



A multi-fidelity data-driven model for highly accurate and computationally efficient modeling of short fiber composites

Downloaded from: <https://research.chalmers.se>, 2026-05-19 04:09 UTC

Citation for the original published paper (version of record):

Cheung, H., Mirkhalaf, M. (2024). A multi-fidelity data-driven model for highly accurate and computationally efficient modeling of short fiber composites. *Composites Science and Technology*, 246. <http://dx.doi.org/10.1016/j.compscitech.2023.110359>

N.B. When citing this work, cite the original published paper.



A multi-fidelity data-driven model for highly accurate and computationally efficient modeling of short fiber composites

Hon Lam Cheung^a, Mohsen Mirkhalaf^{b,*}

^a Department of Mechanics and Maritime Sciences, Chalmers University of Technology, Hörsalsvägen 7A, 41296 Gothenburg, Sweden

^b Department of Physics, University of Gothenburg, Origovägen 6B, 41296 Gothenburg, Sweden

ARTICLE INFO

Dataset link: <https://github.com/hlcheungai/SRC-TransferLearning>

Keywords:

Transfer learning
Short fiber composites
Multi-fidelity data
Recurrent neural networks
Elasto-plastic behavior

ABSTRACT

To develop physics-based models and establish a structure–property relationship for short fiber composites, there are a wide range of micro-structural properties to be considered. To achieve a high accuracy, high-fidelity full-field simulations are required. These simulations are computationally very expensive, and any single analysis could potentially take days to finish. A solution for this issue is to develop surrogate models using artificial neural networks. However, generating a high-fidelity data set requires a huge amount of time. To solve this problem, we used transfer learning technique, a limited amount of high-fidelity full-field simulations, together with a previously developed recurrent neural network model trained on low-fidelity mean-field data. The new RNN model has a very high accuracy (in comparison with full-field simulations) and is remarkably efficient. This model can be used not only for highly efficient modeling purposes, but also for designing new short fiber composites.

1. Introduction

Short Fiber Reinforced Composites (SFRCs) have gained popularity in variety of applications, mainly due to their compatibility with injection molding processes and their interesting properties compared to unfilled polymers [1,2]. The presence of discontinuous fibers in SFRCs prevents load transfer between them, assigning the responsibility of load transfer to the matrix material. With high stiffness, fibers create the potential for elevated stress levels within the composite, often resulting in yielding of the matrix [3–5]. This will eventually result in overall plastic behavior of the composite. Accurate prediction of yield and post-yield behavior is of paramount importance particularly for extreme cases such as crash.

Micro-mechanical modeling approaches have been developed and used to predict non-linear elasto-plastic behavior of SFRCs (see e.g., [5–10]). These models can be generally categorized to two major classes: mean-field and full-field models. In mean-field models (see e.g., [6,7]), an average stress and average strain is considered in the micro-structural constituents. These models are typically computationally efficient. However, it is not possible to obtain detailed information about deformation mechanism at the micro-structure and interactions between different phases. Also, these models are not as accurate as full-field models. On the other hand, full-field models consider microscopic fields at distinct microscopic points. For solving equilibrium equations, typically, Finite Element Method (see e.g., [8,11]) or Fast Fourier

Transform (see e.g., [9,10]) are used. To conduct full-field analysis, it is essential to generate realistic Representative Volume Elements (RVEs) that accurately represent the material's micro-structure. While this method yields high accuracy, it is not universally applicable. Challenges arise for RVE generation, particularly in cases involving high aspect ratios of fibers or high fiber volume fractions [12,13]. Furthermore, this approach is also computationally demanding [5], particularly for coupled multi-scale FE² analyses [14]. Hence, alternative solutions are needed to overcome these issues while keeping a high level of accuracy.

Machine learning techniques have recently been applied to accelerate modeling in materials science (see [15,16] for reviews and [17–22] for some recent developments for composites). Model Order Reduction (MOR) methods are also employed to reduce the dimensionality and complexity of mathematical models used in various engineering and scientific applications. By reducing the degrees of freedom of the models, MOR techniques enable computationally efficient simulations while preserving critical system characteristics. Examples of MOR methods include Non-uniform Transformation Field Analysis (NTFA) (e.g., [23–25]), Proper Orthogonal Decomposition (POD) (e.g., [26,27]), and Self-consistent Clustering Analysis (SCA) (e.g., [28,29]). However, each reduced-order model is built for a *specific* micro-structure, and thus, different micro-structures would require different reduced-order models. Another model reduction approach is the Deep Material Network

* Corresponding author.

E-mail address: mohsen.mirkhalaf@physics.gu.se (M. Mirkhalaf).

(DMN), initially proposed by Liu et al. [30,31]. DMN has the benefits of only requiring linear elastic RVE data and can be used for other materials and loading conditions. However, each DMN can be used only for the micro-structure geometry it was trained on [32]. For different geometries (e.g., different fiber orientations or volume fractions), one can employ transfer learning to adopt pre-trained DMN to new configurations [33] or interpolate between multiple DMNs [32,34].

Recent developments have shown the capability of Artificial Neural Networks (ANNs) for surrogate modeling in materials science including modeling of composites [35]. Mozaffar et al. [17] demonstrated the successful prediction of path-dependent plasticity using Recurrent Neural Networks (RNNs). They conducted a detailed analysis of two-dimensional RVEs of a composite material and built a database to train and validate an RNN model. Another study by Gorji et al. [18] focused on developing an ANN model that accurately reproduced an anisotropic non-linear material model. Their approach utilized RNNs to create a data-driven model, which was then employed to study the multi-axial behavior of a two-dimensional foam. Recently, Bonatti and Mohr [20] proposed a novel self-consistent RNN as a surrogate model for non-linear material behavior. To address issues related to the network performance, they introduced a linearized minimal state cell. The trained and validated network was subsequently integrated into an explicit finite element framework for component-level analysis. Wu et al. [36] developed an efficient ANN surrogate model for micro-mechanical modeling of an elasto-plastic composite using two-dimensional finite element simulations. Wu and Noels [21] extended the previous work by developing an RNN model capable of capturing the evolution of local micro-structure state variables under complex loading paths. More recently, Friemann et al. [37] also developed an RNN model for SFRCs. In this work, contrary to majority of other non-linear ANN models, different micro-structural parameters including variety of fiber orientations and fiber volume fractions were considered. However, to avoid a huge amount of time for data generation process, a mean-field model was used for micro-mechanical simulations.

In this study, we have used *Transfer Learning* approach to develop an RNN model for high-fidelity full-field modeling of SFRCs. Using this method, it is possible to use existing knowledge in an already trained network for developing a new ANN model. Recently, Ghane et al. [38] proposed to use transfer learning to overcome initialization challenges in RNN models for cyclic elasto-plastic behaviour of woven composites. Initially a network was trained using 6-dimensional arbitrary evolving strain paths, and then fine-tuned using another dataset of specific cyclic loadings. Inspired by this work, we are proposing to transfer knowledge from the previously developed RNN model by Friemann et al. [37] (which uses low-fidelity mean-field data) to a new RNN model which uses a small data set of high-fidelity full-field data. The enhanced RNN model was successfully re-trained despite the small data set. The RNN model has a remarkable computational performance and a great accuracy when compared to full-field test data.

The subsequent sections of this paper are organized in the following manner. In Section 2, the process of data set generation is illustrated. Section 3 explains the RNN model development and the use of transfer learning technique. Section 4 gives the obtained results using the developed RNN model, and provides comparisons to other networks, too. Related discussions are given in Section 5. Finally, Section 6 gives some concluding remarks from this study.

2. Data generation

In this study, an elasto-plastic matrix and elastic fibers are considered, and variety of fiber orientations and fiber volume fractions are taken into account. The elasto-plastic matrix was modeled with the J_2 -plasticity model with an isotropic linear-exponential hardening. The J_2 yield function is stated as

$$\Phi(\sigma, \kappa) = \sigma_V - (\sigma_y + \kappa) \leq 0, \quad (1)$$

Table 1

Constitutive properties of elasto-plastic matrix and elastic fibers.

| Fiber properties | | Matrix properties | |
|-----------------------|-------------------|--------------------------|---------|
| Young's modulus | 76 GPa | Young's modulus | 3.1 GPa |
| Poisson's ratio | 0.22 | Poisson's ratio | 0.35 |
| Length | 240 μm | Yield stress | 25 MPa |
| Diameter | 10 μm | Linear hardening modulus | 150 MPa |
| Fiber volume fraction | 0.1–0.15 | Hardening modulus | 20 MPa |
| | | Hardening exponent | 325 |

where σ_y is the yield stress, σ_V is the von Mises equivalent stress, and κ is a hardening stress. The von Mises stress (σ_V) is calculated using the deviatoric stress (σ_{dev}) as

$$\sigma_V = \sqrt{\frac{3}{2} \sigma_{dev} : \sigma_{dev}}, \quad \sigma_{dev} = \sigma - \frac{1}{3} \text{tr}(\sigma) \mathbf{I}, \quad (2)$$

where \mathbf{I} is the second order identity tensor. The hardening stress is calculated by

$$\kappa = H \bar{\epsilon}^p + H_\infty (1 - e^{-m \bar{\epsilon}^p}), \quad (3)$$

where H is the linear hardening modulus, H_∞ is the hardening modulus, m is a hardening exponent, and $\bar{\epsilon}^p$ is the accumulated plastic strain. Constitutive properties of matrix and fibers (see [37,39]) are listed in Table 1. As supervised learning is employed to train the neural network, a full-field data set that covers a wide range of fiber orientations and loading conditions is required. Initially, random orientation tensors, fiber volume fractions, and 6-Dimensional random strain paths are sampled. Then, Digimat-FE is used to generate the RVE geometry and solve the boundary value problem using either a Finite Element (FE) or Fast Fourier Transform (FFT) analysis (see [40,41]). The subsequent subsections provide a detailed explanation of each step involved in this process.

2.1. Random orientation tensor generation

This study utilizes a procedure for generating random orientation tensors developed by Friemann et al. [37]. For a 3D orientation tensor, first, a line segment from 0 to 1 is divided into three segments using two randomly sampled points from the interval (0, 1). The lengths of these three segments are then used as the eigenvalues for a second order diagonal tensor (in matrix format, a 3×3 diagonal matrix). Then, a random rotation tensor is sampled using the algorithm proposed by Arvo [42]. This algorithm involves a single-axis rotation tensor $\mathbf{R}(\theta)$ with an angle θ sampled randomly in $(0, 2\pi)$, and a negatively scaled Householder transformation, given by:

$$-\mathbf{H} = 2\mathbf{v}\mathbf{v}^T - \mathbf{I}, \quad \mathbf{v} = \begin{bmatrix} \cos \varphi \sqrt{z} & \sin \varphi \sqrt{z} & \sqrt{1-z} \end{bmatrix}^T, \quad (4)$$

where angle φ is sampled randomly in $(0, 2\pi)$, and z is sampled randomly in $(0, 1)$. The final rotation tensor is obtained as $\mathbf{M} = -\mathbf{H}\mathbf{R}$. Finally, this rotation tensor \mathbf{M} is used to rotate the diagonal tensor, resulting in the final orientation tensor.

For planar orientation tensors, where only two out of three diagonal components are non-zero, the process differs slightly. First, a line segment from 0 to 1 is divided into two segments using one randomly sampled point from the interval (0, 1). The lengths of these two segments are then used as the components associated with the x and y -axis of a second order diagonal tensor. Then, a z -axis rotation tensor $\mathbf{R}(\theta)$ with an angle θ sampled randomly in $(0, 2\pi)$ is used to rotate the diagonal tensor. Subsequently, the resulting tensor undergoes a 90-degree rotation around a randomly selected x , y , or z -axis, resulting in the final planar orientation tensor.

Uni-Directional (UD) orientation tensors are sampled simply by randomly choosing whether the first, second or third diagonal component equals 1, with all other components being set to 0.

Remark 1. For 3D and 2D orientation distributions, the orientation tensor generated by this procedure will serve as the reference to Digimat-FE to create an RVE, but the actual orientation tensor may not be exactly the same as the reference one. For UD samples, we always get the reference orientation tensor from a generated RVE.

2.2. Random strain path generation

A random strain path generation developed by Friemann et al. [37] is used. Initially, the number of drift directions (n_{Drift}) is randomly selected from 1, 2, 5, and 10. Then, a set of drift directions is sampled from a normal distribution with a mean of 0 and a standard deviation of 1. To obtain 100 time steps ($n_{\text{TimeSteps}}$), each drift direction is repeated $\frac{n_{\text{TimeSteps}}}{n_{\text{Drift}}}$ times. The drift for the entire path can be expressed as

$$\mathbf{D} = [\underbrace{\mathbf{d}_1 \cdots \mathbf{d}_1}_{\frac{n_{\text{TimeSteps}}}{n_{\text{Drift}}}} \underbrace{\mathbf{d}_2 \cdots \mathbf{d}_2}_{\frac{n_{\text{TimeSteps}}}{n_{\text{Drift}}}} \cdots \underbrace{\mathbf{d}_{n_{\text{Drift}}} \cdots \mathbf{d}_{n_{\text{Drift}}}}_{\frac{n_{\text{TimeSteps}}}{n_{\text{Drift}}}}], \quad (5)$$

where \mathbf{d}_i represents the randomly sampled drift direction as a 6D column vector. Additionally, at each time step, a random noise vector is sampled from a normal distribution with a mean of 0 and a standard deviation of 1. Subsequently, these noise vectors are scaled by a factor γ , which ranges from 0 to 1. The noise for the entire path can be expressed as

$$\mathbf{N} = \gamma[n_1 \ n_2 \ \cdots \ n_{n_{\text{TimeSteps}}}], \quad (6)$$

where n_i represents the randomly sampled noise vector as a 6D column vector. Then, a cumulative sum for each strain component is computed by adding both drift and noise from the starting point to the present time step. This results in the unscaled path, which can be expressed as

$$\text{UnscaledPath}_{ij} = \sum_{k=1}^j D_{ik} + N_{ik} \quad \text{for } i = 1, 2, \dots, 6 \text{ and } j = 1, 2, \dots, n_{\text{TimeSteps}}. \quad (7)$$

Finally, this path is scaled such that the largest component is equal to the prescribed maximum strain ϵ_{max} , which ranges from 0.01 – 0.05, to obtain the final scaled path as

$$\text{ScaledPath} = \frac{\epsilon_{\text{max}}}{\max(\text{UnscaledPath})} \text{UnscaledPath}. \quad (8)$$

Table 2 gives the random strain path generation settings.

Table 2
Settings used for random strain path generation.

| Parameter | Distribution | Range/ Value |
|-------------------------|-------------------|---------------|
| $n_{\text{TimeSteps}}$ | – | 100 |
| n_{Drift} | Uniformly sampled | (1, 2, 5, 10) |
| γ | Uniformly sampled | (0–1) |
| ϵ_{max} | Uniformly sampled | (0.01–0.05) |

The choice of 100 time steps, i.e., a time increment of 0.01 per step, was driven by the computational requirements of RVE simulations. This setting enabled the generation of more data with varying properties and loading paths than a higher time steps setting. As for the number of drift directions, a higher number would increase the complexity of the strain path. However, it also leads to potential issues for convergence in the simulation. Therefore, a maximum of 10 drift directions is selected to balance the complexity and convergence considerations.

Remark 2. The time step in this study signifies the progression of load steps within the applied strain path, with no inherent temporal influence on the material behavior.

2.3. RVE size determination

An RVE should contain enough information about the micro-structure, and it is crucial for it to statistically represent the micro-structure. By including sufficient micro-structure details, the RVE can provide meaningful prediction and accurately capture the material's mechanical response. Therefore, it is essential to conduct a study on the RVE size to ensure the representativeness of the obtained data.

To determine the RVE size, different approaches have been proposed (see e.g., [43–45]). Kanit et al. [43] proposed a methodology to find the minimum size needed to compute the effective properties (e.g., elastic modulus) provided the desired precision and the number of realizations. Gitman et al. [44] also proposed a statistical method using the Chi-square criterion to find the minimum RVE size that satisfies the accuracy requirement. Mirkhalaf et al. [45] proposed two criteria: (i) the coefficient of variation of the deformation behavior for different RVE realizations with the same size should be less than a desirable value, (ii) the average behavior of all realizations with the same RVE size should fall within a desirable error range compared to the average response of the next RVE size. This approach [45] is followed in this study for RVE size determination. Each tested RVE size has 4 realizations, subjected to monotonic uniaxial stress loading by applying corresponding strain component from 0–0.05. After conducting a statistical analysis, to balance the computational time and maintain the data accuracy, RVE sizes of 4 times the fiber length in fiber longitudinal direction and 6 times the fiber diameter in fiber transverse direction (4L6D) is selected for UD fiber orientation. For planar fiber orientation, 2 times the fiber length is chosen for the in-plane directions, and 4 times the fiber diameter is selected for the out-of-plane direction (2L4D). With these selections, the coefficient of variation is approximately below 4%, and the difference of the average response compared to the next RVE size is below 6%. For 3D fiber orientation distributions, an RVE size of 1.875 times the fiber length in all directions (1.875L) is selected. This size yields a coefficient of variation of less than 1.5%. More detailed information about this statistical analysis is provided in supplementary information, Section 1.

It should be mentioned that both FE and FFT analyses are used in this study. A comparison is performed to ensure the accuracy of the obtained results. More details about these simulations and comparisons are found in supplementary information, Section 2. Furthermore, the voxel settings for FFT analysis are detailed below. For UD fiber orientation RVEs, there are 960 and 56 voxels in length in the fiber longitudinal and transverse directions, respectively. For planar fiber orientation RVEs, there are 480 and 40 voxels in length in the in-plane and out-of-plane directions, respectively. For 3D fiber orientation RVEs, there are 432 voxels in all directions.

Remark 3. For each distinct material, characterized by a specific fiber orientation distribution and fiber volume fraction, only one simulation, either FE or FFT, is incorporated into the data set. Therefore, the integration of both FE and FFT results within the data set does not induce fluctuations for the neural networks.

2.4. Data post-processing

After finishing each simulation, the actual orientation tensor, fiber volume fraction, and stress–strain response are extracted. All extracted data, except the stress responses, are placed into an input array, and calculated stresses are inserted into a target output array. Then, the data are split into training, validation, and test data sets, accounting for 80%, 15% and 5% of the data, respectively. Note that the data with the same fiber orientation case and employed solver are grouped, and the splitting of data is performed within these groups. Afterwards, the split data is combined to form the final training, validation, and test data sets. This procedure ensures that each data set contains approximately the same percentage of data with a specific fiber orientation case and

Table 3
Number of data samples generated to form the full-field data set.

| Fiber orientation case | Solver | Train | Valid | Test | Total |
|------------------------|--------|-------|-------|------|-------|
| UD | FE | 14 | 2 | 1 | 17 |
| | FFT | 20 | 4 | 1 | 25 |
| Planar | FE | 104 | 20 | 6 | 130 |
| | FFT | 100 | 19 | 6 | 125 |
| 3D | FE | 0 | 0 | 0 | 0 |
| | FFT | 200 | 38 | 12 | 250 |
| All | | 438 | 83 | 26 | 547 |

generated by a particular solver. Table 3 provides the number of data in the data set for different orientation cases and solvers.

Fig. 1 displays a scatter plot representing the eigenvalues of the orientation tensor (λ_1 and λ_2) for all generated samples. The color of each point in the scatter plot represents the fiber volume fraction of the sample. While the random reference orientation tensor was uniformly sampled, the actual distribution of orientation tensors is more densely concentrated around the unidirectional fiber orientation side ($\lambda_1 = 1$). There are fewer data points in the region corresponding to a random 3D distribution ($\lambda_1 = \lambda_2 = \lambda_3 = 0.333$) due to the difficulties encountered in generating complex RVEs.

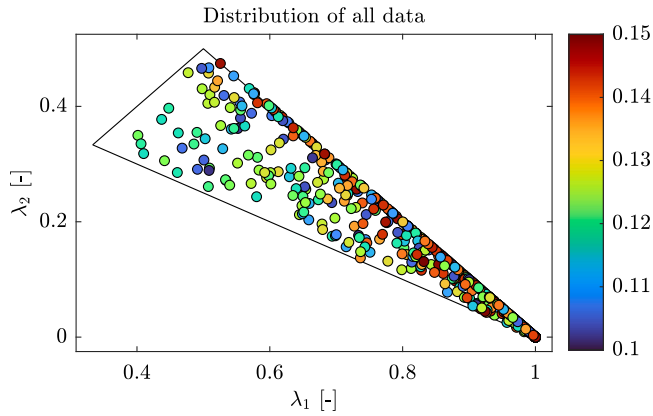


Fig. 1. Distribution of the generated data in terms of their sorted eigenvalues λ_1 and λ_2 ($\lambda_1 \geq \lambda_2 \geq \lambda_3$) of the orientation tensors. Different colors represent different fiber volume fractions.

It is worth noting that the total amount of data generated is approximately 0.07% of the mean-field data set used to develop the original network.

3. RNN model development

The original RNN model was developed by Friemann et al. [37] using mean-field simulations. It has 13 inputs including 6 unique orientation tensor components, a fiber volume fraction, and a strain path (a sequence of 6 unique strain tensor components). The inputs were standardized utilizing the Z-score method with the mean and standard deviation calculated from the training data set. The RNN architecture consisted of 3 Gated Recurrent Unit (GRU) layers [46], with 500 hidden states in each layer. Following the GRU layers, there is a dropout layer [47] with a 50% dropout rate, and a fully connected layer with 6 neurons. The fully connected layer utilized a linear activation function to output the stress prediction (a sequence of 6 unique stress tensor components). The RNN model was developed on a data set consisting of 40,000 samples generated using mean-field simulations. A single combination of a matrix and fiber was considered. However, a variety of different fiber orientation tensors and fiber volume fractions were considered. Also, very complex 6-Dimensional strain paths, with 2000 time

steps, were generated and used. The developed network demonstrates a very good performance, and accurate stress-strain predictions are obtained. However, the obtained accuracy is for low-fidelity mean-field simulations.

In this study, we have used transfer learning approach to enhance the RNN model for high-fidelity full-field simulations. The new network inherits the normalization and trainable parameters from the original network which are further fine-tuned with a newly generated full-field data set, with only 547 data samples, each having 100 time steps. Using this approach, the knowledge learned from the mean-field data set is used and enhanced in the final network using the small full-field data set. Fig. 2 shows an illustration of the architecture and the transfer learning method.

3.1. Training settings

To train the neural network, a loss function needs to be defined. In this study, the networks were developed utilizing the default loss function for time-series regression in Matlab, which is the half mean squared error of the predicted responses for each time step. The loss function is expressed as

$$loss = \frac{1}{2S} \sum_{i=1}^S \sum_{j=1}^R (t_{ij} - O_{ij})^2, \quad (9)$$

where S is the sequence length, R is the number of output, t is the target, and O is the network prediction.

ADAM optimizer was employed to minimize the loss function, as it has demonstrated good performance across a wide range of neural networks. The optimizer parameters, such as gradient decay factor β_1 , squared gradient decay factor β_2 , and denominator offset ϵ were set to their default values. These default values were chosen as they are generally considered suitable for a wide range of neural networks [48]. Also, L_2 -regularization is used to prevent over-fitting [49] and gradient clipping is used to prevent exploding gradient [50].

The hyper-parameters are to be optimized, with the majority of adjustments focused on parameters related to the learning rate, its decay during training, and the number of iterations. Although the gradient threshold is also tuned, it is less likely to be a critical parameter for final neural network performance. The Bayesian optimization function [51] in Matlab was used to optimize the training parameters. Considering the computational time, the maximum number of trials was set to 65, and the objective was set to minimize the validation loss. Subsequently, the iteration corresponding to the minimum validation loss from the best trial was selected as the final network configuration.

3.2. Evaluation metrics

During the neural network training, the loss function is employed to evaluate the network performance. However, the loss function may not be an ideal metric for reviewing the neural network performance, primarily due to its lack of normalization. Therefore, comparing the results between different test samples with varying magnitudes of stress is not justified. To address this issue and enable a proper evaluation of the neural network performance across different test cases, an alternative metric was employed. The equivalent von Mises stress is used to determine the MeRE and MaRE. By considering the von Mises stress, which combines multiple components of the stress tensor into one scalar value with physical meaning, it becomes easier to evaluate the overall performance of the neural network. The von Mises stress is calculated at each time step for both the target and network prediction. Subsequently, MeRE and MaRE are calculated by the following equations:

$$MeRE = \frac{\sqrt{\sum_{t=1}^T (\sigma_V^t - \hat{\sigma}_V^t)^2}}{\max(\sigma_V^t)T} \quad \text{and} \quad MaRE = \frac{\max(\sigma_V^t - \hat{\sigma}_V^t)}{\max(\sigma_V^t)}, \quad (10)$$

where σ_V is the target von Mises stress, $\hat{\sigma}_V$ is the predicted one, the superscript t represents the time step in the sequence, and T represents the number of time steps for the sequence.

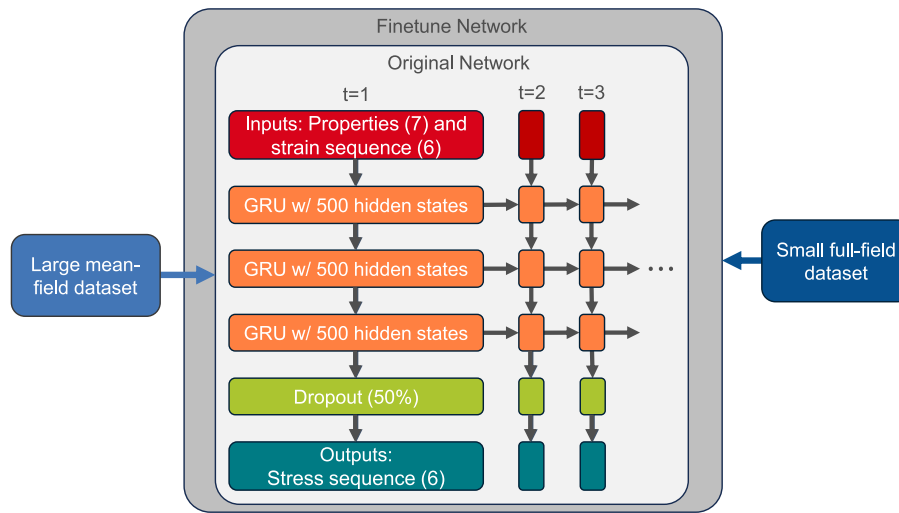


Fig. 2. Network architecture and the transfer learning approach utilizing a large mean-field and a small full-field data set.

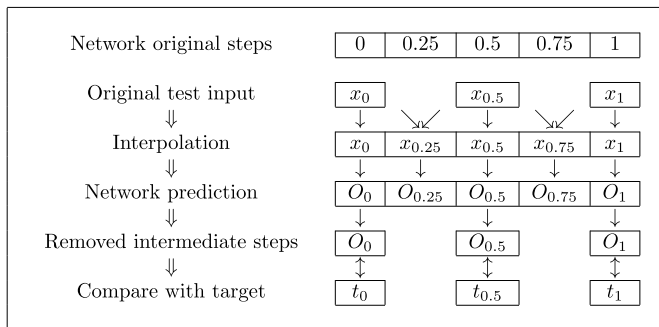


Fig. 3. Illustration of the testing procedure for a network trained with longer time steps compared to the test sample.

4. Results

Different networks are trained utilizing data sets with different time steps, and hence, time steps of the test samples can be a factor when evaluating the networks' performance. To establish a fair performance evaluation, it is reasonable to use the same time steps that the network has trained on for predicting test samples. If the network is trained on a higher number of time steps than the test sample, linear interpolation is applied to the test sample input to allow the network to make predictions with the same time steps used for its training. After obtaining the prediction, the intermediate predictions are removed before comparing them to the target. Fig. 3 illustrates the evaluation procedure. However, if the sample's time steps exceed the time steps the network has trained on, the test input will be fed into the network without any modification. This case involves testing the "From scratch" network under biaxial stress loading cases ($\sigma_{11} + \sigma_{23}$ and $\sigma_{11} + \sigma_{22}$). Full-field simulations for these samples did not converge within 100 time steps (with a time increment of 0.01 per step), so they were generated using 200 time steps (with a time increment of 0.005 per step).

4.1. Training results

Initially, a neural network is trained and validated solely utilizing the full-field data set. This is to investigate the necessity and importance of using a transfer learning approach. For simplicity, the architecture of this network will be kept the same as the original network, since it

Table 4
Range of training hyper-parameters and their final values.

| Parameter | Range | From scratch | Fine-tuned |
|-------------------------------------|----------------|--------------|------------|
| Maximum epochs | [100–800] | 729 | 343 |
| Mini batch size | [5–50] | 11 | 40 |
| Initial learning rate | [0.0001–0.001] | 0.001 | 0.0003 |
| Learning rate drop period | [10–100] | 10 | 23 |
| Learning rate drop factor | [0.9–0.99] | 0.9772 | 0.9735 |
| Gradient threshold | [0.9–1.1] | 1.0906 | 1.0538 |
| Number of time steps | | 100 | 2000 |
| Validation loss (MPa ²) | | 99.37 | 149.5 |
| | | 18.27 | 17.92 |

is optimized for a similar task. The network, trained from scratch using the full-field data set, will be referred to as the "From scratch" network throughout this paper.

For the network enhancement (fine-tuning using transfer learning), an issue is the difference in the number of load increments between mean-field (2000 time steps) and full-field simulations (100 time steps). To address this, linear interpolation was used to obtain a full-field data set with 2000 time steps. Then, this 2000 time steps full-field data set was used to train another version of both "From scratch" and fine-tuned network. Upon evaluation, a slight improvement in validation loss was observed for the fine-tuned network, while no improvement was observed for the "From scratch" network. Therefore, the final version of the fine-tuned network was trained using the 2000 time steps full-field data set, whereas the final version of the "From scratch" network was trained using the original 100 time step. Table 4 shows the optimized training hyper-parameters resulted in a validation loss of 99.37 MPa² and 17.92 MPa² for the "From scratch" and fine-tuned network, respectively.

Figs. 4(a) and 4(b) display the learning curves of the "From scratch" network using the original 100 and the interpolated 2000 time steps data, respectively. Figs. 5(a) and 5(b) show the learning curves of the fine-tuned network using the original 100 and interpolated 2000 time steps data, respectively. Note that in the fine-tuned network training plot, the validation loss appears to be lower than the training loss. This is mainly due to the dropout layer and prior training with a large mean-field data set. Therefore, the performance can be impacted by the dropout during training, leading to a higher training loss compared to the validation loss. On the other hand, for the "From scratch" network, only a small training data set was used, containing less variety of features. Therefore the dropout effect on training loss is less profound, resulting in a lower training loss compared to validation.

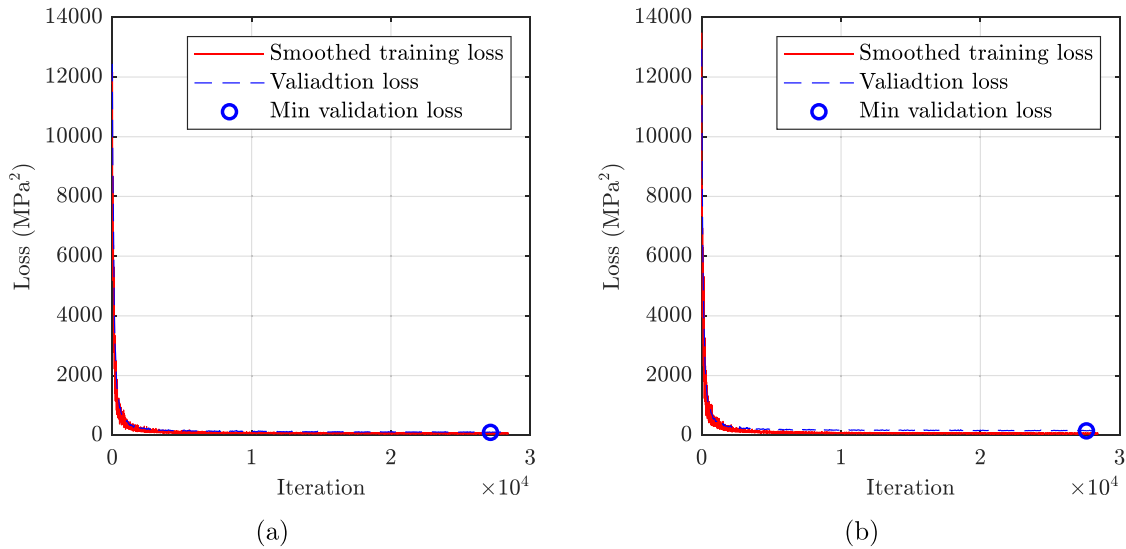


Fig. 4. The learning curve for (a) “From scratch” network trained on original 100 time steps data, (b) “From scratch” network trained on interpolated 2000 time steps data. The iteration with the lowest validation loss, corresponding to the final network, is marked with a blue circle.

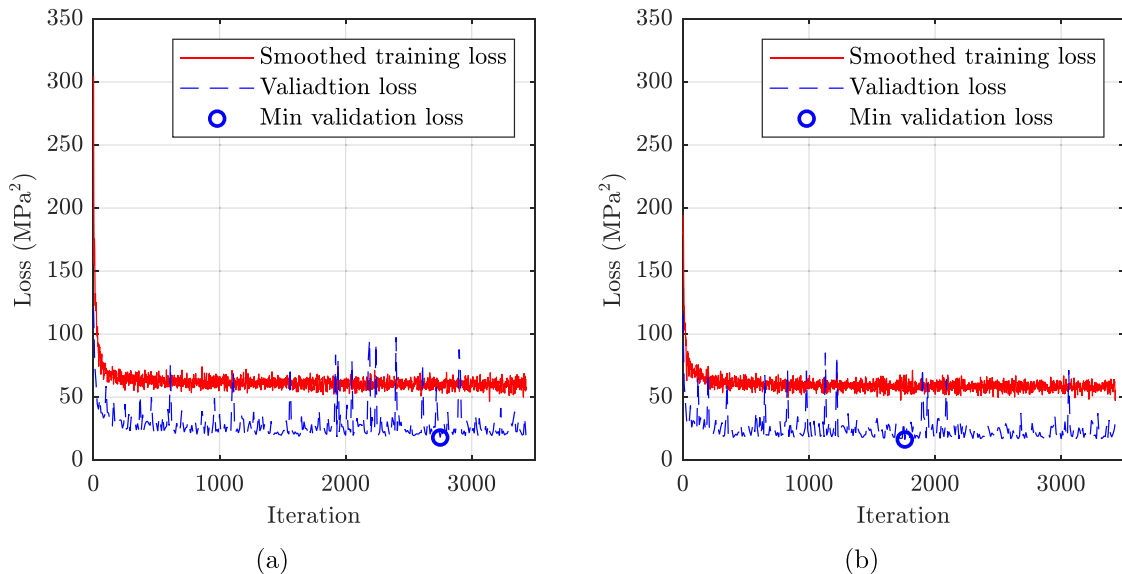


Fig. 5. The learning curve for (a) fine-tuned network trained on original 100 time steps data, and (b) fine-tuned network trained on interpolated 2000 time steps data. The iteration with the lowest validation loss, corresponding to the final network, is marked with a blue circle.

The results for the test data set with random 6D strain path loading can be found in Fig. 6, and their precise values can be found in Table 5 in the supplementary information. In this data set, both Mean Relative Error (MeRE) and Maximum Relative Error (MaRE) values in the original and “From scratch” networks are approximately 3 and 2 times higher than that of the fine-tuned network, respectively. The “From scratch” network also demonstrated better performance compared to the original network in this test. This can be explained by the fact that the test data set consisted of full-field data, whereas the original network was trained on mean-field data. Consequently, there exists a natural divergence between the predictions of the original network and the target values in the test data set.

4.2. Specific loading test result

To evaluate the performance of the neural network under more specific loading conditions, five loading cases were tested. These included: (i) uniaxial stress in normal (σ_{11}) direction, (ii) uniaxial stress

in shear (σ_{12}) direction, (iii) biaxial stress in two normal ($\sigma_{11} + \sigma_{22}$) directions, (iv) biaxial stress in normal and shear ($\sigma_{11} + \sigma_{23}$) directions, (v) plane strain in two normal ($\epsilon_{11} + \epsilon_{22}$) directions. Each loading case involves applying the corresponding strain components for a single cyclic loading from 0 to 0.035, then to -0.035 , and back to 0. For uniaxial and biaxial stress loading states, Digimat-FE will calculate the remaining strain components to keep other stress components null. While for plane strain loading, other strain components are kept at 0. Ten different RVEs were generated from distinct random reference orientation tensors. Additionally, an extra sample with properties close to random 3D orientation was created to demonstrate the models’ performance, given the lack of training data for such cases. Table 5 provides the properties of each RVE. Each RVE was subjected to all five specific loading cases. Samples 1 to 10 were generated from random reference orientation tensors, while Sample 11 represented a random 3D orientation case.

The results for the specific loading test are presented in Fig. 7, and their precise values can be found in Table 6 of the supplementary

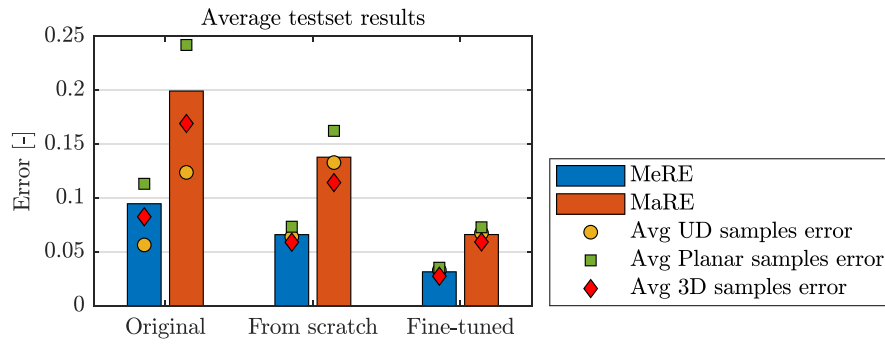


Fig. 6. Average mean relative error (MeRE) and maximum relative error (MaRE) results for the test data set.

Table 5

Orientation tensor components, fiber volume fraction, and sorted eigenvalues ($\lambda_1 \geq \lambda_2 \geq \lambda_3$) for each of the specific test sample.

| Sample | a_{11} | a_{22} | a_{33} | a_{12} | a_{13} | a_{23} | v_f | λ_1 | λ_2 | λ_3 |
|--------|----------|----------|----------|----------|----------|----------|--------|-------------|-------------|-------------|
| #1 | 0.0197 | 0.4926 | 0.4877 | 0.0172 | -0.0223 | 0.2963 | 0.1152 | 0.7865 | 0.1982 | 0.0153 |
| #2 | 0.5193 | 0.3062 | 0.1745 | 0.0613 | -0.1175 | -0.1690 | 0.1320 | 0.6022 | 0.3479 | 0.0499 |
| #3 | 0.6291 | 0.2851 | 0.0858 | 0.3868 | 0.0636 | 0.0443 | 0.1409 | 0.8879 | 0.0785 | 0.0336 |
| #4 | 0.7748 | 0.0946 | 0.1306 | 0.0540 | 0.1060 | -0.0576 | 0.1142 | 0.7946 | 0.1689 | 0.0365 |
| #5 | 0.4435 | 0.2642 | 0.2923 | -0.2709 | -0.0629 | 0.0234 | 0.1314 | 0.6509 | 0.2820 | 0.0670 |
| #6 | 0.2408 | 0.5046 | 0.2546 | -0.0893 | 0.0170 | 0.0649 | 0.1247 | 0.5434 | 0.2661 | 0.1905 |
| #7 | 0.4557 | 0.4060 | 0.1383 | 0.0162 | 0.0231 | -0.0640 | 0.1113 | 0.4606 | 0.4177 | 0.1217 |
| #8 | 0.0644 | 0.5045 | 0.4311 | 0.1398 | -0.1217 | -0.3498 | 0.1289 | 0.8625 | 0.1162 | 0.0213 |
| #9 | 0.2640 | 0.2657 | 0.4703 | -0.1617 | -0.1163 | -0.1418 | 0.1191 | 0.5480 | 0.4243 | 0.0278 |
| #10 | 0.3212 | 0.5478 | 0.1310 | -0.3797 | -0.2013 | 0.2590 | 0.1481 | 0.9604 | 0.0393 | 0.0003 |
| #11 | 0.3261 | 0.3228 | 0.3511 | -0.0061 | -0.0036 | -0.0098 | 0.1250 | 0.3543 | 0.3304 | 0.3153 |

information. In the specific loading test, both average MeRE and MaRE in the original and “From scratch” networks are over 2 and 3.5 times higher, respectively, than that of the fine-tuned network. It is noteworthy that the “From scratch” network performance is worse than the original network, and both MeRE and MaRE are approximately 3 times higher in the specific loading test compared to the test data set.

Fig. 8(a) shows the stress–strain plot for sample 2 under a uniaxial stress loading. The “From scratch” network exhibits poor prediction performance, particularly during the unloading phase. The original network generally outperforms the “From scratch” network. However, inaccuracies are still clear at different stages of the deformation. These inaccuracies can be attributed to the mean-field nature of data used for training the original network. The fine-tuned network demonstrates a significant improvement compared to both other networks. Fig. 8(b) shows the result for sample 4 under a plane strain loading. Once again, the “From scratch” network exhibits poor prediction performance, mainly for the shear stress components (σ_{12} , σ_{23} and σ_{13}). Its predictions fail to capture the general trend of the target, indicating that the amount of data is insufficient for it to handle different types of loading. On the other hand, the original network is capable of making predictions that follow the trend of the target, although it tends to over or under-estimate the stress values. The fine-tuned network gives the most accurate prediction for all stress components.

5. Discussion

To apply the transfer learning technique, different approaches have been explored. Initially, an attempt was given to a partial fine-tuning by freezing some of the parameters of the original model and fine-tuning the remaining trainable parameters using the high-fidelity data. However, it was found that fine-tuning all trainable parameters consistently yields better performance. Both the test data set results and specific loading test results have demonstrated a significant performance gain of the fine-tuned network over “From scratch” and original networks. The specific loading test further highlighted the necessity and benefits of

using the transfer learning approach. When handling loading conditions that are absent from the training data, a large amount of training data is required to provide sufficient information for the network to properly handle these cases. Since the “From scratch” network is trained on a very small full-field data set, its performance is much worse than the fine-tuned network and even the original network. On the other hand, the fine-tuned network was first trained with a significantly larger mean-field data set before fine-tuning with a small full-field data set, enabling it to handle unseen loading conditions with highly accurate predictions. Furthermore, although the training data set lacks samples in the random 3D orientation region, results for the random 3D sample in the specific test demonstrate that the neural network can interpolate the training data and provide enhanced prediction for this sample as well.

In Fig. 8(b), it can be observed that for stress components with low magnitudes, the networks’ relative error is high, however, the absolute error is small. This can be attributed to the use of the half mean square loss function to develop the networks, which emphasizes on large absolute differences between target and output stress values. Using a different loss function, such as mean absolute error or mean absolute percentage error, could potentially reduce the relative error when the stress values are small. However, it is uncertain how different loss functions will impact the overall performance of the networks.

It should be mentioned that to acquire sufficient data (similar to the mean-field data set) to train a network solely relying on full-field data, approximately 90 years of computational time (calculated with 1.875L 3D RVE FFT analysis running on NVIDIA RTX A4500) is needed. Using the transfer learning approach remarkably reduces the amount of required full-field data to develop the neural network.

Using a neural network for modeling SFRC has a few advantages over a traditional full-field method. First, as the neural network uses the orientation tensor and volume fraction as inputs, SFRC can be modeled with desired properties and a full-field level of accuracy without a trial-and-error process of generating a proper RVE realization. Furthermore, the neural network uses weights and biases to store

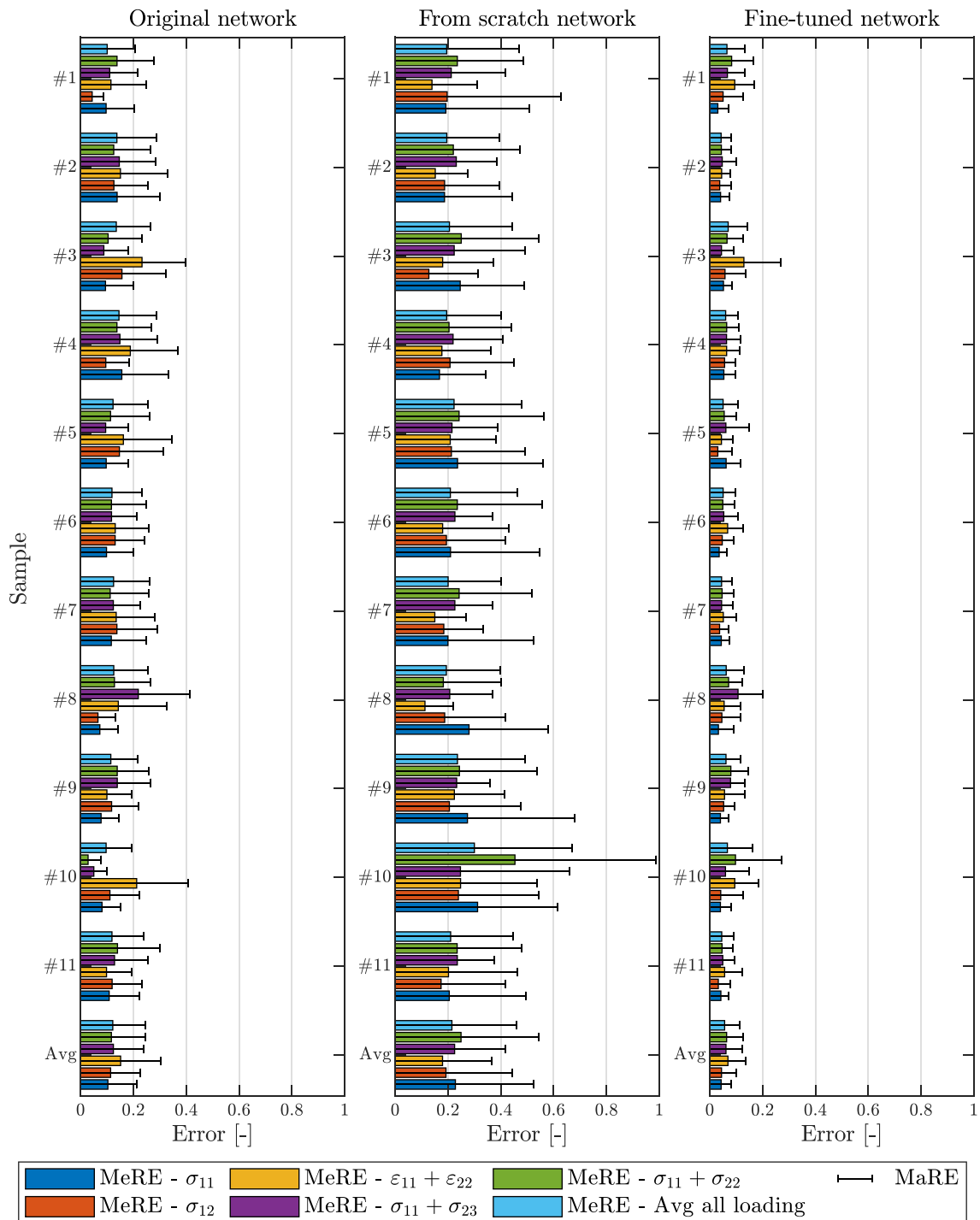


Fig. 7. Results for the specific loading tests for original (left), “From scratch” (middle), and fine-tuned (right) networks. Different colors represent various loading cases, with the line above each bar indicating the MaRE for the corresponding case.

the learned knowledge, significantly reducing the amount of memory and computational time needed. For example, a full-field simulation requires approximately 13,873 MB of graphics memory and 50 min to complete (1.875L 3D RVE FFT analysis running on NVIDIA RTX A4500). In contrast, the size of the neural network only takes up 13.4 MB, and a prediction can be completed within 1 s.

Despite the great performance of the developed network, there are also limitations to be addressed. Firstly, the current deterministic network is unable to address uncertainties. Potential uncertainties could be attributed to the different responses of different RVE realizations,

and existing uncertainties in materials properties (such as properties of natural fibers). There are different possible approaches to capture uncertainties in an RNN, including Bayesian Recurrent Neural Networks (BRNN) (see e.g., [52]) or approximating BRNN using Monte Carlo dropout for RNN [53,54]. Furthermore, although a variety of orientation tensors and fiber volume fractions are considered, the versatility of the network is limited due to the fixed constitutive properties of matrix and fibers. A more comprehensive model should consider a variety of matrix and fiber properties.

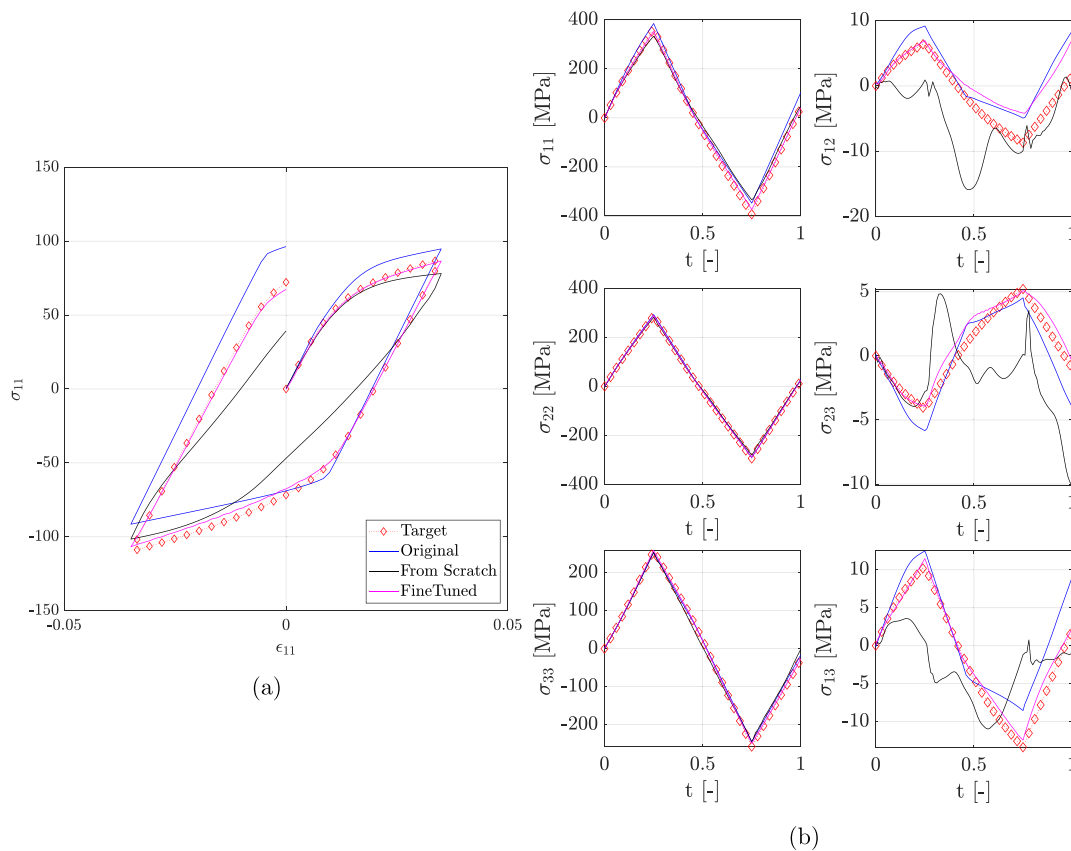


Fig. 8. Specific loading test results for the original, “From scratch” and fine-tuned networks: (a) stress–strain plot for sample 2 under a uniaxial stress (σ_{11}) loading, (b) stress results for sample 4 under a plane strain ($\epsilon_{11} + \epsilon_{22}$) loading.

6. Conclusions

In this paper, we proposed an efficient approach for highly accurate and remarkably efficient surrogate modeling of SFRCs. This approach essentially requires two data sets: (i) a low-fidelity data set with a large amount of data, and (ii) a high-fidelity data set with a limited amount of data. A previously developed RNN model using low-fidelity mean-field simulations were used together with a limited data set from high-fidelity full-field simulations. Transfer Learning technique was used to transfer the knowledge from the initial network to a newly trained network using the high-fidelity data set. Due to the time-series nature of data and the path-dependency of the plastic behavior, a GRU architecture, as a particular form of RNNs, was used. The developed network showed a great performance for predicting high-fidelity full-field simulations in a remarkably quick and efficient manner. Not only accurate analysis of SFRCs is enabled, but also efficient design of new SFRCs (considering required macroscopic properties) is possible using the RNN model. The proposed approach in this paper can be efficiently used for other material systems and other physical problems, as well.

CRedit authorship contribution statement

Hon Lam Cheung: Methodology, Data curation, Formal analysis, Investigation, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Mohsen Mirkhalaf:** Conceptualization, Methodology, Funding acquisition, Investigation, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

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.

Data availability

The codes and a portion of the data generated for this research can be accessed on a GitHub repository (<https://github.com/hlcheungai/SFRC-TransferLearning>). Additional data is available to interested readers upon request.

Acknowledgments

Hon Lam Cheung expresses gratitude to Ehsan Ghane for his insights and engaging discussions that enriched this research. Mohsen Mirkhalaf gratefully acknowledges financial support from the Swedish Research Council (VR grant: 2019-04715).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.compscitech.2023.110359>.

References

- [1] Christoph Unterwiesing, Oliver Brüggemann, Christian Fürst, Effects of different fibers on the properties of short-fiber-reinforced polypropylene composites, *Compos. Sci. Technol.* 103 (2014) 49–55.
- [2] S.M. Mirkhalaf, E.H. Eggels, T.J.H. van Beurden, F. Larsson, M. Fagerström, A finite element based orientation averaging method for predicting elastic properties of short fiber reinforced composites, *Composites B* 202 (2020) 108388.

- [3] P.T. Curtis, M.G. Bader, J.E. Bailey, The stiffness and strength of a polyamide thermoplastic reinforced with glass and carbon fibres, *J. Mater. Sci.* 13 (2) (1978) 377–390.
- [4] N. Sato, T. Kurauchi, S. Sato, O. Kamigaito, Microfailure behaviour of randomly dispersed short fibre reinforced thermoplastic composites obtained by direct SEM observation, *J. Mater. Sci.* 26 (14) (1991) 3891–3898.
- [5] S.M. Mirkhalaf, T.J.H. van Beurden, M. Ekh, F. Larsson, M. Fagerström, An FE-based orientation averaging model for elasto-plastic behavior of short fiber composites, *Int. J. Mech. Sci.* 219 (2022) 107097.
- [6] I. Doghri, L. Brassart, L. Adam, J.S. Gérard, A second-moment incremental formulation for the mean-field homogenization of elasto-plastic composites, *Int. J. Plast.* 27 (3) (2011) 352–371.
- [7] A. Selmi, I. Doghri, L. Adam, Micromechanical simulations of biaxial yield, hardening and plastic flow in short glass fiber reinforced polyamide, *Int. J. Mech. Sci.* 53 (9) (2011) 696–706, cited By 16.
- [8] W. Tian, L. Qi, J. Zhou, J. Liang, Y. Ma, Representative volume element for composites reinforced by spatially randomly distributed discontinuous fibers and its applications, *Compos. Struct.* 131 (2015) 366–373.
- [9] M. Schneider, F. Ospald, M. Kabel, Computational homogenization of elasticity on a staggered grid, *Internat. J. Numer. Methods Engrg.* 105 (9) (2016) 693–720, cited By 86.
- [10] J. Spahn, H. Andrä, M. Kabel, R. Müller, A multiscale approach for modeling progressive damage of composite materials using fast Fourier transforms, *Comput. Methods Appl. Mech. Engrg.* 268 (2014) 871–883.
- [11] L. Qi, W. Tian, J. Zhou, Numerical evaluation of effective elastic properties of composites reinforced by spatially randomly distributed short fibers with certain aspect ratio, *Compos. Struct.* 131 (2015) 843–851.
- [12] S. Bargmann, B. Klusemann, J. Markmann, J.E. Schnabel, K. Schneider, C. Soyarslan, J. Wilmers, Generation of 3D representative volume elements for heterogeneous materials: A review, *Prog. Mater. Sci.* 96 (2018) 322–384.
- [13] S.M. Mirkhalaf, E.H. Eggels, A.T. Anantharanga, F. Larsson, M. Fagerström, Short fiber composites: Computational homogenization vs orientation averaging, in: *ICCM22 2019, Engineers Australia*, 2019, p. 3000.
- [14] B.A. Castricum, M. Fagerström, M. Ekh, F. Larsson, S.M. Mirkhalaf, A computationally efficient coupled multi-scale model for short fiber reinforced composites, *Composites A* 163 (2022) 107233.
- [15] Frederic E. Bock, Roland C. Aydin, Christian J. Cyron, Norbert Huber, Surya R. Kalidindi, Benjamin Klusemann, A review of the application of machine learning and data mining approaches in continuum materials mechanics, *Front. Mater.* 6 (2019).
- [16] Dana Bishara, Yuxi Xie, Wing Kam Liu, Shaofan Li, A state-of-the-art review on machine learning-based multiscale modeling, simulation, homogenization and design of materials, *Arch. Comput. Methods Eng.* 30 (1) (2023) 191–222.
- [17] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M.A. Bessa, Deep learning predicts path-dependent plasticity, *Proc. Natl. Acad. Sci.* 116 (52) (2019) 26414–26420.
- [18] Maysam B. Gorji, Mojtaba Mozaffar, Julian N. Heidenreich, Jian Cao, Dirk Mohr, On the potential of recurrent neural networks for modeling path dependent plasticity, *J. Mech. Phys. Solids* 143 (2020) 103972.
- [19] N. Mentges, B. Dashtbozorg, S.M. Mirkhalaf, A micromechanics-based artificial neural networks model for elastic properties of short fiber composites, *Composit. Part B: Eng.* 213 (2021).
- [20] Colin Bonatti, Dirk Mohr, On the importance of self-consistency in recurrent neural network models representing elasto-plastic solids, *J. Mech. Phys. Solids* 158 (2022) 104697.
- [21] Ling Wu, Ludovic Noels, Recurrent neural networks (RNNs) with dimensionality reduction and break down in computational mechanics; application to multi-scale localization step, *Comput. Methods Appl. Mech. Engrg.* 390 (2022) 114476.
- [22] E Ghane, M Fagerström, SM Mirkhalaf, A multiscale deep learning model for elastic properties of woven composites, *Internat. J. Solids Struct.* 282 (2023) 112452.
- [23] J.C. Michel, P. Suquet, Nonuniform transformation field analysis, *Int. J. Solids Struct.* 40 (25) (2003) 6937–6955, Special issue in Honor of George J. Dvorak.
- [24] S. Rousselet, J.C. Michel, P. Suquet, Nonuniform transformation field analysis of elastic-viscoplastic composites, *Compos. Sci. Technol.* 69 (1) (2009) 22–27, Mechanical Response of Fibre Reinforced Composites.
- [25] Felix Fritzen, Matthias Leuschner, Reduced basis hybrid computational homogenization based on a mixed incremental formulation, *Comput. Methods Appl. Mech. Engrg.* 260 (2013) 143–154.
- [26] J. Yvonnet, Q.-C. He, The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains, *J. Comput. Phys.* 223 (1) (2007) 341–368.
- [27] J. Oliver, M. Caicedo, A.E. Huespe, J.A. Hernández, E. Roubin, Reduced order modeling strategies for computational multiscale fracture, *Comput. Methods Appl. Mech. Engrg.* 313 (2017) 560–595.
- [28] Zeliang Liu, Mark Fleming, Wing Kam Liu, Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials, *Comput. Methods Appl. Mech. Engrg.* 330 (2018) 547–577.
- [29] Cheng Yu, Orion L. Kafka, Wing Kam Liu, Self-consistent clustering analysis for multiscale modeling at finite strains, *Comput. Methods Appl. Mech. Engrg.* 349 (2019) 339–359.
- [30] Zeliang Liu, C.T. Wu, M. Koishi, A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials, *Comput. Methods Appl. Mech. Engrg.* 345 (2019) 1138–1168.
- [31] Zeliang Liu, C.T. Wu, Exploring the 3D architectures of deep material network in data-driven multiscale mechanics, *J. Mech. Phys. Solids* 127 (2019) 20–46.
- [32] Tianyu Huang, Zeliang Liu, C.T. Wu, Wei Chen, Microstructure-guided deep material network for rapid nonlinear material modeling and uncertainty quantification, *Comput. Methods Appl. Mech. Engrg.* 398 (2022) 115197.
- [33] Zeliang Liu, C.T. Wu, M. Koishi, Transfer learning of deep material network for seamless structure–property predictions, *Comput. Mech.* 64 (2) (2019) 451–465.
- [34] Sebastian Gajek, Matti Schneider, Thomas Böhlke, An FE–DMN method for the multiscale analysis of short fiber reinforced plastic components, *Comput. Methods Appl. Mech. Engrg.* 384 (2021) 113952.
- [35] Xin Liu, Su Tian, Fei Tao, Wenbin Yu, A review of artificial neural networks in the constitutive modeling of composite materials, *Composites B* 224 (2021) 109152.
- [36] Ling Wu, Van Dung Nguyen, Nanda Gopala Kilingar, Ludovic Noels, A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths, *Comput. Methods Appl. Mech. Engrg.* 369 (2020) 113234.
- [37] J. Friemann, B. Dashtbozorg, M. Fagerström, SM Mirkhalaf, A micromechanics-based recurrent neural networks model for path-dependent cyclic deformation of short fiber composites, *Internat. J. Numer. Methods Engrg.* 124 (10) (2023) 2292–2314.
- [38] Ehsan Ghane, Martin Fagerström, Mohsen Mirkhalaf, Recurrent neural networks and transfer learning for elasto-plasticity in woven composites, 2023, Pre-print at: <https://doi.org/10.48550/arXiv.2311.13434>, arXiv:2311.13434.
- [39] S. Kammoun, I. Doghri, L. Adam, G. Robert, L. Delannay, First pseudo-grain failure model for inelastic composites with misaligned short fibers, *Composites A* 42 (12) (2011) 1892–1902.
- [40] H. Moulinec, P. Suquet, A numerical method for computing the overall response of nonlinear composites with complex microstructure, *Comput. Methods Appl. Mech. Engrg.* 157 (1) (1998) 69–94.
- [41] J. Zeman, T.W.J. de Geus, J. Vondřejc, R.H.J. Peerlings, M.G.D. Geers, A finite element perspective on nonlinear FFT-based micromechanical simulations, *Internat. J. Numer. Methods Engrg.* 111 (10) (2017) 903–926.
- [42] James Arvo, Iii.4 - fast random rotation matrices, in: David Kirk (Ed.), *Graphics Gems III (IBM Version)*, Morgan Kaufmann, San Francisco, 1992, pp. 117–120.
- [43] T. Kanit, S. Forest, I. Galliet, V. Mounoury, D. Jeulin, Determination of the size of the representative volume element for random composites: statistical and numerical approach, *Int. J. Solids Struct.* 40 (13) (2003) 3647–3679.
- [44] I.M. Gitman, H. Askes, L.J. Sluys, Representative volume: Existence and size determination, *Eng. Fract. Mech.* 74 (16) (2007) 2518–2534.
- [45] S.M. Mirkhalaf, F.M. Andrade Pires, Ricardo Simoes, Determination of the size of the representative volume element (RVE) for the simulation of heterogeneous polymers at finite strains, *Finite Elem. Anal. Des.* 119 (2016) 30–44.
- [46] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734.
- [47] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* 15 (1) (2014) 1929–1958.
- [48] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, in: *Yoshua Bengio, Yann LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*, 2015.
- [49] Xue Ying, An overview of overfitting and its solutions, *J. Phys. Conf. Ser.* 1168 (2) (2019) 022022.
- [50] Razvan Pascanu, Tomas Mikolov, Yoshua Bengio, On the difficulty of training recurrent neural networks, in: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML '13, 2013, pp. III–1310–III–1318*.
- [51] Jasper Snoek, Hugo Larochelle, Ryan P. Adams, Practical Bayesian optimization of machine learning algorithms, in: F. Pereira, C.J. Burges, L. Bottou, K.Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, Vol. 25, 2012.
- [52] Meire Fortunato, Charles Blundell, Oriol Vinyals, Bayesian recurrent neural networks, 2019.
- [53] Yarin Gal, Zoubin Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in: Maria Florina Balcan, Kilian Q. Weinberger (Eds.), *Proceedings of the 33rd International Conference on Machine Learning*, in: *Proceedings of Machine Learning Research*, vol. 48, PMLR, New York, New York, USA, 2016, pp. 1050–1059.
- [54] Yarin Gal, Zoubin Ghahramani, A theoretically grounded application of dropout in recurrent neural networks, 2016.