Neural Computing and Applications (2020) 32:14359–14373
https://doi.org/10.1007/s00521-019-04480-7

A machine-learning-enhanced hierarchical multiscale method for bridging from molecular dynamics to continua

Shaoping Xiao[1] · Renjie Hu[2] · Zhen Li[2] · Siamak Attarian[1] · Kaj-Mikael Björk[3,4] · Amaury Lendasse[3,5]

Received: 1 January 2019 / Accepted: 29 August 2019 / Published online: 18 September 2019

Abstract

In the community of computational materials science, one of the challenges in hierarchical multiscale modeling is information-passing from one scale to another, especially from the molecular model to the continuum model. A machine-learning-enhanced approach, proposed in this paper, provides an alternative solution. In the developed hierarchical multiscale method, molecular dynamics simulations in the molecular model are conducted first to generate a dataset, which represents physical phenomena at the nanoscale. The dataset is then used to train a material failure/defect classification model and stress regression models. Finally, the well-trained models are implemented in the continuum model to study the mechanical behaviors of materials at the macroscale. Multiscale modeling and simulation of a molecule chain and an aluminum crystalline solid are presented as the applications of the proposed method. In addition to support vector machines, extreme learning machines with single-layer neural networks are employed due to their computational efficiency.

Keywords Extreme learning machine · Hierarchical multiscale method · Molecular model · Continuum model

1 Introduction

To accelerate and foster the maturation of technology in designing novel engineering materials and devices, numerical methods [41] play an important role in exploiting new engineering design procedures. Recent developments in nanotechnology demand that molecular building blocks complement and enhance new engineering techniques at the macroscale [69, 71, 72]. Therefore, an aggressive development of new computational methods, including multiscale methods [42], is required to address complex physical phenomena at various length and time scales for the integrated design of multiscale, multifunctional materials and products [48].

Multiscale methods have been categorized into two classes: concurrent and hierarchical multiscale methods. Concurrent multiscale methods [51] employ an appropriate model to couple multiple length/time scales so that simulations at different scales are conducted simultaneously. Most of the developed concurrent multiscale methods are atomistic/continuum coupling methods [1, 6, 13, 51, 62, 66, 70], in which the molecular model is overlapped with the continuum model. One of the main challenges in concurrent multiscale methods

✉ Renjie Hu
renjue-hu@uiowa.edu

Shaoping Xiao
shaoping-xiao@uiowa.edu

Zhen Li
Zhen-li@uiowa.edu

Siamak Attarian
siamak-attarian@uiowa.edu

Kaj-Mikael Björk
kaj-mikael.bjork@arcada.fi

Amaury Lendasse
alendass@Central.UH.EDU

[1] Department of Mechanical Engineering, The University of Iowa, Iowa City, USA

[2] Department of Industrial and System Engineering, The University of Iowa, Iowa City, USA

[3] Arcada University of Applied Sciences, Helsinki, Finland

[4] Hanken School of Economics, Helsinki, Finland

[5] Information and Logistics Technology Department, The University of Houston, Houston, USA

is how to couple the scales [15, 61] without spurious nonphysical phenomena occurring at scale interfaces or overlapping domains. This challenge motivates recent state-of-the-art developments in concurrent multiscale modeling and simulation [21, 54, 63].

On the contrary, the scale-coupling or scale-overlapping challenge in concurrent multiscale methods doesn't exist in hierarchical approaches [60], in which the molecular and continuum models are simulated sequentially. Indeed, researchers pay more attention to how to pass information between scales, especially from the molecular model to the continuum model. Homogenization, including the representative volume element (RVE) techniques, is commonly employed to obtain effective material properties from the molecular model for continuum simulations. The Cauchy–Born (CB) rule [14] is one of the most-used homogenization techniques. It assumes that the lattice vectors deform as line elements within a locally homogeneous deformation so that stress-deformation relationships can be derived. It has been extended to study curved membranes [4] and crystalline solids with temperature effects [74–76]. Recently, Ademiloye et al. [2] proposed a hierarchical multiscale model based on the CB rule to investigate the elastic properties and biomechanical responses of the erythrocyte membrane. Other than the CB rule, Bogdanor et al. [8] adopted a homogenization-based reduced-order multiscale computational model to predict the progressive damage accumulation and failure in composite materials. In addition, Meng et al. [49] developed a cohesive law to characterize the interfacial properties between cellulose nanofibrils by considering the hydrogen bond breaking and reforming at the molecular scale. The developed cohesive model then rendered a superior toughness in continuum simulations.

On the other hand, the RVE techniques utilize a periodic subdomain in the molecular model to calculate effective material properties, which are then passed to the continuum model. Jiang et al. [36] used molecular dynamics (MD) to predict basic mechanical behaviors, including elastic and damage responses to external loading conditions. Then, the MD results were used to generate a preliminary elastodamage model for continuum simulations. Grabowski et al. [20] developed a multiscale electro-mechanical model to study carbon nanotube (CNT) composites. They used MD simulations to provide information about the elastic properties and density of polymeric material and CNTs for simulations at the meso- and macroscales. Subramanian et al. [59] presented a framework of point-information-to-continuum-level analysis to characterize the behavior of CNT composites. In their method, the stochastic distributions obtained from MD simulations provide a basis to simulate local variations of matrix properties in the continuum model. In addition, Ghaffari et al. [17] studied the lubricant between sliding solids via MD simulations and passed the friction coefficient to the continuum model to predict the rolling contact fatigue life.

Recently, due to the development of computer technology and explosive data generation and consumption, data science and analysis via machine learning (ML) has become an efficient tool in science and engineering [55]. Machine learning has been widely applied in the biomedical engineering domain for real-time simulations. Jahya et al. [34] used a validated finite element (FE) model of the prostate and its surrounding structures to generate training data for deep learning (DL), i.e., ML with artificial neural networks (ANNs). Then, the trained ANN could predict a three-dimensional phantom deformation based on given input variables, which include boundary conditions. In another application, Lorente et al. [45] used ML regression models, including three tree-based methods and two simpler ones, to simulate the biomechanical behaviors of the human liver during breathing in real time.

In the community of computational mechanics and materials science, data sciences and informatics [35] have been used to accelerate materials development and deployment. Kalidindi et al. [37] described a few computational protocols to accelerate significantly the process of building microstructure informatics in the integrated computational materials engineering infrastructure. Gupta et al. [23] used a data science approach to establish reduced-order linkages between the material microscale internal structure and its associated macroscale properties. Their training dataset was generated from the mechanical responses of an ensemble of representative microstructures based on FE simulations. In molecular simulations, ML has been used to predict molecular properties [24] for accurate atomistic simulations. Chen et al. [9] presented a highly accurate force field for molybdenum by ML on a large material dataset. In a similar work, Glielmo et al. [18] proposed a novel scheme to predict atomic forces as vector quantities by ML regression. Artrith and Urban [5] implemented ANN potentials in atomistic material simulations to study titanium dioxide (TiO_2). Bélisle et al. [7] evaluated a few ML techniques to predict material properties from training data obtained via MD simulations. In addition, Ibanez et al. [31, 32] discussed the difficulties in the data-driven or data-intensive approaches that link experimental data to numerical simulations. They proposed the solution by using a data-driven inverse approach to generate the whole constitutive manifold from few complex experimental tests.

A few works have been done by using ML in multiscale modeling and simulation. Matouš et al. [47] reviewed predictive nonlinear theories for multiscale modeling of heterogeneous materials. They discussed a predictive image-based multiscale material model, in which statistically representative unit cells were generated via ML to optimally preserve the statistical description of the microstructure. Le et al. [39] employed ANNs and proposed a decoupled computational homogenization method for nonlinear elastic materials. In their method, the training samples of the effective potential were computed through random sampling in the parameter space, and then ANNs were used to approximate the surface response and to derive the macroscopic stress and tangent tensor components. Liu et al. [43] developed a data-driven approach to predict the behavior of general heterogeneous materials under inelastic deformation. One of their innovations was using an unsupervised clustering algorithm to homogenize the local features of the material microstructure into a group of clusters. This research group [44] recently developed a microstructural database based on the self-consistent clustering analysis to accurately predict a nonlinear material response. In another pioneering work, Fritzen and Kunc [16] used a data-driven approach to investigate the nonlinear behavior of materials with a three-dimensional microstructure. They performed finite element method (FEM) simulations on the microstructural level first, and the generated simulation data then served as input for a reduced-order model at the macroscale level.

As reviewed in the above, the information (such as material properties) in a small scale is averaged via homogenization in most existing hierarchical multiscale methods [20–28]. Some physical phenomena may be smeared when such information is passed to a large scale. The motivation of this work is to find an efficient and effective way to pass the information between different scales without losing detailed physical phenomena. Our contribution is to propose a novel data-driven approach by using ML to bridge from the molecular model to the continuum model in a hierarchical multiscale framework. First, MD simulations are conducted to generate the dataset, including training and testing sets, in which the input variables contain deformation and temperature while the output variables are stress components and material failure mode. Then, the generated data is used to train several ML classification and regression models. Finally, well-trained learning machines are directly implemented in continuum simulations to predict material failure mode and stress components. In this approach, neither constitutive relations nor effective material properties are explicitly derived as achieved in existing hierarchical multiscale methods. The learning machines serve as "black boxes" to replace constitutive relations and failure mode decisions in the continuum model. Such "black boxes" are trained based on the dataset from molecular simulations; therefore, the proposed scheme is physical-based and data-driven. As one of the most important emerging ML techniques, DL greatly improved the progress in speech recognition, image recognition, game playing, etc. [46] However, DL is suffering from many limitations, including the expensive training process [19], the requirement for large datasets [19, 46], and a lack of interpretation ability [19, 46]. In our study, extreme learning machines (ELMs) were employed to enhance the multiscale method proposed in this paper because of their speed manner in both training and prediction with high accuracy. We also compared the DL method with the ELMs and showed that the ELMs not only outperform DL in this problem but more importantly has a superior advantage on the training and prediction times, coinciding with previous studies [30, 63].

The outline of this paper is described as follows. After the introduction, MD simulations and data collection are described in Sect. 2. The examples include a one-dimensional molecule chain and an aluminum crystalline solid. Section 3 describes the proposed ML-enhanced hierarchical multiscale modeling as well as the ML regression and classification algorithms. Details about the training processes are also explained. Continuum modeling and simulation with the implementation of ML-trained predictive models are discussed in Sect. 4 followed by conclusions and future outlook.

2 Molecular dynamics simulations

Molecular dynamics has been a powerful tool to elucidate physical phenomena at the nanoscale [50, 68, 71, 73]. In MD simulations, the atoms or molecules in the simulated system follow the laws of classical mechanics. The motion of an atom, e.g., atom i, with mass m_i, is due to its interaction with other atoms in the system according to Newton's second law:

$$m_i \vec{a}_i = \vec{f}_i = -\nabla U\left(\vec{r}_i\right) \tag{1}$$

where $\vec{a}_i$ is the acceleration of atom i , and the interatomic force, $\vec{f}_i$, applied on atom i is derived from the total potential energy U, which is a function of the position vector, $\vec{r}_i$.

In MD simulations, the accelerations are calculated via Eq. 1. The velocity Verlet method is commonly used to conduct time integrations within the time step of Δt to update velocities and displacements:

$$u(t+\Delta t) = u(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 \tag{2}$$

$$a(t+\Delta t) = \frac{f\left(\vec{r}(t+\Delta t)\right)}{m} \tag{3}$$

$$v(t+\Delta t) = v(t) + \frac{\Delta t}{2}[a(t) + a(t+\Delta t)] \tag{4}$$

In this paper, MD simulations are conducted to generate a dataset. Two examples are considered here: a one-dimensional Lennard-Jones molecule chain and an aluminum (AL) crystalline solid. The collected dataset includes training and testing samples, which are used to train predictive models, i.e., learning machines, for predicting stresses and material failure/defect modes. Then, the well-trained predictive models are implemented in the continuum model of the proposed hierarchical multiscale method.

2.1 One-dimensional Lennard-Jones molecule chain

We first consider a one-dimensional molecule chain, which contains 1000 atoms, with periodic boundary conditions. Each atom has a mass of 1.993×10^{-26} kg. The following Lennard-Jones (LJ) 6–12 potential function is employed to describe the interatomic interactions between the nearest neighboring atoms,

$$U(l) = 4\varepsilon\left[\frac{1}{4}\left(\frac{l_0}{l}\right)^{12} - \frac{1}{2}\left(\frac{l_0}{l}\right)^{6}\right] \quad (5)$$

where $l_0 = 1$ nm is the initial bond length, l is the deformed bond length, and $\varepsilon = 1.65 \times 10^{-18}$ J is the depth of the energy well.

At given deformation gradient (F) and temperature (T), a canonical (NVT) MD simulation is conducted until the molecule chain reaches a thermodynamic equilibrium state. Then, the atomic-level Cauchy stress tensor [77] can be calculated via

$$\boldsymbol{\sigma} = \frac{\boldsymbol{1}}{\boldsymbol{V}}\sum\left(\frac{\boldsymbol{1}}{\boldsymbol{2}}\sum_{i\neq j} \boldsymbol{r}_{ij}\otimes\boldsymbol{f}_{ij}\right) \quad (6)$$

where $\boldsymbol{r}_{ij}(= \boldsymbol{r}_i - \boldsymbol{r}_j)$ represents the interatomic distance between atoms i and j, and $\otimes$ denotes the tensor product of two vectors. The cross-sectional area is assumed as 1 nm^2. The sign convention adopted for interatomic forces, $\boldsymbol{f}_{ij}$, is positive for attraction and negative for repulsion. Accordingly, a positive stress indicates tension and a negative stress indicates compression in the one-dimensional case here. It shall be noted that a temperature-related homogenization technique can theoretically derive the stress-deformation gradient relation [74–76]. However, it was developed for crystalline solids only and has difficulties for other materials without regular lattice structures.

Figure 1 shows the collected stress-deformation gradient data at various temperatures: 300 K, 1000 K and 2000 K. Due to the nature of LJ potential, the stress-deformation gradient relations exhibit hyperelasticity in compression. Although the relations, shown in Fig. 1, have no big differences between each other when tensile deformation is small, the failure stresses vary significantly at different temperatures. Figure 2 demonstrates the failure modes in terms of deformation gradient and temperature. At a higher temperature, the LJ molecule chain would be broken at a smaller deformation gradient.

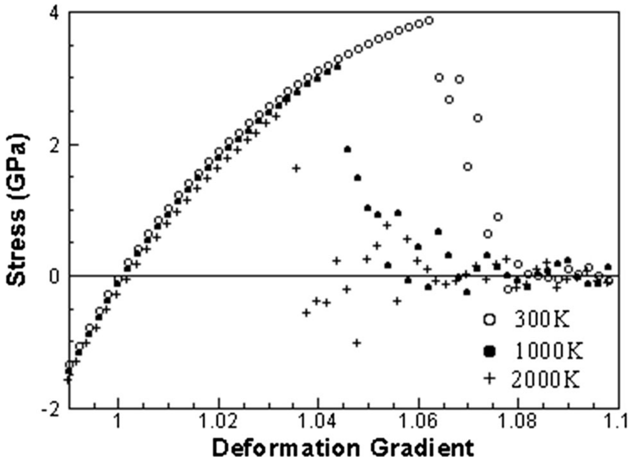


Fig. 1 Stress-deformation gradient data at various temperatures

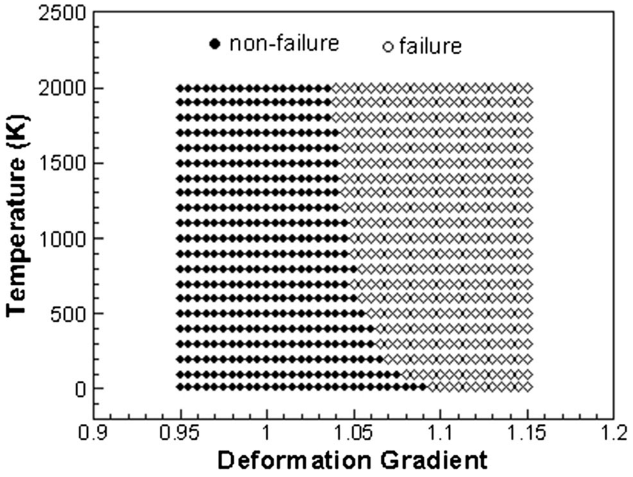


Fig. 2 Failure and non-failure domains

A number of MD simulations are conducted at various deformation gradients and temperatures to generate the dataset. To pass the information from the molecular model to the continuum model in this example, two predictive models need to be trained. One is a material failure predictive model, in which the input variables are deformation gradient and temperature, while the output is a Boolean to represent two different material failure modes: failure or non-failure. The other is a stress predictive model, in which the input variables are deformation gradient and temperature while the output target is stress. The above two predictive models will be trained by using ML classification and regression methods, respectively, based on the corresponding dataset collected from MD simulations.

2.2 Aluminum crystalline solid

In another example, an FCC Al crystalline solid with <100> orientations in the X, Y and Z directions is studied at a temperature of 300 K via MD simulations. The simulated atomistic model contains twelve lattice units in each direction so that the total number of atoms is 6912. The potential function is a many-body interatomic potential developed by Mishin et al. [52]. Molecular dynamics simulations in this study are carried out with the Larger-scale Atomic Molecular Massively Parallel Simulator (LAMMPS) [53]. The time step is 1 fs. Periodic boundary conditions are employed in each direction, and the deformation, represented by the engineering strains, is applied in the $X - Y$ plane only to approximate a two-dimensional simulation model with the plane strain condition.

We first elucidate the mechanical behavior of this FCC Al crystal under uniaxial tension at 0% shear strain. In MD simulations, the first step is equilibration, in which the simulated model is equilibrated in the isothermal-isobaric (NPT) ensemble at a pressure of 0 bar for 20 ps. Then, the model is deformed at a constant strain rate of $5 \times 10^{-7}\ \mathrm{fs}^{-1}$ in the X direction only, while no deformations are applied in the Y and Z directions. This is a strain-free condition, which is different from the stress-free conditions used in Tschopp and McDowell's studies [64]. Various strain rates are tested, and the above strain rate is chosen due to its minimal effect on stress calculation.

Figure 3 shows the stress–strain relation of the FCC Al crystal under uniaxial tension up to 50% strain. Due to the Poisson effect, σ_y is non-zero although $\varepsilon_y = 0$. It can be seen that there are three regions separated by the discontinuities. In Region 1, the Al crystal maintains an almost perfect crystalline structure so that the stress–strain relation represents nearly elasticity until point A, at which a discontinuity occurs. Then, dislocation nucleation and growth are observed in Region 2.

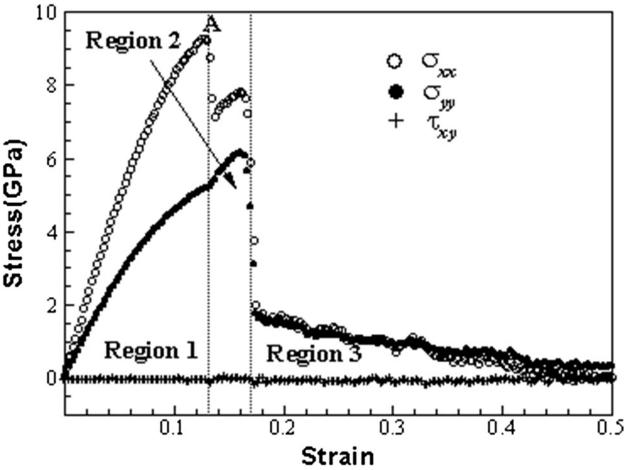


Fig. 3 Stress–strain relation of an FCC Al crystal in uniaxial tension at 300 K

Dislocation is one type of defect in crystals, and it occurs when the atoms are out of position in the crystal structure. Dislocations can be identified by the centro-symmetry parameter (CSP) [38], which is a metric to quantify the local loss of centro-symmetry at an atomic position. The non-centro-symmetry environment is characteristic for most crystal defects, including dislocation. The CSP is calculated as

$$CSP = \sum_{i=1}^{N/2} \left| \boldsymbol{r}_i + \boldsymbol{r}_{i+N/2} \right|^2 \tag{7}$$

where $\boldsymbol{r}_i$ and $\boldsymbol{r}_{i+N/2}$ are position vectors from the central atom to a pair of opposite neighbors, and $N = 12$ is the number of nearest neighbors taken into account for FCC crystals. When an FCC crystal is pulled along a <100> direction, dislocation always occurs on a <111> plane and in a <110> direction as shown in Fig. 4.

When the simulated Al crystal is continuously elongated, voids nucleate and grow. This phenomenon occurs when the stress–strain relationship reaches to Region 3, as shown in Fig. 3. At the microscale, the nucleation and growth of voids can be viewed as the initiation and growth of damage. Once the damage reaches the pre-defined threshold, a microscale crack initiates. Figure 5 illustrates configurations of Al crystal with voids at various strains.

The uniaxial tension simulation described above is conducted at $\varepsilon_{xy} = 0$. We conduct additional biaxial tension, compression and tension-compression simulations with the normal strains ranging from -20% to 50%. The similar physical phenomena of dislocation and void nucleation and growth are observed. The same simulations are repeated at various shear strains ε_{xy} in a range of -20% to 20% to generate the dataset for ML methods to train the predictive models for continuum modeling and simulation.

We only consider the plane strain condition at a room temperature of 300 K in this example. Therefore, in the collected dataset, the input variable of training and testing samples are strains, including ε_x, ε_y, and ε_{xy}. The output variables are either stresses or material defect modes according to the targets of predictive models. Three predictive models are trained in this example. The first model trained by ML classification is to predict material defect modes, including defect-free, dislocation and void modes. The output target is -1, 0, or 1 to represent three different material defect modes, respectively. It shall be noted that Fig. 3 only shows the case of uniaxial tension with $\varepsilon_{xy} = 0$ and $\varepsilon_y = 0$. Indeed, the domains of the material defect modes are three-dimensional in terms of strain components, as shown in Fig. 6. The other two predictive models, trained by ML regression, are for stress prediction. One is

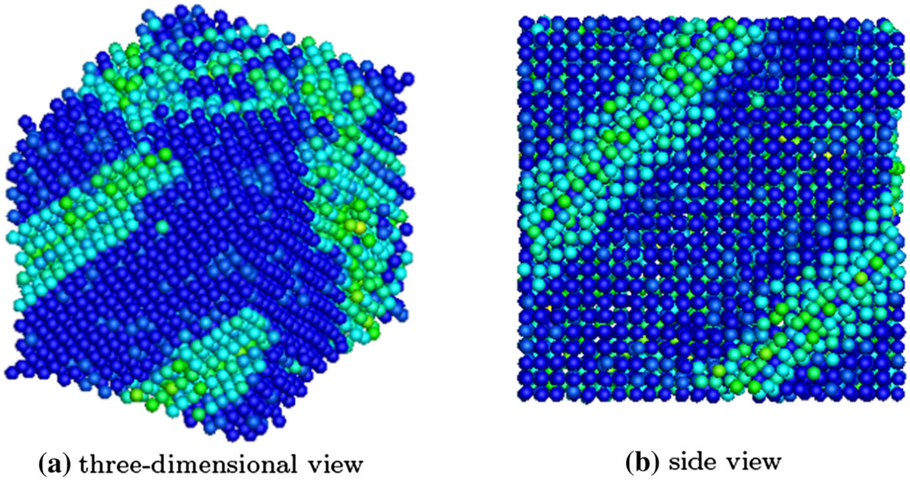


Fig. 4 Nucleation of dislocation in Al crystal at 15% strain

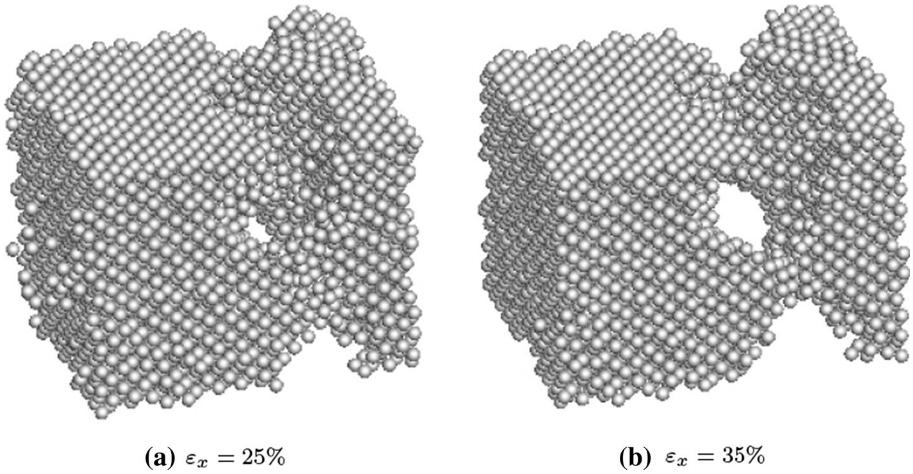

(a) $\varepsilon_x = 25\%$
(b) $\varepsilon_x = 35\%$

Fig. 5 Voids nucleation and growth in the Al crystal

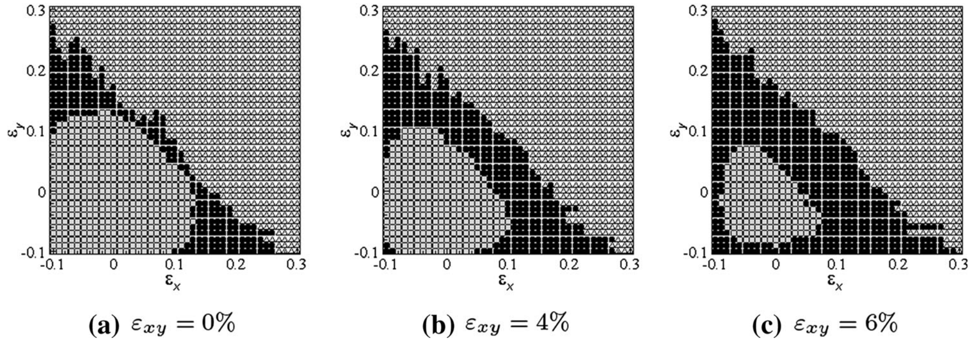


Fig. 6 Domain of material defect modes (○ defect-free; ● dislocation; Δ void)

to predict the stresses when the material is at the defect-free mode, and the other is to predict stresses at the dislocation mode. The output targets of both models are stress components: σ_x, σ_y, and σ_{xy}. We assume that material failure occurs at the location where the void mode is predicted.

3 Hierarchical multiscale modeling with machine learning

Figure 7 illustrates the proposed hierarchical multiscale modeling enhanced by ML for solving dynamic problems of continuum and structural mechanics. As discussed in the

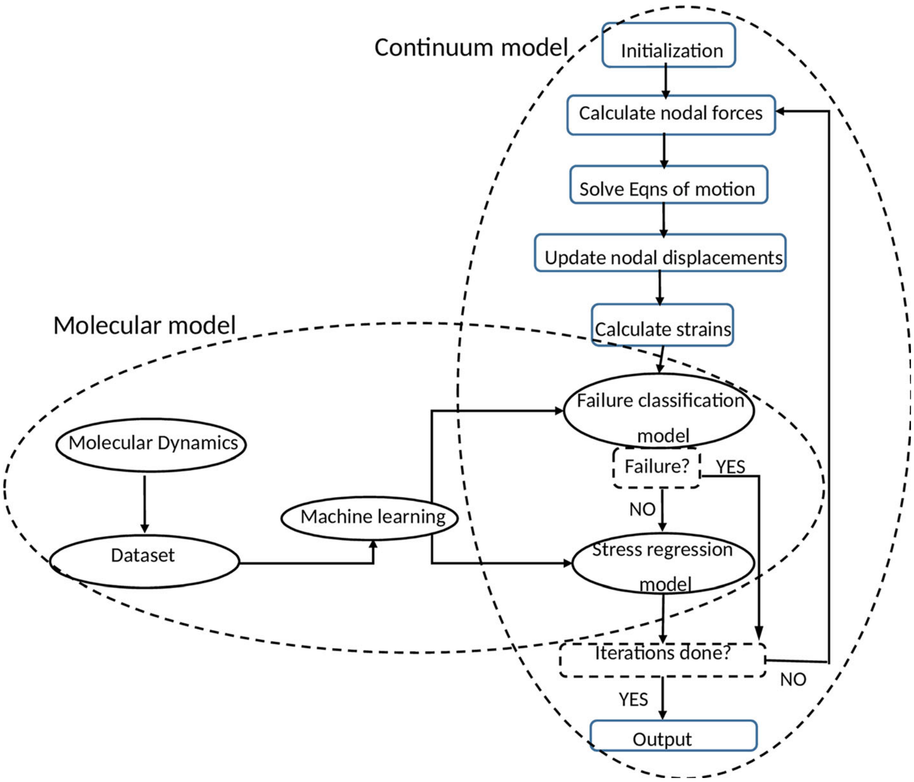


Fig. 7 Hierarchical multiscale modeling enhanced by machine learning

previous section, two types of data are collected via molecular dynamics simulations. In both types of data, the input variables include strain (or deformation gradient) and temperature if the temperature effect is considered. However, the output variables are different. One type of data is for ML to train a material failure classification model, and the output variable is an integer to represent material failure or defect mode. Another type of data is for ML to train stress regression models so that the output variables are stress components. Both failure classification and stress regression models are implemented in the continuum model to substitute the explicit constitutive relations, which are commonly used in conventional continuum simulations. Consequently, information is passed from the molecular model to the continuum model in our hierarchical multiscale method as shown in Fig. 7. The details about training predictive models by ML are described below.

Support vector machines (SVMs) are supervised ML algorithms commonly used for regression [57], classification [11], and outliers detection. The estimated output in an SVM nonlinear regression algorithm can be written as

$$\tilde{y}(\boldsymbol{x}) = \sum_{J=1}^{N} \left(\alpha_J - \alpha_J^*\right) K(\boldsymbol{x}_J, \boldsymbol{x}) + b \tag{8}$$

where N is the number of training samples, α_J and α_J^* are Lagrange multipliers, and b is the bias. K is the kernel function, which transforms the training data from the input to the feature space. There are a few optimization algorithms [57], which can be used to minimize the error function and to generate predictive models in SVM regression. Similarly, SVMs can conduct classification tasks by constructing hyperplanes in a multidimensional space that separates various labeled cases.

SVMs are adopted in the example of the one-dimensional LJ molecule chain. The input and output variables of data samples are:

$$\boldsymbol{x} = \begin{Bmatrix} F \\ T \end{Bmatrix} \quad y_R = \{\sigma\} \quad or \quad y_C = \{-1 \ or \ 1\} \tag{9}$$

where F is the deformation gradient, T is the temperature, and σ is the Cauchy stress. In addition, 1 represents material non-failure mode while -1 represents material failure mode. It shall be noted that 90% of the collected dataset is used as the training set, with the remaining 10% comprising the testing set.

There are a total of 436 data samples of $\{\boldsymbol{x}_I, \ (y_R)_I\}$ used to train and test the stress regression model. The radial basis function (RBF) [10], as the kernel function, in our SVM training is expressed as

$$K(\boldsymbol{x}_J, \boldsymbol{x}) = e^{-\gamma\|\boldsymbol{x}_J - \boldsymbol{x}\|^2} \tag{10}$$

where $\gamma = 0.1$. The Normalized Mean Square Error (NMSE) of testing data is 0.38%.

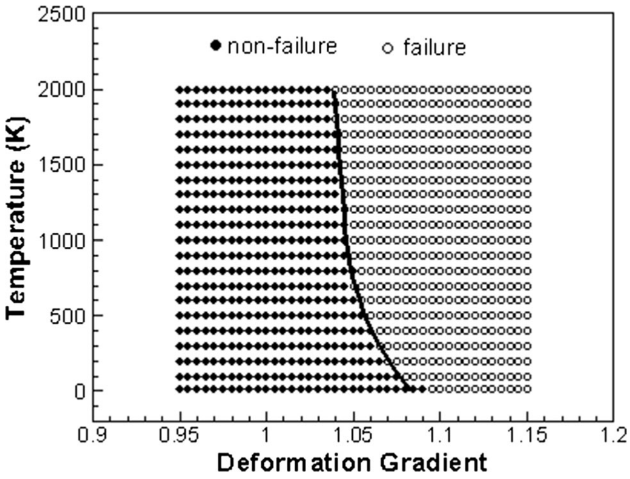


Fig. 8 The material non-failure/failure interface predicted by machine learning

To train the material failure classification model, a total of 861 data samples of $\left\{\boldsymbol{x}_I,\ (y_C)_I\right\}$ are used, and the model accuracy for the testing set is 99%. Figure 8 demonstrates the material non-failure/failure interface predicted via the learning machine and compared with the collected data samples.

In the framework of hierarchical multiscale modeling described in Fig. 7, if the FEM is employed in continuum simulations, the well-trained predictive models will be used to predict material failure mode and stresses at each quadrature point at every iteration. It has been shown that SVMs are computationally intensive [3]. Therefore, in the example of the Al crystalline solid, Extreme Learning Machines (ELMs) [22, 27], one of the important emergent ML techniques, are adopted. It has been shown that an ELM is a fast training method for Single-Layer Feedforward Networks (SLFNs) [30]. Therefore, an SLFN is used in our ELM model, as shown in Fig. 9. Although a SLFN has three layers of neurons, the term of "Single" stands for the only layer of linear/nonlinear neurons in the model, and it is the hidden layer. In addition, the input layer provides input variables, while the output layer targets output variables.

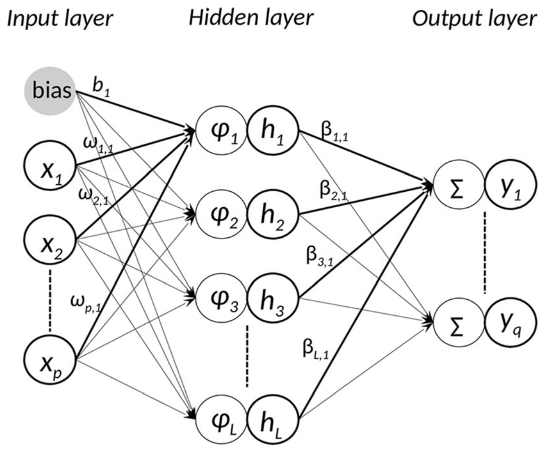


Fig. 9 An ELM model with a single-layer neural network

The ELM model in Fig. 9 is described below. Considering a set of N distinct training samples $(\boldsymbol{x}_I,\ \boldsymbol{y}_I)$ where $I \in [1,\ N]$, input data $\boldsymbol{x}_I \in R^p$ and corresponding output data $\boldsymbol{y}_I \in R^q$. There are L hidden neurons transforming the input data into a different representation $\boldsymbol{h}$, which is used as output layer weights to estimate outputs. There are two steps in the transformation. First, the data is projected into the hidden layer using the input layer weights, $\boldsymbol{\omega}$, and biases, $\boldsymbol{b}$. Then, the projected data is transformed via the transformation functions, φ. As a result, the transformation of the input data can be mathematically expressed as

$$h_j = \varphi_j\left(\boldsymbol{\omega}_j^T\boldsymbol{x} + b_j\right) = \varphi_j\left(\sum_{i=1}^{p}\omega_{ij}x_i + b_j\right) \qquad j = 1\ldots L \tag{11}$$

It is known that the hidden layer is not constrained to have only one type of transformation function, i.e., activation function, in neurons. Different functions can be used: linear, sigmoid, hyperbolic tangent, and some radial basis functions (RBFs) [10, 26]. Particularly, linear neurons learn linear relationships between input and output data. In addition, the RBF neurons use distances between samples and centroids as inputs, and any norm including L^1, L^2, and L^∞ norms of distances can be used. Consequently, the estimated outputs of the kth training sample are then calculated as

$$\begin{aligned}\tilde{y}_k &= \boldsymbol{\beta}_k^T\boldsymbol{h} = \sum_{j=1}^{L}\boldsymbol{\beta_{jk}\varphi_j}\left(\sum_{i=1}^{p}\boldsymbol{\omega_{ij}x_i} + \boldsymbol{b_j}\right)\\ &= y_k + \varepsilon_k \qquad\qquad k = 1\ldots q\end{aligned} \tag{12}$$

where ε_k is the noise, i.e., the estimate residual. The tenfold cross-validation technique [40] is used in ELMs training.

It shall be noted that an ELM model can be used for multi-layered feedforward neural networks [29] as well. As a difference from traditional ML theories, the hidden neurons don't need to be tuned in ELM models, and all the parameters of hidden neurons can be randomly generated and independent of the training data. Indeed, an ELM [3, 28] can universally approximate any continuous function with almost any nonlinear and piecewise hidden neurons. Therefore, it can solve any regression problem with a desired accuracy when enough hidden neurons and training data are given. In addition, multi-label classification problems [65] can be handled similarly. On the other hand, unlike the back-propagation [33] training procedure, there is no dependence between the input and output weights so a non-iterative linear solution for the output weights becomes possible. Therefore, it provides a speedup of 5 orders of magnitude in ELMs compared to Multilayer Perceptron (MLP) [25, 56], or a speedup of 6 orders of magnitude compared to SVMs [12] based on the studies of [3].

In the Al crystal example, the ELM classification model, used to classify material defect modes, is trained with a total of 153,704 data samples, in which the input and output variables are

$$\boldsymbol{x} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{Bmatrix}, y_C = \{1,\ 0\ or - 1\} \tag{13}$$

where ε_{xx}, ε_{yy} and ε_{xy} are engineering strains. Among the three output classes, 1 represents the material defect-free mode, 0 represents the material dislocation mode, and -1 represents the material void mode. In the ELM classification without overfitting, various numbers of neurons with hyperbolic tangent functions are used, and the accuracies are listed in Table 1. It shall be noted that the ELM neural network with 2000 neurons is implemented in the continuum model because it utilizes fewer neurons but achieves a sufficient accuracy.

It has been shown in Fig. 3 that the stress is dramatically reduced once voids are nucleated. Therefore, we consider material failure occurring at the location where material void mode is predicted. Consequently, two ELM regression models are trained to predict stresses. One is for the material defect-free mode, and the other is for the material dislocation mode. The input and output variables of data samples are

$$\boldsymbol{x} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{Bmatrix}, \boldsymbol{y}_R = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} \tag{14}$$

where σ_{xx}, σ_{yy}, and σ_{xy} are Cauchy stresses. There are 25,278 and 46,323 data samples collected to train those two predictive models, respectively. In the ELM regression model for defect-free stress prediction, there are 500 nonlinear neurons with hyperbolic tangent functions. The NMSEs of the three stress components are 0.01%, 0.02%, and 0.12%, respectively. In the ELM regression neural network for dislocation stress prediction, there are a total of 3000 nonlinear neurons with hyperbolic tangent functions in the hidden layer. The NMSEs of the three output stress components are 0.02%, 0.02%, and 0.78%, respectively. It can be seen that the learning machine to predict stresses in the material dislocation mode has larger NMSEs than the learning machine for the material defect-free mode. The reason is that stresses vary due to dislocation nucleation, growth, and movement in the material dislocation mode. More features, including dislocation density and orientation, need to be considered in ELM training. In addition, temperature could be an additional feature if the temperature effect is considered.

Table 1 Accuracies of ELM classifications

Number of neurons	Classification accuracy (%)
100	93.7
500	97.4
1000	97.9
2000	98.5
3000	98.7
20,000	99.1

To demonstrate the advantages of the ELMs over other DL methods in our studies, we trained a Deep Neural Network (DNN) with the same datasets and compared the training/prediction cost with the ELMs. The Deep Learning Toolbox in MATLAB was employed. Given the sizes of datasets in our studies, we chose a DNN with three hidden layers, which has 32 neurons in the first layer, 16 in the second layer and 8 in the last layer, to reach the same level of prediction performances (the same NMSEs for regression or the same accuracy for classification) as the ELMs. The results, shown in Table 2, illustrate that the ELMs were faster than the DNN. Specifically, the training processes of ELMs were on average 33.6 times faster and their prediction times were on average 12.4 times shorter than the DNN.

4 Continuum modeling and simulation

4.1 one-dimensional Lennard-Jones molecule chain

After the predictive models are trained via SVM methods, shock wave propagation in a 200 μm-long L-J molecule chain is modeled as a continua and simulated by using FEM. The chain is discretized with 200 two-node elements. Each element contains 1000 atoms, and the nodal mass is 1.993×10^{-23} kg. The cross-sectional area is still 1 nm^2, and the time step is set as 5.0 fs.

We first study the wave propagation along the molecule chain at 300 K when a square pulse load is applied at the left end of the chain while the right end is free. This compressive pulse has an amplitude of 1 nN and a period of 0.5 ps. Figure 10 shows the configurations of stress shock wave propagation at four various times: $t_1 = 1$ ps, $t_2 = 2$ ps, $t_3 = 3.5$ ps, and $t_4 = 4.5$ ps. After the compressive pulse is applied, there is a compressive stress

Table 2 Comparison between ELM and DNN

	DNN TrTime[d]	ELM TrTime	DNN PrdTime[e]	ELM PrdTime	DNN NMSE/ ACC	ELM NMSE/ ACC	DNN size	ELM size
Stable StrsRgr[a]	13.4	0.51	0.162	0.005	0.0006	0.0005	32, 16, 8	500
Disloc StrsRgr[b]	567.45	8.13	0.21	0.07	0.0068	0.0027	32, 16, 8	3000
MtrlF Clsf[c]	80.43	17.08	0.35	0.21	98% (ACC)	99% (ACC)	32, 16, 8	2000

[a] Stress regression in material stable (defect-free) mode; [b] stress regression in material dislocation mode; [c] material failure classification; [d] training time in s; [e] prediction time in s

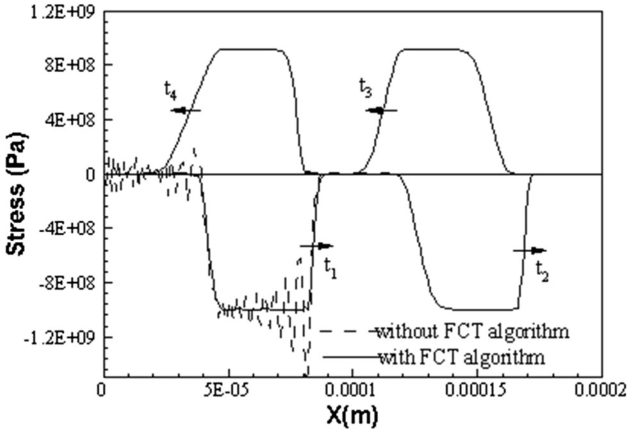


Fig. 10 Stress shock wave propagation in LJ molecule chain subjected to a square pulse load at 300 K

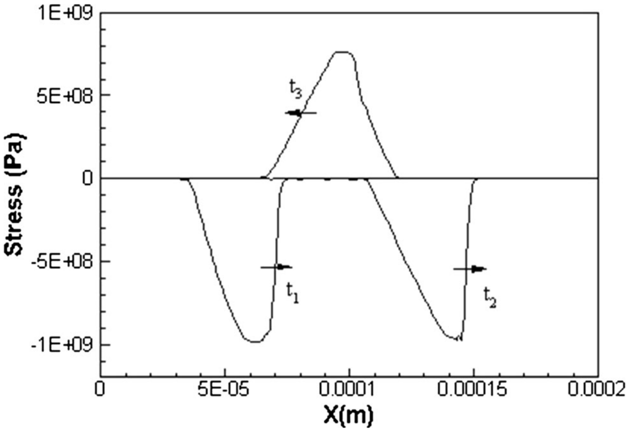


Fig. 11 Stress shock wave propagation in LJ molecule chain subjected to a sine pulse load at 2000 K ($t_1 = 1$ ps, $t_2 = 2$ ps, $t_3 = 4.5$ ps)

shock wave propagating along the chain as shown in Fig. 10. It can be seen that the oscillations are generated behind the shock wave fronts: the loading and unloading wave fronts. Generally, the oscillation occurs because numerical methods have difficulty reproducing strong discontinuities. A common solution is using artificial viscosity to smooth the shock wave fronts. In this paper, the flux-corrected transport (FCT) algorithm [67] is applied to eliminate the oscillations.

Due to the hyperelastic nature of LJ potential when LJ bonds are compressed, the secant modulus is larger at a higher compressive stress, and the wave speed is faster. Consequently, the unloading wave front becomes gentler while the loading wave front remains steep. The phenomena can be observed in the wave profile at time t_2 in Fig. 10. After the stress wave is reflected by the right end, which is free, the compressive stress wave becomes a tensile stress wave, and different phenomena are then observed. Since the stress-deformation gradient relations in Fig. 1 indicate that the secant modulus is lower at a higher tensile stress, the wave speed is slower. Therefore, in the wave profiles at time t_3 and t_4 in Fig. 10, the loading wave becomes gentler while the unloading becomes steeper.

When a sinusoidal pulse load with an amplitude of 10 nN and a period of 0.5 ps is applied, the phenomena similar to those shown in Fig. 11 are observed. After wave reflection at the right end, the tensile wave propagates. Theoretically, when the tensile tress reaches to a critical level, i.e., the failure stress at the deformation gradient threshold, the material failure occurs as well as spallation. In our study, the SVM classification model is used to check each element for failure occurrence. Once the failure is predicted, the spall thickness and speed are then calculated. Table 2 lists the spall thicknesses and speeds under various temperatures between 300 K and 1800 K. Obviously, at a higher temperature, material failure occurs earlier, so that the spall has smaller thickness and higher speed (Table 3).

Table 3 Spall thicknesses and speeds at various temperatures

Temperature (K)	Spall thickness (μm)	Spall speed (m/s)
300	32	4095.2
600	27	4339.1
900	25	4442.8
1200	22	4687.4
1500	21	4887.8
1800	20	5076.3

4.2 Al crystalline solid

Here an Al crystalline solid subjected to uniaxial tension is considered. Plane strain is assumed, and the simulated object has a length of 2 mm and a height of 2 mm. There is a hole with a radius of 0.15 mm located at the center of the solid. 0.01 µs is chosen as the time step, and a prescribed extension of 0.01 µm is applied on the top and bottom surfaces of the solid at every time step. Such a small extension is chosen in order to approximate quasi-static simulations. There are a total of 1766 nodes and 3379 triangular elements in the FEM model as shown in Fig. 12. Since three-node linear triangular elements are used in the continuum model, the calculated strain tensor in each element is a constant at every time step. The failure classification model is used first to identify material failure (defect) mode for each element. In our simulation, if the material void mode is detected in an element, the material failure occurs in this element, and the element becomes a void. Otherwise, for an element in either material defect-free or dislocation mode, the stress regression model is used to predict the stress tensor. The above procedure is repeated on each non-failure element at every time step.

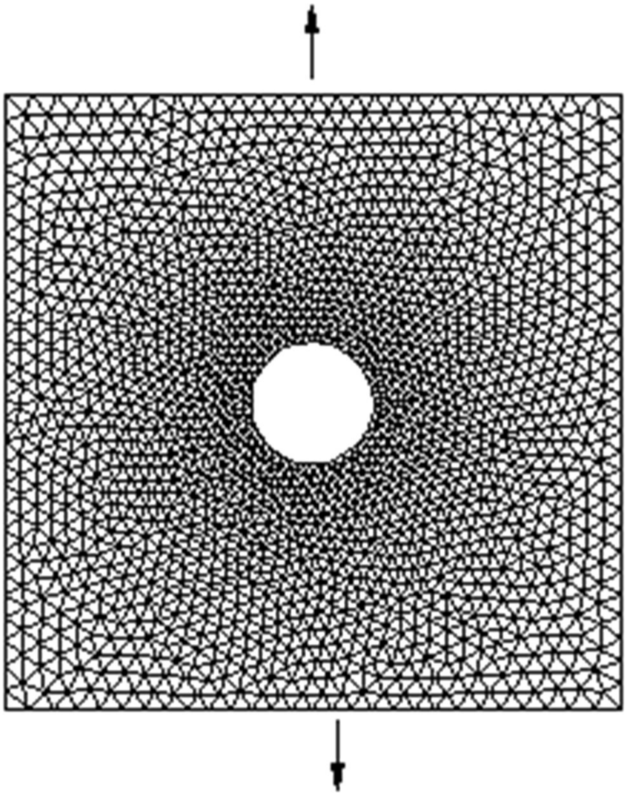


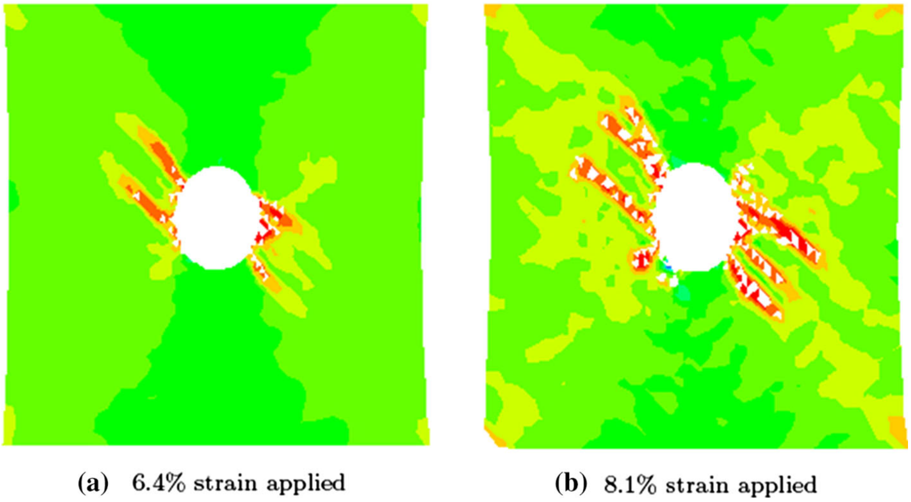
Fig. 12 An Al crystalline solid is subject to uniaxial tension

Figure 13 illustrates the evolution of strain localization in the simulated Al crystalline solid under uniaxial tension. It can be clearly noted that the elements with material dislocation or void modes are detected and shear band paths [58] are observed. Particularly, micro-cracks are initiated in the element with material void modes and then form macro-cracks along the shear band paths.

5 Conclusions

In the proposed hierarchical multiscale method, ML played an important role in bridging different scales. The dataset, collected via MD simulations in the molecular model, was used to train a few predictive models, i.e., learning machines, and then well-trained learning machines were implemented in continuum modeling and simulation. We did molecular dynamics simulations of a one-dimensional molecular chain and an aluminum crystalline solid. The collected datasets represent material physical phenomena at the nanoscale, including stress–strain relations, dislocation phenomenon, and failure occurrence. Based on the

Fig. 13 Evolution of strain localization in a central-holed Al crystalline solid



(a) 6.4% strain applied

(b) 8.1% strain applied

collected molecular database, learning machines were trained for evaluating stresses and determining material failure/defect modes, which were related to the macroscale mechanical behaviors: stress responses, shear band phenomenon, and micro/macro-crack initiation and propagation. The examples of molecule chain and aluminum crystalline solid in the continuum model demonstrated that ML is an alternative approach to enhance multiscale modeling and simulation. Our experimental comparison in this paper confirmed that ELMs were faster and more accurate than SVMs and DNN. It is expected that learning machines can catch detailed physical phenomena in the molecular model and pass the information to the continuum model as long as sufficient neurons without overfitting and proper physical features (i.e., input variables) in the dataset are provided. Although only simple examples were demonstrated in this paper, more potential in-depth research can be done within the same framework. Of course, the larger number of the features involved, the more challenges will be raised in big data collection and mining.

In general, the data science process includes framing the problem, collecting/processing data, exploring data, performing in-depth analysis, and communicating the results of the analysis. In the engineering domain, the first step of framing a problem is transforming a physical problem to a mathematical model, which can be solved by numerical methods, including multiscale methods. In addition, the features of the dataset, which need to be collected in the next step, must be identified based on the original physical problem. In the examples discussed in this paper, only deformation and temperature were considered as the features. Indeed, more features can be considered according to the nanostructured materials to be studied and the physical phenomena to be investigated. For example, to study the mechanical behaviors of nanocomposites, additional features could include inclusion density, orientation, and distribution. In another example of studying the role of defects in mechanics of materials, defect density, defect size, and dislocation orientation could be considered as features as well.

The proposed machine-learning-enhanced multiscale method is closely accompanied by the rest of the data science process. First, the data collection is conducted via MD simulations in the proposed framework. Generally, a set of randomly generated input variables initiates one MD simulation and then generates a single data. Therefore, a large dataset requires intensive MD simulations, especially for high-dimensional feature spaces. Fortunately, those MD simulations can be run independently, and researchers can take advantage of current parallel, grid, or cloud computing techniques. The effort of processing data, i.e., cleaning data, is minimal because the dataset is collected from physical-based molecular simulations. Next, based on the physical phenomena that we intend to investigate at the nanoscale and the messages that we want to pass to the macroscale, data exploration is the step during which the output targets, including stress, damage initiation and growth, and failure occurrence, are identified, as shown in this paper. Then, the key to perform in-depth analysis is employing appropriate ML algorithms. In the proposed multiscale method, the learning machines play an important role in predicting outputs in continuum simulations at each material point and at every iteration after being trained. Consequently, a fast-speed learning machine with high accuracy is required in data analysis to avoid computational intensity in the next step. It is obvious that ELMs, reinforced by parallelization, offers a much better solution than the others, including SVMs. In the last step, communicating results is passing the information from the molecular model to the continuum model via the well-trained learning machines.

Ideally, MD is one way to sample the ensemble by generating configurations deterministically at the nanoscale. The domain average is then used to evaluate mechanical and thermodynamic quantities, including stresses. However, randomly generated initial configurations and temperature regulations can introduce statistical noises. For example, the same strain state may result in different stress states at various MD simulations. To reduce the noises, time averages shall be used as the ensemble averages to generate output data based on the ergodic hypothesis. On the other hand, even for big data with such statistical noises, ELMs can provide confidence intervals around the best predictions.

The challenges of the proposed method include feature extraction, neural network selection, and dataset optimization. Various information from the molecular database can be obtained based on observed physical phenomena. Therefore, features, extracted from the molecular database, should be informative and non-redundant to represent detailed physical phenomena. As discussed in the above, for example, additional features, including reinforcement density and size, are important when studying nanocomposites. On the other hand, given the different nature of the physical problems, it is challenging to select the "right" type of neural network to perform relevant analysis. For instance, if mechanical responses are rate-dependent, a recurrent neural network may be needed. An alternative solution is to add deformations at the previous time as additional features if a feedforward neural network is used. In addition, optimal sampling is a challenge if we want to study the sensitivity of the data samples in order to reduce the MD simulations and avoid generating unnecessary data samples. Different data samples have different importance. Some data samples are more relevant to the training process of ML models than others. For example, data samples that are on the boundary of two classes are more important than the data points at the center area of each class. How to automatically optimize the sampling process is a challenge. One possible approach for ELMs is to apply the weighted linear regression together with the data validation to find the optimal weights for each data sample.

In this paper, only the nano- and macroscales are considered, and learning machines are employed to pass the information from the molecular model to the continuum model. Indeed, the hierarchical multiscale model can be extended to include various scales, and massages can be passed in the same manner. The quantum scale can be added as the smallest scale. Not only the interatomic forces but also bond breaking and reforming in the molecular model can be determined by learning machines, which are trained based on the dataset collected from quantum calculations. In addition, the microscale or the mesoscale can be added between the nano- and the macroscales to link nanomechanics and structure mechanics by micromechanics. Such a bottom-up multiscale strategy will enhance novel material design coupled with engineering product design, which is usually a top-down process.

6 Potential conflict of interests

Such conflict of interest may include (but are not limited to the following):

- Research grants from funding agencies
- Honoraria for speaking at symposia
- Financial support for attending symposia
- Financial support for educational programs
- Employment or consultation
- Support from a project sponsor
- Position on advisory board or board of directors or other type of management relationships
- Multiple affiliations
- Financial relationships, for example equity ownership or investment interest
- Intellectual property rights (e.g., patents, copyrights and royalties from such rights)
- Holdings of spouse and/or children that may have financial interest in the work.

References

1. Abraham FF, Broughton JQ, Bernstein N, Kaxiras E (1998) Spanning the length scales in dynamic simulation. Comput Phys 12(6):538–546
2. Ademiloye A, Zhang L, Liew K (2017) Atomistic-continuum model for probing the biomechanical properties of human erythrocyte membrane under extreme conditions. Comput Methods Appl Mech Eng 325:22–36
3. Akusok A, Bjork KM, Miche Y, Lendasse A (2015) High-performance extreme learning machines: a complete toolbox for big data applications. IEEE Access 3:1011–1025
4. Arroyo M, Belytschko T (2003) A finite deformation membrane based on inter-atomic potentials for the transverse mechanics of nanotubes. Mech Mater 35(3–6):193–215
5. Artrith N, Urban A (2016) An implementation of artificial neural-network potentials for atomistic materials simulations: performance for TiO 2. Comput Mater Sci 114:135–150
6. Badia S, Bochev P, Lehoucq R, Parks ML, Fish J, Nuggehally MA, Gunzburger M (2007) A Force-Based Blending Model forAtomistic-to-Continuum Coupling. Int J Multiscale Comput Eng 5(5):387–406
7. Bélisle E, Huang Z, Le Digabel S, Gheribi AE (2015) Evaluation of machine learning interpolation techniques for prediction of physical properties. Comput Mater Sci 98:170–177
8. Bogdanor MJ, Oskay C, Clay SB (2015) Multiscale modeling of failure in composites under model parameter uncertainty. Comput Mech 56(3):389–404
9. Chen C, Deng Z, Tran R, Tang H, Chu IH, Ong SP (2017) Accurate force field for molybdenum by machine learning large materials data. Phys Rev Mater 1(4):043603
10. Chen S, Cowan C, Grant P (1991) Orthogonal least squares learning algorithm for radial basis function networks. IEEE Trans Neural Netw 2(2):302–309
11. Hsu Chih-Wei, Lin Chih-Jen (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13(2):415–425
12. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
13. Curtin WA, Miller RE (2003) Atomistic/continuum coupling in computational materials science. Modell Simul Mater Sci Eng 11(3):R33–R68
14. Ericksen J (1984) The Cauchy and Born hypotheses for crystals. In: Phase Transformations and Material Instabilities in Solids. Elsevier, pp 61–77
15. Fish J (2006) Bridging the scales in nano engineering and science. J Nanopart Res 8(5):577–594
16. Fritzen F, Kunc O (2018) Two-stage data-driven homogenization for nonlinear solids using a reduced order model. Eur J Mech A Solids 69:201–220
17. Ghaffari MA, Zhang Y, Xiao S (2018) Multiscale modeling and simulation of rolling contact fatigue. Int J Fatigue 108:9–17
18. Glielmo A, Sollich P, De Vita A (2017) Accurate interatomic force fields via machine learning with covariant kernels. Phys Rev B 95(21):214302
19. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
20. Grabowski K, Zbyrad P, Uhl T, Staszewski WJ, Packo P (2017) Multiscale electro-mechanical modeling of carbon nanotube composites. Comput Mater Sci 135:169–180
21. Gracie R, Belytschko T (2011) An adaptive concurrent multiscale method for the dynamic simulation of dislocations. Int J Numer Meth Eng 86(4–5):575–597
22. Huang G-B, Zhu Q-Y, Siew C-K (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE international joint conference on neural networks (IEEE Cat. No.04CH37541), vol 2. IEEE, pp 985–990
23. Gupta A, Cecen A, Goyal S, Singh AK, Kalidindi SR (2015) Structure-property linkages using a data science approach: application to a non-metallic inclusion/steel composite system. Acta Mater 91:239–254
24. Hansen K, Biegler F, Ramakrishnan R, Pronobis W, von Lilienfeld OA, Müller KR, Tkatchenko A (2015) Machine

learning predictions of molecular properties: accurate many-body potentials and nonlocality in chemical space. J Phys Chem Lett 6(12):2326–2331

25. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366
26. Hu R, Roshdibenam V, Johnson HJ, Eirola E, Akusok A, Miche Y, Björk KM, Lendasse A (2018) Elm-som: A continuous self-organizing map for visualization. In: International joint conference on neural networks (IJCNN). IEEE, pp 1–8
27. Huang G, Huang GB, Song S, You K (2015) Trends in extreme learning machines: a review. Neural Netw 61:32–48
28. Huang GB (2014) An insight into extreme learning machines: random neurons random features and kernels. Cogn Comput 6(3):376–390
29. Huang GB (2015) What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. Cogn Comput 7:263–278
30. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501
31. Ibañez R, Abisset-Chavanne E, Aguado JV, Gonzalez D, Cueto E, Chinesta F (2018) A manifold learning approach to data-driven computational elasticity and inelasticity. Arch Comput Methods Eng 25(1):47–57
32. Ibañez R, Borzacchiello D, Aguado JV, Abisset-Chavanne E, Cueto E, Ladeveze P, Chinesta F (2017) Data-driven non-linear elasticity: constitutive manifold construction and problem discretization. Comput Mech 60(5):813–826
33. Igelnik B, Pao YH (1995) Stochastic choice of basis functions in adaptive function approximation and the functional-link net. IEEE Trans Neural Networks 6(6):1320–1329
34. Jahya A, Herink M, Misra S (2013) A framework for predicting three-dimensional prostate deformation in real time. Int J Med Robot Comput Assist Surg 9(4):e52–e60
35. Jain A, Hautier G, Ong SP, Persson K (2016) New opportunities for materials informatics: resources and data mining techniques for uncovering hidden relationships. J Mater Res 31(08):977–994
36. Jiang S, Tao J, Sewell TD, Chen Z (2017) Hierarchical multiscale simulations of crystalline β-octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine (β-HMX): Generalized interpolation material point method simulations of brittle fracture using an elastodamage model derived from molecular dynamics. Int J Damage Mech 26(2):293–313
37. Kalidindi SR, Niezgoda SR, Salem AA (2011) Microstructure informatics using higher-order statistics and efficient data-mining protocols. JOM 63(4):34–41
38. Kelchner CL, Plimpton SJ, Hamilton JC (1998) Dislocation nucleation and defect structure during surface indentation. Phys Rev B 58(17):11085–11088
39. Le BA, Yvonnet J, He QC (2015) Computational homogenization of nonlinear elastic materials using neural networks. Int J Numer Meth Eng 104(12):1061–1084
40. Lendasse A, Wertz V, Verleysen M (2003) Model selection with cross-validations and bootstraps—application to time series prediction with RBFN models. In: Xu L, Kaynak O, Alpaydin E, Oja E (ed) Artificial neural networks and neural information processing—ICANN/ICONIP 2003. Lecture notes in computer science, vol 2714. Springer, Berlin, pp 573–580
41. Liu W, Karpov E, Zhang S, Park H (2004) An introduction to computational nanomechanics and materials. Comput Methods Appl Mech Eng 193(17–20):1529–1578
42. Liu WK, Qian D, Gonella S, Li S, Chen W, Chirputkar S (2010) Multiscale methods for mechanical science of complex materials: Bridging from quantum to stochastic multiresolution continuum. Int J Numer Meth Eng 83(8–9):1039–1080
43. Liu Z, Bessa M, Liu WK (2016) Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials. Comput Methods Appl Mech Eng 306:319–341
44. Liu Z, Fleming M, Liu WK (2018) Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials. Comput Methods Appl Mech Eng 330:547–577
45. Lorente D, Martínez-Martínez F, Rupérez M, Lago M, Martínez-Sober M, Escandell-Montero P, Martínez-Martínez J, Martínez-Sanchis S, Serrano-López A, Monserrat C, Martín-Guerrero J (2017) A framework for modelling the biomechanical behaviour of the human liver during breathing in real time using machine learning. Expert Syst Appl 71:342–357
46. Marcus G (2018) Deep learning: a critical appraisal. arXiv preprint arXiv:1801.00631
47. Matouš K, Geers MG, Kouznetsova VG, Gillman A (2017) A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. J Comput Phys 330:192–220
48. McDowell DL, Panchal J, Choi HJ, Seepersad C, Allen J, Mistree F (2010) Integrated design of multiscale, multifunctional materials and products, 1st edn. Butterworth-Heinemann, Oxford
49. Meng Q, Li B, Li T, Feng XQ (2017) A multiscale crack-bridging model of cellulose nanopaper. J Mech Phys Solids 103:22–39
50. Mielke SL, Troya D, Zhang S, Li JL, Xiao S, Car R, Ruoff RS, Schatz GC, Belytschko T (2004) The role of vacancy defects and holes in the fracture of carbon nanotubes. Chem Phys Lett 390(4–6):413–420
51. Miller RE, Tadmor EB (2009) A unified framework and performance benchmark of fourteen multiscale atomistic/continuum coupling methods. Modell Simul Mater Sci Eng 17(5):053001
52. Mishin Y, Farkas D, Mehl MJ, Papaconstantopoulos DA (1999) Interatomic potentials for monoatomic metals from experimental data and ab initio calculations. Phys Rev B 59(5):3393–3407
53. Plimpton S (1995) Fast parallel algorithms for short-range molecular dynamics. J Comput Phys 117(1):1–19
54. Rahman MM, Feng Y, Yankeelov TE, Oden JT (2017) A fully coupled space-time multiscale modeling framework for predicting tumor growth. Comput Methods Appl Mech Eng 320:261–286
55. Reich Y, Barai S (1999) Evaluating machine learning models for engineering problems. Artif Intell Eng 13(3):257–272
56. Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386–408
57. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222
58. Song JH, Areias PMA, Belytschko T (2006) A method for dynamic crack and shear band propagation with phantom nodes. Int J Numer Meth Eng 67(6):868–893
59. Subramanian N, Rai A, Chattopadhyay A (2015) Atomistically informed stochastic multiscale model to predict the behavior of carbon nanotube-enhanced nanocomposites. Carbon 94:661–672
60. Tadmor E, Phillips R, Ortiz M (2000) Hierarchical modeling in the mechanics of materials. Int J Solids Struct 37(1–2):379–389
61. Tadmor EB, Miller RE (2017) Benchmarking, validation and reproducibility of concurrent multiscale methods are still needed. Modell Simul Mater Sci Eng 25(7):071001
62. Tadmor EB, Ortiz M, Phillips R (1996) Quasicontinuum analysis of defects in solids. Philos Mag A 73(6):1529–1563
63. Talebi H, Silani M, Rabczuk T (2015) Concurrent multiscale modeling of three dimensional crack and dislocation propagation. Adv Eng Softw 80:82–92
64. Tschopp M, McDowell D (2008) Influence of single crystal orientation on homogeneous dislocation nucleation under uniaxial loading. J Mech Phys Solids 56(5):1806–1830

65. Tsoumakas G, Katakis I (2007) Multi-label classification: an overview. Int J Data Wareh Min 3(3):1–13
66. Wagner GJ, Liu WK (2003) Coupling of atomistic and continuum simulations using a bridging scale decomposition. J Comput Phys 190(1):249–274
67. Xiao S (2006) A non-oscillatory method for spallation studies. Int J Numer Meth Eng 66:364–380
68. Xiao S, Andersen DR, Han R, Hou W (2006) Studies of carbon nanotube-based oscillators using molecular dynamics. J Comput Theor Nanosci 3(1):142–147
69. Xiao S, Andersen DR, Yang W (2008) Design and analysis of nanotube-based memory cells. Nanoscale Res Lett 3:416–420
70. Xiao S, Belytschko T (2004) A bridging domain method for coupling continua with molecular dynamics. Comput Methods Appl Mech Eng 193(17–20):1645–1669
71. Xiao S, Hou W (2006) Fracture of vacancy-defected carbon nanotubes and their embedded nanocomposites. Phys Rev B 73(11):115406
72. Xiao S, Hou W (2007) Studies of nanotube-based resonant oscillators through multiscale modeling and simulation. Phys Rev B 75(12):125414
73. Xiao S, Wang S, Ni J, Briggs R, Rysz M (2008) Reliability analysis of carbon nanotubes using molecular dynamics with the aid of grid computing. J Comput Theor Nanosci 5(4):528–534
74. Xiao S, Yang W (2006) Temperature-related Cauchy-Born rule for multiscale modeling of crystalline solids. Comput Mater Sci 37(3):374–379
75. Xiao S, Yang W (2007) A temperature-related homogenization technique and its implementation in the meshfree particle method for nanoscale simulations. Int J Numer Meth Eng 69(10):2099–2125
76. Yang W, Xiao S (2008) Extension of the temperature-related Cauchy–Born rule: material stability analysis and thermo-mechanical coupling. Comput Mater Sci 41(4):431–439
77. Zhou M (2003) A new look at the atomic level virial stress: on continuum-molecular system equivalence. Proc R Soc A: Math Phys Eng Sci 459(2037):2347–2392

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.