Contents lists available at ScienceDirect

Materials Today Communications

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

Neural network-driven framework for efficient microstructural modeling of particle-enriched composites

Shishir Barai [a], Feihong Liu [a], Manik Kumar [a], Christian Peco [a,b,*]

[a] Department of Engineering Science and Mechanics, Pennsylvania State University, University Park, State College, PA, 16802, USA
[b] Institute for Computational and Data Sciences, Pennsylvania State University, University Park, State College, PA, 16802, USA

ARTICLE INFO

Keywords:
Autoencoders
Dimensionality reduction
Finite element analysis
High performance computing
Microstructure optimization

ABSTRACT

Simulating materials with complex, variable microstructures like nanoparticle-enriched matrices presents significant challenges for FEM, particularly in defining constitutive equations across diverse loading conditions and internal structures. These complexities often make purely physics-based FEM simulations impractical for larger macroscopic domains. We present a novel approach that significantly reduces the dimensionality of the FEM data, while preserving the ability to interpolate and extrapolate across various nanoparticle configurations. Our method combines raw data projection, neural network surrogates, and convolutional autoencoders to geometrically normalize the data in a reduced-dimensional space. This reduction enables efficient training and prediction, while retaining flexibility for different microstructural variations. Compared to direct training on unprocessed data, our approach reduces dimensionality and computational costs without sacrificing accuracy, allowing robust virtual testing of particle-enriched composites. This framework can be applied to other multiscale composite systems for optimized material design.

1. Introduction and background

The optimization of advanced composite materials holds immense promise for enabling transformative technologies in structural, energy, biomedical, and other critical application domains [1,2]. For instance, the development of lightweight structural composites with tailored microstructures is important for enhancing the fuel efficiency and performance of aerospace components [2]. In the biomedical field, the controlled design of microstructures in scaffolds plays a vital role in tissue engineering [3,4]. However, realizing their full promise hinges upon tailored optimization of complex internal microstructures governing functional performance.

Physics-based computational modeling then becomes invaluable in elucidating the multiscale behaviors and complex interactions within these heterogeneous composites, either synthetic or biological [7–9]. The finite element method (FEM) in particular has been widely applied for understanding the localized phenomena and advancing materials characterization [10–13]. Microstructural details play a vital role in governing the functional behaviors of these composites. The critical role of microstructural characteristics in dictating the functional performance of composites underscores the need for their optimization. This optimization is crucial not only for achieving desired material properties but also for pioneering tailored designs that meet specific application requirements. However, multiscale modeling faces challenges when addressing composites with variable microstructures at each point. These complexities arise due to the need to accurately represent varieties of material behaviors and interactions at different scales, from the macroscopic down to the atomic level. Traditional FEM and similar numerical methods, while providing detailed insights, struggle with the computational intensity required to simulate each distinct microstructure across the material [14,15]. The necessity to compute separate responses for varied microstructures introduces severe computational demands, limiting the scalability and efficiency of these methods [16–19]. Consider the recently developed cryo-ultrasonic testing (Cryo-UT) [5] (Fig. 1a), where metal parts are encapsulated in ice to address the shortcomings of traditional immersion testing methods. The incorporation of metallic particles results in more refined polycrystalline ice microstructures which can be illustrated by Fig. 1b. However, the freezing process is affected by multiple factors, such as the particle size and fraction of colloidal [20,21] leading to various representative configurations and high-dimensional microstructural data. These complex microstructural characteristics significantly influence the overall functional behavior. Similarly, in the design of short fiber-reinforced polymer composites (SFRPCs) for aerospace, and automotive applications, the intricate microstructural characteristics, comprising fiber orientation, volume, and distribution influence the

* Corresponding author.
E-mail address: christian.peco@psu.edu (C. Peco).

https://doi.org/10.1016/j.mtcomm.2024.111278
Received 8 October 2024; Received in revised form 4 December 2024; Accepted 9 December 2024
A v a i loanbll1e6Dee c e m2b0e2r4
2 3 5 2 - 4 9 2 2 0 8 2 E 4 0 s e l v t i e r l i g a n t , s ' - ' , i / n ' c & ! u t d h i o n s g e r e a n d d a t a +) A + l (! %) a n d (m i t l e a c h n o l o g i e s .

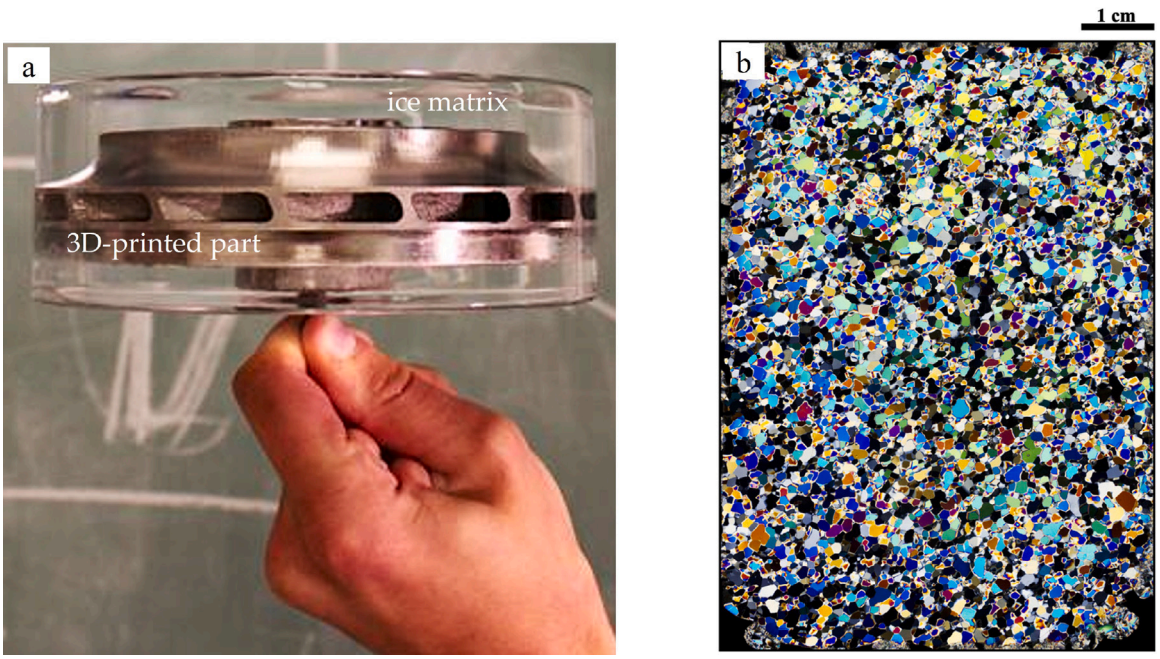

Fig. 1. (a) Cryo-ultrasonic: A Ti-6AL-4V shrouded impeller surrounded by a block of clear ice [5], (b) Cross-polarized optical micrograph of polycrystalline ice [6].

composite material's overall functional behavior [22]. The resulting high-dimensional data, along with the intricate interactions between microstructural features, presents significant computational challenges for FEM simulations. Accurately capturing these complex relationships between microstructural details and functional responses poses difficulties for traditional modeling approaches [23–26]. Strategies that can accurately model the governing physics relating microstructures to responses without needing repeated explicit simulation offer pathways to significantly enhance the feasibility.

In this context, we introduce machine learning and deep learning methodologies that seek not merely to alleviate computational cost in multiscale modeling, but to introduce new strategies in composite design. Using neural networks, high-dimensional microstructure datasets can be effectively encoded and analyzed, circumventing the computational bottlenecks associated with conventional methods, some of them requiring quadrature point-wise simulation in heterogeneous microstructure layouts. Various machine-learning (ML) approaches have demonstrated promise in serving as efficient surrogates for complex computational tasks. For instance, regression models have been effectively utilized to predict alloy energies across diverse systems, offering insights with accuracy comparable to *ab initio* methods [27,28]. Deep learning has emerged as a versatile tool for identifying intricate relationships amid high-parametric spaces associated with microstructures [29,30]. Additionally, physics-informed neural networks incorporate conserved physical principles to enhance deep learning for computational mechanics [31]. These approaches train on simulation data, encoding complex phenomena into model representations. Deep neural networks have also been integrated with traditional numerical methods like finite element analyses to couple deep learning and first-principles-based techniques [32–34]. Specifically, physics-informed neural networks have been developed to approximate micromechanical behaviors in finite element analyses, showcasing their potential to significantly enhance computational efficiency in multiscale modeling [35]. Tailored machine learning pipelines further enable accelerated material characterization by directly linking design parameters to desired responses, effectively bypassing the need for exhaustive simulations [36,37]. Particularly, the work by Maia et al. [35] underscores the adoption of machine learning for accelerating multiscale finite element analysis. By employing surrogate models that approximate micromechanical behavior, this approach marks a significant leap forward, offering a pathway to overcome traditional computational bottlenecks and streamline the optimization of composite materials.

Nevertheless, the training of such surrogate models introduces a new set of challenges. Some of these challenges include the structure and dimensionality of the data set, which generation and purpose are tailored to FEM frameworks and not that of ML. Among deep learning tools, autoencoders have emerged as a powerful tool for learning compressed representations of high-dimensional data [38,39]. Various studies have demonstrated the effectiveness of convolutional autoencoders (CAE) for reducing complex microstructures and deriving deep material property descriptors to lower the dimensionality in physics-based simulations [40–42]. Other than autoencoder, Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) methods have also shown effectiveness in dimensionality reduction of high dimensional material data [43–47]. While PCA has been extensively used for dimensionality reduction, its effectiveness is limited to linear feature spaces [48]. Nonlinear techniques like t-SNE can capture more complex data relationships [49]. However, t-SNE predominantly serves an exploratory visualization role rather than enabling the reconstruction of original inputs from reduced dimensions. On the other hand, autoencoders, particularly convolutional autoencoders (CAE), excel at learning nonlinear embeddings from microstructure data, offering better feature preservation and reconstruction capabilities [50]. Rather than relying on matrix factorization of the data covariance, autoencoders learn deep nonlinear transformations using neural network encoding and decoding stages. By projecting microstructure details and physics-based simulation data into tailored compact latent spaces, autoencoders can capture essential nonlinear relationships lost by linear PCA. Additionally, the decoder network of the autoencoders empowers reconstruction and analysis from the compressed representations, unlike visual-focused algorithms like t-SNE. Overall, the versatile dimensionality reduction and generation capabilities make autoencoders well-suited for microstructure optimization.

Recent advancements in the application of autoencoders to finite element analysis and microstructure modeling have shown promising results. For instance, Hu et al. [51] demonstrated the use of a convolutional neural network (CNN) and an autoencoder to predict the mode shape of bridges based on a large mode shape database generated by FEM. The autoencoder was employed to encode the

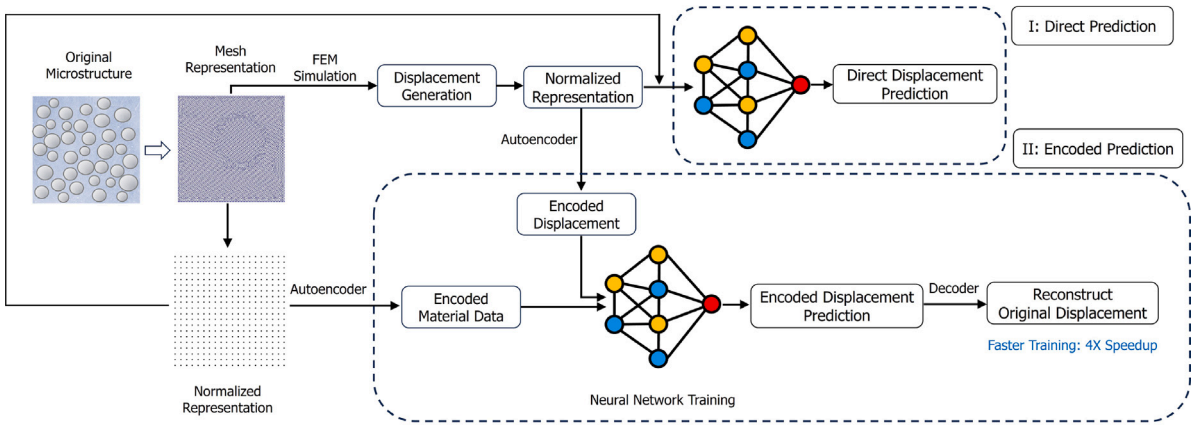

Fig. 2. Schematic overview of the proposed accelerated hybrid deep learning framework for microstructure response prediction.

mode shape tensor into a low-dimensional latent space representation, enabling efficient training of the CNN. This approach, while effective for lower order mode shapes, still faces challenges in predicting higher-order mode shapes and may not fully replace traditional finite element methods. Similarly, Kohar et al. [52] presented a framework combining 3D CNN autoencoders with long short-term memory neural networks (LSTM-NN) to process FE model data for predicting the force–displacement response and mesh deformation in crashworthiness applications. The autoencoder was used to automatically determine relevant features from the unstructured FE data in an unsupervised manner, demonstrating its ability to handle complex FEM data. The application of autoencoders has also extended to stress-state prediction, as demonstrated by Lee et al. [53]. They introduced a domain-adaptive convolutional variational autoencoder (VAE) for stress-state prediction. The approach aimed to transfer knowledge from a source domain of FE analysis data to a target domain of real-world test specimen images, enabling accurate predictions in both domains. This study highlighted the potential of autoencoders in addressing the scarcity of labeled data in material mechanics applications. Furthermore, Kim et al. [54] employed a VAE to generate a continuous microstructure space based on synthetic microstructural images. Their work showcased the potential of combining autoencoders with Gaussian process regression for predicting mechanical properties from microstructural features. Autoencoders have also been applied to predict fracture patterns in highly nonlinear brittle fracture problems. Shinde et al. [50] developed a deep learning framework based on an autoencoder neural network to construct a nonlinear reduced-order model. The autoencoder efficiently compressed highly nonlinear data, enabling the prediction of crack propagation patterns directly from loading conditions. This study showcased the capabilities of autoencoders in handling complex fracture mechanics problems. More recently, Koopas et al. [55] introduced a microstructure-embedded autoencoder approach for reconstructing high-resolution solution field data from a reduced parametric space, significantly improving computational efficiency. Their method, which incorporates microstructural information into the autoencoder architecture, showed superior performance in upscaling low-fidelity solutions to high fidelity while preserving critical details at sharp interfaces.

While these studies have made significant advancements in applying autoencoders to finite element analysis and microstructure modeling, there remains a gap in the application of these techniques to particle-reinforced composites. Connecting a particular microstructure to its response in nanoparticle-reinforced composites presents a significant challenge due to the complexity of the system and its spatial variability. Moreover, the treatment of raw FEM geometrical data to streamline its usage as a surrogate block in a multiscale model, in contrast with traditional experimental images, poses additional problems.

In this study, we tackle the aforementioned challenges by employing a methodology that leverages neural networks and deep learning to rapidly predict the microstructure response. Our approach involves generating an encoded surrogate model exploiting the FEM data structure, which resides entirely within the simulation space. We start by preprocessing raw FEM data geometry and displacement fields by projecting them onto a structured, normalized sampling space. Subsequently, we introduce convolutional autoencoders to reduce the dimensionality of high-fidelity microstructure datasets, targeting the accelerated simulation-based design of microstructures. This process involves compressing high-fidelity simulation data, encompassing both microstructure representations and displacement solutions. Following this, a separate predictive model, trained on the encoded data, is utilized to anticipate compressed displacement fields from these encoded microstructures. Fig. 2 illustrates the overall methodology employed in this study, from the generation of complex mesh structures to the prediction of displacement fields. The illustration encompasses both the direct predictive modeling technique as well as the accelerated framework leveraging convolutional autoencoders. By demonstrating an approach to overcome key computational barriers, this work aims to pave the way for optimized, tailored improvements via rapid virtual testing of microstructure–property relationships. This could lead to faster development of advanced materials for aerospace, automotive, and other critical industries where tailored microstructures are crucial in achieving desired material properties. The proposed framework has the potential to greatly accelerate simulation-driven discovery by reducing computational complexity.

The remainder of the manuscript is organized as follows. In Section 2, we present the continuum modeling methodologies that underpin our study, detailing the process of finite element discretization. In Section 3, we focus on the operational dynamics of the neural network modeling employed. Section 4 is dedicated to presenting our numerical findings, encompassing analysis of the simulation outcomes, evaluating the efficacy of autoencoder acceleration in predicting mechanical responses, and investigating the impact of CPU core parallelization on model performance. We draw our paper to a close with Section 5, which encapsulates our conclusions, findings, and outlines prospective directions for future research.

2. Data generation: microstructure and continuum modeling

This section describes the methodology for generating the dataset used in this study. It involves creating synthetic microstructure samples of two-phase composites and performing FEM to simulate their mechanical behavior under a specific loading condition. The subsections detail the processes of microstructure representative volume element (RVE) generation and finite element formulation, providing the data for training the machine learning model.

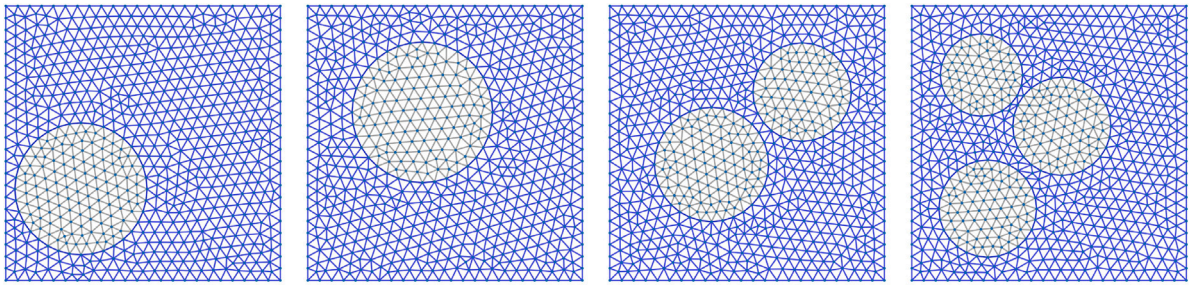

Fig. 3. A subset of 400 microstructure samples with varying inclusion sizes and locations. The meshes shown are coarsened for visualization purposes; smaller and more densely packed mesh elements were used in the actual simulations to accurately capture the microstructural behaviors.

2.1. Microstructure RVE generation

While several approaches are focused on treating real microstructure imaging (obtained, for example, from experimental techniques like micro-computed tomography scans), this study emphasizes the synthetic sample environment. Such an approach is used in engineering to explore a diverse range of possible geometries, and its raw data structures are typically linked to the virtual framework in which they will be used (i.e., FEM in our case [56]). We generate two-phase composites, simulating an enriched nanoparticle medium, comprising a matrix phase and circular particle inclusion regions, as depicted in Fig. 3. To ensure consistency and facilitate comparison across samples, a unit square $[0,1]\times[0,1]$ domain was utilized for all microstructure geometries. Randomized variations in the locations and sizes of the inclusions were generated across numerous geometry structures to mimic realistic microstructural configurations where inclusion phase area fractions varied between 13% and 39%. To increase variability and sample characteristics, we also generated RVEs with multiple inclusions, ranging from two to four inclusions of different sizes and positions. The matrix and inclusions were assigned specific mechanical properties to reflect the material heterogeneity. This methodology allows us to consider these matrix-inclusion composites as a benchmark for evaluating complex microstructures and their mechanical responses.

We used a custom Python script to generate the two-phase microstructures. The generated microstructures were discretized into finite element meshes using *Gmsh*, a widely recognized open-source finite element meshing software [57]. We employed the non-structured mesh scheme for all generated geometries of microstructure. This mesh scheme provides flexibility in ensuring the mesh resolution and capturing of material interfaces. The Delaunay triangulation algorithm was employed, connecting nodal topology by 3-node triangle elements. An element size factor of 1 and a well-refined characteristic spacing length were utilized to ensure a quasi-uniformed discretization as illustrated by Fig. 3. The mesh element size can vary depending on the specific problem and loading conditions. For the current study, a mesh element size factor of 1 was found to effectively capture the displacements with good accuracy for the specific loading and microstructure configurations analyzed, while maintaining computational feasibility. Problems involving more complex microstructure geometries, higher-stress gradients, or different loading scenarios could benefit from finer mesh resolutions as high-quality simulations with refined meshes can help improve the accuracy of both the FEM simulations and the machine learning models. With the microstructure geometries precisely represented, these discretized models were then used in finite element modeling to collect the data on mechanical responses, providing the foundation for the subsequent deep learning strategies.

2.2. Finite element formulation

After acquiring the representative microstructures, the FEM is employed to model and generate the dataset of the displacement field, denoted as $u_i = u_i(\mathbf{x})$, where i represents the i-th component of the field. We denote the computational domain of the microstructures as Ω, with Dirichlet and Neumann boundaries represented by Γ_{g_i} and Γ_{h_i}, respectively. The strong form of the equilibrium in the continuum can be formulated as:

$$\begin{aligned} &\sigma_{ij,j} + l_i = 0, \text{ in } \Omega, \\ &u_i = g_i, \text{ on } \Gamma_{g_i}, \\ &\sigma_{ij} n_j = h_i, \text{ on } \Gamma_{h_i}, \end{aligned} \tag{1}$$

where $\sigma_{ij} = \sigma_{ij}(u_i)$ represents the Cauchy stress; l_i denotes the body force; g_i and h_i specify the values at the essential and natural boundaries, respectively; and n_i indicates the normal direction vector at the natural boundary. In this study, we assume the body force l_i and surface force h_i are zero. The modeled matrix-inclusion microstructure, specifically under a uniaxial tension state with essential boundary conditions applied, is depicted in Fig. 4a. In addition to this example, simulations encompassing multiple inclusions and subjected to various loading conditions, such as biaxial tension and uniaxial compression, have also been conducted. After applying the FEM discretization, we aim to solve for and obtain a deformed displacement field $u_i(\mathbf{x})$, as exhibited in Fig. 4b.

With the essential boundary conditions considered, the finite element solutions search trial solutions $u_i \in \mathcal{U}$ ($\mathcal{U} = \{u_i \mid u_i \in \mathcal{H}^1,\ u_i = g_i \text{ on } \Gamma_{g_i}\}$), such that for all test functions $w_i \in \mathcal{V}$ ($\mathcal{V} = \{w_i \mid w_i \in \mathcal{H}^1,\ w_i = 0 \text{ on } \Gamma_{g_i}\}$)

$$\int_\Omega w_{(i,j)} \sigma_{ij} d\Omega = 0. \tag{2}$$

Note that the body force l_i and surface force h_i are zero, and $(\cdot)_{(i,j)} = \frac{1}{2}[(\cdot)_{i,j} + (\cdot)_{j,i}]$. This paper discusses the linear elasticity problem, thus the Cauchy stress tensor can be expressed through the generalized Hooke's law by $\sigma_{ij} = c_{ijkl}\epsilon_{kl}$ where $\epsilon_{kl} = u_{(k,l)}$ is the infinitesimal strain tensor.

Using finite-dimensional approximation, we search the approximated u_i and w_i, expressed by $u_i^h \in \mathcal{U}^h$ ($\mathcal{U}^h \subset \mathcal{U}$) and $w_i^h \in \mathcal{V}^h$ ($\mathcal{V}^h \subset \mathcal{V}$) where u_i^h can be decomposed by

$$u_i^h = v_i^h + g_i^h, \tag{3}$$

where $v_i^h \in \mathcal{V}^h$ and g_i^h completes the approximation with the Dirichlet boundary conditions. Consider the finite element shape function $N_j(\mathbf{x})$ (short by $N_{\mathbf{x}_j}(\mathbf{x})$), we have

$$v_i^h(\mathbf{x}) = \Sigma_{j=1}^{nn} N_j(\mathbf{x}) d_i(\mathbf{x}_j), \tag{4}$$

$$g_i^h(\mathbf{x}) = \Sigma_{j=1}^{nn} N_j(\mathbf{x}) g_i(\mathbf{x}_j), \tag{5}$$

where nn denotes the number of nodes connected by an element, $d_i(\mathbf{x})$ refers to the nodal displacements, and $g_i(\mathbf{x})$ corresponds to the imposed displacements (Dirichlet boundary conditions). This paper employs the 3-node triangular element as $\mathbf{x} \in \mathbb{R}^2$ for all numerical tests, which is given by

$$N_i(x,y) = \frac{1}{2A}(a_i + b_i x + c_i y),\ i = 1,\ 2,\ 3, \tag{6}$$

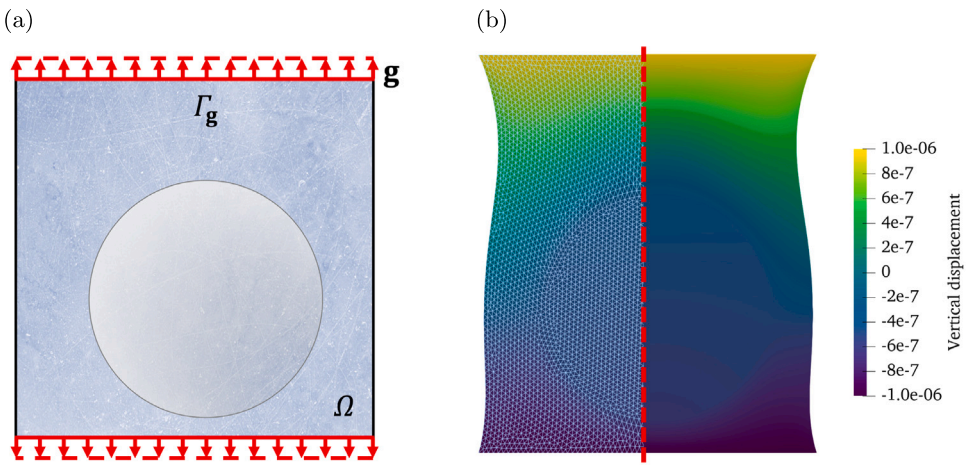

Fig. 4. Microstructure under the vertical uni-axial tension: (a) Undeformed microstructure; (b) Vertical displacement $u_2(\mathbf{x})$.

where area $A = \frac{1}{2}(x_1(y_2 - y_3) + x_1(y_3 - y_2) + x_3(y_1 - y_2))$ and coefficients

$$\begin{aligned} a_i &= x_j y_k - x_k y_j, \\ b_i &= y_j - y_k, \\ c_i &= x_k - x_j. \end{aligned} \tag{7}$$

The index above follows the cyclic permutation: $(i, j, k) \rightarrow (j, k, i) \rightarrow (k, i, j)$. The Bubnov–Galerkin equation then goes to

$$\int_{\Omega} w^h_{(i,j)} c_{ijkl} v^h_{(k,l)} d\Omega = -\int_{\Omega} w^h_{(i,j)} c_{ijkl} g^h_{(k,l)} d\Omega. \tag{8}$$

Eq. (8) thus can be written in the matrix form

$$\mathbf{Kd} = \mathbf{F}, \tag{9}$$

where **K**, **F**, and **d** are global stiffness matrix, force vector, and displacement vector, respectively. The global matrix (vector) assembling can be expressed by

$$\mathbf{K} = \Sigma_{e=1}^{ne}(\mathbf{k}^e),\ \mathbf{k}^e = [k^e_{pq}],\ k^e_{pq} = \mathbf{e}_i^T \int_{\Omega_e} \mathbf{B}_a^T \mathbf{D} \mathbf{B}_b d\Omega \mathbf{e}_j, \tag{10}$$

$$\mathbf{F} = \Sigma_{e=1}^{ne}(\mathbf{f}^e),\ \mathbf{f}^e = [f^e_p],\ f^e_p = -k^e_{pq} g_q, \tag{11}$$

where *ne* denotes the total number of elements. $\mathbf{e}_i$ is the basis vector. **D** is the elasticity matrix under Voigt notation. $\mathbf{B}_i$ is called the strain matrix with the form

$$\mathbf{B}_i = \begin{bmatrix} N_{i,1} & 0 & 0 \\ 0 & N_{i,2} & 0 \\ 0 & 0 & N_{i,3} \\ 0 & N_{i,3} & N_{i,2} \\ N_{i,3} & 0 & N_{i,1} \\ N_{i,2} & N_{i,1} & 0 \end{bmatrix}. \tag{12}$$

While the current work focuses on linear elasticity, this choice was made to efficiently generate the large dataset necessary for training machine learning models. The goal of this study is to demonstrate the feasibility of using surrogate modeling techniques like autoencoders to accelerate FEM simulations. Linear elasticity offers a reasonable approximation, allowing us to evaluate the effectiveness of our surrogate models.

In this study, all finite element computations were conducted using the MOOSE (Multiphysics Object Oriented Simulation Environment) framework, a versatile and open-source platform for modeling physics-based problems [58]. MOOSE supports large-scale parallel computations that efficiently facilitate finite element analysis, complex micromechanical investigations, and the generation of large-scale data sets. These features make it highly suitable for preparing large data sets for deep learning models.

3. Deep learning strategy for accelerated response prediction

Deep learning surrogate models present an alternative to the use of FEM hierarchical problem solving, leveraging offline FEM datasets to provide rapid predictions. However, even these surrogate models can become computationally expensive when dealing with high-dimensional data, such as meshes and FEM results associated with complex microstructures that change over very heterogeneous samples. To address this computational challenge, we introduce autoencoders that efficiently encode high-fidelity datasets into a lower-dimensional space while retaining essential information. This strategic encoding substantially streamlines the training process, minimizing computational costs and facilitating a significant acceleration in model development. Fig. 5 illustrates the schematic workflow of training a neural network model for deep learning applications. During training, inputs propagate forward through the network to generate output predictions. The optimization process then compares predictions against known targets to calculate loss, updating parameters through backpropagation to minimize this loss. This iterative gradient descent process allows the model to learn the complex relationships within the training data [59]. The following subsections focus on applying deep learning models to predict displacements and using autoencoders to accelerate the prediction process in microstructure optimization.

3.1. Deep learning model for displacement prediction

A hybrid deep neural network with convolutional (CNN) and fully connected dense layers is developed to predict displacement fields from input mesh microstructures. CNN layers extract localized features and patterns from the input data, while dense layers learn the complex input–output mapping for accurate prediction [60,61].

The heterogeneity in mesh sizes and shapes from finite element simulations introduces significant challenges in data preparation for machine learning. To address this, the raw simulation data first underwent processing into compatible input–output pairs. The mesh structures were projected onto a fixed sampling space of 120 × 120 and material properties were assigned to each point, serving as the input features to the machine learning model. These included Young's moduli and Poisson's ratios defined across the microstructures for the FEM simulations. For target outputs, nodal displacements were linearly interpolated onto the same fixed grid. This formatting ensures consistent spatial input material property and output displacement fields for each microstructure sample. For each grid point in the sampling space, if its position fell within the radius of inclusion, it was assigned Young's modulus and Poisson's ratio of the inclusion phase; otherwise, it was assigned the properties of the matrix phase. This

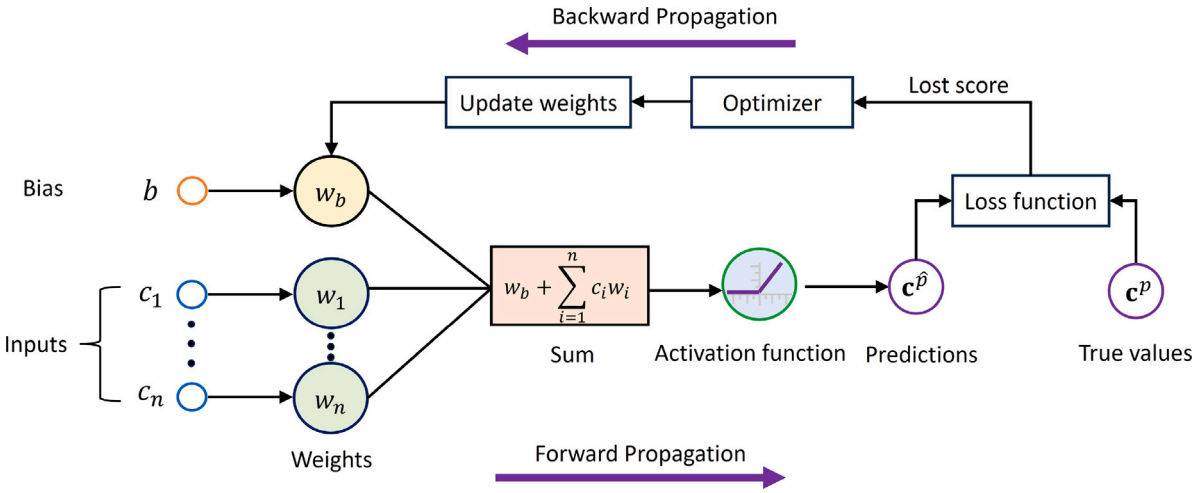

Fig. 5. Schematic diagram illustrating forward and backpropagation to update weights and minimize loss during neural network training. The diagram simplifies the model to one input and one output layer; the actual model comprises multiple hidden layers.

linear interpolation technique maintains the original material property distribution by mapping data from the finite element simulation to the structured grid using the exact location of each point relative to the inclusion boundaries. To project displacements, linear interpolation was used to map nodal displacements, which is consistent with the linear finite element formulation employed in the original simulation. For cases requiring higher-order geometries or displacement fields with nonlinear behavior, the interpolation degree could be increased to enhance accuracy. This approach accurately captures both material properties and displacement fields as long as the grid resolution is sufficiently high relative to the original mesh size to accurately capture the geometry of inclusions and the detailed behavior at the inclusion boundary. This 120×120 resolution was indeed necessary as smaller sampling spaces were found to lose important features at the sharpest matrix-inclusion interfaces when representing complex geometries with multiple inclusions. The processed inputs and outputs are then used to train deep neural networks for displacement prediction. This approach enabled efficient training of deep neural networks for surrogate modeling, simplifying complex physics into data-driven models to optimize materials.

The neural network models utilize non-linear activation functions which allow them to capture complex data patterns. For our neural network models, we chose ReLU (rectified linear unit) activation function for its ability to handle vanishing gradient problems making it a suitable choice for CNN and densely connected layers [62,63]. The equation for ReLU is given by:

$$f(x) = \max(0, x). \tag{13}$$

In Eq. (13), the variable x represents the input to the activation function, and $f(x)$ represents the output. The ReLU function outputs the input value if it is positive, and zero otherwise. This operation ensures that ReLU preserves the positive features of the input while discarding negative values, leading to faster convergence during training and reduced computational complexity.

For efficient training, we employed the Adam (adaptive moment estimation) optimizer which outperforms traditional optimizers like SGD (stochastic gradient descent) [64]. Adam combines the benefits of both adaptive learning rates and momentum, resulting in faster convergence and improved generalization.

The equations for Adam optimizer update rules are given by:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2,$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}, \quad \theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t. \tag{14}$$

Here, in Eq. (14), m_t and v_t represent the first and second-moment estimates, respectively. The variable g_t denotes the gradient at time t, while $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of m_t and v_t. θ_t signifies the parameters at time t, η stands for the learning rate, and β_1 and β_2 are the decay rates for the moment estimates. Additionally, ϵ serves as a small constant to prevent division by zero. This formulation allows Adam to adaptively adjust the learning rates for each parameter based on the magnitude of its gradients and history, leading to efficient optimization in a wide range of scenarios.

To calculate the difference between the actual displacements and the model's predictions, we utilized the mean squared error (MSE) cost function, known for its effectiveness in regression tasks. The equation for MSE is given by:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2. \tag{15}$$

In this Eq. (15), y_i represents the actual target value, $\hat{y}_i$ represents the predicted value, and n represents the number of samples. MSE computes the average squared difference between predicted and actual values across all samples and penalizes larger deviations more heavily. This cost function guides the training process by minimizing the overall prediction error, leading to improved model performance and generalization.

As shown in Fig. 6, the neural network model for displacement prediction combines CNN layers with max pooling to process $120 \times 120 \times 2$ input material data, extracting features at multiple levels. This is followed by multiple dense layers to map the high-level features to the output displacement predictions. This hybrid deep neural network uses a 2×2 kernel size for the CNN layers to handle its complexity and large dataset efficiently. The architecture consists of three convolutional layers with 2×2 kernels, using 32, 64, and 128 filters, respectively, followed by max-pooling layers after each convolution. The output from the convolutional layers is flattened and passed through two dense layers, with the final layer reshaping the output to match the original $120 \times 120 \times 2$ displacement field. To effectively handle multiple load cases without increasing model complexity, we employed separate training for each load scenario using identical model architectures. This approach allows for more efficient training and easier model selection for specific cases, avoiding the need for additional parameters or one-hot encoding to differentiate between load cases. It also results in more manageable and less computationally expensive models compared to a single, larger model trained on all cases simultaneously.

A comprehensive discussion of the training methodology is presented in Section 4.2. This direct displacement prediction model is

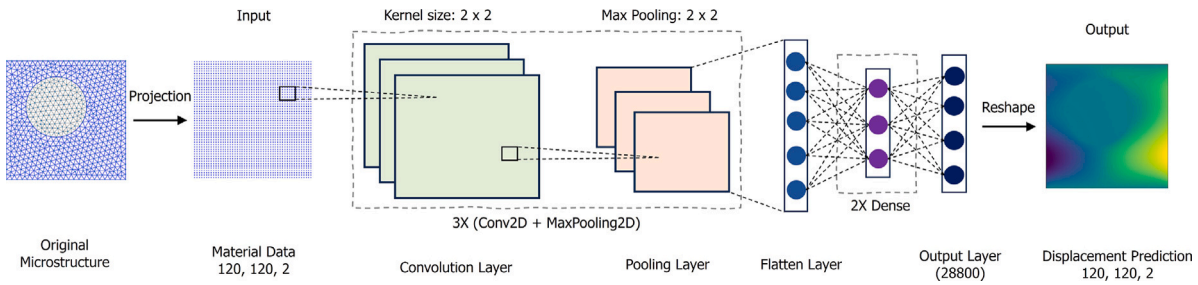

Fig. 6. Illustration of a neural network architecture designed to predict displacement within the original space. The architecture consists of multiple convolutional and pooling layers, followed by fully connected layers.

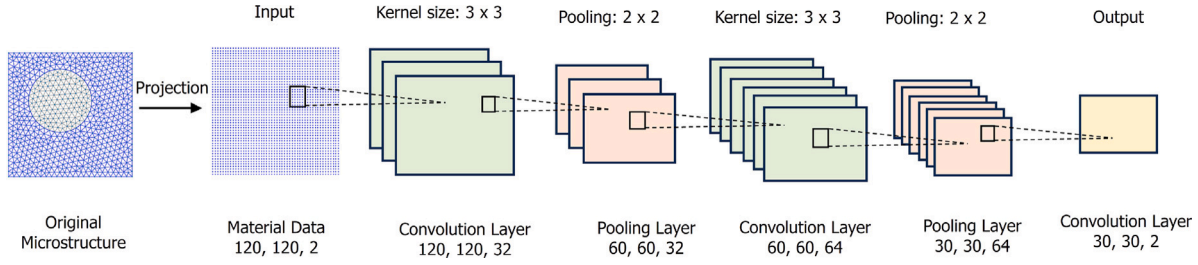

Fig. 7. Diagram illustrating the encoder network of a material autoencoder. The architecture comprises of combination of convolutional layers and pooling layers.

implemented in TensorFlow and trained on the curated finite element simulation data to predict displacement fields rapidly [65]. However, despite the efficiency gains offered by this surrogate model, the training process can still become computationally intensive when dealing with high-dimensional microstructure data and displacement fields, requiring further optimization strategies. To address these challenges, we utilize convolutional autoencoders for dimensionality reduction, focusing on compressing the simulation data into lower-dimensional latent representations. This approach accelerates the training process as well as streamlines materials optimization.

3.2. Autoencoder for dimensionality reduction

Given the distinct complexities of material properties and displacement data, we trained each data separately using the same autoencoder architecture. Both the material and displacement autoencoders consist of an encoder and a decoder network. We employed convolutional neural network (CNN) layers in this architecture to preserve and leverage the spatial information inherent in our material and displacement data, which is crucial for capturing fine details and enabling accurate predictions from the encoded representations.

To determine the optimal encoded dimension, we investigated three different encoding architectures. In all the cases, the material and displacement autoencoders share similar structures but are trained separately. The first architecture encodes both material and displacement data to a $60 \times 60 \times 2$ latent space, the second further reduces the dimension to $30 \times 30 \times 2$, and finally, the third architecture reduces the dimension to $15 \times 15 \times 2$. This approach allows us to explore the trade-off between dimensionality reduction and information preservation.

The material encoder, as illustrated in Fig. 7, uses a 3×3 kernel size in its CNN layers to capture detailed features within the compressed data space. The encoder architecture consists of three convolutional layers: the first with 32 filters, the second with 64 filters, and the final layer with 2 filters, all using 3×3 kernels. Each of the first two convolutional layers is followed by a max pooling layer with a 2×2 pool size, effectively reducing the spatial dimensions. The final convolutional layer produces the encoded output with a shape of $30 \times 30 \times 2$. The displacement autoencoder shares a similar structure with the material autoencoder. The decoder networks of the autoencoders mirror the encoder architectures, using transposed convolutions and upsampling layers to reconstruct the original high-fidelity data from the compressed latent representations, ensuring minimal loss of critical information.

The encoded displacement prediction process utilizing both autoencoders is outlined in Fig. 8. First, the separately trained autoencoders compress the preprocessed input–output simulation data pairs. The resulting encoded material properties and encoded displacement data, both with dimensions of $30 \times 30 \times 2$, are used to train a secondary deep neural network predictor model. In this model, the input data is the encoded material data and the target output data is the encoded displacement field. This encoded prediction model, while similar in structure to the direct prediction model shown in Fig. 6, is less complex and computationally less expensive due to the reduced input dimensions. It consists of two convolutional layers with 32 and 64 filters respectively, each followed by max pooling, and a connected layer with 512 units before the final output layer. This architecture efficiently processes the encoded data, leveraging the dimensionality reduction achieved by the autoencoders. Finally, predicted encoded displacement data goes through the decoder network to reconstruct the original displacement field. This method focuses on learning from compressed data, ensuring accuracy while significantly reducing training time and addressing computational challenges in material optimization.

4. Numerical examples: model testing and validation

This section presents the results of testing and validating the hypothesis that implementing low-dimensional training can effectively accelerate the prediction of mechanical responses for nanoparticle-reinforced microstructures. We demonstrate the successful integration of FEM and advanced machine learning techniques, leveraging the strengths of both approaches. The results of the finite element simulations are detailed, followed by the performance evaluation of the deep learning models developed for displacement prediction. Particular emphasis is placed on the effectiveness of dimensionality reduction using convolutional autoencoders, which compress the high-dimensional data into a lower-dimensional latent space, thereby improving computational efficiency. This investigation demonstrates the viability of the proposed dimensionality reduction method for enhancing computational efficiency. Additionally, we explore the optimal configuration of machine learning architectures necessary to achieve acceleration.

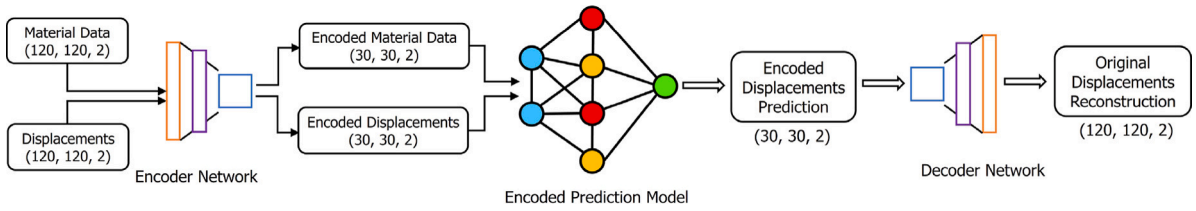

Fig. 8. Schematic of encoded displacement prediction and reconstruction.

Furthermore, we evaluate the extrapolation capability of our autoencoder and encoded prediction models by testing their performance on microstructures with more than two inclusions, despite being trained only on one and two inclusion data. This assessment provides insights into the models' ability to generalize to more complex microstructures. Lastly, we highlight the benefits of parallel processing in further accelerating the training and prediction processes.

4.1. Simulation data preparation

In preparing the training data set, we generated 400 matrix-inclusion composite samples with various positions and area fractions of reinforced single and multi-particle inclusions as described in Section 2.1. The choice of 400 samples was made to ensure a sufficiently large dataset for training and evaluating the machine learning models while maintaining computational feasibility for the finite element simulations. By generating a diverse set of microstructures, we aimed to capture a wide range of possible configurations and behaviors, enabling the machine learning models to learn and generalize mechanical responses effectively. This study primarily investigates the variations in microstructure configurations, where the matrix and inclusions phases were assigned distinct mechanical properties—Young's moduli of 9 GPa and 210 GPa, and Poisson's ratios of 0.33 and 0.22 respectively.

In this study, we primarily focus on simulating the uni-axial tensile problem, specifically oriented in the y-direction, to comprehensively test and validate our direct and encoded prediction approach. To enhance the robustness of our model and provide a more diverse set of scenarios, we also incorporate biaxial tension (in both x and y directions) and uniaxial compression (in the y-direction) load cases. For all cases, the magnitude of applied displacement remains consistent at 1×10^{-6} m, with directions adjusted appropriately for each loading scenario. The generated positions of reinforced inclusions are random, causing displacements to occur in both the x and y directions due to the material's Poisson ratio. This diverse set of loading conditions effectively underscores the complexity of the microstructural behavior, providing a rigorous benchmark for our model testing and data collection. As detailed in Section 2.2, the uni-axial tensile test can be considered as the case with essential boundary conditions exclusively. For nodes $\mathbf{x}$ where the vertical component $y = 0$ m, we define the essential boundary vector as $\mathbf{g} = [0, -1 \times 10^{-6}]^T$ m. Conversely, when the vertical component is $y = 1$ m, the essential boundary vector is set to $\mathbf{g} = [0, 1 \times 10^{-6}]^T$ m. The Preconditioned Jacobian-Free Newton–Krylov (PJFNK) solver is utilized for computation. The displacement field $\mathbf{u}(\mathbf{x})$ is subsequently extracted from the results file for further analysis. The same displacement magnitude was applied for other load cases as well, including biaxial tension in both x and y directions (with $[-1 \times 10^{-6}, 1 \times 10^{-6}]$ boundary conditions for both axes) and uniaxial compression in the y-direction.

As discussed in Section 2.2, the MOOSE framework is utilized for obtaining finite element modeling solutions from the microstructures. We chose MOOSE for its flexibility in integrating microstructure modeling, stochastic inputs, and machine learning couplings in a modular Python-based workflow [66]. The scalability and customization options in MOOSE make it an ideal tool for creating digital twins of new composite materials.

4.2. Prediction performance using original data

The initial phase of our study focused on utilizing the original material data and interpolated displacement data as input and target variables, respectively, for the development of a predictive deep learning model. This direct prediction model, illustrated in Fig. 6, was specifically developed to predict displacements in their original configurations. Prior to model training, the raw simulation data which includes the mesh structures and the nodal displacement underwent a preprocessing step involving projection and interpolation into a fixed grid of 120 × 120 points. This approach, detailed in Section 3.1, was essential for standardizing the representation of microstructural features across samples. This proposed projection strategy was crucial in handling the heterogeneity of mesh sizes and shapes from the finite element simulations, enabling the deep learning model to effectively learn from the consistent spatial representations of the microstructural configurations and material properties.

Following this preprocessing step, all input data were normalized using a Min-Max scaler to ensure uniformity across the dataset's range. Such normalization is critical for preventing any single feature from disproportionately influencing the model's learning process due to scale differences [67]. Following this, an extensive hyperparameter tuning process was conducted. This iterative process involved evaluating the model's performance over a range of hyperparameters to identify the optimal configuration. The final model architecture incorporated key components previously discussed in Section 3.1: the Rectified Linear Unit (ReLU) as the activation function, Adam as the optimizer, and Mean Squared Error (MSE) as the loss function. The learning rate and batch size significantly impact training efficiency and model convergence. The learning rate was set to 0.001, with a batch size of 32, to balance the trade-off between training speed and model accuracy. The selected hyperparameters enabled the model to capture the underlying patterns in the data while maintaining generalization capabilities. The dataset was randomized and partitioned into an 80% training set and a 20% test set, with 20% of the training data further allocated as a validation set. This stratification ensures that the model's performance is not artificially inflated by having seen all available data during training, thus maintaining its generalizability to unseen datasets. Training proceeded for up to 100 epochs, with the entire process of data loading, preprocessing, training, and evaluation being timed to facilitate subsequent comparisons with autoencoder-enhanced models. After predicting on the test set, the displacements were inverse-transformed back to the original scales for analysis. The whole process including the data loading, preprocessing, training the model, and evaluating performance took 1,536.47 s to complete, shown in Table 2.

The direct prediction model demonstrated a remarkable capability to predict displacements based on the material data accurately. As depicted in Fig. 9, which includes results for both uniaxial tension with one inclusion (a–d) and biaxial tension with two inclusions (e–h), the comparison between original and predicted displacements in both x and y directions highlights the model's precision across multiple load cases. Notably, the inclusion regions, represented as areas with higher Young's modulus and lower Poisson's ratio, exhibited lower displacements relative to the matrix regions. This observation aligns with physical expectations, given the stiffer nature of the inclusion material.

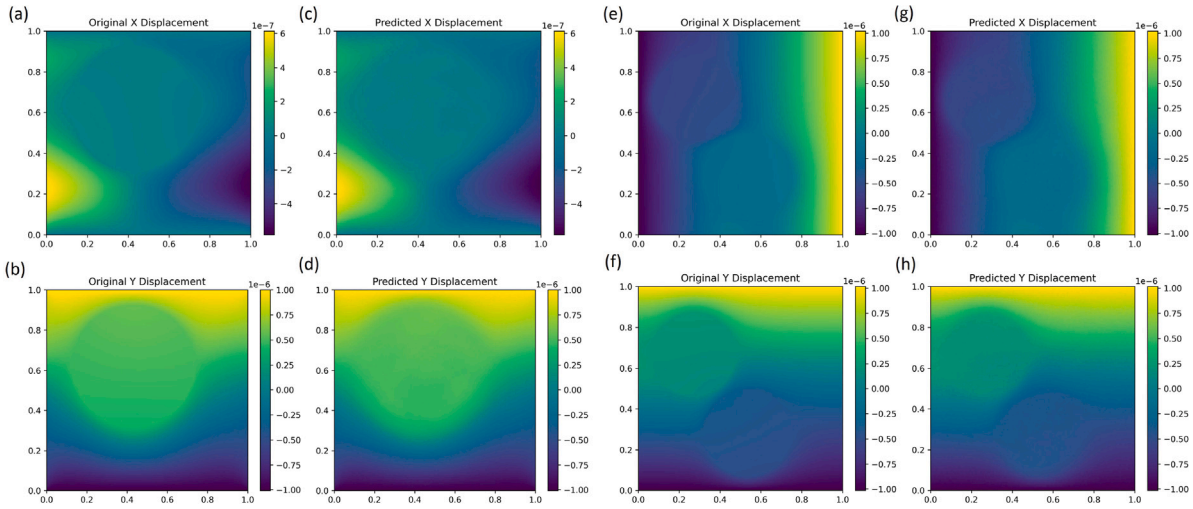

Fig. 9. Comparison of original and predicted displacements from the neural network model for direct displacement prediction: (a–d) show the original and predicted X and Y displacement fields for a mesh with one inclusion under uniaxial tension, while (e–h) present the original and predicted X and Y displacement fields for a mesh with two inclusions under biaxial tension.

Table 1
Reconstruction error for different encoded dimensions.

Original size	Encoded size	Reconstruction error (material data)	Reconstruction error (displacement data)
120 × 120 × 2	60 × 60 × 2	2.66×10^{-4}	7.39×10^{-5}
	30 × 30 × 2	6.33×10^{-4}	1.10×10^{-4}
	15 × 15 × 2	4.43×10^{-3}	9.76×10^{-4}

The similarity between the predicted and actual displacements within these inclusion regions underscores the model's ability to capture the nuanced effects of material properties on displacement behavior. As shown in Table 2, quantitatively, the model achieved an average mean absolute error (MAE) of 1.13×10^{-8} and a MSE of 2.96×10^{-16}, evidencing its high accuracy in displacement predictions. The model's capacity to handle multiple load scenarios further highlights its robustness in studying complex microstructures. Such low error metrics not only affirm the predictive prowess of the model but also suggest its potential applicability in simulating complex material behaviors with minimal computational resources. It is important to note that all prediction plots presented in this study, including Figs. 9, 10, 12, 16 and 17, R^2 plots in Fig. 13 as well as the error metrics in Tables 1–3 are generated using test datasets not seen by the model during training, providing a robust evaluation of its generalization capabilities.

4.3. Autoencoder performance analysis

Separate convolutional autoencoder networks were developed to encode the material property fields and displacement solutions. CNN layers enable compact feature extraction optimized for each data modality, by learning hierarchical filter representations that preserve information critical for reconstruction [61]. All the models underwent training up to 100 epochs, with a batch size of 32, employing the ReLU activation function, Adam optimizer (learning rate = 0.001), and mean squared error as the loss function.

To determine the optimal level of compression, we explored three different encoded dimensions: $60 \times 60 \times 2$, $30 \times 30 \times 2$, and $15 \times 15 \times 2$, corresponding to compression rates of approximately 75%, 94%, and 98%, respectively, for both the material and displacement data. Fig. 10 illustrates the reconstruction performance for each compression level, with subfigures (a–b) showing the original $120 \times 120 \times 2$ data, (c–d) the reconstruction from $60 \times 60 \times 2$ encoding, (e–f) from $30 \times 30 \times 2$ encoding, and (g–h) from $15 \times 15 \times 2$ encoding.

Our analysis revealed that the material property data, due to its binary nature (each point representing two material properties), could be accurately reconstructed even at the highest compression rate of $15 \times 15 \times 2$. However, the displacement data, which exhibits more complex variations, showed a gradual loss of accuracy as compression increased. We found that the $30 \times 30 \times 2$ encoding provided an optimal balance between compression and reconstruction accuracy for displacement data. This compression-accuracy trade-off is visually demonstrated in Fig. 11, which illustrates the first feature of both encoded material and displacement data at various compression levels. The figure illustrates encoded data with three levels of compression: $60 \times 60 \times 2$, $30 \times 30 \times 2$, and $15 \times 15 \times 2$. Notably, even after compression, the encoded data retains key features visible in the original data, validating the effectiveness of our autoencoder approach. As expected, the encoded representations at $60 \times 60 \times 2$ closely resemble the original data, preserving fine details and spatial relationships. The $30 \times 30 \times 2$ encoding strikes a balance, maintaining the most significant features while achieving substantial compression. However, at the highest compression level of $15 \times 15 \times 2$, we observe a gradual fading of finer details, particularly in the more complex displacement data.

Table 1 presents the reconstruction errors for all three encoded dimensions. The results show that while compression beyond $30 \times 30 \times 2$ leads to higher reconstruction errors, especially for displacement data, the $30 \times 30 \times 2$ encoding maintains a good balance between data reduction and accuracy. Consequently, we chose the $30 \times 30 \times 2$ encoding (94% compression) for both material and displacement data in our subsequent encoded prediction model. This approach achieves significant dimensionality reduction while preserving the essential features required for accurate reconstruction and prediction, demonstrating the effectiveness of our autoencoder-based compression technique in handling complex microstructural data.

When compared to existing literature where CAEs were employed on image data – for instance, achieving a compression ratio of 32 (equivalent to retaining just 3.125% of the original data) without significant loss of microstructural information [68], or the reduction of MNIST dataset dimensions [69] by 66% with an 84% accuracy – our work stands out for its significant dimensional reduction while maintaining high fidelity in data reconstruction. This comparison is notable given the intrinsic complexity and variability of 2D spatial coordinate data relative to standardized image datasets. Fig. 10 demonstrates the autoencoders' ability to reconstruct Young's modulus, and Y displacement fields with high fidelity, indicating minimal information

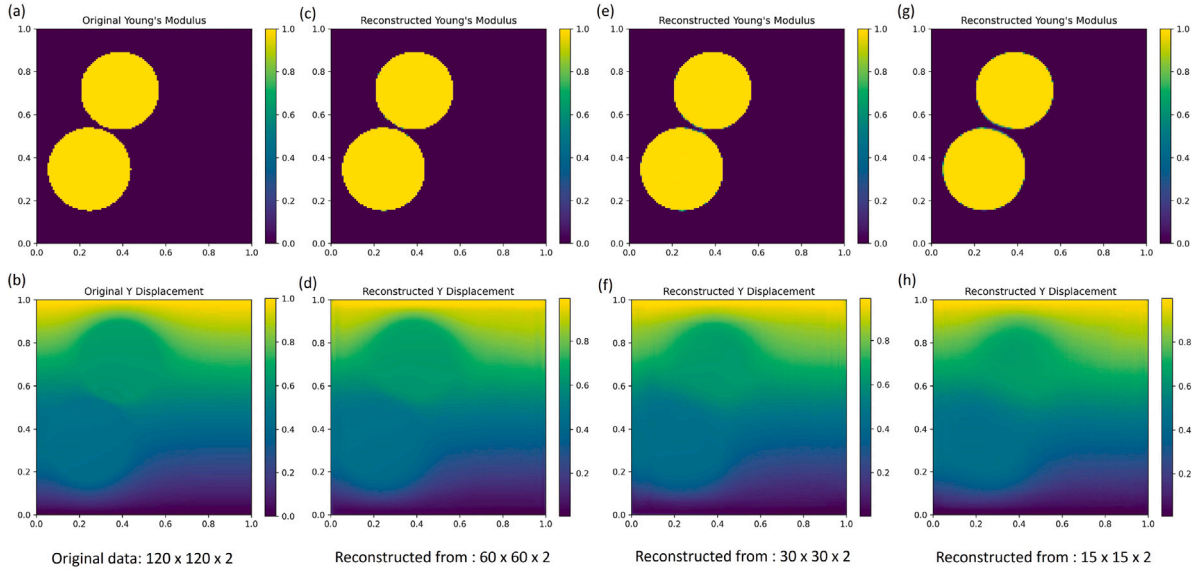

Fig. 10. Visualization of the autoencoder's reconstruction performance on material properties and displacement data at various compression levels: (a–b) original spatial dimension (120 × 120 × 2) of the material and displacement data, scaled between 0 and 1. (c–d) reconstructed material and displacement data from encoded dimension of 60 × 60 × 2. (e–f) reconstructed material and displacement data from encoded dimension of 30 × 30 × 2. (g–h) reconstructed material and displacement data from encoded dimension of 15 × 15 × 2.

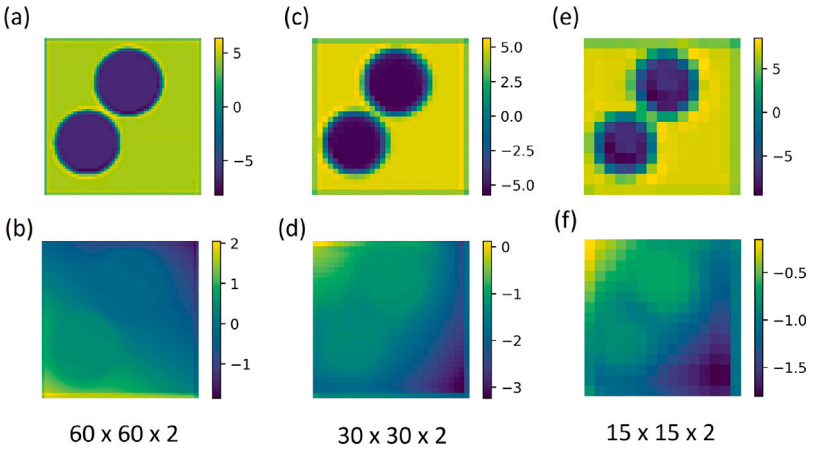

Fig. 11. Visualization of the encoded material and displacement data at various compression levels: (a–b) encoded features at 60 × 60 × 2. (c–d) encoded features at 30 × 30 × 2. (e–f) encoded features at 15 × 15 × 2.

loss during compression. The color bar represents the scaled magnitude of both the material properties and the displacement, ranging from 0 to 1. This scaling ensures consistency in data representation and facilitates efficient training of the model. The ability of both autoencoder models to achieve substantial reductions in data dimensionality while ensuring minimal loss of information indicates their advanced capability to capture and preserve crucial spatial relationships and properties inherent in the data. The training durations for the material and displacement autoencoders (encoded dimensions: 30 × 30 × 2) were recorded at 48.25 s and 103.11 s, showcasing the efficiency of these models. The specialized models effectively encode and reconstruct the detailed features unique to each data type with minimal information loss, despite achieving significant compression ratios.

4.4. Accelerated modeling through encoded data

In this subsection, we present the outcomes of implementing our advanced computational strategy, aimed at accelerating the modeling process through the application of autoencoder. This approach aligns with our primary objective: to improve computational efficiency in material modeling and optimization. Our strategy employs a multi-model architecture, utilizing the strengths of different neural network components. As discussed in Section 4.3, we utilized convolutional autoencoders to efficiently encode the high-dimensional material property distributions and displacement fields into compact latent representations. Then, a secondary hybrid neural network model, combining convolutional and dense layers, was developed to predict the encoded displacement field from the corresponding encoded material property, bypassing the need to operate on the original high-dimensional data. By operating on these compressed latent representations, we hypothesized that the training process could be significantly accelerated while retaining the essential physics and achieving accurate predictions. This multi-model approach complements our initial surrogate model, which predicted the original displacements from the original material data. While the surrogate model provided a baseline for performance comparison, our strategy involving autoencoders aimed to enhance computational efficiency by leveraging the dimensionality reduction capabilities of the encoders.

Adhering to the previously optimized hyperparameters, the model underwent training for 100 epochs, with a learning rate of 0.001 and batch size of 32. To prevent overfitting, early stopping was utilized

Table 2
Comparison of training times and error metrics for direct and encoded displacement prediction.

Model	Process	Run time (seconds)	Mean absolute error	Mean squared error
Direct Displacement Prediction	Data pre-processing	172.67	1.13×10^{-8}	2.96×10^{-16}
	Displacements Prediction Training	1361.51		
	Inference	2.29		
	Total	**1536.47**		
Encoded Displacement Prediction	Data pre-processing	168.33	4.02×10^{-8}	2.58×10^{-15}
	Material Autoencoder Training	48.25		
	Displacement Autoencoder Training	103.11		
	Displacements Prediction Training	68.94		
	Inference	2.31		
	Total	**390.94**		

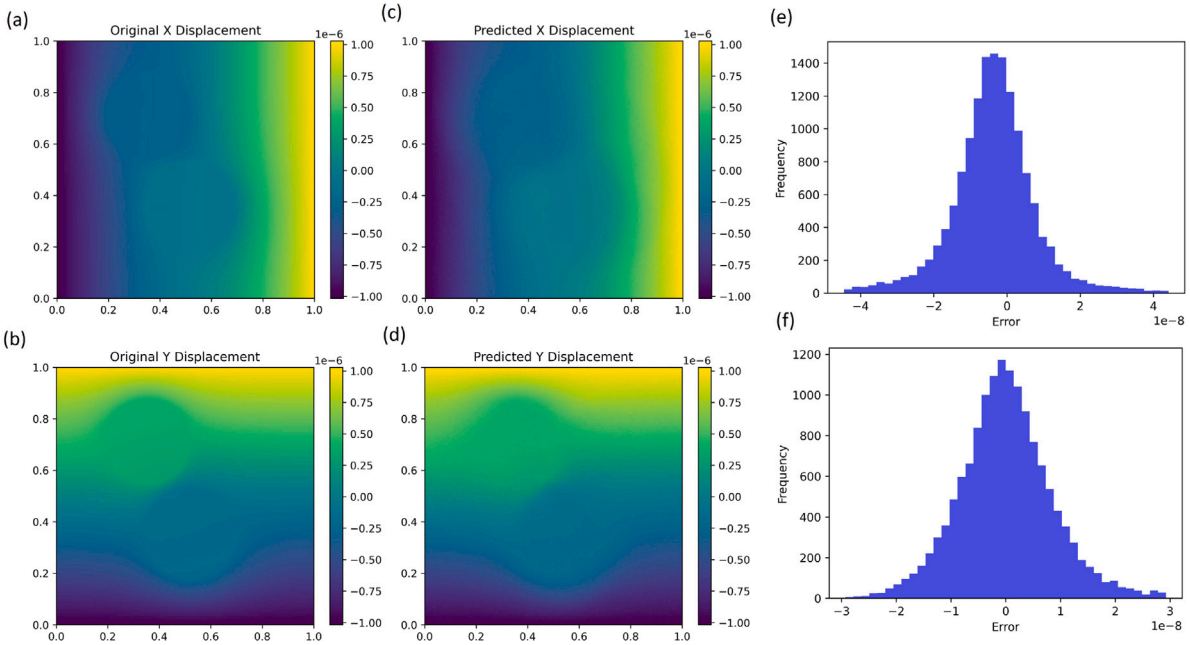

Fig. 12. Comparison of actual and predicted displacements via autoencoders: subfigures (a–d) show the original and predicted displacements in the x and y directions for a mesh with 2 inclusions under biaxial tension load case. Subfigures (e) and (f) present histograms displaying the distribution of plain errors, defined as the difference between the original and predicted displacement values for the x and y displacements, respectively, illustrating how the prediction errors are distributed. The overall process is illustrated in Fig. 8.

to stop the training if the validation loss did not improve for 10 consecutive epochs. This approach helps ensure that the model maintains generalization capability without overfitting to the training data. The training and validation loss trends shown in Fig. 13c consistently declined, indicating smooth training convergence without evidence of overfitting or underfitting. This demonstrates the model's precise learning of relationships within the compact data distributions.

The post-training evaluation involved decoding the predicted encoded displacements back to their original scale for direct comparison with the original displacements. Fig. 12(a–d) showcases the close resemblance between the predicted and actual displacement fields, demonstrating the model's accuracy. The model predictions match finite element solutions closely, confirming encoded data retention of the crucial physics. Additionally, Fig. 12(e–f) present error histograms for the x and y displacements, respectively, which exhibit a normal distribution with most errors concentrated around zero. This distribution indicates that the model's predictions are highly accurate, with minimal deviations from the actual displacements. The computed average MAE and MSE were 4.02×10^{-8} and 2.58×10^{-15}, respectively, shown in Table 2. Moreover, the prediction model achieved an exceptional average R^2 value of 0.99 across both x and y displacement components. As illustrated in Fig. 13a and 13b, plotting original versus predicted displacements shows nearly all points lying along the parity line, further underscoring the precise predictive capability despite the multi-stage encoding, latent-space training, and decoding pipeline.

These metrics are comparable to the performance of the initial model that utilized original data without dimensionality reduction. The entire modeling process, inclusive of data preparation, autoencoder training, and predictive modeling, was completed in 390.94 s—a 75% reduction in time compared to the 1,536.47 s required by the original model. This significant improvement in computational efficiency, alongside maintained accuracy, highlights the effectiveness of autoencoders in accelerating multiscale modeling tasks [70]. Our findings align with recent works in the field, such as Fetni et al.'s [45] study, which demonstrated that combining autoencoders with principal component analysis could achieve a remarkable 1/196 reduction ratio

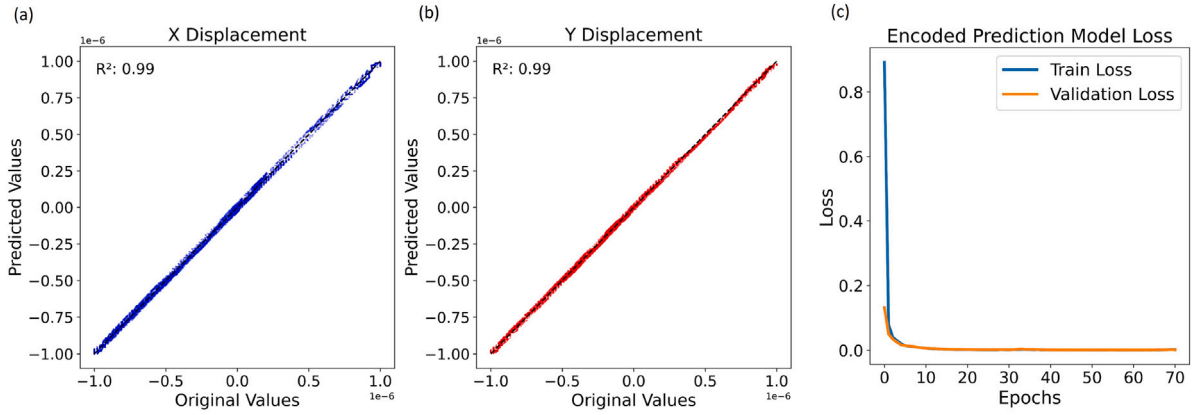

Fig. 13. Evaluation of model performance and training convergence: (a) R^2 scores for predicted x displacements, (b) R^2 scores for predicted y displacements, and (c) training and validation loss over epochs.

while maintaining over 80% accuracy in microstructural image analysis. Similarly, Ahmad et al. [71] successfully combined autoencoders with convolutional long short-term memory (ConvLSTM) networks to accelerate phase-field modeling of microstructure evolution, emphasizing the potential of machine learning techniques to significantly reduce computational costs in materials science simulations. However, while these studies validate the potential of autoencoders, they do not provide exact quantifications of speed-ups. Our approach not only confirms the effectiveness of autoencoder-based methods in compressing high-dimensional FEM data but also quantifies this acceleration, with our model achieving a $\sim 4x$ speed-up in microstructure response prediction compared to the direct prediction model. This explicit comparison further underscores the computational advantages offered by our framework. The total time, including mesh file generation and FEM simulation for 400 datasets, was approximately 1,730 s, making the autoencoder-accelerated model approximately $\sim 4.5x$ faster than the FEM simulations. However, the true power of our autoencoder-based approach becomes evident in the prediction phase. For prediction on 400 microstructures, our trained model requires only 3.1 s, compared to 1,730 s for traditional FEM simulations. This translates to a remarkable $\sim 558x$ speedup while maintaining high accuracy. All the fem simulations and the training of deep learning models were run on a High Performance Computing (HPC) server. This acceleration is particularly significant given that our study utilized linear elasticity and small deformation assumptions. For more complex scenarios, such as non-linear or large deformation simulations, where FEM simulations could potentially take hours or days, the relative speedup of our method would be even more pronounced. In such cases, the autoencoder based predictive model would still deliver rapid results, as the training time does not scale proportionally with FEM simulation times. This underscores the scalability of our approach, making it exceptionally powerful for applications in more complex settings. Furthermore, the potential for transfer learning to adapt our pre-trained models to more complex, non-linear cases offers an avenue for further acceleration, as fine-tuning an existing model typically requires less computational resources than training from scratch.

4.5. Influence of material properties contrast and transfer learning acceleration

The elasticity contrast of 210/9 was chosen as a benchmark to validate the framework under typical high-contrast conditions. To further validate the models' capability to handle geometries with different Young's modulus for the inclusion regions. Specifically, we generated 50 samples each for inclusion Young's moduli of 90 GPa and 20 GPa, while maintaining the matrix Young's modulus at 9 GPa. This allowed us to explore the impact of reducing the material contrast from the original 210/9 GPa benchmark case. Moose framework was used to run bi-axial tensile FEM simulations on all the new microstructures keeping all other parameters same. The results of this investigation are summarized in Figs. 14 and 15. From Fig. 14(d–f), when the inclusion Young's modulus is 90 GPa, the interpolated displacement fields remain distinguishable between the inclusion and matrix regions, as the contrast remains significant (90/9 = 10x). However, as the inclusion Young's modulus is further reduced to 20 GPa, Fig. 14(g–i), the difference in displacements between the inclusion and matrix becomes less pronounced. This is because the material properties are much closer (20/9 = 2.2x), making it more challenging to differentiate the regions based on the displacement field. This observation further validates our choice to use the 210/9 contrast as a benchmark, as it provides a clearer distinction between inclusion and matrix regions for evaluation purposes. For these new datasets, we first fine-tuned our pre-trained material and displacement autoencoder models. The first few layers of the autoencoders were frozen, and only the last 2 layers were trained for 20 epochs with a batch size of 8. This transfer learning approach enabled the autoencoder models to adapt to the new material property variations without the need to retrain from scratch. Next, we fine-tuned the pre-trained prediction model using the updated autoencoder representations. Again, the first few layers were frozen, and only the last 2 layers were trained for 20 epochs. The resulting predictions, shown in Fig. 15, demonstrate that our framework can effectively handle the reduced material contrast. The mean absolute error (MAE) and mean squared error (MSE) values for the prediction model were 5.68×10^{-8} and 6.02×10^{-15}, respectively, which are comparable to the original 210/9 benchmark case. However, it is important to note that transfer learning works best when the new datasets are similar to the original training data. For datasets with vastly different material properties or displacement behaviors, additional training may be required to achieve similar performance.

This approach of working in low-dimensional spaces for faster optimization offers a method for rapid analysis in high-dimensional microstructure studies. By avoiding the high costs of sampling complex morphologies extensively, we pave the way for fast and effective development of tailored composites, with potential applications ranging from rapid prototyping to real-time material response prediction in complex engineering scenarios. Additionally, we explored the influence of CPU core parallelization on optimizing model performance and reducing computational costs. Determining computational needs is vital in shared high-performance computing environments with scheduling and cost considerations [72]. This benchmarking was particularly focused on a model that had been enhanced with autoencoder acceleration, chosen for its previously demonstrated ability to significantly improve performance. The necessity of this benchmarking becomes apparent when considering the original model without autoencoder

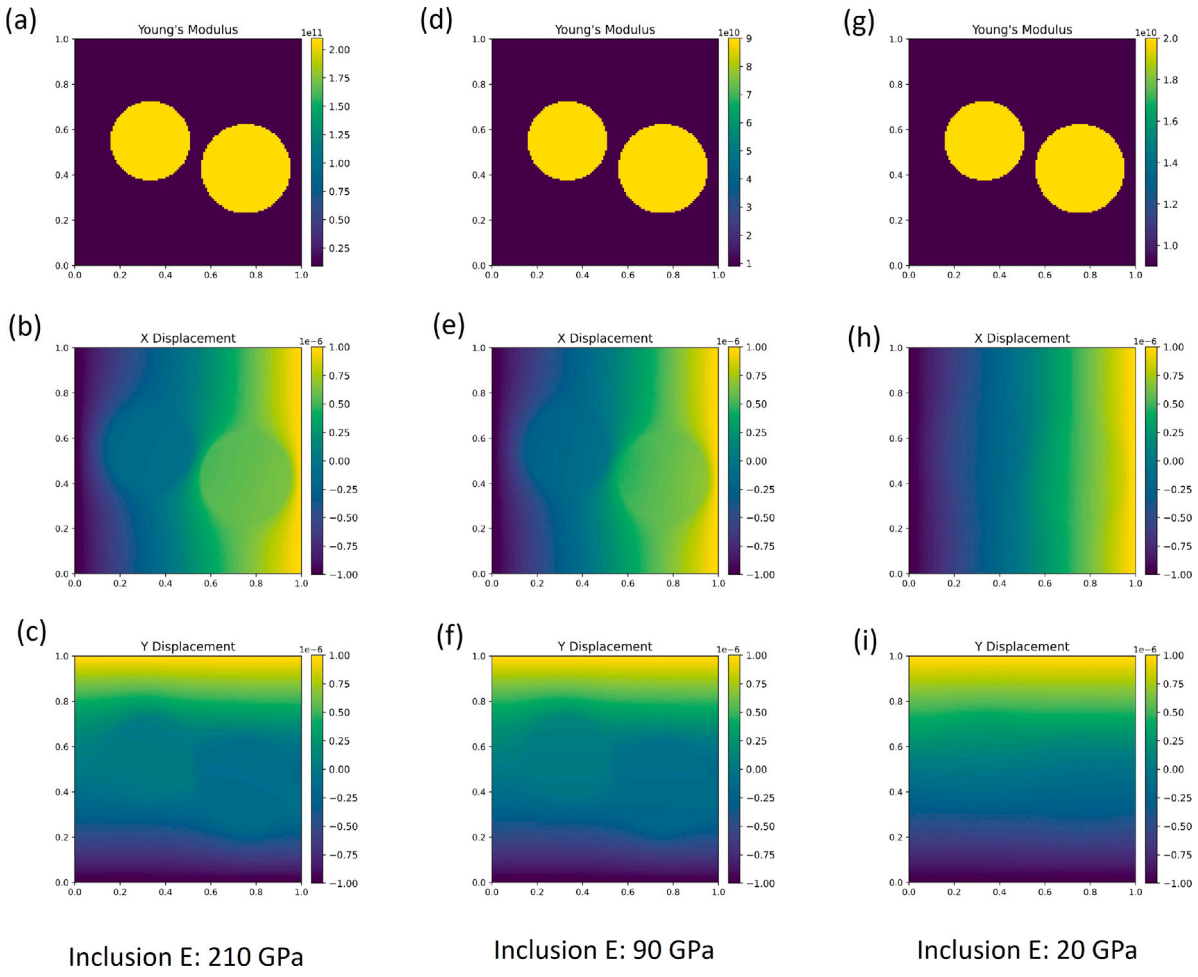

Fig. 14. Effect of different material properties on interpolated displacement fields. Interpolated materials and displacement fields for inclusion Young's moduli of (a–c) 210 GPa, (d–f) 90 GPa, and (g–i) 20 GPa. Young's modulus for the matrix was kept constant 9 GPa for all cases.

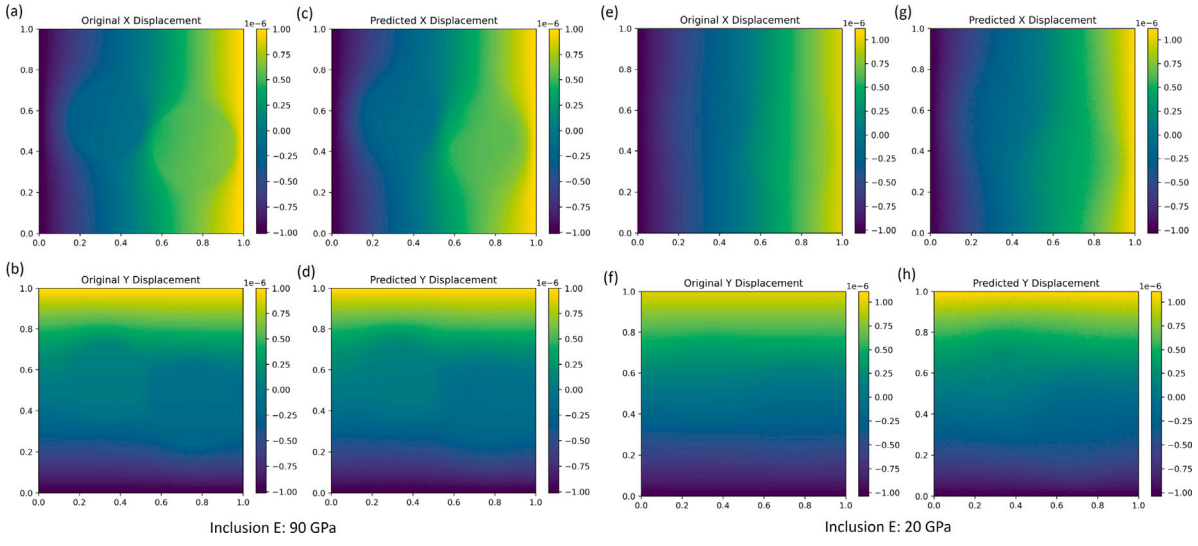

Fig. 15. Comparison of original and predicted displacement fields for different inclusion—matrix Young's modulus values. (a–d) Original and predicted displacement fields for elasticity contrast of 90/9 GPa, (e–h) Original and predicted displacement fields for elasticity contrast of 20/9 GPa.

acceleration, which required 1,536.47 s to complete its run using all 24 cores of a high-performance computing (HPC) server. To determine the most resource-efficient approach to training, we conducted a series of benchmarks on a Slurm HPC server varying the number of CPU cores utilized. The server uses an Intel(R) Xeon(R) Gold 6226R CPU with a clock speed of 2.90 GHz. These tests demonstrated a trend of diminishing returns on training time reduction with an increase in CPU cores. Initially, employing a single core resulted in a training

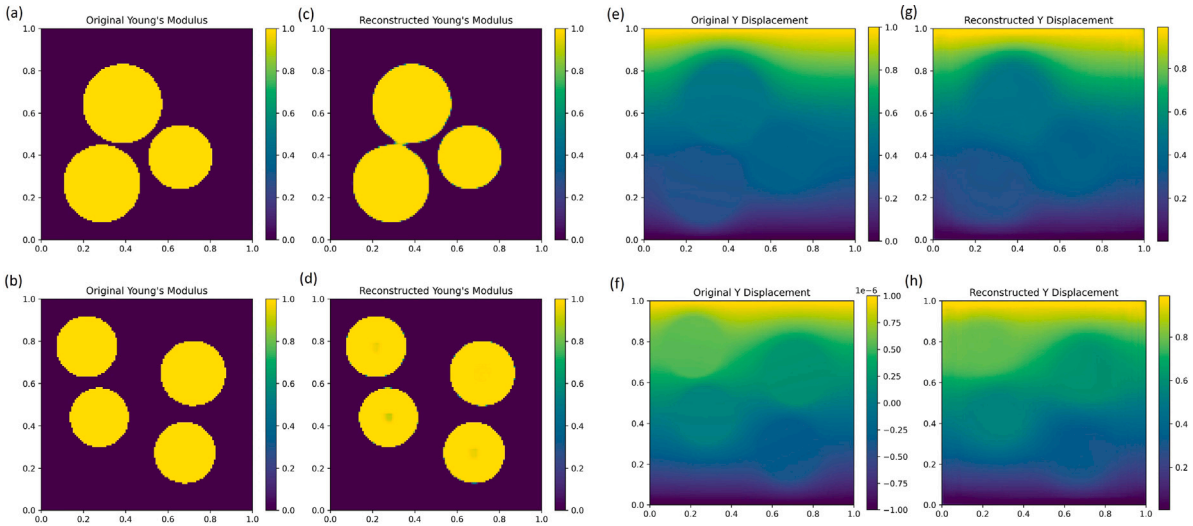

Fig. 16. Evaluation of autoencoder performance on reconstructing material data and displacement field under biaxial tension: (a–d) shows the original and reconstructed Young's Modulus for a microstructure with three inclusions, while (e–h) shows the original and reconstructed Y displacement fields for a microstructure with four inclusions.

time of 1480.15 s. Upgrading to 4 cores reduced this to 932.60 s, yielding a 37% improvement. Further expansions to 8 cores and 16 cores decreased the training times to 489.66 s and 416.62 s, reflecting 48% and 15% enhancements, respectively, with the final expansion to 24 cores marking the training time down to 390.94 s.

These results indicate that using up to 8 cores achieves a balance between computational efficiency and resource utilization, highlighting an efficient strategy for resource allocation without overspending. This analysis supports strategic planning for computational resource deployment and highlights the significant advantages of integrating autoencoder acceleration into our modeling process. By enhancing parallelization techniques and adopting autoencoder acceleration, we significantly streamline the process of conducting complex simulations, paving the way for faster development and optimization of advanced composite microstructures.

4.6. Extrapolation capabilities for complex particle configurations

To evaluate the robustness and generalization capabilities of our models, we conducted an extrapolation study using microstructures with three, four, and five inclusions — configurations not seen during the initial training phase, which utilized only one and two inclusion samples. This assessment aimed to test the models' ability to handle increasingly complex microstructures and their mechanical responses under various loading scenarios.

We first evaluated the performance of the autoencoder models. Fig. 16 illustrates the material and displacement autoencoders' ability to reconstruct original material properties and displacement fields from the compact $30 \times 30 \times 2$ representation. The results demonstrate that both models can reconstruct the data with high accuracy, showcasing their capacity to learn and retain detailed spatial features. This suggests that the autoencoders are well-suited to handle more complex microstructures and their corresponding responses. However, in Fig. 16d, a small region of spurious values is observed at the center of a few inclusions, suggesting a slight deviation from the expected values. While such deviations are more expected at boundary regions (Fig. 16c), where the boundaries of nearby inclusions are very close to each other, errors at the center of inclusions are likely to arise due to extrapolation on more complex geometries. This observation highlights the importance of including more diverse and complex configurations in the training dataset to further improve the model's generalization capabilities for multi-inclusion systems.

Subsequently, we evaluated the encoded prediction model on multi-inclusion microstructures, even though the model was trained only on one- and two-inclusion cases. Fig. 17 shows the original and predicted displacement fields for microstructures with three and four inclusions under biaxial tension. The model performed well on the three-inclusion cases, maintaining a high degree of accuracy. As the complexity increased to four inclusions, a slight reduction in accuracy was observed, but the model still produced reasonable results. The overall performance remained quite strong, even for these more complex structures. This trend is further reflected in Table 3, where the mean absolute error (MAE) and mean squared error (MSE) show a gradual increase from the training data to the three- and four-inclusion cases. This minor loss of accuracy can be attributed to the two-step process of encoding the material and displacement data into a compact representation and then decoding the predicted displacements back into the original $120 \times 120 \times 2$ space. While each step introduces some minor loss of detail, the model still handles increased microstructural complexity well. Notably, the model's ability to extrapolate to unseen, more complex cases, such as the three- and four-inclusion microstructures, demonstrates its robustness and potential for practical applications, where a wide range of microstructural complexities are likely to be encountered. While the current framework generalizes well to new configurations, it is currently limited to fixed domain dimensions (1×1 in the x-y directions). Increasing the complexity, such as incorporating samples with different aspect ratios, would require adjustments to the sampling space, modifications to the model architecture, including adapting the input and output shapes, and retraining the model. To address this efficiently, transfer learning techniques could be utilized. By freezing the initial layers of the trained model on the current sample and retraining only the later layers on new samples, the model could leverage pre-learned features. This approach is computationally more efficient than training entirely new models and has been shown to be effective for adapting pre-trained models to new, related tasks [73]. However, as discussed in Section 4.5, the effectiveness of transfer learning depends significantly on the similarity of the data distributions between the original and new datasets. If the data distributions are

Table 3
MAE and MSE error for extrapolation.

Dataset	Mean absolute error (MAE)	Mean squared error (MSE)
One and Two Inclusions	4.02×10^{-8}	2.58×10^{-15}
Three Inclusions	8.72×10^{-8}	7.01×10^{-15}
Four Inclusions	1.74×10^{-7}	2.13×10^{-14}

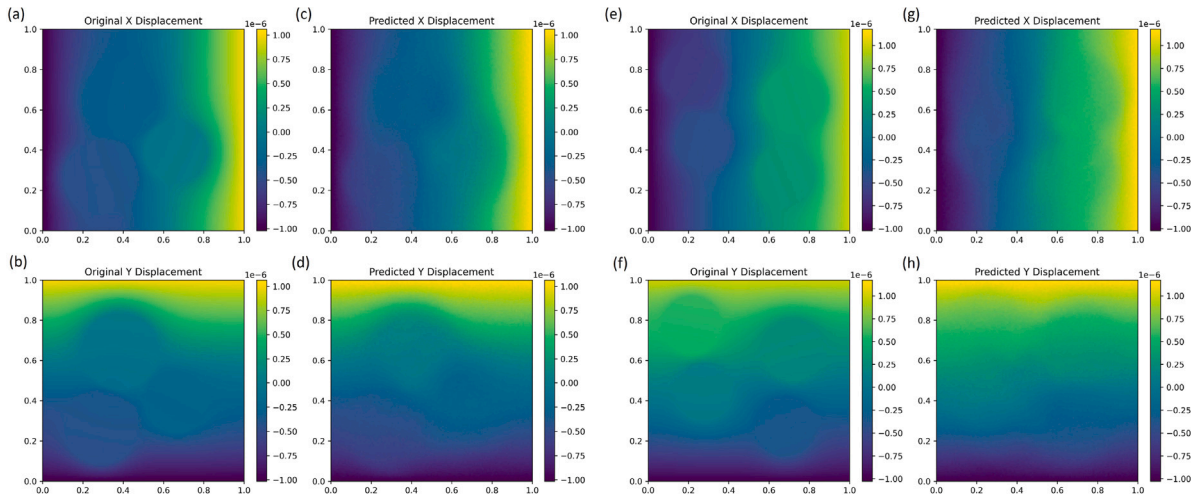

Fig. 17. Evaluation of encoded model performance on more complex microstructure under biaxial tension: (a–d) shows the original and predicted X and Y displacement fields for a microstructure with three inclusions, while (e–h) shows the original and predicted X and Y displacement fields for a microstructure with four inclusions.

vastly different, the pre-trained features may not generalize well, and additional fine-tuning or even retraining of the entire model may be required.

In conclusion, our models demonstrate a notable ability to extrapolate to more complex microstructures, particularly evident in the autoencoder's reconstruction capabilities and the prediction model's performance on three-inclusion samples. While accuracy decreases for even more complex structures, the results suggest that our approach provides a solid foundation for predicting mechanical responses across a range of microstructural complexities, with potential for further refinement through targeted fine-tuning strategies.

5. Conclusions

This work demonstrates an integrated materials modeling framework combining finite element microstructure generation, autoencoder-enabled dimensionality reduction, and accelerated displacement prediction to advance multiphysics characterization and design optimization. Physics-based simulations, while providing detailed insights, pose computational barriers for exhaustive microstructure evaluation. We overcome this through convolution autoencoders compressing spatial material properties and displacement solutions by 93.75% and 93.75% respectively. Encoded representations retain detailed features, enabling precise reconstruction through integrated decoding. A secondary neural network model trained on encoded data matches direct predictions closely while reducing training time by 75%. In more complex scenarios, such as plasticity, crack formation, or 3D geometry, the acceleration benefits of surrogate models would likely be even greater, given that such simulations are typically much more computationally expensive. Future work will involve extending our approach to nonlinear deformation and 3D geometry. Additionally, parallelization benchmarks reveal optimized resource allocations, with 8 CPU cores attaining significant acceleration without wasted capacity. This comprehensive approach to overcoming simulation and hardware barriers facilitates rapid virtual testing toward property improvements. This finding is instrumental in guiding the strategic deployment of computational resources for material multiscale modeling projects. The foundations laid by this work pave the way for future research avenues. Exploring transfer learning to utilize pre-trained models on similar tasks could reduce the need for large training datasets and computational resources, enhancing the model's applicability across different material systems. Further, integrating generative models like Generative Adversarial Networks (GANs) promises to transform material microstructure synthesis and optimization. By generating novel, detailed microstructural designs tailored to specific mechanical properties, these methods extend the current study's impact and offer innovative directions for applying machine learning in materials science, contributing to the advancement of next-generation materials.

CRediT authorship contribution statement

Shishir Barai: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Feihong Liu:** Writing – original draft, Visualization, Validation, Software, Investigation. **Manik Kumar:** Visualization, Validation, Software. **Christian Peco:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

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

Acknowledgments

This work has been funded by the National Science Foundation grants 2339373 and 1943899. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Computations for this research were performed on the Pennsylvania State University's Institute for Computational and Data Sciences' Roar supercomputer.

Data availability

Data will be made available on request.

References

[1] Rachid Hsissou, Rajaa Seghiri, Zakaria Benzekri, Miloudi Hilali, Mohamed Rafik, Ahmed Elharfi, Polymer composite materials: A comprehensive review, Compos. Struct. 262 (2021) 113640.

[2] Harsh Sharma, Ajay Kumar, Sravendra Rana, Nanda Gopal Sahoo, Muhammad Jamil, Rajeev Kumar, Shubham Sharma, Changhe Li, Abhinav Kumar, Sayed M Eldin, et al., Critical review on advancements on the fiber-reinforced composites: Role of fiber/matrix modification on the performance of the fibrous composites, J. Mater. Res. Technol. (2023).

[3] Shuo Zhang, Sanjairaj Vijayavenkataraman, Wen Feng Lu, Jerry YH Fuh, A review on the use of computational methods to characterize, design, and optimize tissue engineering scaffolds, with a potential in 3D printing fabrication, J. Biomed. Mater. Res. B: Appl. Biomater. 107 (5) (2019) 1329–1351.

[4] Olivia Lowe, Christian Peco, Fariborz Tavangarian, Modeling of the Bending Behavior to Study Nested-Cylinder Structure in Spicules, in: TMS 2023 152nd Annual Meeting & Exhibition Supplemental Proceedings, Springer Nature Switzerland, Cham, 2023, pp. 1215–1221.

[5] F. Simonetti, M. Fox, Experimental methods for ultrasonic testing of complex-shaped parts encased in ice, NDT & E Int. 103 (2019) 1–11.

[6] Nicolás González-Santacruz, Patricia Muñoz-Marzagon, Miguel Bartolomé, Ana Moreno, Jennifer Huidobro, Sérgio Henrique Faria, Effects of impurities on the ice microstructure of Monte Perdido Glacier, Central Pyrenees, NE Spain, Ann. Glaciol. (2023) 1–14.

[7] Koenraad George Frans Janssens, Dierk Raabe, Ernest Kozeschnik, Mark A Miodownik, Britta Nestler, Computational Materials Engineering: An Introduction to Microstructure Evolution, Academic Press, 2010.

[8] F. Ghanbari, F. Costanzo, D.P. Hughes, C. Peco, Phase-field modeling of constrained interactive fungal networks, J. Mech. Phys. Solids 145 (2020) 104160.

[9] Farshad Ghanbari, Joe Sgarrella, Christian Peco, Emergent dynamics in slime mold networks, J. Mech. Phys. Solids 179 (2023) 105387.

[10] Solahuddin Bin Azuwa, Fadzil Bin Mat Yahaya, Experimental investigation and finite element analysis of reinforced concrete beams strengthened by fibre reinforced polymer composite materials: A review, Alex. Eng. J. 99 (2024) 137–167.

[11] Alexandre Guével, Yue Meng, Christian Peco, Ruben Juanes, John E Dolbow, A Darcy–Cahn–Hilliard model of multiphase fluid-driven fracture, J. Mech. Phys. Solids 181 (2023) 105427.

[12] Yingjie Liu, Christian Peco, John Dolbow, A fully coupled mixed finite element method for surfactants spreading on thin liquid films, Comput. Methods Appl. Mech. Engrg. 345 (2019) 429–453.

[13] B.W. Spencer, W. Jiang, J.E. Dolbow, C. Peco, Pellet cladding mechanical interaction modeling using the extended finite element method, in: Top Fuel 2016: LWR Fuels with Enhanced Safety and Performance, 2016, pp. 929–938.

[14] Siddhant Kumar, Dennis M. Kochmann, What machine learning can do for computational solid mechanics, in: Current Trends and Open Problems in Computational Mechanics, Springer, 2022, pp. 275–285.

[15] Dipjyoti Nath, Ankit, Debanga Raj Neog, Sachin Singh Gautam, Application of machine learning and deep learning in finite element analysis: a comprehensive review, Arch. Comput. Methods Eng. (2024) 1–40.

[16] Tamara G Grossmann, Urszula Julia Komorowska, Jonas Latz, Carola-Bibiane Schönlieb, Can physics-informed neural networks beat the finite element method? IMA J. Appl. Math. (2024) hxae011.

[17] Xiaolong He, Qizhi He, Jiun-Shyan Chen, Deep autoencoders for physics-constrained data-driven nonlinear materials modeling, Comput. Methods Appl. Mech. Engrg. 385 (2021) 114034.

[18] C. Peco, Y. Liu, C. Rhea, J.E. Dolbow, Models and simulations of surfactant-driven fracture in particle rafts, Int. J. Solids Struct. 156–157 (2019) 194–209.

[19] Xiang-Long Peng, Mozhdeh Fathidoost, Binbin Lin, Yangyiwei Yang, Bai-Xiang Xu, What can machine learning help with microstructure-informed materials modeling and design? 2024, arXiv preprint arXiv:2405.18396.

[20] Sylvain Deville, Eric Maire, Guillaume Bernard-Granger, Audrey Lasalle, Agnès Bogner, Catherine Gauthier, Jérôme Leloup, Christian Guizard, Metastable and unstable cellular solidification of colloidal suspensions, Nature Mater. 8 (12) (2009) 966–972.

[21] Francesco Simonetti, Michael D. Uchic, Equiaxed polycrystalline ice for ultrasonic testing of solids, Phys. Rev. Appl. 18 (1) (2022) 014034, Publisher: APS.

[22] Yunmei Zhao, Zhenyue Chen, Xiaobin Jian, A high-generalizability machine learning framework for analyzing the homogenized properties of short fiber-reinforced polymer composites, Polymers 15 (19) (2023) 3962.

[23] Sarah David Müzel, Eduardo Pires Bonhin, Nara Miranda Guimarães, Erick Siqueira Guidi, Application of the finite element method in the analysis of composite materials: A review, Polymers 12 (4) (2020) 818.

[24] Eduardo G. Rodriguez, Christian Peco, Daniel Millán, The influence of material properties distribution of waves in 1d: Application to cryoultrasonics, Mecánica Computacional 39 (7) (2022) 217.

[25] Eduardo G. Rodriguez, Daniel Millán, Christian Peco, Representative Volume Element (RVE) Analysis for mechanical characterization of ice with metallic's inclusion of micro/nano particles, in: XXXIX Congreso Argentino de Mecánica Computacional y I Congreso Argentino Uruguayo de Mecánica Computacional, 40, 2023, p. 506.

[26] Farshad Ghanbari, Eduardo G Rodriguez, Daniel Millán, Francesco Simonetti, Andrea P Argüelles, Christian Peco, Modeling of wave propagation in polycrystalline ice with hierarchical density gradients, Finite Elem. Anal. Des. 217 (2023) 103916.

[27] Chandramouli Nyshadham, Matthias Rupp, Brayden Bekker, Alexander V Shapeev, Tim Mueller, Conrad W Rosenbrock, Gábor Csányi, David W Wingate, Gus LW Hart, Machine-learned multi-system surrogate models for materials prediction, npj Comput. Mater. 5 (1) (2019) 51.

[28] Lianping Wu, Teng Li, A machine learning interatomic potential for high entropy alloys, J. Mech. Phys. Solids 187 (2024) 105639.

[29] Fadi Aldakheel, Elsayed S Elsayed, Tarek I Zohdi, Peter Wriggers, Efficient multiscale modeling of heterogeneous materials using deep neural networks, Comput. Mech. (2023) 1–17.

[30] Jonathan Schmidt, Mário RG Marques, Silvana Botti, Miguel AL Marques, Recent advances and applications of machine learning in solid-state materials science, npj Comput. Mater. 5 (1) (2019) 83.

[31] Maziar Raissi, Paris Perdikaris, George E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.

[32] Lei Zhang, Lin Cheng, Hengyang Li, Jiaying Gao, Cheng Yu, Reno Domel, Yang Yang, Shaoqiang Tang, Wing Kam Liu, Hierarchical deep-learning neural networks: finite elements and beyond, Comput. Mech. 67 (2021) 207–230.

[33] Sebastian K. Mitusch, Simon W. Funke, Miroslav Kuchta, Hybrid FEM-NN models: Combining artificial neural networks with the finite element method, J. Comput. Phys. 446 (2021) 110651.

[34] Y.M.A. Hashash, Sungmoon Jung, Jamshid Ghaboussi, Numerical implementation of a neural network based material model in finite element analysis, Int. J. Numer. Methods Eng. 59 (7) (2004) 989–1005.

[35] M.A. Maia, IB.C.M. Rocha, P. Kerfriden, F.P. van der Meer, Physically recurrent neural networks for path-dependent heterogeneous materials: Embedding constitutive models in a data-driven surrogate, Comput. Methods Appl. Mech. Engrg. 407 (2023) 115934.

[36] Chi Chen, Dan Thien Nguyen, Shannon J Lee, Nathan A Baker, Ajay S Karakoti, Linda Lauw, Craig Owen, Karl T Mueller, Brian A Bilodeau, Vijayakumar Murugesan, et al., Accelerating computational materials discovery with machine learning and cloud high-performance computing: from large-scale screening to experimental validation, J. Am. Chem. Soc. (2024).

[37] Miguel A Bessa, Ramin Bostanabad, Zeliang Liu, Anqi Hu, Daniel W Apley, Catherine Brinson, Wei Chen, Wing Kam Liu, A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality, Comput. Methods Appl. Mech. Engrg. 320 (2017) 633–667.

[38] Kamal Berahmand, Fatemeh Daneshfar, Elaheh Sadat Salehi, Yuefeng Li, Yue Xu, Autoencoders and their applications in machine learning: a survey, Artif. Intell. Rev. 57 (2) (2024) 28.

[39] Tianshui Chen, Liang Lin, Wangmeng Zuo, Xiaonan Luo, Lei Zhang, Learning a wavelet-like auto-encoder to accelerate deep neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[40] Zouhair Nezhi, Marcus Stiemer, Morten Schierholz, Christian Schuster, Dimensional reduction by auto-encoders in machine learning based power integrity analysis, in: 2024 IEEE 28th Workshop on Signal and Power Integrity, SPI, IEEE, 2024, pp. 1–4.

[41] Ye Ji, Arnd Koeppe, Patrick Altschuh, Deepalaxmi Rajagopal, Yinghan Zhao, Weijin Chen, Yi Zhang, Yue Zheng, Britta Nestler, Towards automatic feature extraction and sample generation of grain structure by variational autoencoder, Comput. Mater. Sci. 232 (2024) 112628.

[42] Stefan Petscharnig, Mathias Lux, Savvas Chatzichristofis, Dimensionality reduction for image features using deep learning and autoencoders, in: Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, 2017, pp. 1–6.

[43] Xin Liu, Yanping Bao, Lihua Zhao, Chao Gu, Establishment and application of steel composition prediction model based on t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction algorithm, J. Sustain. Metall. 10 (2) (2024) 509–524.

[44] Norbert Huber, A strategy for dimensionality reduction and data analysis applied to microstructure–property relationships of nanoporous metals, Materials 14 (8) (2021) 1822.

[45] Seifallah Fetni, Thinh Quy Duc Pham, Truong Vinh Hoang, Hoang Son Tran, Laurent Duchêne, Xuan-Van Tran, Anne Marie Habraken, Capabilities of auto-encoders and principal component analysis of the reduction of microstructural images; application on the acceleration of phase-field simulations, Comput. Mater. Sci. 216 (2023) 111820.

[46] Do-Won Kim, Myeong-Seok Go, Jae Hyuk Lim, Seungchul Lee, Data-driven stress and strain curves of the unidirectional composites by deep neural networks with principal component analysis and selective-data augmentation, Compos. Struct. 313 (2023) 116902.

[47] Binu Melit Devassy, Sony George, Dimensionality reduction and visualisation of hyperspectral ink data using t-SNE, Forensic Sci. Int. 311 (2020) 110194.

[48] P. Hajibabaee, F. Pourkamali-Anaraki, M.A. Hariri-Ardebili, Dimensionality reduction techniques in structural and earthquake engineering, Eng. Struct. 278 (2023) 115485.

[49] Thomas A. Ujas, Veronica Obregon-Perko, Ann M. Stowe, A guide on analyzing flow cytometry data using clustering methods and nonlinear dimensionality reduction (tsne or umap), in: Neural Repair: Methods and Protocols, Springer, 2023, pp. 231–249.

[50] Krushna Shinde, Vincent Itier, José Mennesson, Dmytro Vasiukov, Modesar Shakoor, Dimensionality reduction through convolutional autoencoders for fracture patterns prediction, Appl. Math. Model. 114 (2023) 94–113.

[51] Kejian Hu, Xiaoguang Wu, Mode shape prediction based on convolutional neural network and autoencoder, in: Structures, vol. 40, Elsevier, 2022, pp. 127–137.
[52] Christopher P Kohar, Lars Greve, Tom K Eller, Daniel S Connolly, Kaan Inal, A machine learning framework for accelerating the design process using CAE simulations: An application to finite element analysis in structural crashworthiness, Comput. Methods Appl. Mech. Engrg. 385 (2021) 114008.
[53] Sang Min Lee, Sang-Youn Park, Byoung-Ho Choi, Application of domain-adaptive convolutional variational autoencoder for stress-state prediction, Knowl.-Based Syst. 248 (2022) 108827.
[54] Yongju Kim, Hyung Keun Park, Jaimyun Jung, Peyman Asghari-Rad, Seungchul Lee, Jin You Kim, Hwan Gyo Jung, Hyoung Seop Kim, Exploration of optimal microstructure and mechanical properties in continuous microstructure space using a variational autoencoder, Mater. Des. 202 (2021) 109544.
[55] Rasoul Najafi Koopas, Shahed Rezaei, Natalie Rauter, Richard Ostwald, Rolf Lammering, Introducing a microstructure-embedded autoencoder approach for reconstructing high-resolution solution field from reduced parametric space, 2024, arXiv preprint arXiv:2405.01975.
[56] Joe Sgarrella, Farshad Ghanbari, Christian Peco, I-STL2MOOSE: From STL data to integrated volumetrical meshes for MOOSE, SoftwareX 21 (2023) 101273.
[57] Christophe Geuzaine, Jean-François Remacle, Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities, Int. J. Numer. Methods Eng. 79 (11) (2009) 1309–1331.
[58] Alexander D. Lindsay, Derek R. Gaston, Cody J. Permann, Jason M. Miller, David Andrš, Andrew E. Slaughter, Fande Kong, Joshua Hansel, Robert W. Carlsen, Casey Icenhour, Logan Harbour, Guillaume L. Giudicelli, Roy H. Stogner, Peter German, Jacob Badger, Sudipta Biswas, Leora Chapuis, Christopher Green, Jason Hales, Tianchen Hu, Wen Jiang, Yeon Sang Jung, Christopher Matthews, Yinbin Miao, April Novak, John W. Peterson, Zachary M. Prince, Andrea Rovinelli, Sebastian Schunert, Daniel Schwen, Benjamin W. Spencer, Swetha Veeraraghavan, Antonio Recuero, Dewen Yushu, Yaqi Wang, Andy Wilkins, Christopher Wong, 2.0 - MOOSE: Enabling massively parallel multiphysics simulation, SoftwareX 20 (2022) 101202.
[59] Bogdan M. Wilamowski, Neural network architectures and learning algorithms, IEEE Ind. Electron. Mag. 3 (4) (2009) 56–63.
[60] Waseem Rawat, Zenghui Wang, Deep convolutional neural networks for image classification: A comprehensive review, Neural Comput. 29 (9) (2017) 2352–2449.
[61] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
[62] Abien Fred Agarap, Deep learning using rectified linear units (relu), 2018, arXiv preprint arXiv:1803.08375.
[63] Hong Hui Tan, King Hann Lim, Vanishing gradient mitigation with deep learning neural network optimization, in: 2019 7th International Conference on Smart Computing & Communications, ICSCC, IEEE, 2019, pp. 1–4.
[64] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[65] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, Software available from tensorflow.org.
[66] Derek Gaston, Chris Newman, Glen Hansen, Damien Lebrun-Grandie, MOOSE: A parallel computational framework for coupled systems of nonlinear equations, Nucl. Eng. Des. 239 (10) (2009) 1768–1778.
[67] SGOP.A.L. Patro, Kishore Kumar Sahu, Normalization: A preprocessing stage, 2015, arXiv preprint arXiv:1503.06462.
[68] Dharanidharan Arumugam, Ravi Kiran, Compact representation and identification of important regions of metal microstructures using complex-step convolutional autoencoders, Mater. Des. 223 (2022) 111236.
[69] B Janakiramaiah, G Kalyani, S Narayana, T Bala Murali Krishna, Reducing dimensionality of data using autoencoders, in: Smart Intelligent Computing and Applications: Proceedings of the Third International Conference on Smart Computing and Informatics, Volume 2, Springer, 2020, pp. 51–58.
[70] John E. Herr, Kevin Koh, Kun Yao, John Parkhill, Compressing physics with an autoencoder: Creating an atomic species representation to improve machine learning models in the chemical sciences, J. Chem. Phys. 151 (8) (2019).
[71] Owais Ahmad, Naveen Kumar, Rajdip Mukherjee, Somnath Bhowmick, Accelerating microstructure modeling via machine learning: A method combining autoencoder and convlstm, Phys. Rev. Mater. 7 (8) (2023) 083802.
[72] Stephen D. Kleban, Scott H. Clearwater, Fair share on high performance computing systems: What does fair really mean? in: CCGrid 2003. 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003. Proceedings, IEEE, 2003, pp. 146–153.
[73] Yifan Tang, M. Rahmani Dehaghani, G. Gary Wang, Review of transfer learning in modeling additive manufacturing processes, Addit. Manuf. 61 (2023) 103357.