, the finetuned model achieves a relative L^2 error of 2.87 % on the corresponding test set, demonstrating strong predictive performance despite the geometric shift introduced by corner singularities. This result further confirms the model’s ability to handle challenging geometrical variations via transfer learning. To illustrate the quality of predictions after fine-tuning, we present representative examples for all three cases in Figs. 9–11, which show close agreement with the corresponding ground truth.

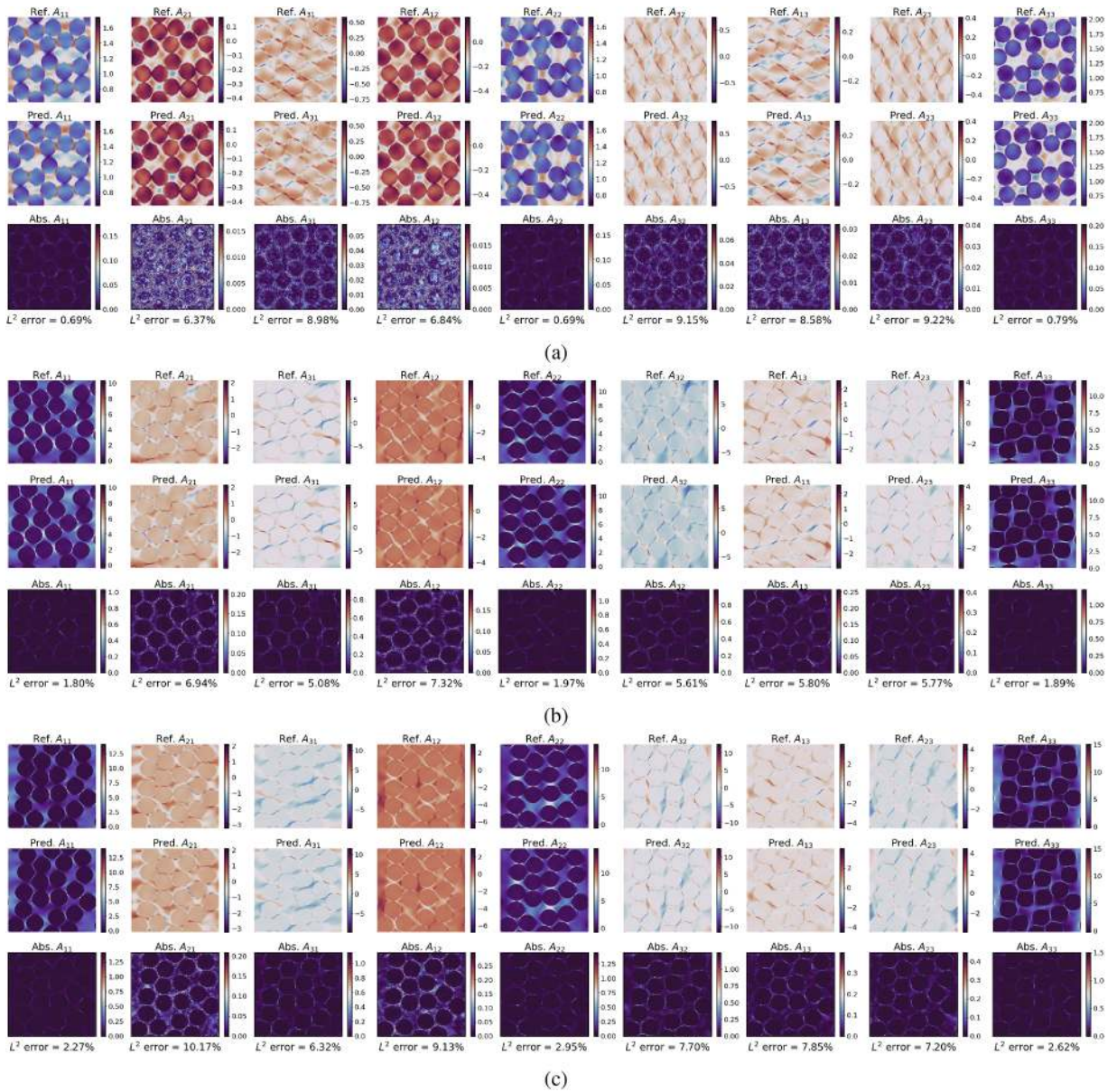


Fig. 9. Transfer Learning (Case I): Representative predictions by Micrometer for microstructures with varying volume fractions and Young's modulus ratios between fiber and matrix. (a) Low Young's modulus ratio. (b) medium Young's modulus ratio. (c) High Young's modulus ratio.

6. Discussion

We introduced Micrometer, a deep learning model for predicting the mechanical response of heterogeneous materials. This model achieves both high accuracy and computational efficiency. Our approach begins with the formulation of a novel operator learning problem, focusing on learning the solution operator of the parametric Lippmann-Schwinger equation. Specifically, we learn the mapping from the fourth-order elastic tensor to the strain concentration tensor. This formulation enables efficient evaluation of mechanical responses across a diverse range of microstructural configurations and material properties, under any external loading conditions.

To address the scarcity of high-quality data in this field, we have generated and released CSMBench/CMME, the first large-scale, comprehensive, high-fidelity, and high-resolution dataset specifically focused on linear elasticity. Leveraging this comprehensive dataset, we demonstrate that Micrometer outperforms several popular PDE surrogates, exhibiting strong scalability with respect to data size, model parameters, and computational resources. Moreover, we validated Micrometer's effectiveness through applications in computational homogenization and multiscale modeling, where it achieves accuracy comparable to traditional numerical methods

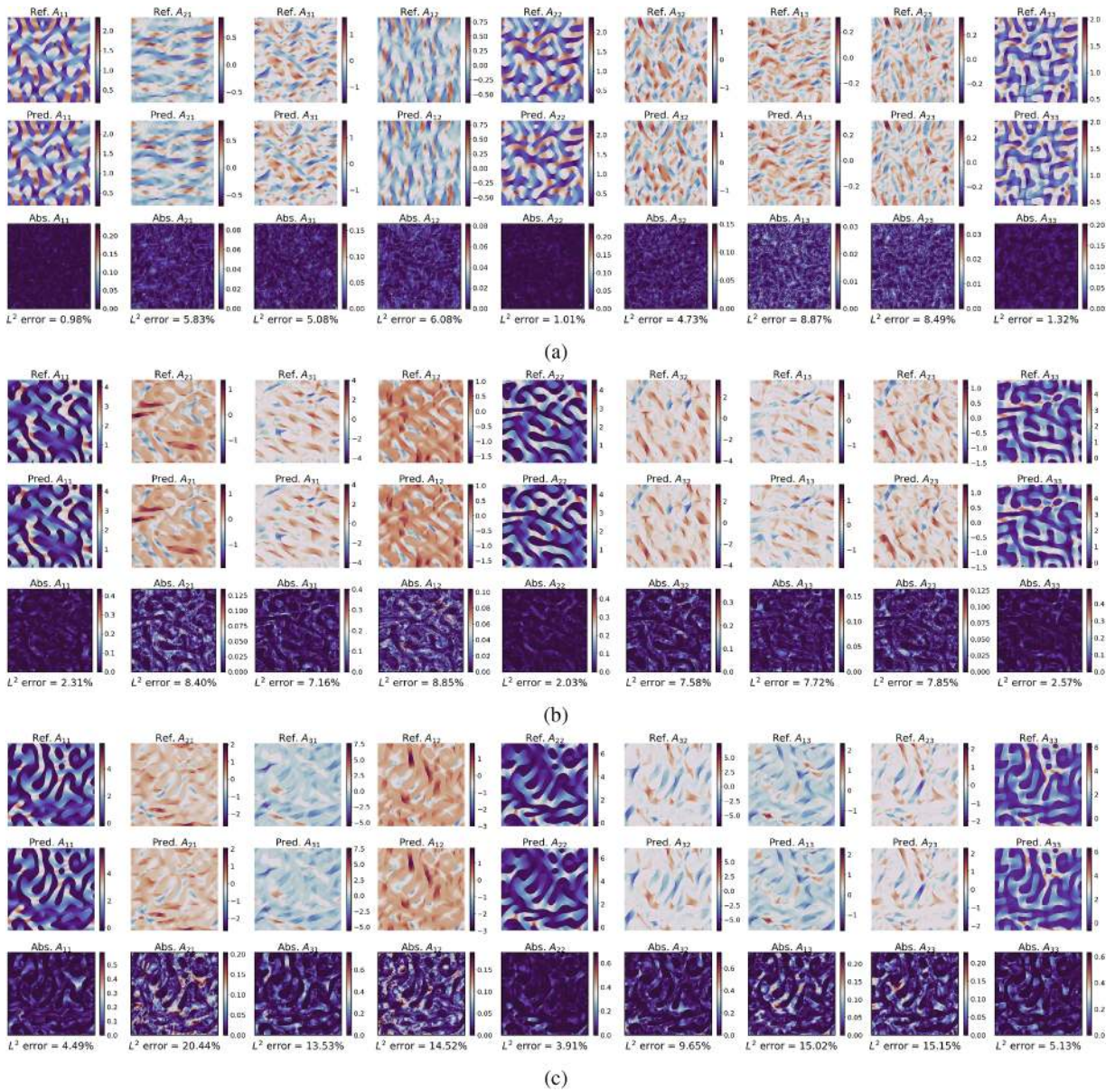


Fig. 10. Transfer Learning (Case II): Representative predictions by Micrometer for microstructures with varying volume fractions and Young's modulus ratios between fiber and matrix. (a) Low Young's modulus ratio. (b) Medium Young's modulus ratio. (c) High Young's modulus ratio.

while reducing computational time by up to two orders of magnitude. Finally, we have showcased Micrometer's adaptability through transfer learning experiments on new materials with limited data, highlighting its potential to tackle diverse scenarios in computational solid mechanics with minimal additional training.

While Micrometer represents a notable advancement in computational micromechanics, there remain several avenues for improvement and expansion. Our future work will focus on three key areas. First, we aim to extend the model to three-dimensional representations, which will enable more accurate modeling of complex microstructures and material behaviors, thus broadening its applicability to real-world materials and structural components. Second, we plan to move beyond linear elasticity by incorporating more sophisticated material behaviors such as hyperelasticity, plasticity, and fracture into the Micrometer framework. This expansion will significantly enhance the model's capability to simulate a wider range of mechanical responses under various conditions. Third, we will continue to develop and curate high-quality datasets that capture these complex phenomena, while simultaneously advancing the model architecture to handle non-linear and time-dependent mechanical responses. It is important to acknowledge that these more complex settings often come with substantially increased computational costs. However, by leveraging transfer learning with our accurate pre-trained model as a starting point, we can potentially accelerate future research and mitigate the computational burden associated with exploring these intricate material behaviors. This approach not only promises to extend the capabilities of Micrometer but also to make the exploration of complex material physics more computationally feasible.

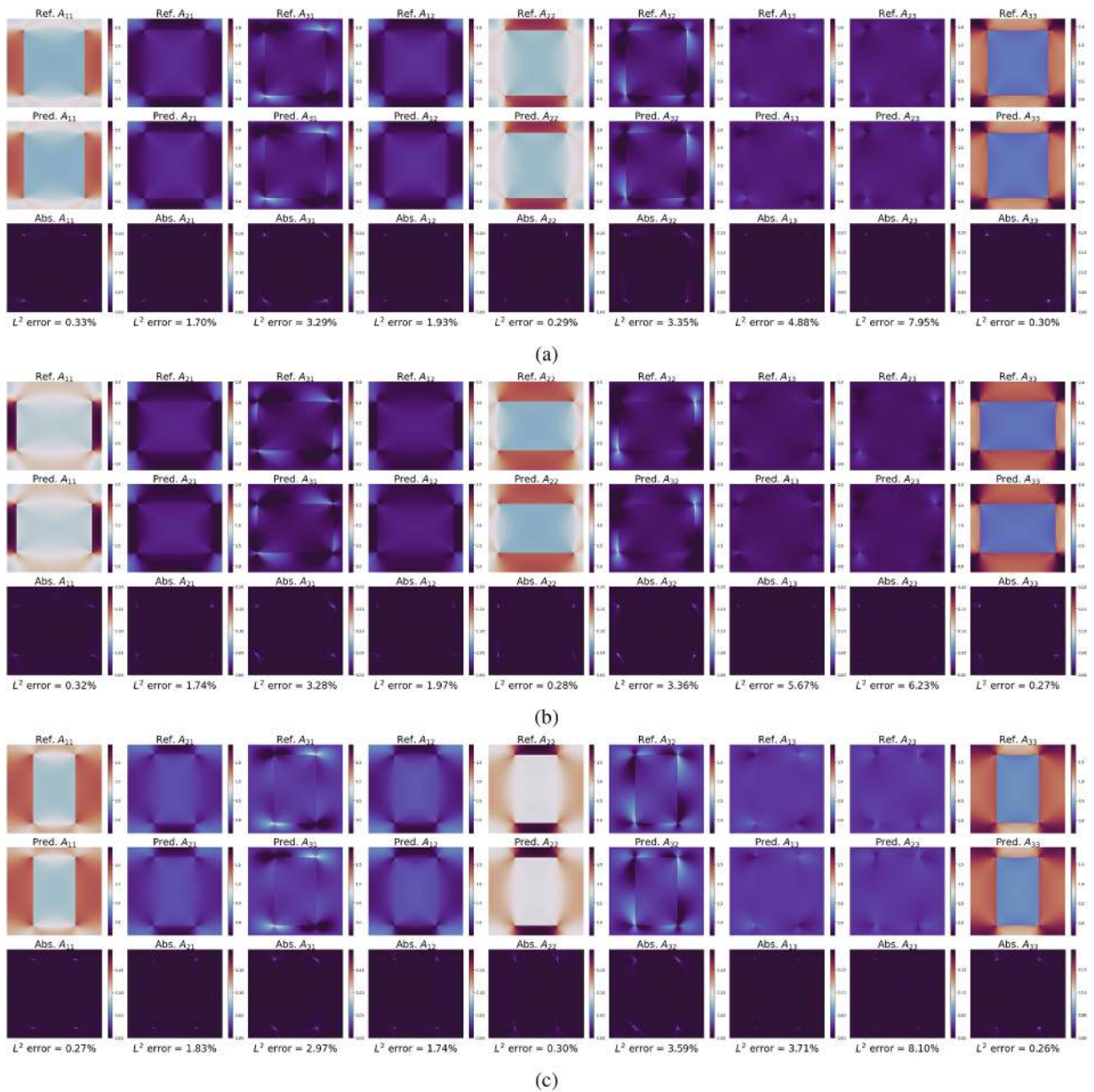


Fig. 11. Transfer Learning (Case III): Representative predictions of strain concentration tensors by Micrometer for different rectangular microstructures.

Another exciting avenue for future research is the integration of physics-informed machine learning techniques [8,86]. This approach could further enhance Micrometer’s predictive accuracy and data efficiency by embedding fundamental physical laws directly into the model training process. Such a methodology could be particularly beneficial for simulating complex phenomena like crack propagation or phase transition, where adherence to physical principles is crucial and data generation is computationally expensive. Augmenting the Micrometer training objective with additional physics-informed constraints is feasible thanks to the flexible design of the proposed continuous decoder that readily enables the computation of PDE constraints via automatic differentiation. By integrating data-driven learning with physics-based modeling, we anticipate that these future developments will greatly improve Micrometer’s versatility and applicability, potentially accelerating material discovery, optimizing structural designs, facilitating more efficient and accurate multiscale simulations in materials science and engineering.

CRedit authorship contribution statement

Sifan Wang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Tong-Rui Liu:** Writing – original draft, Software, Methodology, Investigation,

Formal analysis, Data curation, Conceptualization; **Shyam Sankaran**: Visualization, Software, Investigation; **Paris Perdikaris**: Writing – review & editing, Supervision, Investigation, Funding acquisition, Conceptualization.

Data availability

Code and data is publicly available at: <https://github.com/sifanexisted/micrometer>.

Declaration of competing interest

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

Acknowledgements

We would like to acknowledge support from the [US Department of Energy under the Advanced Scientific Computing Research program](#) (grant DE-SC0024563), and the [Institute for Foundations of Data Science at Yale University](#). We thank Dr. António Rui Melro, Dr. Zeliang Liu and Dr. Yanhong Chen for fruitful discussions and code development perspectives. We also thank the developers of the software that enabled our research, including JAX [87], Matplotlib [88], NumPy [89], and rvesimulator [52].

Appendix A. Nomenclature

Table A.6 summarizes the main symbols and notations used in this work. The fourth-order and second-order tensors are represented in Voigt notation. Accordingly, second-order tensors are expressed in vector form for computational convenience.

Appendix B. Green's function for Lippmann-Schwinger equation

B.1. Periodical isotropic Green's function in fourier space

Recall that Lippmann-Schwinger equation is given by

$$\varepsilon_{ij}(\mathbf{x}) + \int_{\Omega} G_{ijkl}^{(0)}(\mathbf{x}, \mathbf{x}') : \tau_{kl}(\mathbf{x}') d\mathbf{x}' - \bar{\varepsilon}_{ij} = 0. \quad (\text{B.1})$$

To solve the above equations, one can choose strain or stress controlled PBC applied to the RVE

$$\frac{1}{|\Omega|} \int_{\Omega} \varepsilon_{ij} d\mathbf{x} = \bar{\varepsilon}_{ij} = \varepsilon_{ij,M} \quad \text{or} \quad \frac{1}{|\Omega|} \int_{\Omega} \sigma_{ij} d\mathbf{x} = \sigma_{ij,M}, \quad (\text{B.2})$$

where $\varepsilon_{ij,M}$ and $\sigma_{ij,M}$ denote macroscopic homogenized strain and stress, respectively. Mixed boundary conditions can also be imposed in general.

Here the Green's function $G_{ijkl}^{(0)}(\mathbf{x}, \mathbf{x}')$ is a non-local function, which represents the strain field at point \mathbf{x} generated by a concentrated external stress field at point \mathbf{x}' in homogeneous reference material. Due to the periodic assumption of the RVE, $\mathbb{G}^{(0)}(\mathbf{x})$ can be expressed in Fourier space as $\hat{\mathbb{G}}^{(0)}(\boldsymbol{\xi})$, which can be determined by either continuous or discrete scheme. $\boldsymbol{\xi}$ is the frequency vector in Fourier space corresponding to \mathbf{x} in Euclidean space while the domain Ω is discretized with pixel grid, for instance:

$$\xi_i = \frac{2\pi}{h_i T_i} \begin{cases} \left[0, 1, \dots, \frac{T_i}{2} - 1, -\frac{T_i}{2}, \dots, -1 \right], & \text{if } T_i \text{ is even.} \\ \left[0, 1, \dots, \frac{T_i - 1}{2}, -\frac{T_i - 1}{2}, \dots, -1 \right], & \text{if } T_i \text{ is odd.} \end{cases} \quad (\text{B.3})$$

where $i = 1, 2$ represents the indices of spatial coordinate in 2D. h_i and T_i denotes the pixel size and the number of pixels along each spatial coordinate, respectively.

In general, the continuous formulation of above isotropic Green's function $\hat{\mathbb{G}}^{(0)}(\boldsymbol{\xi})$ in Fourier space is written as:

$$\begin{cases} \hat{G}_{ijkl}^{(0)}(\boldsymbol{\xi}) = 0, & \text{if } \boldsymbol{\xi} = \mathbf{0}, \\ \hat{G}_{ijkl}^{(0)}(\boldsymbol{\xi}) = \frac{1}{4\mu^0} \hat{G}_{ijkl}^{(1)}(\boldsymbol{\xi}) + \frac{\mu^0 + \lambda^0}{\mu^0(2\mu^0 + \lambda^0)} \hat{G}_{ijkl}^{(2)}(\boldsymbol{\xi}), & \text{if } \boldsymbol{\xi} \neq \mathbf{0}. \end{cases} \quad (\text{B.4})$$

$\hat{G}_{ijkl}^{(1)}(\boldsymbol{\xi})$ and $\hat{G}_{ijkl}^{(2)}(\boldsymbol{\xi})$ can be expressed as follows:

$$\hat{G}_{ijkl}^{(1)}(\boldsymbol{\xi}) = \frac{\delta_{ik}\xi_j\xi_l + \delta_{il}\xi_j\xi_k + \delta_{jk}\xi_i\xi_l + \delta_{jl}\xi_i\xi_k}{\|\boldsymbol{\xi}\|^2}, \quad \hat{G}_{ijkl}^{(2)}(\boldsymbol{\xi}) = -\frac{\xi_i\xi_j\xi_k\xi_l}{\|\boldsymbol{\xi}\|^4}. \quad (\text{B.5})$$

The mathematical proof of Eqs. (B.4) and (B.5) is provided in Appendix B.2.

Table A.6
Summary of the main symbols and notation used in this work.

Notation	Description
Micromechanics	
RVE	Representative volume element
PBC	Periodic boundary condition
FRP	Fiber reinforced composite
:	Tensor contraction between fourth-order tensor and second-order tensor
::	Tensor contraction between two fourth-order tensors
*	Convolution operator
$\Omega \in \mathbb{R}^2$	RVE domain in 2D
$\mathbf{x} \in \Omega$	Spatial coordinate of a microscale material point inside RVE domain Ω
ξ	Frequency vector in Fourier space corresponding to \mathbf{x} in Euclidean space
s, \tilde{s}, \bar{s}	Absolute, fluctuation and average displacement vector
$\epsilon, \tilde{\epsilon}, \bar{\epsilon}$	Absolute, fluctuation and average strain vector
σ	Stress vector
τ	Polarization stress vector
C	Fourth-order elastic tensor
A	Strain concentration tensor
$\mathbb{G}^{(0)}, \hat{\mathbb{G}}^{(0)}$	Green's function of Lippmann-Schwinger equation in Euclidean and Fourier space
E, ν	A pair of Young's modulus and Poisson ratio
λ, μ	A pair of Lamé constants
T	RVE discretization resolution
h	Pixel size in discretization of RVE
χ	Characteristic function determines the microstructural configuration of the RVE
Operator Learning	
\mathcal{X}	The input function space
\mathcal{Y}	The output function space
$u \in \mathcal{X}$	Input function
$v \in \mathcal{Y}$	Output function
y	Query coordinate in the input domain of v
$\Phi : \mathcal{X} \rightarrow \mathcal{Y}$	The operator mapping between function spaces
$\mathcal{E} : \mathcal{X} \rightarrow \mathbb{R}^n$	Encoder mapping
$\mathcal{D} : \mathbb{R}^n \rightarrow \mathcal{Y}$	Decoder mapping
ϕ	Activation function
Micrometer	
PE	Positional embedding
MSA	Multi-head self-attention
MHA	Multi-head attention
LN	Layer normalization
P	Patch size of Vision Transformer
D	Embedding dimension of Vision Transformer
β	Locality of interpolated latent grid features
Hyperparameters	
B	Batch size
Q	Number of query coordinates in each batch
$H \times W$	Resolution of spatial discretization

B.2. Mathematical proof

The aim of this subsection is to find the Green's function \mathbb{G}^0 corresponding to isotropic homogeneous reference material \mathbb{C}^0 in the Lippmann-Schwinger equation (see Eq. (2.12)). Recalling that the periodic boundary condition (PBC) is applied to the RVE, the periodical physical fields can be denoted by Neumann series with a single wave vector [90,91], such as:

$$\square^*(\mathbf{x}) = \hat{\square}^*(\xi) \exp(i\xi \cdot \mathbf{x}), \quad (\text{B.6})$$

where i denotes to $\sqrt{-1}$ and \square^* can be referred to $\sigma^*, \epsilon^*, \tau^*$ and s^* . Substitute Eq. (2.8) into Eq. (2.1) and express each physical field periodically, we can arrive at:

$$\begin{cases} \frac{\partial \sigma_{ij}^*(\mathbf{x})}{\partial x_i} = 0, \\ \sigma_{ij}^*(\mathbf{x}) = C_{ijkl}^0(\mathbf{x}) : \epsilon_{kl}^*(\mathbf{x}) + \tau_{ij}^*(\mathbf{x}), \\ \epsilon_{ij}^*(\mathbf{x}) = \frac{1}{2} \left(\frac{\partial s_i^*(\mathbf{x})}{\partial x_j} + \frac{\partial s_j^*(\mathbf{x})}{\partial x_i} \right). \end{cases} \quad (\text{B.7})$$

Recalling that the strain field and displacement field can be decomposed into an average and fluctuation part, we have

$$\begin{cases} s_i^*(\mathbf{x}) = \bar{s}_i^*(\mathbf{x}) + \hat{s}_i, \\ \varepsilon_{ij}^*(\mathbf{x}) = \bar{\varepsilon}_{ij}^*(\mathbf{x}) + \hat{\varepsilon}_{ij}. \end{cases} \tag{B.8}$$

By Substituting Eq. (B.6) into Eq. (B.7) and eliminating σ_{ij}^* , we have

$$C_{ijkl}^0 \xi_j \xi_l \hat{s}_k^*(\xi) = i \hat{\tau}_{ij}^*(\xi) \xi_j. \tag{B.9}$$

Denote the coefficient of \hat{s}_k^* in the left hand side of above equation by:

$$K_{ik}(\xi) = C_{ijkl}^0 \xi_l \xi_j, \tag{B.10}$$

which is the acoustic tensor of the homogeneous reference material. Since the fourth order elastic tensor C_{ijkl}^0 can be expressed as:

$$C_{ijkl}^0 = \lambda^0 \delta_{ij} \delta_{kl} + \mu^0 (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}). \tag{B.11}$$

Therefore, $K_{ik}(\xi)$ can be expressed as:

$$K_{ik}(\xi) = (\mu^0 + \lambda^0) \xi_i \xi_k + \mu^0 \|\xi\|^2 \delta_{ik}. \tag{B.12}$$

We can denote the inverse of $K_{ik}(\xi)$ by $N_{ki}(\xi)$, such as:

$$N_{ki}(\xi) = \frac{1}{\mu^0 \|\xi\|^2} \left(\delta_{ki} - \frac{\mu^0 + \lambda^0}{\mu^0 + 2\lambda^0} \frac{\xi_k \xi_i}{\|\xi\|^2} \right). \tag{B.13}$$

Using the above equation, \hat{s}_k^* can be expressed as:

$$\hat{s}_k^*(\xi) = i N_{ki}(\xi) \hat{\tau}_{ij}^*(\xi) \xi_j. \tag{B.14}$$

According to the symmetrical property of polarization stress $\hat{\tau}_{ij}^*$, \hat{s}_k^* can be further denoted by:

$$\hat{s}_k^*(\xi) = \frac{i}{2} (N_{ki}(\xi) \xi_j + N_{kj}(\xi) \xi_i) \hat{\tau}_{ij}^*(\xi). \tag{B.15}$$

Taking the symmetric gradient of \hat{s}_k^* , the strain fluctuation can be expressed as:

$$\hat{\varepsilon}_{kl}^*(\xi) = -\frac{1}{4} (N_{ki}(\xi) \xi_j \xi_l + N_{kj}(\xi) \xi_i \xi_l + N_{li}(\xi) \xi_j \xi_k + N_{lj}(\xi) \xi_i \xi_k) \hat{\tau}_{ij}^*(\xi). \tag{B.16}$$

The Green's function is defined as a symmetric fourth order tensor which creates a mapping from polarization stress $\hat{\tau}_{ij}^*(\xi)$ to strain fluctuation $\hat{\varepsilon}_{kl}^*(\xi)$:

$$\hat{G}_{kl ij}^0(\xi) = -\frac{1}{4} (N_{ki}(\xi) \xi_j \xi_l + N_{kj}(\xi) \xi_i \xi_l + N_{li}(\xi) \xi_j \xi_k + N_{lj}(\xi) \xi_i \xi_k). \tag{B.17}$$

Substitute Eq. (B.13) into above equation, the Green's function can be computed by:

$$\hat{G}_{ijkl}^0(\xi) = \frac{1}{4\mu^0 \|\xi\|^2} (\delta_{ki} \xi_j \xi_l + \delta_{il} \xi_j \xi_k + \delta_{jl} \xi_i \xi_k + \delta_{jk} \xi_i \xi_l) - \frac{(\mu^0 + \lambda^0)}{\mu^0 (2\mu^0 + \lambda^0)} \frac{\xi_i \xi_j \xi_k \xi_l}{\|\xi\|^4}, \tag{B.18}$$

which is corresponding to Eqs. (B.4) and (B.5). Therefore, the proof ends.

B.3. Expression of Green's function in fourier space by Voigt notation

Following Eq. (B.5) in Section 2, the Green's function in Fourier Space is expressed by:

$$\hat{\mathbb{G}}^{(0)}(\xi) = \frac{1}{4\mu^0} \hat{\mathbb{G}}^{(1)}(\xi) + \frac{(\mu^0 + \lambda^0)}{\mu^0 (2\mu^0 + \lambda^0)} \hat{\mathbb{G}}^{(2)}(\xi). \tag{B.19}$$

Each component in $\hat{\mathbb{G}}^{(1)}(\xi)$ and $\hat{\mathbb{G}}^{(2)}(\xi)$ in terms of Voigt notation is given by:

$$\hat{\mathbb{G}}^{(1)}(\xi) = \begin{bmatrix} \frac{4\xi_1^2}{\|\xi\|^2} & 0 & \frac{4\xi_1 \xi_2}{\|\xi\|^2} \\ 0 & \frac{4\xi_2^2}{\|\xi\|^2} & \frac{4\xi_1 \xi_2}{\|\xi\|^2} \\ \frac{4\xi_1 \xi_2}{\|\xi\|^2} & \frac{4\xi_1 \xi_2}{\|\xi\|^2} & \frac{4(\xi_1^2 + \xi_2^2)}{\|\xi\|^2} \end{bmatrix}, \tag{B.20}$$

$$\hat{\mathbb{G}}^{(2)}(\xi) = \begin{bmatrix} \frac{\xi_1^4}{\|\xi\|^4} & \frac{\xi_1^2 \xi_2^2}{\|\xi\|^4} & \frac{2\xi_1^3 \xi_2}{\|\xi\|^4} \\ \frac{\xi_1^2 \xi_2^2}{\|\xi\|^4} & \frac{\xi_2^4}{\|\xi\|^4} & \frac{2\xi_1 \xi_2^3}{\|\xi\|^4} \\ \frac{2\xi_1^3 \xi_2}{\|\xi\|^4} & \frac{2\xi_1 \xi_2^3}{\|\xi\|^4} & \frac{4\xi_1^2 \xi_2^2}{\|\xi\|^4} \end{bmatrix}. \tag{B.21}$$

Appendix C. Data generation

C.1. FFT based homogenization

Here we summarize the numerical method and process for generating the outputs of training and test dataset, typically, the point-wise strain concentration tensor $\mathbb{A}(\mathbf{x})$ for each RVE. Throughout this work, these datasets are generated via solving Eq. (2.11) by fast Fourier transformation (FFT) based homogenization [6], which is an accurate and efficient numerical method. The systematic solution scheme is provided in Algorithm 1. Compared to other element based numerical methods, the convolution term between Green's function and polarization stress is transferred into dot product and computed in Fourier space via discrete Fourier transform (DFT) rather than in Euclidean space, which exhibits more time and memory efficiency. Rather, the point-wise stress is updated in Euclidean space by local constitutive law. Concerning the convergence test in Algorithm 1, we check the equilibrium state of $\sigma^{(n)}(\mathbf{x})$, referred to the local stress field $\sigma(\mathbf{x})$ at iteration counter n . Following the same principle in references [6,92], an error index Tol^n to check convergence is defined as:

$$\text{Tol}^n = \sqrt{\frac{\sum_{\xi} |\xi \cdot \hat{\sigma}^{(n)}(\xi)|^2}{T_1 T_2 [\hat{\sigma}_{ij}^{(n)}(\mathbf{0}) : \hat{\sigma}_{ij}^{(n)}(\mathbf{0})]}} \ll 1. \quad (\text{C.1})$$

By virtue of Parseval's theorem [93], the convergence criteria can be checked in either Euclidean space or Fourier space. $\hat{\sigma}^{(n)}(\mathbf{0})$ refers to stress in Fourier space at zero frequency, corresponding to the macroscopic average stress in Euclidean space. The fixed-point iteration stops while Tol^n reaches a prescribed value (e.g., 10^{-6}). The convergence rate of the fixed-point iteration solver in Algorithm 1 is highly dependent on the choice of reference material \mathbb{C}^0 , which is determined by λ_0 and μ_0 . Following the work of Moulinec and Suquet [6], to achieve the best convergence property, a pair of λ_0 and μ_0 is chosen as

$$\begin{cases} \lambda^0 = \frac{1}{2} \left(\inf_{\mathbf{x} \in \Omega} \lambda(\mathbf{x}) + \sup_{\mathbf{x} \in \Omega} \lambda(\mathbf{x}) \right), \\ \mu^0 = \frac{1}{2} \left(\inf_{\mathbf{x} \in \Omega} \mu(\mathbf{x}) + \sup_{\mathbf{x} \in \Omega} \mu(\mathbf{x}) \right), \end{cases} \quad (\text{C.2})$$

where $\lambda(\mathbf{x})$ and $\mu(\mathbf{x})$ are computed through $E(\mathbf{x})$ and $\nu(\mathbf{x})$ following Eq. (2.2). Besides, the usage of a continuous Green's function in FFT based homogenization yields numerical defects in local physical fields, such as Ringing artifacts, stress oscillations, to name few [94,95]. To alleviate those negative effects, here we adopt the discrete Green's function based on modified frequency vectors, which can be derived by using the finite difference method (FDM) on a rotated grid. The modified frequency vectors $\tilde{\xi}$ are expressed as [96]

$$\begin{cases} \tilde{\xi}_1 = \frac{2}{h_1} \sin\left(\frac{\xi_1}{2}\right) \cos\left(\frac{\xi_2}{2}\right), \\ \tilde{\xi}_2 = \frac{2}{h_2} \cos\left(\frac{\xi_1}{2}\right) \sin\left(\frac{\xi_2}{2}\right), \end{cases} \quad (\text{C.3})$$

where $h_i (i = 1, 2)$ denotes the pixel size along direction i . The numerical implementation is achieved by replacing ξ with $\tilde{\xi}$ in Eq. (B.4). For a mathematical proof of the discrete Green's function, interested readers are referred to [92,95,96] for details.

Algorithm 1: Basic scheme of FFT based homogenization [6].

Step.1: Initialization, set the strain controlled PBC using Eq.(3.3). For each $\mathbf{x} \in \Omega$, initial local strain and stress fields are set as:

$$\epsilon^{(0)}(\mathbf{x}) = \bar{\epsilon} = \epsilon_M, \quad \sigma^{(0)}(\mathbf{x}) = \mathbb{C}(\mathbf{x}) : \epsilon^{(0)}(\mathbf{x}).$$

Step.2: Solve Eq.(2.11) by fixed point iteration to obtain $\epsilon^{(n+1)}(\mathbf{x})$, $\sigma^{(n+1)}(\mathbf{x})$.

- a. Compute the polarization stress in Euclidean space: $\tau^{(n)}(\mathbf{x}) = \sigma^{(n)}(\mathbf{x}) - \mathbb{C}^0 : \epsilon^{(n)}(\mathbf{x})$.
- b. Fast Fourier transform on polarization stress: $\hat{\tau}^{(n)}(\xi) = \mathcal{F}(\tau^{(n)}(\mathbf{x}))$.
- c. Compute local strain in Euclidean space: $\epsilon^{(n+1)}(\mathbf{x}) = \bar{\epsilon} - \mathcal{F}^{-1}(\hat{\mathbb{G}}^{(0)}(\xi) : \hat{\tau}^{(n)}(\xi))$.
- d. Update local stress in Euclidean space: $\sigma^{(n+1)}(\mathbf{x}) = \mathbb{C}(\mathbf{x}) : \epsilon^{(n+1)}(\mathbf{x})$.

Step.3: Convergence test using Eq.(C.1), if not convergence, repeating **Step.2** with iteration $n = n + 1$.

Appendix D. FE-Micrometer

Algorithm 3 outlines the framework of FE-Micrometer for concurrent multiscale modeling. This framework integrates the finite element method (FEM) to solve the macroscale boundary value problem (MBVP) with Micrometer for resolving the microscale boundary value problem (mBVP) within the RVE, which is associated with a macroscale solid element. For the macroscale FEM, we employ plane strain quadrilateral elements with reduced (one-point) integration scheme (CPE4R).

Algorithm 2: Pipeline of data generation.

Input: N_p : Number of RVE domains, $T_1 \times T_2$: RVE resolutions, R_d : Fiber radius, Std : Standard deviation of fiber radius

Output: RVE_image , $labels_mat$, A_{ten}

Step.1: Sampling of material properties

begin

a. Generate material properties for each RVE and store them in an array $labels_mat \in \mathbb{R}^{N_p \times 4}$ using Latin Hypercube

Sampling (LHS), where row i contains $[E_f^i, \nu_f^i, E_m^i, \nu_m^i]$ for Ω^i :

$[labels_mat] = \text{LHS_design}(N_p, E_{f,max}, \nu_{f,max}, E_{m,max}, \nu_{m,max}, E_{f,min}, \nu_{f,min}, E_{m,min}, \nu_{m,min})$

where $E_{\square,\square}$ and $\nu_{\square,\square}$ are referred to the upper and lower bounds of Young's modulus and Poisson ratio for matrix and fiber materials, respectively. These ranges of material properties are summarized in Table 1.

end

Step.2: RVE and fourth order elastic tensor generation

begin

a. Initialize a 1D array $labels_Vof$ to store the fiber volume fraction Vof^i for all Ω^i .

b. Divide N_p into 20 groups with fiber volume fraction uniformly distributed ranging from 40 % to 60 %.

c. Set array $labels_Vof$: $[labels_Vof] = \text{reshape}(\text{repmat}(40 : 1 : 60, N_p/20, 1), N_p, 1)$

d. Initialize a 3D array $RVE_image \in \mathbb{R}^{T_1 \times T_2 \times N_p}$ to store the RVE information, where $RVE_image(:, :, i)$ is a boolean array representing the i -th RVE's microstructure (0: matrix, 1: fiber). Meanwhile, initialize a 5D array $C_{ten} \in \mathbb{R}^{T_1 \times T_2 \times 3 \times 3 \times N_p}$ to store the fourth order elastic tensor, where $C_{ten}(:, :, :, :, i)$ corresponds to the pixel-wise fourth order elastic tensor for i -th RVE.

e. Generate RVEs and fourth order elastic tensor, and store them in RVE_image and C_{ten} , respectively:

for $i \in [1, N_p]$ **do**

$RVE_image(:, :, i) = \text{RAND_uSTRU_GEN}(R_d, labels_Vof(i), Std, T_1, T_2)$

for $p \in [1, T_1]$ **do**

for $q \in [1, T_2]$ **do**

$[C_{ten}(p, q, :, :, i)] = C_tensor(RVE_image(p, q, i), labels_mat(i, :))$ # Eqs. (2.2) and (2.3)

end

end

end

end

Step.3: Calculation of strain concentration tensor

begin

a. Initialize a 5D array $A_{ten} \in \mathbb{R}^{T_1 \times T_2 \times 3 \times 3 \times N_p}$ to store the strain concentration tensor, where $A_{ten}(:, :, :, :, i)$ corresponds to the full-field strain concentration tensor for i -th RVE.

b. Run Algorithm 1 to compute A_{ten} :

for $i \in [1, N_p]$ **do**

$[A_{ten}(:, :, :, :, i)] = \text{FFT_homogenization}(C_{ten}(:, :, :, :, i))$

end

end

Appendix E. Overview of Fourier Neural Operator (FNO)

Integral Kernel Neural Operators form a general class of neural operators, which learn mappings between function spaces by approximating the integral kernel of an operator using neural networks. They were introduced as a generalization of the Neural Operator framework proposed by Li et al. [97]. The key idea here is to represent the underlying operator $\Phi : \mathcal{X} \rightarrow \mathcal{Y}$ as a convolutional integral operator, $\Phi(u)(y) = \int_{\Omega} \kappa(x - y)u(x)dx$, where κ is a kernel function. In practice, the kernel function κ is parameterized by a neural network, and the integral is then approximated using a quadrature rule, such as Monte Carlo integration, Gaussian quadrature, or via spectral convolution in the frequency domain.

FNO is one representative example of this paradigm, defined as:

$$\Phi(u; \theta) = Q \circ H_L \circ \dots \circ H_2 \circ H_1 \circ \mathcal{R}(u), \quad (\text{E.1})$$

where each hidden layer $H_{\ell}(v)(x; \theta_{\ell})$ is given by:

$$H_{\ell}(v)(x; \theta) = \phi(W_{\ell}v(x) + b_{\ell} + \mathcal{K}(v)(x; \theta_{\ell})), \quad (\text{E.2})$$

with a component-wise activation function ϕ . The architecture consists of input and output layers, \mathcal{R} and Q , respectively, which are typically pointwise compositions using either shallow neural networks or linear transformations. The hidden layers H_{ℓ} involves an

Algorithm 3: Concurrent multiscale modeling using FE-Micrometer.**Step.1: Prepare the macroscale FE mesh, material properties and model inputs & outputs****begin**

- Generate the macroscale FE mesh with N_{node} nodes and set N_p as the number of RVEs in FE mesh.
- Set macroscale material properties (for each RVE domain Ω^i) by using Eq.(5.7), and store them in an array $labels\ mat \in \mathbb{R}^{N_p \times 4}$, where row i contains $[E_f^i, \nu_f^i, E_m^i, \nu_m^i]$ for Ω^i .
- Following Step.2 in Algorithm 2, generate RVEs and fourth order elastic tensor, and store them in $RVE_image \in \mathbb{R}^{T_1 \times T_2 \times N_p}$ and $C_{ten} \in \mathbb{R}^{T_1 \times T_2 \times 3 \times 3 \times N_p}$, respectively.
- Inference the *Micrometer* and store the predicted strain concentration tensor into $A_{ten} \in \mathbb{R}^{T_1 \times T_2 \times 3 \times 3 \times N_p}$:

$$[A_{ten}] = \text{Micrometer}(C_{ten})$$

end**Step.2: Quasi static analysis: FE-Micrometer two-scale calculation at arbitrary time interval $[t_n, t_{n+1}]$**

Input: $s^n \in \mathbb{R}^{2N_{node}}$ and $C_{M,stan}^n \in \mathbb{R}^{3 \times 3 \times N_p}$: Displacement and macroscale consistent tangent stiffness at t_n , A_{ten} : strain concentration tensor, C_{ten} : fourth order elastic tensor, $F_{ext}^n \in \mathbb{R}^{2N_{node}}$: External force, Dof_I : label of DOFs in stress free status

Output: $s^{n+1} \in \mathbb{R}^{2N_{node}}$, $C_{M,stan}^{n+1} \in \mathbb{R}^{3 \times 3 \times N_p}$, $\sigma_M^{n+1} \in \mathbb{R}^{3 \times 1 \times N_p}$, $\epsilon_M^{n+1} \in \mathbb{R}^{3 \times 1 \times N_p}$, $\sigma^{n+1} \in \mathbb{R}^{T_1 \times T_2 \times 3 \times 1 \times N_p}$, $\epsilon^{n+1} \in \mathbb{R}^{T_1 \times T_2 \times 3 \times 1 \times N_p}$ and $F_{int}^{n+1} \in \mathbb{R}^{2N_{node}}$: Displacement, macroscale consistent tangent stiffness, macrostress, macrostrain, microstress, microstrain and internal force at t_{n+1}

begin

Set $s^{n,k} \leftarrow s^n$, $C_{M,stan}^{n,k} \leftarrow C_{M,stan}^n$ and apply the macroscale boundary conditions.

repeat

- Compute the macroscale global stiffness matrix K_u and internal force vector F_{int} using macroscale consistent tangent stiffness $C_{M,stan}^{n,k}$ and displacement field $s^{n,k}$.
- Get displacement field $s^{n,k+1}$ and macrostrain $\epsilon_M^{n,k+1}$:

$$R^{n,k+1} = F_{ext}^n - F_{int}, \quad s^{n,k+1} = s^{n,k} - [K_u]^{-1} R^{n,k+1} \quad \text{and} \quad \epsilon_M^{n,k+1} = \nabla^{\text{sym}} s^{n,k+1} \quad (\text{D.1})$$

- Get microstress $\sigma^{n,k+1}$, microstrain $\epsilon^{n,k+1}$, macrostress $\sigma_M^{n,k+1}$ and consistent tangent stiffness $C_{M,stan}^{n,k+1}$. The function Mat below corresponds to Eqs. (2.2), (2.3) and (3.2):

for $i \in [1, N_p]$ **do****for** $p \in [1, T_1]$ **do****for** $q \in [1, T_2]$ **do**

$$[\sigma^{n,k+1}(p, q, :, :, i), \epsilon^{n,k+1}(p, q, :, :, i)] = \text{Mat}(C_{ten}(p, q, :, :, i), A_{ten}(p, q, :, :, i), \epsilon_M^{n,k+1}(:, :, i))$$

end**end**

Compute the macrostress $\sigma_M^{n,k+1}(:, :, i)$ by averaging microstress $\sigma^{n,k+1}(:, :, :, i)$ in Ω_i

Compute the macroscale tangent stiffness $C_{M,stan}^{n,k+1}(:, :, i)$ by averaging the pagewise production between $C_{ten}(p, q, :, :, i)$ and $A_{ten}(p, q, :, :, i)$ in Ω_i , see Eq.(5.5)

end

Compute $F_{int}^{n+1,k}$ and get the reaction force by summing up $F_{int}^{n+1,k}(Dof_I)$

Set $k \leftarrow k + 1$ **until** $|R^{n,k}(Dof_f)|_2 \leq TOL$, with $TOL = 10^{-7}$;**end**Updating the solution: Set $s^{n,k} \rightarrow s^{n+1}$

affine transformation and incorporates a convolutional integral operator \mathcal{K} , parameterized by θ_ℓ . This operator can be efficiently evaluated using the Fourier transform \mathcal{F} , resulting in a matrix-valued Fourier multiplier:

$$\mathcal{K}(v)(x; \theta_\ell) = \int \kappa(x-y; \theta_\ell) v(y) dy = \mathcal{F}^{-1}(\mathcal{F}(\kappa(\cdot; \theta_\ell)) \mathcal{F}(v)), \quad (\text{E.3})$$

where the Fourier transform \mathcal{F} is computed component-wise. More specifically, if $\kappa(x) = (\kappa_{ij}(x))_{ij=1}^{d_c}$ represents the components of $\kappa(x)$, and $\hat{\kappa}_{k,ij}$ denotes the k -th Fourier coefficient of $\kappa_{ij}(x)$, the i -th component $\mathcal{K}(v)_i$ of the output function $\mathcal{K}(v)$ is expressed as:

$$[\mathcal{K}(v)_i](x; \theta_\ell) = \frac{1}{(2\pi)^d} \sum_{k \in \mathbb{Z}^d} \left(\sum_{j=1}^{d_c} \hat{\kappa}_{k,ij} \mathcal{F}(v_j)(k) \right) e^{ik \cdot x}. \quad (\text{E.4})$$

Here the Fourier coefficients $\hat{\kappa}_{k,ij}$ are the trainable parameters of the convolutional operator. In practice, a Fourier cut-off k_{\max} is introduced, restricting a finite sum over k up to wavenumbers $|k|_{\ell^\infty} \leq k_{\max}$, where $|\cdot|_{\ell^\infty}$ is the ℓ^∞ -norm. This results in a finite set of trainable parameters, $\theta_\ell = \{\hat{\kappa}_{k,ij} : |k|_{\ell^\infty} \leq k_{\max}, i, j = 1, \dots, d_\ell\}$.

While the FNO is theoretically formulated directly on the function space without reduction to a finite-dimensional latent space, practical implementations typically discretize it by identifying the input and output functions with their point-values on an equidistant grid. This allows for efficient evaluation of the discrete Fourier transform using the fast Fourier transform algorithm (FFT), facilitating straightforward implementation in popular deep learning libraries.

Appendix F. Model details

Fourier Neural Operator (FNO) We employ a Fourier Neural Operator (FNO) model with four layers, where the number of channels increases from 32 to 128, and the Fourier modes range from 16 to 32. The GeLU activation function is employed, and no normalization scheme is applied. As shown in Table G.8, the performance of FNO is inferior to that of Micrometer, even when using a comparable number of parameters.

Vision Transformer (ViT) We employ various configurations of the Vision Transformer (ViT), as detailed in Table F.7, and explore different patch sizes of 8 and 16. As shown in Table G.8, the model performance is highly sensitive to the patch size, with smaller patches resulting in significantly better test errors. However, even with these adjustments, the accuracy of ViT remains lower than that of Micrometer when comparing models with similar numbers of parameters.

Table F.7
Details of Vision Transformer model variants.

Model	Encoder layers	Embedding dim	MLP width	Heads	# Params
ViT-S	6	384	384	6	8 M
ViT-B	12	512	1024	16	26 M
ViT-L	18	768	1536	12	87 M

UNet: We adhere to the conventional implementation of the conventional UNet [66], with group normalization. We vary the embedding dimension of UNets, ranging from 32 to 128. As shown in Table G.8, the UNet consistently underperforms compared to the other models.

Appendix G. Additional results

Table G.8

Performance of various deep learning models for predicting full microstress fields. Models include U-Net, Fourier Neural Operator (FNO), Vision Transformer (ViT), and Micrometer with different configurations. Metrics include relative L^1 and L^2 errors, Root Mean Square Error (RMSE). Lower values (\downarrow) indicate better performance for error metrics.

Model	# Params	Rel. L^1 error (\downarrow)	Rel. L^2 error (\downarrow)	RMSE (\downarrow)	Training time (hours)
UNet-16	2 M	16.33 %	16.99 %	0.0837	8.5
UNet-32	8 M	15.34 %	15.80 %	0.0797	8.5
UNet-64	31 M	15.11 %	15.56 %	0.0795	8.5
UNet-96	70 M	14.91 %	15.44 %	0.0789	12.5
UNet-128	124 M	14.29 %	14.75 %	0.0752	20.5
FNO-32-16 modes	4 M	8.81 %	11.22 %	0.0541	10.2
FNO-32-32 modes	17 M	7.47 %	9.84 %	0.0477	10.2
FNO-64-16 modes	16 M	6.50 %	8.87 %	0.0437	10.2
FNO-64-32 modes	67 M	5.72 %	7.97 %	0.0399	10.2
FNO-128-16 modes	67 M	5.49 %	7.85 %	0.0400	14.5
FNO-128-32 modes	268 M	4.79 %	7.19 %	0.0373	14.5
ViT-S-8	8 M	9.32 %	11.96 %	0.0543	9.2
ViT-S-16	8 M	16.82 %	20.85 %	0.1047	8.0
ViT-B-8	26 M	7.24 %	9.21 %	0.0437	11.8
ViT-B-16	26 M	14.64 %	18.45 %	0.0911	10.0
ViT-L-8	87 M	4.95 %	6.86 %	0.0346	15.2
ViT-L-16	87 M	11.77 %	15.42 %	0.0731	12.4
Micrometer-S-8	20 M	5.32 %	7.24 %	0.0359	9.2
Micrometer-S-16	20 M	5.46 %	7.34 %	0.0364	8.5
Micrometer-B-8	70 M	4.39 %	6.42 %	0.0337	17.0
Micrometer-B-16	70 M	5.13 %	7.05 %	0.0354	14.5
Micrometer-L-8	292 M	3.61 %	5.95 %	0.0303	18.2
Micrometer-L-16	292 M	4.14 %	6.12 %	0.0312	15.5

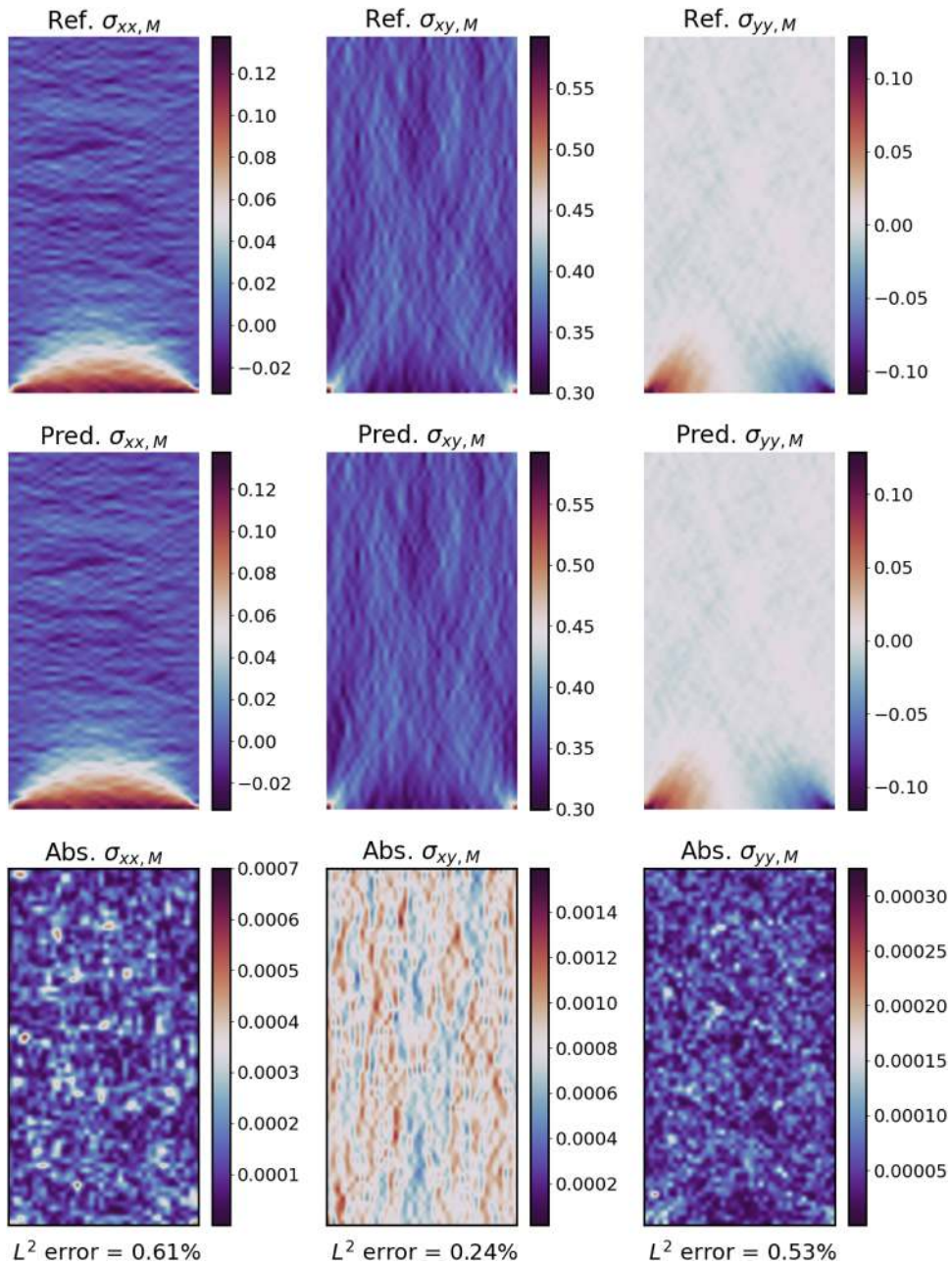


Fig. G.12. Concurrent Multiscale Modeling: Comparison of macrostress for a representative composite plate with low-Young's modulus ratio between FE-FFT and FE-Micrometer.

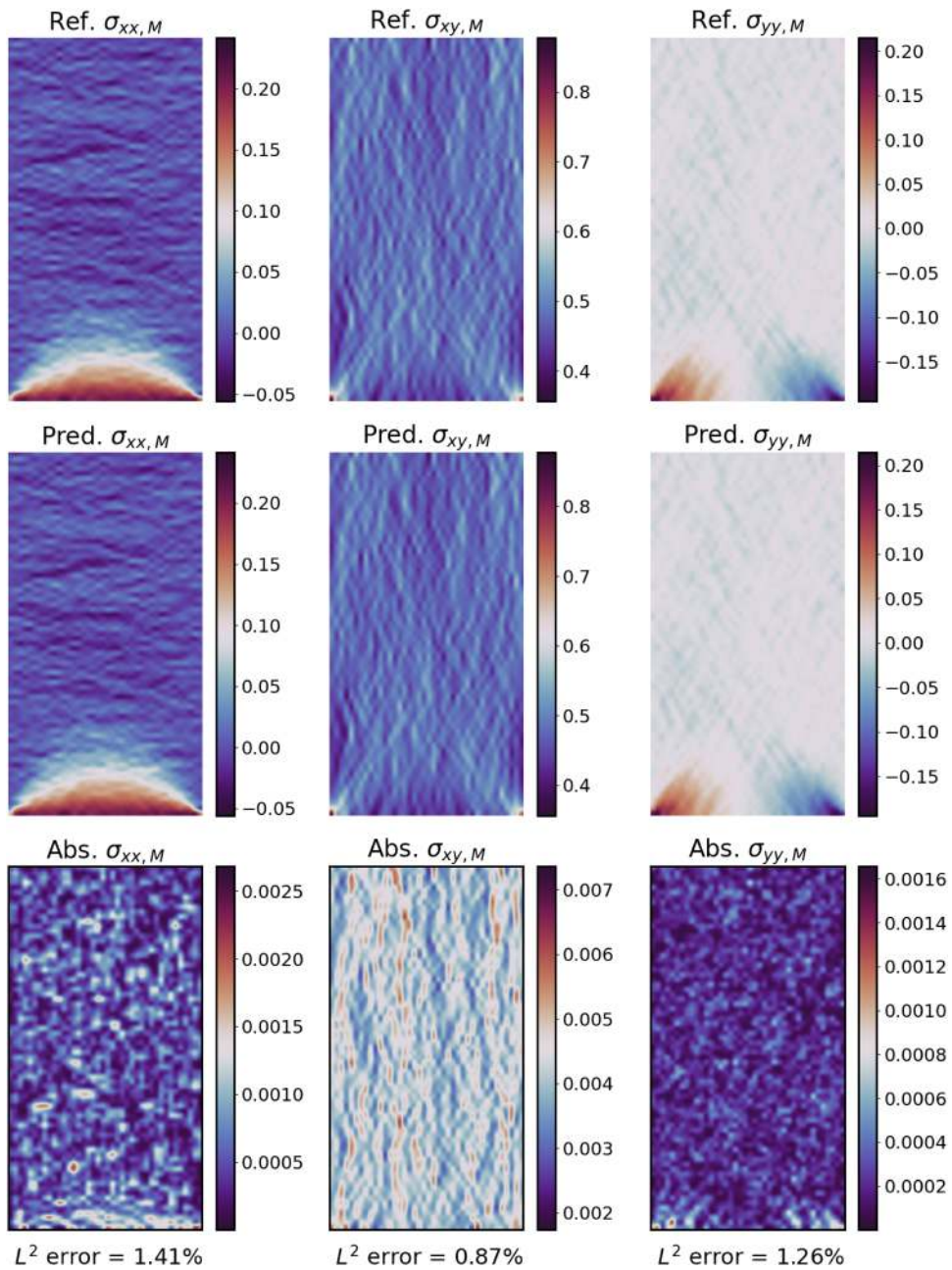


Fig. G.13. Concurrent Multiscale Modeling: Comparison of macrostress for a representative composite plate with medium-Young’s modulus ratio between FE-FFT and FE-Micrometer.

Appendix H. Extension to 3D fiber reinforced composites

To demonstrate the extensibility of our method to three-dimensional problems, we present a proof-of-concept study involving 3D fiber-reinforced composite materials. While the primary focus of this paper is on 2D microstructures for clarity and computational tractability, this additional example illustrates that our framework naturally generalizes to 3D with minimal architectural changes.

Problem setup. We consider three-dimensional fiber-reinforced composite RVEs, modeled as extrusions of 2D cross-sections along the out-of-plane direction. which can be regarded as extrusions of two-dimensional transverse cross sections along the out-of-plane direction. Regarding the geometrical configuration, we set the RVE size to $50\mu\text{m} \times 50\mu\text{m} \times 12.5\mu\text{m}$, with a $128 \times 128 \times 32$ discretization resolution.

We generate 2000 training and 100 test RVE samples with random distribution of fiber, where the radius and its standard deviation are provided in Table 1. The volume fraction of fiber is set as 45%. The material properties of fiber and matrix are fixed among RVE samples, which are provided in Table H.9. The matrix material is considered isotropic, while the fiber is modeled to be transversely isotropic, with distinct Young's modulus and Poisson's ratio values in the out-of-plane direction compared to the in-plane directions. This anisotropy introduces a directional dependence on the mechanical response, making the 3D case more representative of practical composite systems. One sample of 3D fiber reinforced composite RVE is given in Fig. H.14.

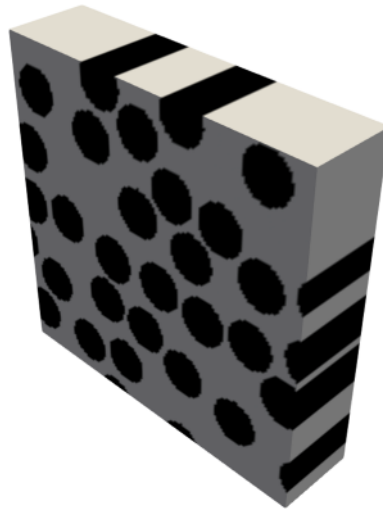


Fig. H.14. A sample of 3D fiber reinforced composite microstructure.

Table H.9

Elastic property of matrix and fiber material [98].

	E_{11} (GPa)	E_{33} (GPa)	ν_{12}	ν_{13}	G_{13} (GPa)
Fiber	19.80	245.00	0.32	0.28	29.20
Matrix	3.80	-	0.39	-	-

In this example, our aim is to learn the mapping from spatially varying fourth-order elastic tensors to the diagonal components of the strain concentration tensor. The A_{33} component is excluded from the training, as it is identical unit value throughout the domain due to boundary conditions. To this end, we extend the 2D version of our Micrometer model to 3D by directly replacing the 2D Fourier Neural Operator (FNO) encoder with its 3D counterpart. We train the model to follow the same procedure as described in the main text. On the held-out test set, the model achieves a relative L^2 error of 2.75%. Fig. H.15 visualizes comparisons between the predicted and reference strain fields. The model demonstrates excellent agreement with the ground truth, accurately capturing both the spatial distribution and magnitude of the fields. This 3D experiment confirms that our operator learning framework is naturally generalized to higher-dimensional problems and remains effective in modeling the mechanical response of anisotropic composite materials.

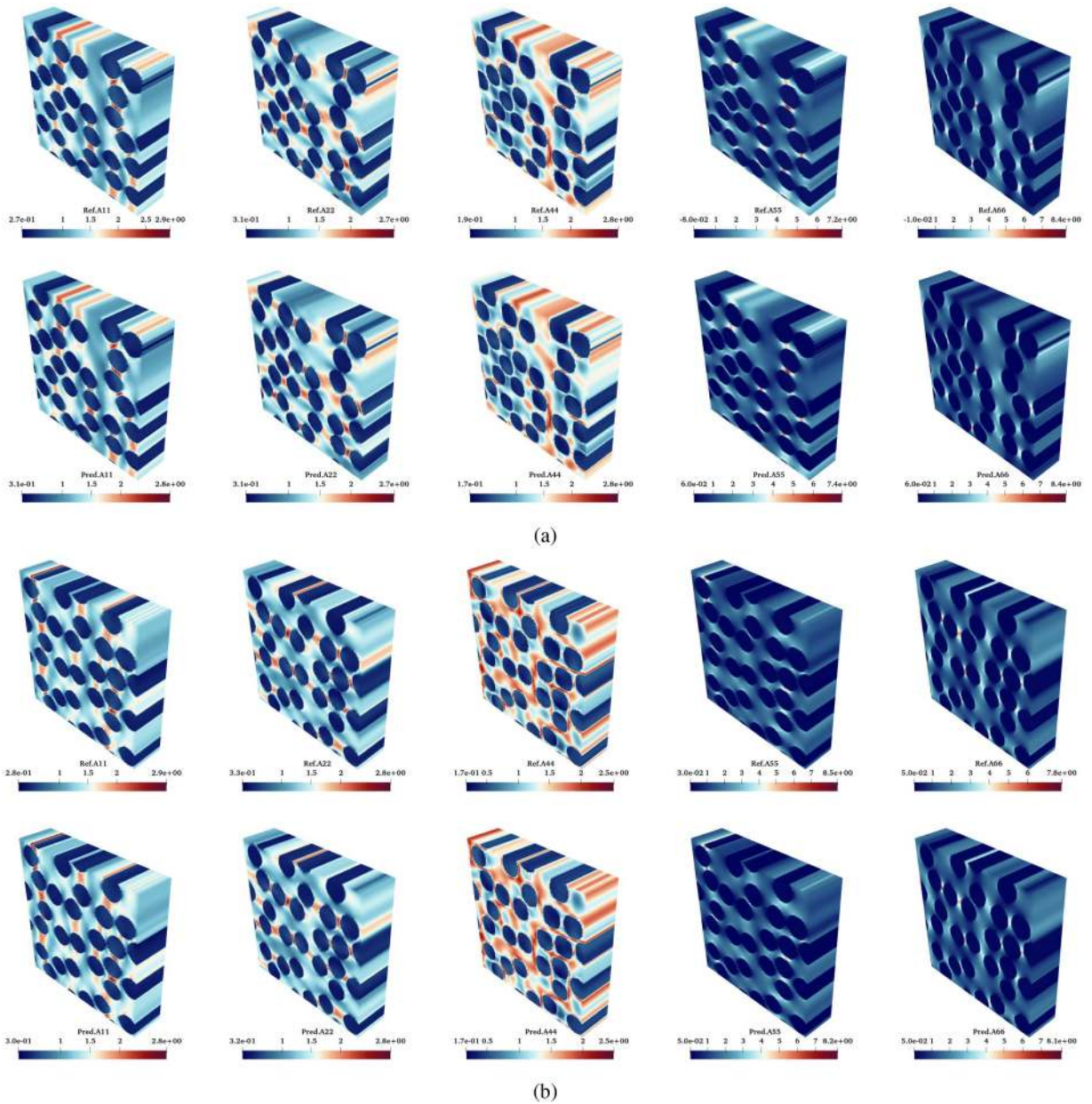


Fig. H.15. Representative predictions by Micrometer for 3D microstructures.

References

- [1] M.F. Ashby, D.R.H. Jones, *Engineering Materials 1: An Introduction to Properties, Applications and Design*, 1, Elsevier, Oxford, 2012.
- [2] M.F. Ashby, *Materials Selection in Mechanical Design*, Elsevier, Oxford, 2011.
- [3] J. Fish, G.J. Wagner, S. Keten, Mesoscopic and multiscale modelling in materials, *Nat. Mater.* 20 (6) (2021) 774–786.
- [4] T.J.R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Courier Corporation, Mineola, NY, 2012.
- [5] P. Wriggers, *Nonlinear Finite Element Methods*, Springer Science & Business Media, Berlin, 2008.
- [6] H. Moulinec, P. Suquet, A numerical method for computing the overall response of nonlinear composites with complex microstructure, *Comput. Methods Appl. Mech. Eng.* 157 (1–2) (1998) 69–94.
- [7] W.H. Müller, Mathematical vs. experimental stress analysis of inhomogeneities in solids, *J. Phys. IV* 6 (C1) (1996) 139–148.
- [8] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.* 3 (6) (2021) 422–440.
- [9] J. Linghu, H. Dong, J. Cui, Ensemble wavelet-learning approach for predicting the effective mechanical properties of concrete composite materials, *Comput. Mech.* 70 (2) (2022) 335–365.
- [10] J. Linghu, W. Gao, H. Dong, Y. Nie, Higher-order multi-scale physics-informed neural network (HOMS-PINN) method and its convergence analysis for solving elastic problems of authentic composite materials, *J. Comput. Appl. Math.* 456 (2025) 116223.

- [11] Q. Chen, G. Chatzigeorgiou, F. Meraghni, X. Chen, Z. Yang, Physics-informed deep homogenization approach for random nanoporous composites with energetic interfaces, *Eng. Appl. Artif. Intell.* 140 (2025) 109735.
- [12] S.A. Niaki, E. Haghighat, T. Campbell, A. Poursartip, R. Vaziri, Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture, *Comput Methods Appl Mech Eng* 384 (2021) 113959.
- [13] C.S. Ha, et al., Rapid inverse design of metamaterials based on prescribed mechanical behavior through machine learning, *Nat. Commun.* 14 (1) (2023) 5765.
- [14] L. Zheng, K. Karapiperis, S. Kumar, D.M. Kochmann, Unifying the design space and optimizing linear and nonlinear truss metamaterials by generative modeling, *Nat. Commun.* 14 (1) (2023) 7563.
- [15] H. Li, et al., Clustering discretization methods for generation of material performance databases in machine learning and design optimization, *Comput. Mech.* 64 (2019) 281–305.
- [16] J. Wu, O. Sigmund, J.P. Groen, Topology optimization of multi-scale structures: a review, *Struct. Multidiscip. Optim.* 63 (2021) 1455–1480.
- [17] J. Fish, *Practical Multiscale Modeling*, John Wiley & Sons, Chichester, UK, 2013.
- [18] A. Clément, C. Soize, J. Yvonnet, Uncertainty quantification in computational stochastic multiscale analysis of nonlinear elastic materials, *Comput. Methods Appl. Mech. Eng.* 254 (2013) 61–82.
- [19] F. Feyel, J.-L. Chaboche, FE² multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials, *Comput. Methods Appl. Mech. Eng.* 183 (3–4) (2000) 309–330.
- [20] N.B. Kovachki, S. Lanthaler, A.M. Stuart, Operator learning: algorithms and analysis, [arXiv preprint arXiv:2402.15715](https://arxiv.org/abs/2402.15715). (2024).
- [21] Z. Li, N.B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, in: *International Conference on Learning Representations*, 2021. <https://openreview.net/forum?id=c8P9NQVtmnO>.
- [22] J. Guibas, M. Mardani, Z. Li, A. Tao, A. Anandkumar, B. Catanzaro, Adaptive fourier neural operators: Efficient token mixers for transformers, [arXiv preprint arXiv:2111.13587](https://arxiv.org/abs/2111.13587). (2021).
- [23] G. Wen, Z. Li, K. Azizzadenesheli, A. Anandkumar, S.M. Benson, U-FNO—an enhanced fourier neural operator-based deep-learning model for multiphase flow, *Adv. Water Resour.* 163 (2022) 104180.
- [24] V.S. Fanaskov, I.V. Oseledets, Spectral neural operators, in: *Doklady Mathematics*, 108, Springer, 2023, pp. S226–S232.
- [25] H. You, Q. Zhang, C.J. Ross, C.-H. Lee, Y. Yu, Learning deep implicit fourier neural operators (IFNOs) with applications to heterogeneous material modeling, *Comput. Methods Appl. Mech. Eng.* 398 (2022) 115296.
- [26] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via deepONet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* 3 (3) (2021) 218–229.
- [27] S. Wang, H. Wang, P. Perdikaris, Improved architectures and training algorithms for deep operator networks, *J. Sci. Comput.* 92 (2) (2022) 35.
- [28] S. Wang, H. Wang, P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed deepONets, *Sci. Adv.* 7 (40) (2021) eabi8605.
- [29] S. Wang, P. Perdikaris, Long-time integration of parametric evolution equations with physics-informed deepONets, *J. Comput. Phys.* 475 (2023) 111855.
- [30] J. Seidman, G. Kissas, P. Perdikaris, G.J. Pappas, NOMAD: nonlinear manifold decoders for operator learning, *Adv. Neural Inf. Process. Syst.* 35 (2022) 5601–5613.
- [31] M. Zhu, S. Feng, Y. Lin, L. Lu, Fourier-deepONet: fourier-enhanced deep operator networks for full waveform inversion with improved accuracy, generalizability, and robustness, *Comput. Methods Appl. Mech. Eng.* 416 (2023) 116300.
- [32] S. Goswami, M. Yin, Y. Yu, G.E. Karniadakis, A physics-informed variational deepONet for predicting crack path in quasi-brittle materials, *Comput. Methods Appl. Mech. Eng.* 391 (2022) 114587.
- [33] M.J. Buehler, Modeling atomistic dynamic fracture mechanisms using a progressive transformer diffusion model, *J. Appl. Mech.* 89 (12) (2022) 121009.
- [34] E.L. Buehler, M.J. Buehler, End-to-end prediction of multimaterial stress fields and fracture patterns using cycle-consistent adversarial and transformer neural networks, *Biomed. Eng. Adv.* 4 (2022) 100038.
- [35] A.J. Lew, M.J. Buehler, Single-shot forward and inverse hierarchical architected materials design for nonlinear mechanical properties using an attention-diffusion model, *Mater. Today* 64 (2023) 10–20.
- [36] M.J. Buehler, MeLM, a generative pretrained language modeling framework that solves forward and inverse mechanics problems, *J. Mech. Phys. Solids* 181 (2023) 105454.
- [37] M. Yang, Y. Wen, W. Chen, Y. Chen, K. Jia, Deep optimized priors for 3D shape modeling and reconstruction, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3269–3278.
- [38] Y. Chen, T. Dodwell, T. Chuaqui, R. Butler, Full-field prediction of stress and fracture patterns in composites using deep learning and self-attention, *Eng. Fract. Mech.* 286 (2023) 109314.
- [39] A. Bhaduri, A. Gupta, L. Graham-Brady, Stress field prediction in fiber-reinforced composite materials using a deep learning approach, *Compos. Part B* 238 (2022) 109879.
- [40] M.M. Rashid, T. Pittie, S. Chakraborty, N.M.A. Krishnan, Learning the stress-strain fields in digital composites using fourier neural operator, *iScience* 25 (11) (2022) 105452.
- [41] M.M. Rashid, S. Chakraborty, N.M.A. Krishnan, Revealing the predictive power of neural operators for strain evolution in digital composites, *J. Mech. Phys. Solids* 181 (2023) 105444.
- [42] A. Gupta, A. Bhaduri, L. Graham-Brady, Accelerated multiscale mechanics modeling in a deep learning framework, *Mech. Mater.* 184 (2023) 104709.
- [43] I. Saha, A. Gupta, L. Graham-Brady, Prediction of local elasto-plastic stress and strain fields in a two-phase composite microstructure using a deep convolutional neural network, *Comput. Methods Appl. Mech. Eng.* 421 (2022) 116816.
- [44] Z. Hashin, S. Shtrikman, A variational approach to the theory of the elastic behaviour of multiphase materials, *J. Mech. Phys. Solids* 11 (2) (1963) 127–140.
- [45] R. Hill, A self-consistent mechanics of composite materials, *J. Mech. Phys. Solids* 13 (4) (1965) 213–222.
- [46] E. Kröner, *Statistical Continuum Mechanics*, Springer International Publishing, Vienna, 1971.
- [47] R. Hill, Elastic properties of reinforced solids: some theoretical principles, *J. Mech. Phys. Solids* 11 (5) (1963) 357–372.
- [48] J.-C. Michel, P. Suquet, Nonuniform transformation field analysis, *Int. J. Solids Struct.* 40 (25) (2003) 6937–6955.
- [49] J. Yvonnet, *Computational Homogenization of Heterogeneous Materials with Finite Elements*, Springer International Publishing, Cham, 2019.
- [50] T.I. Zohdi, P. Wriggers, *An Introduction to Computational Micromechanics*, Springer, Berlin, 2005.
- [51] A.R. Melro, P.P. Camanho, S.T. Pinho, Generation of random distribution of fibres in long-fibre reinforced composites, *Compos. Sci. Technol.* 68 (9) (2008) 2092–2102.
- [52] J. Yi, M.A. Bessa, Rvesimulator: an automated representative volume element simulator for data-driven material discovery, in: *AI for Accelerated Materials Design - NeurIPS 2023 Workshop*, 2023. <https://openreview.net/forum?id=511z1DGjPi>.
- [53] K. Chen, C. Wang, H. Yang, Deep operator learning lessens the curse of dimensionality for PDEs, [arXiv preprint arXiv:2301.12227](https://arxiv.org/abs/2301.12227). (2023).
- [54] Z. Li, W. Peng, Z. Yuan, J. Wang, Fourier neural operator approach to large eddy simulation of three-dimensional turbulence, *Theor. Appl. Mech. Lett.* 12 (6) (2022) 100389.
- [55] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale, [arXiv preprint arXiv:2010.11929](https://arxiv.org/abs/2010.11929). (2020).
- [56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [57] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, T. Liu, On layer normalization in the transformer architecture, in: *International Conference on Machine Learning*, PMLR, 2020, pp. 10524–10533.
- [58] S. Wang, J.H. Seidman, S. Sankaran, H. Wang, G.J. Pappas, P. Perdikaris, CViT: Continuous Vision Transformer for Operator Learning, [arXiv preprint arXiv:2405.13998](https://arxiv.org/abs/2405.13998). (2024).
- [59] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, A. Geiger, Convolutional occupancy networks, in: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, Springer, 2020, pp. 523–540.

- [60] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, T. Funkhouser, et al., Local implicit grid representations for 3D scenes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6001–6010.
- [61] I. Mehta, M. Gharbi, C. Barnes, E. Shechtman, R. Ramamoorthi, M. Chandraker, Modulated periodic activations for generalizable local functional representations, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 14214–14223.
- [62] E.A. Nadaraya, On estimating regression, *Theory Prob. Appl.* 9 (1) (1964) 141–142.
- [63] G.S. Watson, Smooth regression analysis, *Sankhyā Indian J. Stat. Ser. A* (1964) 359–372.
- [64] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, A. Courville, On the spectral bias of neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 5301–5310.
- [65] Y. Wang, X. Li, Z. Yan, Y. Du, J. Bai, B. Liu, T. Rabczuk, Y. Liu, HomoGenius: a foundation model of homogenization for rapid prediction of effective mechanical properties using neural operators, *arXiv preprint arXiv:2404.07943*. (2024).
- [66] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, part III 18, Springer, 2015, pp. 234–241.
- [67] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16X16 words: transformers for image recognition at scale, in: International Conference on Learning Representations, 2021. <https://openreview.net/forum?id=YicbFdNTTy>.
- [68] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv:1412.6980* (2014).
- [69] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, *arXiv preprint arXiv:1711.05101*. (2017).
- [70] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, C.D. de Las, L.A. Hendricks, J. Welbl, A. Clark, et al., Training compute-optimal large language models, *arXiv preprint arXiv:2203.15556*. (2022).
- [71] J. Kaplan, S. McCandlish, T. Henighan, T.B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling laws for neural language models, *arXiv preprint arXiv:2001.08361*. (2020).
- [72] X. Zhai, A. Kolesnikov, N. Houlsby, L. Beyer, Scaling vision transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12104–12113.
- [73] I. Alabdulmohsin, X. Zhai, A. Kolesnikov, L. Beyer, Getting vit in shape: scaling laws for compute-optimal model design, Thirty-seventh Conference on Neural Information Processing Systems (2023) <https://openreview.net/forum?id=en4LGxpD9E>.
- [74] J. Yvonnet, Computational Homogenization of Heterogeneous Materials with Finite Elements, 258, Springer, Cham, 2019.
- [75] A.R. Melro, P.P. Camanho, F.M. Andrade Pires, S.T. Pinho, Micromechanical analysis of polymer composites reinforced by unidirectional fibres: Part II-micromechanical analyses, *Int. J. Solids Struct.* 50 (2012) 1906–1915.
- [76] P.D. Soden, M.J. Hinton, A.S. Kaddour, Lamina properties, lay-up configurations and loading conditions for a range of fibre reinforced composite laminates, *Compos. Sci. Technol.* 58 (1998) 1011–1022.
- [77] B.T. Werner, I.M. Daniel, Characterization and modeling of polymeric matrix under multi-axial static and dynamic loading, *Compos. Sci. Technol.* 102 (2014) 113–119.
- [78] T.J. Vaughan, C.T. McCarthy, Micromechanical modelling of the transverse damage behaviour in fibre reinforced composites, *Compos. Sci. Technol.* 71 (2011) 388–396.
- [79] C. He, J. Ge, B. Zhang, J. Gao, S. Zhong, W.K. Liu, D. Fang, A hierarchical multiscale model for the elastic-plastic damage behavior of 3D braided composites at high temperature, *Compos. Sci. Technol.* 196 (2020) 108230.
- [80] A.G. Izard, J. Bauer, C. Crook, V. Turlo, L. Valdevit, Ultrahigh energy absorption multifunctional spinodal nanoarchitectures, *Small* 15 (45) (2019) 1903834.
- [81] M.-T. Hsieh, M.R. Begley, L. Valdevit, Architected implant designs for long bones: advantages of minimal surface-based topologies, *Mater. Des.* 207 (2021) 109838.
- [82] C.M. Portela, B.W. Edwards, D. Veyssat, Y. Sun, K.A. Nelson, D.M. Kochmann, J.R. Greer, Supersonic impact resilience of nanoarchitected carbon, *Nat. Mater.* 20 (11) (2021) 1491–1497.
- [83] L. Chen, J. Shen, Applications of semi-implicit fourier-spectral method to phase field equations, *Comput Phys Commun* 108 (2–3) (1998) 147–158.
- [84] Y.V. Obnosov, Periodic heterogeneous structures: new explicit solutions and effective characteristics of refraction of an imposed field, *SIAM J. Appl. Math.* 59 (1999) 1267–1287.
- [85] C. Bellis, H. Moulinec, P. Suquet, Eigendecomposition-based convergence analysis of the Neumann series for laminated composites and discretization error estimation, *Int. J. Numer. Methods Eng.* 121 (2020) 201–232.
- [86] S. Wang, S. Sankaran, H. Wang, P. Perdikaris, An expert’s guide to training physics-informed neural networks, *arXiv preprint arXiv:2308.08468*. (2023).
- [87] J. Bradbury, R. Frostig, P. Hawkins, M.J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: composable transformations of Python + NumPy programs, 2018. <http://github.com/google/jax>.
- [88] J.D. Hunter, Matplotlib: a 2D graphics environment, *Comput. Sci. Eng.* 9 (03) (2007) 90–95.
- [89] C.R. Harris, K.J. Millman, S.J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, et al., Array programming with NumPy, *Nature* 585 (7825) (2020) 357–362.
- [90] T. Mura, *Micromechanics of Defects in Solids*, Springer International Publishing, Dordrecht, 1982.
- [91] S. Li, G. Wang, *Introduction to Micromechanics and Nanomechanics*, World Scientific, Singapore, 2008.
- [92] F. Willot, Fourier-based schemes for computing the mechanical response of composites with accurate local fields, *Comptes Rendus Mécanique* 343 (3) (2015) 232–245.
- [93] E.M. Stein, R. Shakarchi, *Fourier Analysis: An Introduction*, Princeton University Press, Princeton, New Jersey, 2003.
- [94] T.-R. Liu, Y. Yang, O.R. Bacarrezza, S. Tang, M.H. Aliabadi, An extended full field self-consistent cluster analysis framework for woven composite, *Int. J. Solids Struct.* 281 (2023) 112407.
- [95] M. Schneider, A review of nonlinear FFT-based computational homogenization methods, *Acta Mech.* 232 (6) (2020) 2051–2100.
- [96] L. Gélébart, A modified FFT-based solver for the mechanical simulation of heterogeneous materials with Dirichlet boundary conditions, *Comptes Rendus Mécanique* 348 (8–9) (2020) 693–704.
- [97] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: graph kernel network for partial differential equations, *arXiv preprint arXiv:2003.03485*. (2020).
- [98] J. Gao, S. Mojumder, W. Zhang, H. Li, D. Suarez, C. He, J. Cao, W.K. Liu, Concurrent n-scale modeling for non-orthogonal woven composite, *Comput. Mech.* 70 (2022) 853–866.