Noname manuscript No.
(will be inserted by the editor)

Transfer learning of deep material network for seamless structure-property predictions

Zeliang Liu · C.T. Wu · M. Koishi

Received: date / Accepted: date

Abstract Modern materials design requires reliable and consistent structure-property relationships. The paper addresses the need through transfer learning of deep material network (DMN). In the proposed learning strategy, we store the knowledge of a pre-trained network and reuse it to generate the initial structure for a new material via a naive approach. Significant improvements in the training accuracy and learning convergence are attained. Since all the databases share the same base network structure, their fitting parameters can be interpolated to seamlessly create intermediate databases. The new transferred models are shown to outperform the analytical micromechanics methods in predicting the volume fraction effects. We then apply the unified DMN databases to the design of failure properties, where the failure criteria are defined upon the distribution of microscale plastic strains. The Pareto frontier of toughness and ultimate tensile strength is extracted from a large-scale design space enabled by the efficiency of DMN extrapolation.

Keywords Multiscale modeling · Machine learning · Micromechanics · Nonlinear plasticity · Failure analysis · Materials design

Zeliang Liu ✉
Livermore Software Technology Corporation (LSTC), Livermore, CA 94551, USA
E-mail: zlliu@lstc.com

C.T. Wu
Livermore Software Technology Corporation (LSTC), Livermore, CA 94551, USA

M. Koishi
Koishi Laboratory, The Yokohama Rubber Co., LTD., Kanagawa 254-8601, Japan

1 Introduction

Many applications in material science and engineering require good structure-property relationship across different length scales [1,2]. It becomes the key to understanding and quantifying the physical interactions between fine-scale structures in the task of materials design, where the ultimate goal is to determine what microstructure will produce the desired macroscopic properties [3]. Advances in experimental techniques [4] allow designers to gather extensive data on microstructure images and mechanical responses, while finding a consistent and reliable modeling technique for extracting the structure-property relationship from the data remains an elusive challenge.

Classical empirical models with well-designed formulations provide nice approximation to the stress-strain data for a given material, but the fitted parameters usually lose the link to microscale physics and fail to guide materials design. Computational homogenization offers a reliable way to obtain macroscopic material responses from the analysis of a microscale representative volume element (RVE) [5]. The RVE model is constructed in such a way that it can statistically represent the microstructural morphologies and interactions inside a material. Direct numerical simulation (DNS) of RVE based on finite element method (FEM) [6], meshfree methods [7,8] or fast Fourier transform (FFT)-based method [9,10] is accurate, but usually too time-consuming to drive a design process.

Data-driven methods for RVE analysis and multiscale material modeling have attracted increasing attentions in computational mechanics and physics. They have shown great potential to accelerate the online prediction by intelligently utilizing the "big" data from experiments or *a prior* DNS. Many methods used ma-

chine learning models as alternatives to classical empirical models for fitting complex unknown functions, such as artificial neural networks [11,12] and Gaussian process regression [13,14,15]. Efforts have also been made to embed physics into the data-driven models. Oliver et al [16] adopted the proper orthogonal decomposition (POD) to find a low-dimension representation of the fluctuating strain field for an RVE fracture problem. Kalidindi and co-workers [17,18] developed the materials knowledge systems (MKS) for fast RVE analysis by calibrating the Green's function-based convolution kernels from the statistical continuum theories. Liu et al. [19,20,21] proposed the self-consistent clustering analysis (SCA) to model nonlinear history-dependent materials both efficiently and accurately. It uses k-means clustering to group material points with similar mechanical responses and solves the reduced-order integral equation in a self-consistent scheme. The SCA method was reformulated by Yu et al. [22] for elasto-viscoplastic materials under finite deformation.

Recently, Liu et al. [23,24] proposed a mechanistic data-driven multiscale modeling method called deep material network (DMN) which creates high-fidelity reduced representations of both 2D and 3D RVEs. The key ingredients of DMN are a network structure for capturing the complexity of microstructural interactions, and a simple two-layer building block for reproducing the material physics. Due to its efficiency and capability of extrapolation to unknown materials, DMN has been applied to addressing various RVE challenges, such as hyperelastic rubber composite under large deformation, polycrystalline materials with rate-dependent crystal plasticity and carbon fiber reinforced polymer (CFRP) composites. The network structure with physically based parameters also provides a promising tool for materials design.

Although DMN finds proper reduced representations of a given RVE through stochastic gradient descent (SGD), the trained networks for RVEs with different microstructures are not related if the fitting parameters are randomly initialized. This gap between DMN databases is usually not desired by a material designer since it prevents smooth variations of microstructural design parameters. On the other hand, more layers are needed to model complex materials. However, deeper networks become harder to be optimized because there are more fitting parameters and the learning process may encounter more traps with poor local minima.

The concept of transfer learning has been introduced to resolve similar issues appeared in other machine learning tasks. Thrun [25] discussed the knowledge transfer for multi-task training and found that the learning for object recognition becomes easier with knowledge from past task. Raina et al. [26] presented a transfer learning algorithm that constructs an informative Bayesian prior based on previous similar learning problems, and observed large error reduction in binary text classification. In terms of materials informatics, Lubbers et al. [27] used the activations in the VGG-19 model [28] pre-trained from the ImageNet database to characterize the microstructure images. Li et al. [29] also adopted the VGG-19 model to calibrate the difference between original and reconstructed images, and their transfer learning approach outperforms many existing methods in microstructure reconstruction.

In this paper, we present a transfer learning strategy of DMN that constructs a unified set of databases for multiple RVEs. The next section gives a brief overview of the DMN framework and the learning task on 2D particle-reinforced RVEs. In Section 3, a naive approach is proposed to migrate the pre-trained database, and the continuous structure-property relationship is derived via the database interpolation technique. Section 4 provides the training results and compares the transferred models to those from analytical micromechanics methods. In Section 5, the databases from transfer learning are further applied to the design of failure properties of nonlinear plastic materials. Concluding remarks are given in Section 6.

2 Deep material network

2.1 An overview

We first introduce the basic concepts in the framework of deep material network (DMN). As shown in Figure 1, a typical material network is created based on a binary-tree structure connected by two-layer mechanistic building blocks. The complexity of network is determined by the number of layers, denoted by the depth N. Since each node has two child nodes, the bottom layer N initially has 2^{N-1} nodes. However, nodes can be deactivated during the training process. The number of active nodes in the bottom layer is counted by the number N_a, and we have $N_a \leq 2^{N-1}$. During the computation, these active nodes act as the degrees of freedom (DOFs) in the model. An intriguing feature of DMN is that its computational complexity is proportional to N_a.

To track the physical fractions of nodes in the network, each node is associated with a weight w. For the k-th node at Layer i, its weight w_i^k is given by

$$w_i^k = \sum_{j=2^{N-i}(k-1)+1}^{2^{N-i}k} w_N^j = \sum_{j=2^{N-i}(k-1)+1}^{2^{N-i}k} a(z^j), \qquad (1)$$

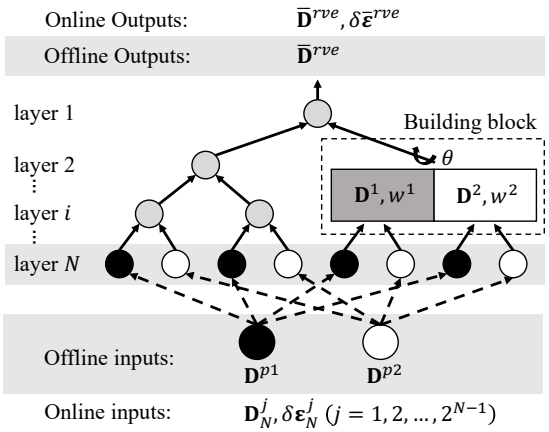


Fig. 1 Illustration of 2D DMN with a binary-tree structure (modified from Liu et al. 2019 [23]). The two-layer building block is presented in the dashed box. Inputs and outputs are demonstrated for small-strain cases.

where the weight at the bottom layer w_N^j is determined by the activation z^j, and the activation function a is chosen as the rectified linear unit (ReLU). As one can see from Eq. (1), all the weights in the network can be computed from z^j. Other than the homogenization operation defined by the weights, a rotation operation is introduced in each building block to enhance model adaptivity, which is controlled by the rotation angle θ_i^k. As a result, the fitting parameters in a 2D DMN with depth N are $z^{j=1,2,\ldots,2^{N-1}}$ and $\theta_{i=1,2,\ldots,N}^{k=1,2,\ldots,2^{i-1}}$. Different from the weights and biases in a generic neural network, all the DMN fitting parameters have physical meanings; hence they should be valid under model exploration.

Analytical homogenization and rotation functions can be derived for the two-layer building block. In this paper, we will focus on 2D small-strain problems under plane strain condition, where the quantities to propagate in the network are the compliance matrix $\mathbf{D}$ and the residual strain $\delta\boldsymbol{\varepsilon}$. Details on the derivations for these analytical functions are provided in Appendix A. Noted that the current 2D small-strain DMN framework can be extended to solve general finite-strain RVE problems with material nonlinearities [23]. We have also shown that a 3D RVE with complex morphology can be effectively described by a network of 3D two-layer building blocks in [24].

Offline stage For a two-phase RVE, the offline inputs of DMN are the elastic compliance matrices $\mathbf{D}^{p1}$ and $\mathbf{D}^{p2}$, which are assigned to the bottom-layer nodes under the following scheme,

$$\mathbf{D}_N^j = \begin{cases} \mathbf{D}^{p1}, & \text{if } j \text{ is odd} \\ \mathbf{D}^{p2}, & \text{if } j \text{ is even} \end{cases}. \tag{2}$$

By applying the homogenization and rotation functions at each building block repeatedly, the information of the compliance matrix is propagated from the bottom layer to the output layer, and the overall quantity $\bar{\mathbf{D}}^{rve}$ can be expressed as

$$\bar{\mathbf{D}}^{rve} = \mathbf{h}\left(\mathbf{D}^{p1}, \mathbf{D}^{p2}, z^{j=1,2,..,2^{N-1}}, \theta_{i=1,\ldots,N}^{k=1,2,\ldots,2^{i-1}}\right). \tag{3}$$

To determine the fitting parameters z^j and θ_i^k, an optimization problem is formulated based on the mean square error (MSE), and its cost function is given by

$$C(z,\theta) = \frac{1}{2N_s}\sum_s \frac{||\bar{\mathbf{D}}_s^{dns} - \bar{\mathbf{D}}_s^{rve}||^2}{||\bar{\mathbf{D}}_s^{dns}||^2} + \lambda L(z), \tag{4}$$

with

$$L(z) = \left(\sum_j a(z^j) - 2^{N-2}\right)^2.$$

where $\bar{\mathbf{D}}_s^{dns}$ is the s-th data points in the training dataset generated by DNS, and N_s is the total number of training samples. The regularization term $\lambda L(z)$ is introduced to constrain the magnitude of z so that the optimization problem is well-posed. Design of experiments (DoE) based on the random sampling is used to generate the offline dataset $\{\mathbf{D}^{p1}, \mathbf{D}^{p2}; \bar{\mathbf{D}}^{dns}\}$. More details on DoE can be found in Appendix B.

Stochastic gradient descent (SGD) is adopted to minimize the cost function. Due to the existence of analytical functions in the building block, the gradients of the cost function with respect to the fitting parameters ∇C can be computed through the backpropagation algorithm. In addition, two model compression methods are developed to accelerate the training speed and improve the convergence [23]:

1. Deletion of nodes with only one child node;
2. Merge of subtrees based on similarity search.

Online stage Although the material network is trained with only linear elastic data, the essential microstructural interactions have been captured by the connected mechanistic building blocks with properly fitted parameters. As a result, the trained network can be extrapolated to unknown material and loading spaces with arbitrary nonlinearities. For 2D small-strain problems with material nonlinearity (e.g. plasticity), the inputs of DMN are $\mathbf{D}_N^j$ and $\delta\boldsymbol{\varepsilon}_N^j$ as shown in Figure 1. In the online stage, each active node in the bottom layer is regarded as an independent material point, whose tangent compliance matrix and residual strain are given by the local constitutive law. The outputs $\bar{\mathbf{D}}^{rve}$ and $\delta\bar{\boldsymbol{\varepsilon}}^{rve}$

of the forward homogenization process can be written as functions of the inputs and the fitting parameters:

$$\bar{\mathbf{D}}^{rve} = \mathbf{h}_D \left(\mathbf{D}_N^{j=1,2,..,2^{N-1}}, z^j, \theta_i^k \right), \tag{5}$$

and

$$\delta\bar{\boldsymbol{\varepsilon}}^{rve} = \mathbf{h}_\varepsilon \left(\delta\boldsymbol{\varepsilon}_N^{j=1,2,..,2^{N-1}}, z^j, \theta_i^k \right). \tag{6}$$

We can then apply macroscopic boundary conditions at the output layer, and extract the unknown components within the macroscopic stress or strain. In the de-homogenization process, the stress and strain data are propagated backward from the output layer to the active nodes in the bottom layer. Newton's method is used to solve the nonlinear system, and the homogenization and de-homogenization processes are iterated by turns until the convergence.

2.2 Problem description

The concept of transfer learning on DMN is demonstrated for 2D RVEs with identical circular particles embedded in the matrix phase. As shown in Figure 2, we generate six RVEs with the volume fraction of particle phase vf_2 equal to 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6. The nearest distance between the particles is explicitly specified in the RVE generation algorithm [30] so that the particles are about evenly distributed in the matrix. Implicit FEM with conforming mesh is used to perform the DNS. For all the RVEs, the ratio of the RVE size to the mesh size is set to be 400, which guarantees that there are at least three layers of element between any two particles. For the RVE with $vf_2 = 0.6$, there are in total 199014 FE nodes and 198212 4-node 2D plane strain elements. To reduce the RVE boundary effect, periodic boundary conditions are used in our analysis.

In the offline stage, we generate 500 DNS samples for each RVE. The first 400 samples are selected as the training dataset, and the remaining 100 samples become the test dataset. Three orthogonal loading cases are needed to determine the 2D compliance matrix of each sample. Since the materials are linear elastic, only one loading step is required for each case. Performance of the SGD training algorithm can be evaluated by the training error e^{tr} and the test error e^{te}. For a given dataset, the error of its s-th sample e_s is calculated as

$$e_s = \frac{||\bar{\mathbf{D}}_s^{dns} - \bar{\mathbf{D}}_s^{rve}||}{||\bar{\mathbf{D}}_s^{dns}||}, \tag{7}$$

and the average error of a dataset with S samples is

$$\bar{e} = \frac{1}{S}\sum_s e_s. \tag{8}$$

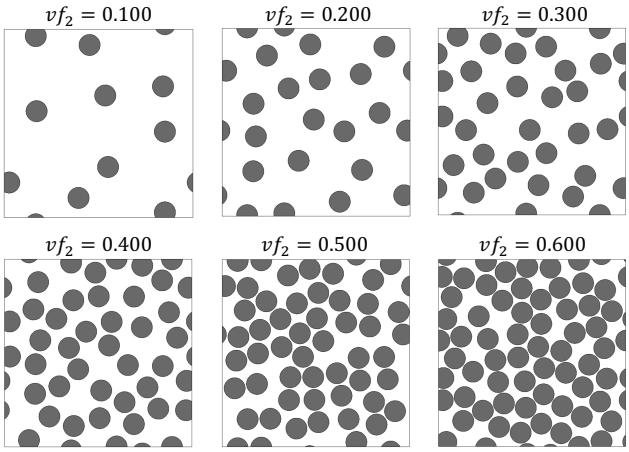


Fig. 2 DNS microstructures of particle-reinforced 2D RVEs with the volume fraction of particle phase vf_2 ranging from 0.1 to 0.6.

Matrix plasticity will be considered in the online stage, while the particle phase remains elastic. The DNS can be time-consuming due to the large amount of DOFs involved in the full-field model. Therefore, it is necessary to improve the efficiency of RVE analysis via model reduction. In Section 5, we will demonstrate a materials design example of toughness and ultimate tensile strength, which is made possible by DMN and the proposed transfer learning technique.

3 Transfer learning

3.1 Motivations

As a common practice, the fitting parameters of DMN are initiated randomly in the optimization process. Since the SGD algorithm converges to a band of local minima close to the "global minimum" , the structure of trained DMN strongly depends on the initial parameters. Meanwhile, the stochasticity of SGD in choosing the mini-batches also introduces uncertainty of the final structure at the early stage of training. With random initialization, the DMN databases trained for different RVEs are not analogous to each other in terms of the topological structure, so that a continuous migration between different database can not be realized through direct interpolation of the fitting parameters. In this aspect, transfer learning of DMN is more advantageous as all the databases can be originated from the same pre-trained database, and hence converge to similar network structures. With the DMN interpolation technique (see Section 3.4), we are able to generate a unified set of databases covering the full-range structure-property relationship.

Another motivation of transfer learning is to improve the convergence of training by reusing the knowledge from pre-trained networks. The benefit from model transfer is usually more evident when the RVEs have similar morphologies, which is exactly the case for our particle-reinforced RVEs. In general, the learning of an RVE with small particle volume fraction vf_2 converges faster and reaches a more compressed model, so that we expect the knowledge of the dilute model can help to ease the training of RVEs with denser particles.

3.2 A naive approach

Before going deep into transfer learning, let us first present a naive approach to creating a continuous set of DMN databases. It starts from a pre-trained network of a base RVE and directly modifies the activations z to achieve the targeted volume fraction for a new RVE. This idea emerges from the observation that a trained material network can accurately represent the volume fraction of its corresponding DNS model.

Assume the volume fraction of particle phase of the pre-trained network is $vf_2^{(b)}$, and the targeted volume fraction of the new database is $vf_2^{(t)}$. Mathematically, the activations are modified as

$$z^{j(t)} = \begin{cases} \dfrac{1-vf_2^{(t)}}{1-vf_2^{(b)}}\, a(z^{j(b)}) & \text{if } j \text{ is odd} \\ \dfrac{vf_2^{(t)}}{vf_2^{(b)}}\, a(z^{j(b)}) & \text{if } j \text{ is even} \end{cases}, \tag{9}$$

where a is the ReLU activation function. Meanwhile, all the rotation angles θ remains unaltered,

$$\theta_i^{k(t)} = \theta_i^{k(b)}. \tag{10}$$

Although no extra training is required (pure extrapolation). the accuracy of the naive approach is limited to some extent, as will be shown in Section 4.2. Therefore, instead of using it as the primary method for database migration, we will adopt the naive approach mainly for the initialization of transfer learning.

3.3 The transfer-learning strategy

The trained DMN of the dilute RVE with $vf_2 = 0.1$ is selected as the starting point of transfer learning. It is first transferred to an approximated model for $vf_2 = 0.2$ through the naive approach and then trained with the DNS dataset of the RVE with $vf_2 = 0.2$. The new DMN database for $vf_2 = 0.2$ obtained from this transfer learning process can be further updated to generate the database for $vf_2 = 0.3$ in the same manner. By repeating these steps following the order below,

$$vf_2 = 0.1 \to 0.2 \to 0.3 \to 0.4 \to 0.5 \to 0.6,$$

the DMN databases for all the RVEs can be generated on the same base with $vf_2 = 0.1$.

The SGD algorithm remains the same for transfer learning, except that all the compression operations, including the node deletion and subtree merging, need to be turned off to keep the network structure consistent across different databases. In specific, the reordering step in subtree merging may completely change the order of the fitting parameters, and hence needs to be avoided in the optimization for transfer learning. However, nodes can still be deactivated during the training, and the number of active nodes in the bottom layer N_a would decrease in the progress of learning. There are other possible paths to perform the transfer-learning task. We start from the DMN for $vf_2 = 0.1$ because it is a better trained model with a more compressed structure, comparing to those for higher volume fractions. As a result, its descendant networks have less chance to get stuck in poor local minima during the optimization.

3.4 Database interpolation

The databases obtained from transfer learning can be interpolated for intermediate volume fractions. To ease the interpolation of activations z, databases for different RVEs are modified so that they share the same total weight W. Since the magnitude of the vector $\{a(z^j)\}$ does not affect the physical properties of DMN, we can scale the activations simultaneously to meet the criterion:

$$\tilde{z}^j = \frac{W}{\sum_i a(z^i)} a(z^j), \tag{11}$$

where $\tilde{z}^j$ represents the activation after the scaling. In this work, we use linear interpolation to determine the intermediate values for both activations and rotational angles. If the volume fractions of the two known databases are given by $vf_2^{(0)}$ and $vf_2^{(1)}$, the fitting parameters for a new database with the volume fraction $vf_2^{(*)}$ in the interval $[vf_2^{(0)}, vf_2^{(1)}]$ are given by

$$\tilde{z}^{j(*)} = N_0 a(\tilde{z}^{j(0)}) + (1-N_0) a(\tilde{z}^{j(1)}) \tag{12}$$

and

$$\theta_i^{k(*)} = N_0 \theta_i^{k(0)} + (1-N_0)\theta_i^{k(1)}, \tag{13}$$

where the linear interpolation function N_0 is

$$N_0 = \frac{vf_2^{(1)} - vf_2^{(*)}}{vf_2^{(1)} - vf_2^{(0)}}. \tag{14}$$

Note that the outputs of network will not transit linearly between two databases, and this is the main difference between the proposed database interpolation and the direct interpolation of outputs (e.g. the compliance matrices $\bar{\mathbf{D}}^{rve(0)}$ and $\bar{\mathbf{D}}^{rve(1)}$). We expect the database interpolation offers higher fidelity since each intermediate point is associated with a physical model based on DMN. It is also possible to introduce other microstructural descriptors, such as the particle shapes, into the sampling space, and the interpolation would become multivariate in higher dimensions.

4 Results and Discussion

4.1 Training results from random initialization

Figure 3 shows the training results with random initialization of fitting parameters for all the RVEs. For each network, the depth N is set to 8, so that there are initially 128 active nodes in the bottom layer. After 10000 epochs of training, the average training errors $\bar{e}^{tr}$ of all six DMNs are reduced to be less than 1.0%. The test dataset is further used to check the quality of a fitted model, and it can be observed from Figure 3 (b) that all the mean test errors $\bar{e}^{te}$ are also less than 1.0%, indicating that there is no over-fitting issue. The maximum sample error appears at the dataset for $vf = 0.6$, which is only $e^{te} = 2.7\%$. It is concluded that DMN is capable of representing the particle-reinforced RVEs with vf_2 ranging from 0.1 to 0.6.

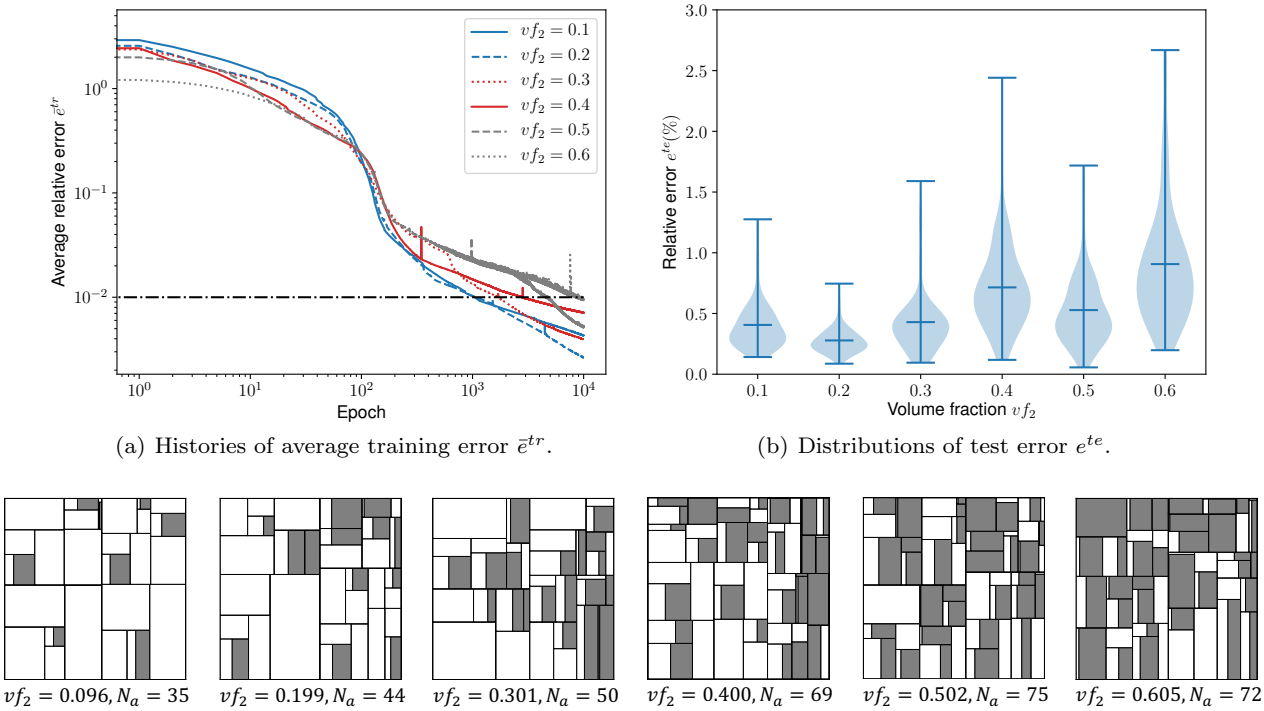
Treemaps of the trained DMNs are presented in Figure 3 (c), with the predicted vf_2 and N_a given under each plot. Although the training dataset only contains mechanical information, DMN is able to extract the volume fraction accurately. The network for $vf_2 = 0.1$ is the most compressed one, with only 35 active nodes remained in the bottom layer. Training for higher vf_2 appears to be less efficient. For $vf_2 = 0.6$, the network has 72 active nodes after the training, while its mean test error is still more than twice the one for $vf_2 = 0.1$. Despite low training and test errors, the network structures for various vf_2 are quite different from each other, making it hard to find a smooth transition between the databases.

4.2 Training results from transfer learning

We first evaluate the accuracy of the naive approach based on the DMN for $vf_2 = 0.1$ with no training, so that it can be considered as an extrapolation problem in terms of the volume fraction. Figure 4 shows the errors on the test datasets of different RVEs. The prediction accuracy decreases as vf_2 moves way from 0.1. At $vf_2 = 0.6$, the mean relative error is around 17.9%, while the maximum error reaches 44.0%. The large errors are contributed by the data points with high contrast of properties between the two phases, for example, the ratio between the elastic moduli could rise up to 1000 in the DoE (see Appendix B). In spite of its simplicity, the naive approach still require SGD training to further improve the accuracy.

The training results from the transfer learning strategy described in Section 3.3 are shown in Figure 5. With the knowledge of pre-trained network for $vf_2 = 0.1$, the convergence of learning for all the RVEs are greatly improved, in the sense that less epochs are required to attain a certain accuracy. For $vf_2 = 0.2, 0.3, 0.4$, the average training errors $\bar{e}^{tr}$ are reduced to be less than 1.0% within only 100 epochs, while the trainings with random initialization cost more than 1000 epochs as we can see from Figure 3 (a). For $vf_2 = 0.6$, the transferred network requires around 1800 epochs to reach 1% average training error, in contrast to nearly 10000 epochs with random initialization. By comparing Figure 5 (b) to Figure 3 (b), one can also observe that the test error e^{te} of DMN after 10000 epochs of training becomes smaller for every RVE. The maximum sample error across all the test datasets reduces from 2.7% to 1.8%.

Figure 5 (c) shows the treemaps of DMN for different RVEs, together with the predicted vf_2 and N_a. Since all the compression operations are turned off during transfer learning, the order of blocks are kept the same during the training, while the block sizes vary with the phase fractions. As all the networks are originated from the one for $vf_2 = 0.1$, their structures represented by the treemaps are analogical to each other. Moreover, the trained networks are also more compressed than those with random initialization. For example, the transfer-learned DMN for $vf_2 = 0.6$ has $N_a = 31$, while the previous one shown in Figure 3 (c) has $N_a = 72$ after trained for the same number of epochs. Since the computational complexity of DMN is proportional to N_a, we expect that the one from transfer learning will be more than two times faster in both offline training and online prediction stages.



(a) Histories of average training error $\bar{e}^{tr}$.

(b) Distributions of test error e^{te}.

(c) Treemaps of DMN after 10000 epochs of training.

Fig. 3 Offline results based on random initialization of learning. The number of layers is $N = 8$. The predicted volume fraction of particle phase vf_2 and the number of active nodes in the bottom layer N_a are provided under each plot in (c).

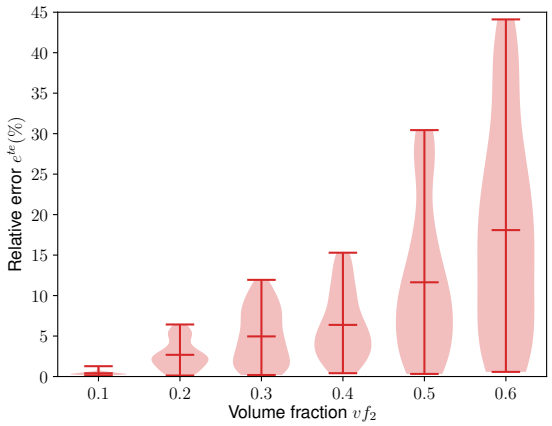


Fig. 4 Distributions of errors on the test datasets from the naive approach. The network for $vf_2 = 0.1$ is chosen as the base of migration.

4.3 Micromechanics: continuous structure-property relationship

Here we apply the transfer learning technique to a classical micromechanics problem: determining the effect of volume fraction on the elastic properties of matrix-inclusion materials. With the DMN database interpolation method, a continuous structure-property relationship can be derived. The naive approach extrapolated from $vf_2 = 0.1$ is adopted to provide coarse predictions. Meanwhile, two analytical micromechanics methods are also evaluated: 1) the Mori-Tanaka method [31] and 2) the self-consistent method [32]. Both methods extend Eshelby's solution [33] for a single inclusion to multi-inclusion systems using mean-field theories.

For the existence of Eshelby's solution in analytical form, both phases are assumed to be isotropic linear elastic in this study. The Young's modulus and Poisson's ratio of the matrix phase are

$$E_1 = 1 \text{ MPa}, \quad \nu_1 = 0.3. \tag{15}$$

Two cases are considered for the particle phase. For hard particle phase, the elastic constants are

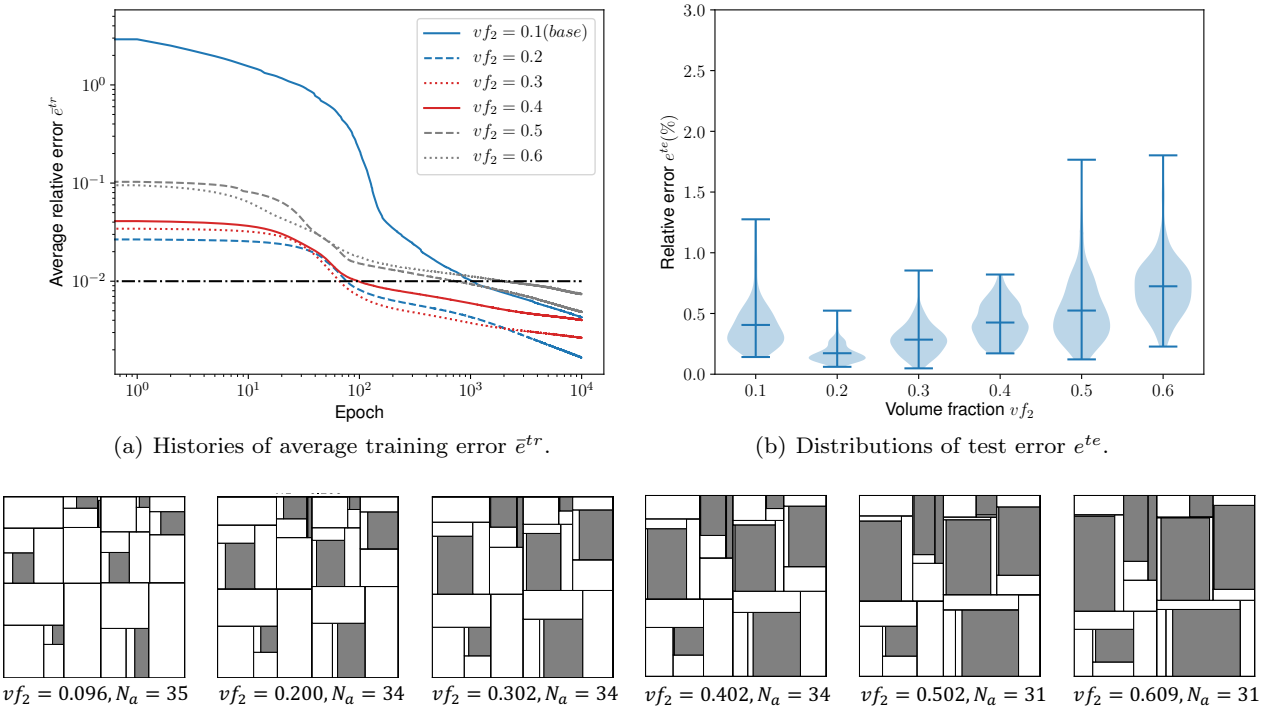
$$E_2 = 1000 \text{ MPa}, \quad \nu_2 = 0.3. \tag{16}$$

For soft particle phase, we have

$$E_2 = 0.01 \text{ MPa}, \quad \nu_2 = 0.3. \tag{17}$$

Although the material inputs fall in the DMN offline sampling space, they don not appear as data points in the training datasets.

The predictions of transverse elastic properties for the 2D RVEs with hard and soft particles are presented

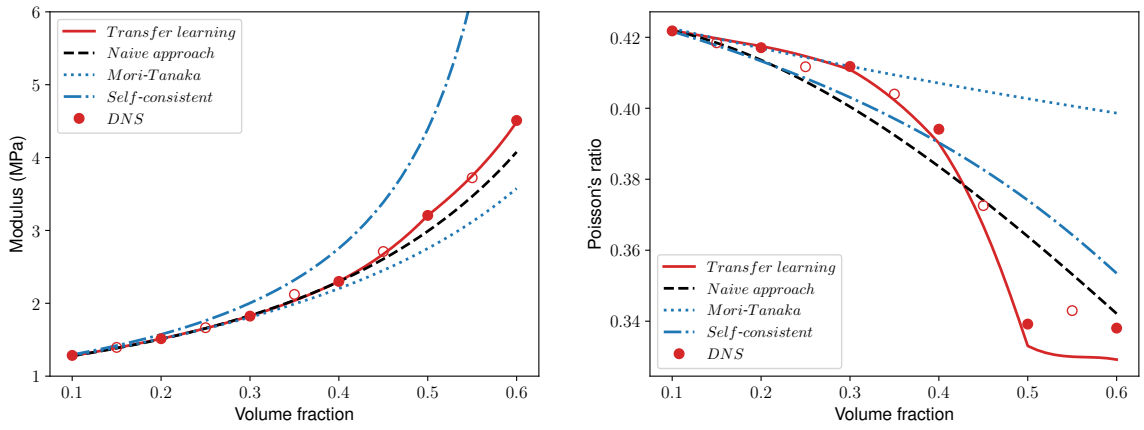


(a) Histories of average training error $\bar{e}^{tr}$.

(b) Distributions of test error e^{te}.

(c) Treemaps of DMN after 10000 epochs of training.

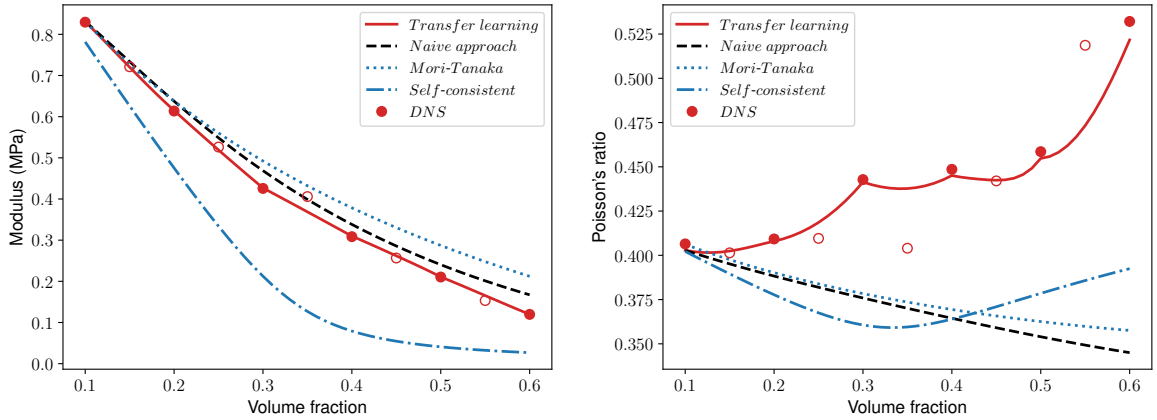
Fig. 5 Transfer-learning results from the pre-trained network for $vf_2 = 0.1$ and $N = 8$. The predicted volume fraction of particle phase vf_2 and the number of active nodes in the bottom layer N_a are provided under each plot in (c).



(a) Transverse Young's modulus.

(b) Transverse Poisson's ratio.

Fig. 6 Effects of vf_2 on the overall elastic properties for RVEs with hard particles: $E_1 = 1$ MPa and $E_2 = 1000$ MPa. Results predicted by transfer learning (solid), naive approach (dashed), Mori-Tanaka (dotted) and self-consistent (dash-dotted) micromechanics methods are provided. DNS references are marked by ∘ and •, where the circles ∘ represent the results for intermediate volume fractions not in the training set.



(a) Transverse Young's modulus. (b) Transverse Poisson's ratio.

Fig. 7 Effects of vf_2 on the overall elastic properties for RVEs with soft particles: $E_1 = 1$ MPa and $E_2 = 0.01$ MPa. Results predicted by transfer learning (solid), naive approach (dashed), Mori-Tanaka (dotted) and self-consistent (dash-dotted) micromechanics methods are provided. DNS references are marked by ∘ and •, where the circles ∘ represent the results for intermediate volume fractions not in the training set.

in Figure 6 and 7, respectively. Other than the RVEs used for training, new RVEs with intermediate volume fractions (e.g. $vf_2 = 0.15$) are also generated and analyzed by DNS. For both cases, DMNs from transfer learning can accurately capture the effects of vf_2 on the transverse Young's modulus and Poisson's ratio in DNS, including the interpolated points (shown as circles ∘ in the plots).

The Mori-Tanaka method predicts the modulus well for small vf_2, but its results diverge from DNS when vf_2 becomes larger than 0.3. The self-consistent method appears to be least accurate. For soft particles, it drastically underestimates the modulus even for a dilute system. The reason is because the self-consistent method treats the whole material as the effective medium, which is suitable for amorphous materials, but not for the particle-reinforced RVEs where the overall properties are dominated by the matrix phase. In this regard, one may consider the generalized self-consistent scheme [34] in which an inclusion is first surrounded by the matrix material and then embedded in the effective medium.

It is noteworthy that the naive approach without any training effort outcompetes both Mori-Tanaka and self-consistent methods, suggesting that it can be treated as an alternative to analytical micromechanics methods for fast estimations. Other than the DMN with transfer learning, all the methods miss the trends of transverse Poisson's ratio, especially for the RVEs with soft particles. One can also observe that the Poisson's ratio has more variation relative to its overall trend. To make the interpolated curves from transfer learning smoother, we can either increase the RVE size or average the homogenization data from multiple RVE realizations for each volume fraction.

5 Design of material toughness and ultimate tensile strength

Up to this point, we have demonstrated how to construct the unified DMN databases through machine learning with knowledge transfer and use them to predict the overall properties of an RVE with given volume fraction and phase properties. In this section, the unified DMN databases are further applied to the design of material toughness and ultimate tensile strength for elasto-plastic RVEs with failure. A common bottleneck of materials design is in the data generation process based on DNS [13], whereas DMN provides a viable way of accelerating the computation without compromising accuracy for each design point. Once the output space is constructed, one can then extract the Pareto frontier of toughness and ultimate strength to guide the selection of volume fraction and phase properties for materials design.

5.1 Nonlinear plasticity

We will first extrapolate the DMNs to nonlinear RVEs with matrix plasticity. The information of effective plastic strain in the matrix will be further used to define the failure criteria. Recall that each individual active node in the bottom layer will be treated as a material point, where the internal history-dependent variables (if any)

are stored. The matrix properties in the elastic regime are

$$E_1 = 100 \text{ MPa}, \quad \nu_1 = 0.3. \tag{18}$$

In the plastic regime, a von Mises yielding law is considered,

$$f_1 = \bar{\sigma}_1 - \sigma_1^Y \leqslant 0, \tag{19}$$

where $\bar{\sigma}_1$ is the equivalent stress. The yield stress σ_1^Y is given by a piece-wise linear hardening law that depends on the equivalent plastic strain ε_1^p:

$$\sigma_1^Y(\varepsilon_1^p) = \begin{cases} 0.1 + 5\varepsilon_1^p & \varepsilon_1^p \in [0, 0.008) \\ 0.124 + 2\varepsilon_1^p & \varepsilon_1^p \in [0.008, \infty) \end{cases} \text{ MPa}. \tag{20}$$

The particle phase is assumed to be isotropic linear elastic with the Poisson's ratio equal to 0.3,

$$\nu_2 = 0.3. \tag{21}$$

Its Young's modulus E_2 is treated as a design parameter that varies during the sampling.

Only uniaxial tension is considered in this paper, and the loading step is set to a relatively small value $\Delta\varepsilon_{11} = 0.0004$ for the accuracy of failure predictions. Regarding DMN evaluations under more complex loading conditions, interested readers are referred to [23].

Figure 8 shows the stress-strain curves given by DNS and DMN with transfer learning under uniaxial tension for $vf_2 = 0.2, 0.4, 0.6$. Two particle moduli are investigated: 1) Hard particles $E_2 = 5000$ MPa; 2) Soft particles $E_2 = 2$ MPa. For the RVE with hard particles and $vf_2 = 0.6$, the corresponding DMN slightly underestimates the yielding stress, while for all the other cases the DMN predictions agree with DNS results very well. All the analyses contain 100 loading steps. A typical DNS took about 1 hour 36 min on 10 CPUs, while the DMN with $N_a = 35$ only took 11.5 s on 1 CPU. This improvement of efficiency enabled by DMN becomes necessary for the materials design problem as will be shown in Section 5.3,

5.2 Failure predictions

Based upon the effective plastic strain in the matrix phase, we define that the composite RVE fails if any of the two conditions is meet:

(a) 10% of the matrix phase has an effective plastic strain ε_1^p above 0.07;
(b) The mean effective plastic strain in the matrix phase $\bar{\varepsilon_1^p}$ is above 0.05.

Note that no progressive damage model is included here, and no failure is considered in the particle phase. Since the distribution of ε_1^p is usually dispersed in a composite RVE, Criterion (a) is more often triggered ($> 95\%$) in the DoE procedure as will be presented in Section 5.3. However, Criterion (b) is introduced mainly for the completeness of the failure criteria. Although not in our design space, an extreme case when Criterion (b) should be used is that the RVE is consist of pure matrix material and the field of ε_1^p is uniform. According to our study, it will only be triggered for RVEs with high fractions of pore-like soft particles, where the strain concentrations are accommodated mostly by the particle phase.

Two variables are used to quantify the failure properties of an RVE. The ultimate tensile strength σ^{TS} is defined as the maximum stress that an RVE can withstand under uniaxial tension loading,

$$\sigma^{TS} = \max\{\sigma_{11}(\varepsilon_{11}), \varepsilon_{11} \in [0, \varepsilon^f]\}, \tag{22}$$

where ε^f is the failure strain. Additionally, the material toughness U^T is defined as the energy per unit volume absorbed before material failure due to load in a fixed loading direction. In the uniaxial tension loading case, the material toughness is simply the area under a stress-strain curve:

$$U^T = \int_0^{\varepsilon^f} \sigma_{11} d\varepsilon_{11}. \tag{23}$$

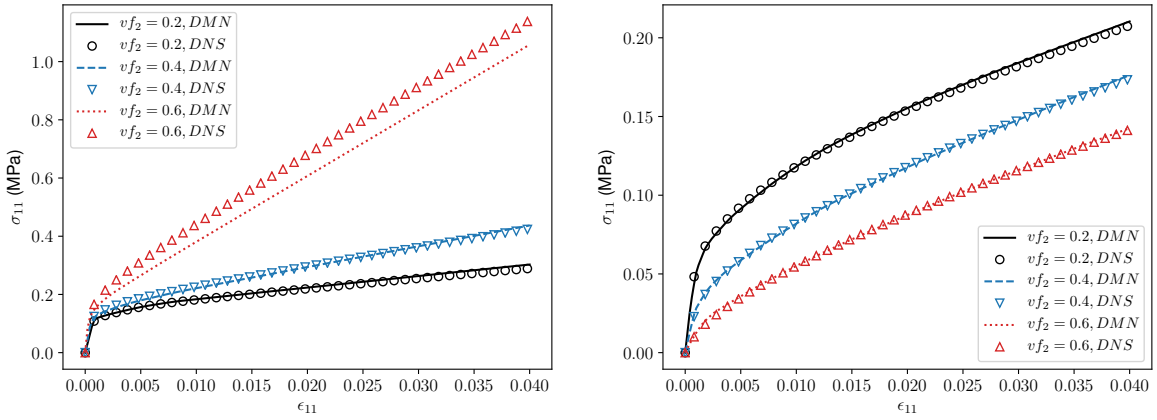
The failure strain ε^f defined by Criterion (b) can be directly computed by taking the weighted average of effective plastic strain in all the active nodes in a network. However, Criterion (a) requires a smooth distribution of the effective plastic strain, while the reduced DMN model can only provide very discrete predictions based on a small number of material points. Therefore, we propose to fit the profile of effective plastic strain from DMN to a continuous log-normal distribution, whose probability density function is given by

$$f_\varepsilon(\varepsilon, s, b, \eta) = \frac{1}{sy\sqrt{2\pi}} \exp\left(-\frac{1}{2}(\frac{\ln y}{s})^2\right)$$

with

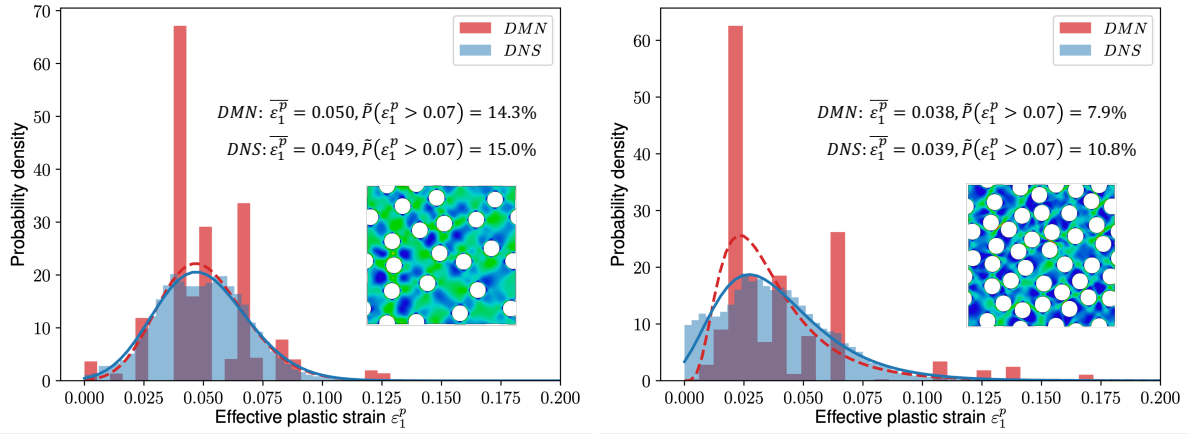
$$y = \frac{\varepsilon - b}{\eta}, \tag{24}$$

where s is the shape parameter, and the distribution is shifted and scaled by the parameters b and η, respectively. We can then determine the critical point on the fitted cumulative distribution function. The fitting process may encounter some difficulties when the overall deformation is small and all local strains ε_1^p are close to



(a) Hard particles: $E_2 = 5000$ MPa. (b) Soft particles: $E_2 = 2$ MPa.

Fig. 8 Stress-strain curves given by DMN and DNS under uniaxial tension for RVEs with (a) hard or (b) soft particles. The elastic modulus of the matrix phase is $E_1 = 100$ MPa. Volume fractions of the particle phase vf_2 are 0.2, 0.4 and 0.6.



(a) RVE with $vf_2 = 0.2$ loaded at $\varepsilon_{11} = 0.036$. (b) RVE with $vf_2 = 0.4$ loaded at $\varepsilon_{11} = 0.020$.

Fig. 9 Distributions of local effective plastic strain ε_1^p in the matrix phase given by DNS and DMN for two typical RVEs with hard particles $E_2 = 5000$ MPa. The DNS contour plot with color range [0.0, 0.2] is presented in each figure. The mean value of effective plastic strain $\overline{\varepsilon_1^p}$ and the fraction $\tilde{P}$ of matrix phase with $\varepsilon_1^p > 0.07$ are also provided.

0. In these cases, the discrete distribution function will be used instead. Fortunately, to our experience, the fittings always converge well at the critical failure points.

Figure 9 shows the distributions of ε_1^p and the corresponding fitted log-normal functions from DNS and DMN for two RVEs with $vf_2 = 0.2, 0.4$, both reinforced by hard particles $E_2 = 5000$ MPa. As we can see from the figures, the DNS profiles can be captured by the log-normal distributions. The DMNs accurately predict the mean values of effective plastic strain. Although the original distributions of ε_1^p from DMNs are discrete with higher local densities, the fitted probability density functions match the ones from DNS well, especially for the tail properties which are important for evaluating the failure Criterion (a).

Table 1 Ultimate tensile strength σ^{TS} (MPa) for RVEs with different particle moduli E_2 and volume fractions vf_2.

	$E_2 = 5000$ MPa			$E_2 = 2$ MPa		
vf_2	0.2	0.4	0.6	0.2	0.4	0.6
DNS	0.275	0.298	0.417	0.244	0.228	0.207
DMN	0.278	0.302	0.381	0.246	0.230	0.198
Error	+1.1%	+1.3%	-8.6%	+0.8%	+0.9%	-4.3%

Table 2 Material toughness U^T (KJ/m^3) for RVEs with different particle moduli E_2 and volume fractions vf_2.

	$E_2 = 5000$ MPa			$E_2 = 2$ MPa		
vf_2	0.2	0.4	0.6	0.2	0.4	0.6
DNS	6.81	4.43	2.61	9.22	9.01	8.26
DMN	6.92	4.65	2.58	9.01	8.83	7.27
Error	+1.6%	+5.0%	-1.1%	-2.3%	-2.0%	-12.2%

Results of ultimate tensile strength and toughness from DMN for all the RVEs evaluated in Figure 8 are summarized in Table 1 and 2, respectively. Note that all DNS results, including those in Figure 9, are calculated based on the original discrete distribution without log-normal fitting. The relative errors of DMN to the DNS results are also provided. One can observe from the tables that the approximation errors of σ^{TS} and U^T are small for low volume fractions $vf_2 = 0.2$ and 0.4, while the errors increase to around 10% at $vf_2 = 0.6$. Nevertheless, the DMN method correctly predicts the dependences of σ^{TS} and U^T on the particle modulus and volume fraction. For a constant E_2 at 5000 MPa, the ultimate tensile strength increases with vf_2, while the trend reverses for soft particles with $E_2 = 2$ MPa. In both cases, the material toughness decreases with vf_2. For the same microstructure with a constant vf_2, the RVE with hard particles typically yields higher ultimate tensile strength, but loses the toughness since it may fail at a lower strain. Recall that these analyses are very challenging because the RVE failure depends on highly localized deformations. The combination of DMN and the profile fitting strategy provides an optimal way of both capturing the global stress-strain responses and reproducing the local information accurately with much less discretization points.

5.3 Materials informatics

Finally, we apply the DMNs with transfer learning to a multi-objective design problem for finding the Pareto frontier of the material toughness and ultimate tensile strength. The design parameters are the modulus E_2 and volume fraction vf_2 of the particle phase considering 3000 points with the following bounds:

$$E_2 = [1, 10000] \text{ MPa}, \quad vf_2 = [0.10, 0.60]. \tag{25}$$

DoE based on Monte Carlo sampling is used to explore the design space.

The Pareto frontier is a set of data points that are "Pareto efficient". It provides all the potential optimal design solutions that one can focus on to maximize the toughness and ultimate tensile strength without searching the whole design space. Mathematically, a data point $X = \{E_2', vf_2'\}$ is defined to be Pareto efficient if there does not exist such a point $X'' = \{E_2'', vf_2''\}$ $(X'' \neq X')$ that the conditions

$$\sigma^{TS}(X'') > \sigma^{TS}(X') \text{ and } U^T(X'') > U^T(X')$$

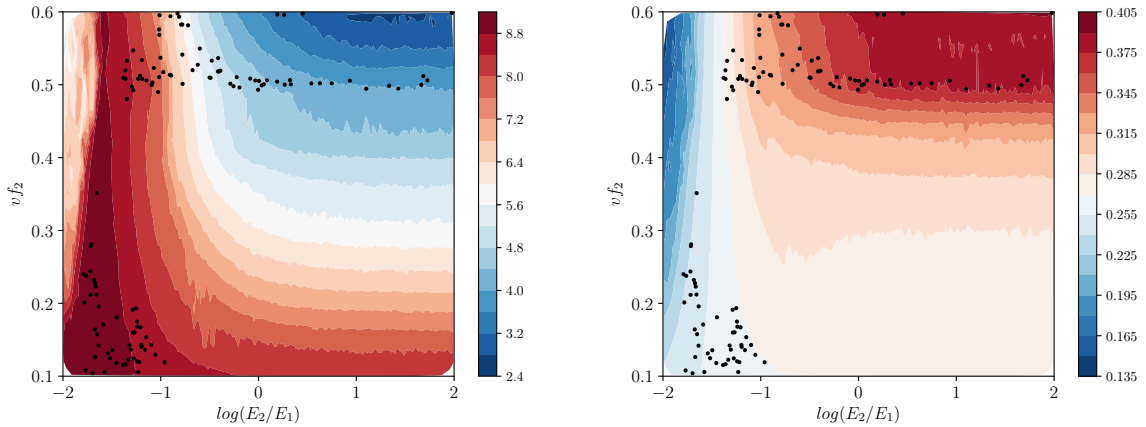
are both satisfied.

Figure 10 presents the contour plots of material toughness and ultimate tensile strength in the design space defined by Eq. (25). The Pareto efficient points are marked by black dots on the plots. As we can see from Figure 10, the region with high toughness locates around $E_2/E_1 = 0.02$. High ultimate tensile strength typically appears at the region with $vf_2 > 0.5$. As long as the particle phase is harder than the matrix phase $E_2 > E_1$, increasing the modulus of particles could not effectively improve the ultimate tensile strength.

Figure 11 shows the output space for all 3000 DoE points, where the Pareto frontier is highlighted by the red dashed line. Interestingly, a large amount of output points are concentrated along a line different from the Pareto frontier. These points are found to be mostly from RVEs with hard particles $E_2/E_1 > 5$, indicating that the failure properties are not sensitive to the modulus of the hard reinforcing particles. One can also observe a transition point with a good trade-off between ultimate tensile strength and toughness at $\sigma^{TS} = 0.32$ MPa and $U^T = 7.86$ KJ/m^3, and its design parameters are $E_2 = 9.62$ MPa and $vf_2 = 0.51$. With the large amount of data generated by DMN, it is possible to fit a machine learning model (e.g. Gaussian process regression, neural network) for inverse design of microstructures, which will be investigated in our future work.

The computational times for DNS and DMN in the offline and online stages are summarized in Table 3. All the calculations are evaluated on 10 Intel® Xeon® E5-2640 CPUs. The DNS analyses are performed using the RVE package based on implicit finite element method in LS-DYNA®. Python libraries are developed for the offline training and online prediction of DMN. Due to the large computational time required by DNS, the prediction stage based on DNS is not feasible and the time is estimated by multiplying the single simulation time by the number of DoE points. Note that the preprocessing time (e.g. microstructure and mesh generation) is not included, but it will become nontrivial if the RVE morphology is very complex.

In DMN, the most time-consuming part is the offline sampling using DNS, which took 71 h 30 min for 6 RVEs with 500 data points per RVE, or 3000 points in total. The offline training based on the transfer learning strategy in Section 3.3 took 12 h 40 min (10000 epochs for each RVE). Fortunately, the offline stage only needs to be performed once, and the created DMN databases can be extrapolated to various material and loading spaces. By virtue of the efficiency of DMN, its online prediction stage for 3000 DoE points was completed in 38 min.



(a) Material toughness KJ/m^3. (b) Ultimate tensile strength (MPa).

Fig. 10 Contour plots of material toughness and ultimate tensile strength as a function of two design parameters E_2 and vf_2. The black dots represent the design points on the Pareto frontier.

Table 3 Computational times for DMN with transfer learning and DNS on 10 Intel® Xeon® E5-2640 CPUs.

	Offline stage for 6 RVEs		Prediction stage (3000 DoE points)
	Sampling (3000 data points)	Training (60000 epochs)	
DNS	0	0	≈ 3000 h (N/A)
DMN	71 h 30 min	12 h 40 min	38 min

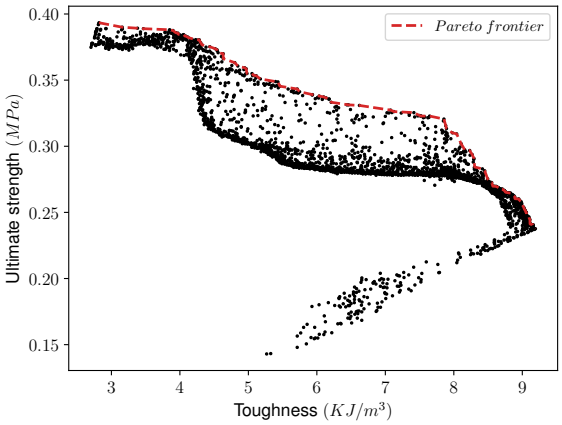


Fig. 11 Output space of material toughness and ultimate tensile strength for a population of 3000 design points from DMN extrapolation. The Pareto frontier is highlighted by the dashed line.

6 Conclusions

In this paper, we have proposed the transfer learning strategy of DMN for structure-property predictions. The knowledge of a pre-trained network is stored and then reused to generate the initial structure for a new RVE through a naive approach. For a set of multiple 2D particle-reinforced RVEs with different phase volume fractions, it is demonstrated that the accuracy and learning convergence can be greatly improved based on the pre-trained DMN with the most compressed structure ($vf_2 = 0.1$).

Since the databases created from transfer learning share the same base structure, intermediate databases can be generated by interpolating the fitting parameters. The unified databases show encouraging micromechanical results of predicting the volume fraction effect on elastic properties, and the idea of database interpolation opens the possibility of incorporating multiple microstructural descriptors to derive a more general design map. Interestingly, the naive approach without SGD training also outperforms the Mori-Tanaka and self-consistent micromechanics methods, suggesting that it can be a good alternative for fast evaluation of unknown RVEs.

The DMN databases are further extrapolated to simulate nonlinear elasto-plastic RVEs, whose failure criteria are defined upon the distribution of local effective plastic strain. With the proposed log-normal fitting procedure, their predictions on the failure properties are shown to be accurate compared to the expensive DNS results for various RVEs. Importantly, the efficiency of DMN enables us to generate a large output space for a multi-objective materials design problem: extracting the Pareto frontier of the material toughness and ultimate tensile strength from 3000 DoE points. From our perspective, it is promising to use the DMN framework with transfer learning in a broad class of materials

design problems where reliable and efficient structure-property relationships are desired.

Acknowledgements The authors give warmly thanks to Dr. John O. Hallquist of LSTC for his support to this research. The support from the Yokohama Rubber Co., LTD under the Yosemite project is also gratefully acknowledged. This is a preprint version of an article published in Computational Mechanics. The final authenticated version is available online at: `http://dx.doi.org/10.1007/s00466-019-01704-4`.

Appendix A Analytical solutions of 2D building block

The 2D DMN framework is originally proposed in our previous work [23]. Analytical solutions are available for the two-layer structure shown in the dashed box within Figure 1, which are derived based on the equilibrium condition

$$\sigma_2^1 = \sigma_2^2, \quad \sigma_3^1 = \sigma_3^2, \tag{A.1}$$

and kinematic constraint

$$\varepsilon_1^1 = \varepsilon_1^2, \tag{A.2}$$

with direction 1 tangential to the interface between the two materials and direction 2 orthogonal to direction 1. Expressions of the components in the compliance matrix $\bar{\mathbf{D}}^r$ after the homogenization operations are

$$\bar{D}_{11}^r = \frac{1}{\Gamma}(D_{11}^1 D_{11}^2), \tag{A.3}$$

$$\bar{D}_{12}^r = \frac{1}{\Gamma}(f_1 D_{12}^1 D_{11}^2 + f_2 D_{12}^2 D_{11}^1),$$

$$\bar{D}_{13}^r = \frac{1}{\Gamma}(f_1 D_{13}^1 D_{11}^2 + f_2 D_{13}^2 D_{11}^1),$$

$$\bar{D}_{22}^r = f_1 D_{22}^1 + f_2 D_{22}^2 - \frac{1}{\Gamma} f_1 f_2 (D_{12}^1 - D_{12}^2)^2,$$

$$\bar{D}_{23}^r = f_1 D_{23}^1 + f_2 D_{23}^2 - \frac{1}{\Gamma} f_1 f_2 (D_{13}^1 - D_{13}^2)(D_{12}^1 - D_{12}^2),$$

$$\bar{D}_{33}^r = f_1 D_{33}^1 + f_2 D_{33}^2 - \frac{1}{\Gamma} f_1 f_2 (D_{13}^1 - D_{13}^2)^2,$$

where

$$\Gamma = f_1 D_{11}^2 + f_2 D_{11}^1 \quad \text{and} \quad f_2 = 1 - f_1.$$

After the homogenization operation, the two-layer structure is rotated. The matrix $\mathbf{R}$ defines the rotation of a second-order tensor through the angle θ under Mandel notation,

$$\mathbf{R}(\theta) = \begin{Bmatrix} \cos^2\theta & \sin^2\theta & \sqrt{2}\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & -\sqrt{2}\sin\theta\cos\theta \\ -\sqrt{2}\sin\theta\cos\theta & \sqrt{2}\sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{Bmatrix}. \tag{A.4}$$

After the rotation operation, the new compliance matrix $\bar{\mathbf{D}}$ is obtained as

$$\bar{\mathbf{D}} = \mathbf{g}(\bar{\mathbf{D}}^r, \theta) = \mathbf{R}(-\theta)\bar{\mathbf{D}}^r\mathbf{R}(\theta). \tag{A.5}$$

In the global network structure, it will become the input of another building block in the upper level.

Similarly, the analytical forms of the residual strain $\delta\bar{\boldsymbol{\varepsilon}}^r$ after the homogenization operation are

$$\delta\bar{\varepsilon}_{11}^r = \frac{1}{\Gamma}(f_1 D_{11}^2 \delta\varepsilon_{11}^1 + f_2 D_{11}^1 \delta\varepsilon_{11}^2), \tag{A.6}$$

$$\delta\bar{\varepsilon}_{22}^r = f_1\delta\varepsilon_{22}^1 + f_2\delta\varepsilon_{22}^2 - \frac{1}{\Gamma} f_1 f_2 (D_{12}^1 - D_{12}^2)(\delta\varepsilon_{11}^1 - \delta\varepsilon_{11}^2),$$

$$\delta\bar{\varepsilon}_{12}^r = f_1\delta\varepsilon_{12}^1 + f_2\delta\varepsilon_{12}^2 - \frac{1}{\Gamma} f_1 f_2 (D_{13}^1 - D_{13}^2)(\delta\varepsilon_{11}^1 - \delta\varepsilon_{11}^2).$$

The overall residual strain $\delta\bar{\boldsymbol{\varepsilon}}$ after the rotation operation is given by

$$\delta\bar{\boldsymbol{\varepsilon}} = \mathbf{R}(-\theta)\delta\bar{\boldsymbol{\varepsilon}}^r. \tag{A.7}$$

Appendix B Design of experiments for DMN training

For the two-phase RVE, the elastic compliance matrices of the two materials are denoted by $\mathbf{D}^{p1}$ and $\mathbf{D}^{p2}$. Both materials are assumed to be orthotropic linear elastic during the sampling. Therefore, each material has four independent design variables: E_{11}, E_{22}, ν_{12} and G_{12}. The compliance matrices in Mandel notation can be expressed as

$$\mathbf{D}^{p1} = \begin{Bmatrix} 1/E_{11}^{p1} & -\nu_{12}^{p1}/E_{22}^{p1} & \\ & 1/E_{22}^{p1} & \\ & & 1/(2G_{12}^{p1}) \end{Bmatrix} \tag{B.1}$$

and

$$\mathbf{D}^{p2} = \begin{Bmatrix} 1/E_{11}^{p2} & -\nu_{12}^{p2}/E_{22}^{p2} & \\ & 1/E_{22}^{p2} & \\ & & 1/(2G_{12}^{p2}) \end{Bmatrix}.$$

To remove the redundancy due to the scaling effect, we have

$$E_{11}^{p1}E_{22}^{p1} = 1, \quad \log_{10}(E_{11}^{p2}E_{22}^{p2}) \in U[-6, 6]. \tag{B.2}$$

The other variables are selected randomly as

$$\log_{10}(E_{22}^{p1}/E_{11}^{p1}) \in U[-1, 1], \quad \log_{10}(E_{22}^{p2}/E_{11}^{p2}) \in U[-1, 1],$$

$$\frac{G_{12}^{p1}}{\sqrt{E_{22}^{p1}E_{11}^{p1}}} \in U[0.25, 0.5], \quad \frac{G_{12}^{p2}}{\sqrt{E_{22}^{p2}E_{11}^{p2}}} \in U[0.25, 0.5],$$

where U represents the uniform distribution. The Poisson's ratios are selected to guarantee that the compliance matrices are always positive definite,

$$\frac{\nu_{12}^{p1}}{\sqrt{E_{22}^{p1}/E_{11}^{p1}}} \in U[0.3, 0.7], \frac{\nu_{12}^{p2}}{\sqrt{E_{22}^{p2}/E_{11}^{p2}}} \in U[0.3, 0.7].$$

Design of experiments are performed based on the Monte Carlo sampling.

References

1. Olson, G.B.: Computational design of hierarchically structured materials. Science **277**(5330), 1237–1242 (1997)
2. Panchal, J.H., Kalidindi, S.R., McDowell, D.L.: Key computational modeling issues in integrated computational materials engineering. Computer-Aided Design **45**(1), 4–25 (2013)
3. McVeigh, C., Vernerey, F., Liu, W.K., Brinson, L.C.: Multiresolution analysis for material design. Computer Methods in Applied Mechanics and Engineering **195**(37), 5053–5076 (2006)
4. Buljac, A., Jailin, C., Mendoza, A., Neggers, J., Taillandier-Thomas, T., Bouterf, A., Smaniotto, B., Hild, F., Roux, S.: Digital volume correlation: review of progress and challenges. Experimental Mechanics **58**(5), 661–708 (2018)
5. Hill, R.: Elastic properties of reinforced solids: some theoretical principles. Journal of the Mechanics and Physics of Solids **11**(5), 357–372 (1963)
6. Feyel, F., Chaboche, J.L.: Fe 2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials. Computer methods in applied mechanics and engineering **183**(3), 309–330 (2000)
7. Wu, C.T., Koishi, M.: Three-dimensional meshfree-enriched finite element formulation for micromechanical hyperelastic modeling of particulate rubber composites. International Journal for Numerical Methods in Engineering **91**(11), 1137–1157 (2012)
8. Wu, C.T., Guo, Y., Askari, E.: Numerical modeling of composite solids using an immersed meshfree galerkin method. Composites Part B: Engineering **45**(1), 1397–1413 (2013)
9. Moulinec, H., Suquet, P.: A numerical method for computing the overall response of nonlinear composites with complex microstructure. Computer Methods in Applied Mechanics and Engineering **157**(1–2), 69 – 94 (1998)
10. De Geus, T., Vondřejc, J., Zeman, J., Peerlings, R., Geers, M.: Finite strain fft-based non-linear solvers made simple. Computer Methods in Applied Mechanics and Engineering **318**, 412–430 (2017)
11. Yvonnet, J., Monteiro, E., He, Q.C.: Computational homogenization method and reduced database model for hyperelastic heterogeneous structures. International Journal for Multiscale Computational Engineering **11**(3), 201–225 (2013)
12. Yang, Z., Yabansu, Y.C., Al-Bahrani, R., Liao, W.k., Choudhary, A.N., Kalidindi, S.R., Agrawal, A.: Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets. Computational Materials Science **151**, 278–287 (2018)
13. Bessa, M., Bostanabad, R., Liu, Z., Hu, A., Apley, D., Brinson, C., Chen, W., Liu, W.: A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality. Computer Methods in Applied Mechanics and Engineering **320**, 633–667 (2017)
14. Raissi, M., Karniadakis, G.E.: Hidden physics models: Machine learning of nonlinear partial differential equations. Journal of Computational Physics **357**, 125–141 (2018)
15. Chen, Z., Huang, T., Shao, Y., Li, Y., Xu, H., Avery, K., Zeng, D., Chen, W., Su, X.: Multiscale finite element modeling of sheet molding compound (smc) composite structure based on stochastic mesostructure reconstruction. Composite Structures **188**, 25–38 (2018)
16. Oliver, J., Caicedo, M., Huespe, A., Hernández, J., Roubin, E.: Reduced order modeling strategies for computational multiscale fracture. Computer Methods in Applied Mechanics and Engineering **313**, 560–595 (2017)
17. Kalidindi, S.R.: Hierarchical materials informatics: novel analytics for materials data. Elsevier (2015)
18. Latypov, M.I., Toth, L.S., Kalidindi, S.R.: Materials knowledge system for nonlinear composites. Computer Methods in Applied Mechanics and Engineering **346**, 180–196 (2019)
19. Liu, Z., Bessa, M., Liu, W.K.: Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials. Computer Methods in Applied Mechanics and Engineering **306**, 319–341 (2016)
20. Liu, Z., Fleming, M., Liu, W.K.: Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials. Computer Methods in Applied Mechanics and Engineering **330**, 547–577 (2018)
21. Liu, Z., Kafka, O.L., Yu, C., Liu, W.K.: Data-driven self-consistent clustering analysis of heterogeneous materials with crystal plasticity. In: Advances in Computational Plasticity, pp. 221–242. Springer (2018)
22. Yu, C., Kafka, O.L., Liu, W.K.: Self-consistent clustering analysis for multiscale modeling at finite strains. Computer Methods in Applied Mechanics and Engineering **349**, 339 – 359 (2019)
23. Liu, Z., Wu, C., Koishi, M.: A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials. Computer Methods in Applied Mechanics and Engineering **345**, 1138–1168 (2019)
24. Liu, Z., Wu, C.: Exploring the 3d architectures of deep material network in data-driven multiscale mechanics. Journal of the Mechanics and Physics of Solids **127**, 20 – 46 (2019)
25. Thrun, S.: Is learning the n-th thing any easier than learning the first? In: Advances in neural information processing systems, pp. 640–646 (1996)
26. Raina, R., Ng, A.Y., Koller, D.: Constructing informative priors using transfer learning. In: Proceedings of the 23rd international conference on Machine learning, pp. 713–720. ACM (2006)
27. Lubbers, N., Lookman, T., Barros, K.: Inferring low-dimensional microstructure representations using convolutional neural networks. Physical Review E **96**(5), 052111 (2017)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
29. Li, X., Zhang, Y., Zhao, H., Burkhart, C., Brinson, L.C., Chen, W.: A transfer learning approach for microstructure reconstruction and structure-property predictions. Scientific reports **8**(1), 13461 (2018)

30. Melro, A., Camanho, P., Pinho, S.: Generation of random distribution of fibres in long-fibre reinforced composites. Composites Science and Technology **68**(9), 2092–2102 (2008)
31. Mori, T., Tanaka, K.: Average stress in matrix and average elastic energy of materials with misfitting inclusions. Acta metallurgica **21**(5), 571–574 (1973)
32. Hill, R.: A self-consistent mechanics of composite materials. Journal of the Mechanics and Physics of Solids **13**(4), 213–222 (1965)
33. Eshelby, J.D.: The determination of the elastic field of an ellipsoidal inclusion, and related problems. Proc. R. Soc. Lond. A **241**(1226), 376–396 (1957)
34. Christensen, R., Lo, K.: Solutions for effective shear properties in three phase sphere and cylinder models. Journal of the Mechanics and Physics of Solids **27**(4), 315–330 (1979)