Learning constitutive relations of plasticity using neural networks and full-field data

Yin Zhang[a], Qing-Jie Li[a], Ting Zhu[b], Ju Li[a,c*]

[a] Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA

[b] George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

[c] Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA

Abstract

Neural networks (NNs) have demonstrated strong capabilities of learning constitutive relations from big data. However, most NN-based constitutive models require experimental data from a considerable number of stress-strain paths that are expensive to collect. Here, we develop a hybrid finite element method – NN (FEM-NN) framework for learning the constitutive relations from full-field data. As a result, the non-uniform displacement field from a deformed sample with geometrical inhomogeneities can be used for training NNs. Such full-field data have the advantage of providing many different stress-strain paths at different locations in the sample by a single test, thereby enabling the highly efficient training of NNs. We apply FEM-NN simulations to learn the constitutive relations of several model materials characterized by rate-independent J_2 plasticity. These FEM-NN studies demonstrate that the trained NNs produce the constitutive relations of plasticity with high accuracy and efficiency.

Keywords: Machine learning, Neural networks, Plasticity, Finite element method

*Corresponding author email: liju@mit.edu

1. Introduction

Constitutive relations of plasticity are critically important for representing the mechanical behavior of materials beyond their elastic limits. They are commonly established based on either phenomenological [1, 2] or mechanistic models [3]. The development of a phenomenological plasticity model usually requires a comprehensive, but expensive experimental program to calibrate the stress-strain responses along many loading paths. In contrast, the development of a mechanistic model, such as a crystal plasticity model, relies on some physical understanding of the microscopic deformation processes that can be difficult to obtain. In recent years, a drastically different approach has emerged that pursues the data-driven learning of constitutive relations through neural networks (NNs) [4-6]. Such a machine learning approach does not necessarily require much human input and is particularly amenable to the big data generated from experiments.

The NN-based learning of constitutive relations has been applied to study problems involving heterogeneous elasticity [7, 8], rate-independent plasticity [9-11], temperature- and rate-dependent plasticity [12-14], path-dependent plasticity [15-17], isotropic plasticity [18-20], thermodynamically-informed plasticity [21] and crystal plasticity [22]. To train these NNs, most studies require experimental measurements from a considerable number of stress-strain paths, which are expensive to collect. It can become much more expensive to study the effects of multiple principal elements on constitutive relations when exploring the nearly infinite compositional space of high-entropy alloys [23, 24]. Moreover, when the conventional tests of dog-bone-shaped tensile specimens are used, only the total force-displacement data are collected for training these NNs. This mode of machine learning does not represent an efficient use of materials and testing effort. Much unused information, such as the spatial distribution of displacements, could be exploited to expedite the learning of constitutive relations.

Recently, a hybrid finite element method − NN (FEM-NN) framework was developed to augment machine learning with physical constraints in the form of partial differential equations [25], and it was implemented as an extension of the open-source FEM framework *FEniCS* [26]. This framework makes it possible to learn constitutive relations from the observations of full-field data. In this work, we adapt the hybrid FEM-NN framework [25] to machine-learn constitutive relations of plasticity from the full-field displacement data of deformed samples, together with the sample-level force-displacement data. The constitutive relations are learned through the hybrid

FEM-NN simulations with an exemplar physical constraint in the form of rate-independent J_2 plasticity. Namely, the yield stress is taken as a function of equivalent plastic strain and learned through a NN. The experimentally measurable full-field data can be used as the ground truth. Each training iteration involves the following steps: (i) generation/update of the NN-based plasticity relation; (ii) solving a nonlinear FEM problem for obtaining the non-uniform displacement field of a deformed sample using the plasticity relation under training; (iii) computing the cost and its gradient by reverse mode algorithmic differentiation [27] for backpropagation toward minimizing discrepancies between the FEM-NN simulation results with the ground truth [28].

We emphasize that the non-uniform displacement field is an example of the full-field data used for training NNs, and it can be generated by geometrical inhomogeneities such as holes and surface undulations that are intentionally introduced into a tensile sample. As such, many different stress-strain paths are produced at different locations in the sample by a single test and thus enable the highly efficient training of NNs. To expedite the development of the FEM-NN framework, we generate the ground truth through a surrogate computational model, which provides the reference full-field displacement data of a deformed sample from the FEM simulation based on a conventional rate-independent J_2 plasticity model. In future studies, the ground truth from this surrogate model can be replaced by the experimental full-field data obtained from digital image correlation (DIC) [29] measurements. In this work, we further test the trained constitutive relations of plasticity using sample geometries that are not included in the training set. We also study materials with different strain hardening behaviors to demonstrate the general applicability of the hybrid FEM-NN framework for learning constitutive relations of plasticity.

2. A hybrid FEM-NN framework for learning constitutive relations

2.1 Displacement-based finite element method

The hybrid FEM-NN framework is developed by extending the general displacement-based finite element analysis. Here we describe its governing equations by representing tensors and vectors as bold symbols. The strong form of equilibrium equations is given by

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \tag{1}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and **b** is the body force per unit volume. Focusing on small strain analysis, the strain tensor $\boldsymbol{\varepsilon}$ is related to the displacement vector **u** by

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left[\nabla \mathbf{u} + \left(\nabla \mathbf{u}\right)^{\mathrm{T}}\right] \tag{2}$$

Since $\boldsymbol{\sigma}$ depends on $\boldsymbol{\varepsilon}$ through the constitutive relation used, the combination of Eq. (1) and (2) allows one to solve the displacement field **u** that satisfies the strong form of stress equilibrium. In contrast, the weak form of equilibrium equations can be expressed in terms of the virtual work principle over a volume of the material V bounded by its surface S

$$\int_V \boldsymbol{\sigma} : \delta\dot{\boldsymbol{\varepsilon}} dV = \int_S \mathbf{t} \cdot \delta\mathbf{v} dS + \int_V \mathbf{b} \cdot \delta\mathbf{v} dV \tag{3}$$

where $\delta\mathbf{v}$ is the virtual velocity, $\delta\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\left[\nabla\delta\mathbf{v} + \left(\nabla\delta\mathbf{v}\right)^{T}\right]$ is the virtual strain rate, and **t** is the surface traction vector on S. In Eq. (3), $\boldsymbol{\sigma}$ also depends on the displacement field **u** through the constitutive relation used. The virtual work equation of Eq. (3) can be discretized through finite elements to solve the displacement field **u** by Newton's method [30].

In this work, the FEM procedure is implemented using the open-source FEM library *FEniCS* [26] in Python. The variational problem could be specified in *FEniCS* using the domain-specific language UFL [31], such as the following Python statement

$$F = \mathrm{inner}\left(\boldsymbol{\sigma}, \delta\boldsymbol{\varepsilon}\right) d\mathbf{x} \tag{4}$$

where F stands for the formula of the variational problem, the "inner" represents the scalar product of two tensors through the double dot operator in Eq. (3), and $d\mathbf{x}$ indicates the integration over the whole volume V. In Eq. (4), the stress $\boldsymbol{\sigma}$ depends on an unknown displacement field **u**, and $\delta\boldsymbol{\varepsilon}$ is the "virtual" strain field given by the trial function $\delta\mathbf{v}$. Here, we assume only the displacement boundary conditions are applied, and no body force is present, such that both terms on the right-hand side of Eq. (3) become zeros. After defining the variational form of stress equilibrium in UFL, we use the *FEniCS* interface to automatically assemble and solve the nonlinear variational system $F = 0$ with Newton's method. The Jacobian of such a system is calculated analytically by automatic differentiation in UFL.

2.2 Constitutive relations of plasticity

The constitutive relations are learned through an exemplar physical constraint in the form of rate-independent J_2 plasticity. Specifically, the total strain rate is decomposed into the elastic strain rate $\dot{\boldsymbol{\varepsilon}}^{\mathrm{e}}$ and plastic strain rate $\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}}$

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{\mathrm{e}} + \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} \tag{5}$$

The elastic strain rate determines the stress rate according to

$$\dot{\boldsymbol{\sigma}} = \lambda \mathrm{tr}\left(\dot{\boldsymbol{\varepsilon}}^{\mathrm{e}}\right) + 2\mu\dot{\boldsymbol{\varepsilon}}^{\mathrm{e}} \tag{6}$$

where the Lamé constants λ and μ are related to Young's modulus E and Poisson's ratio ν by $\lambda = \dfrac{E\nu}{(1+\nu)(1-2\nu)}$ and $\mu = \dfrac{E}{2(1+\nu)}$. Assuming associated plastic flow, the plastic strain rate is given by

$$\dot{\boldsymbol{\varepsilon}}^{p} = \dot{\bar{\varepsilon}}^{\mathrm{p}} \frac{3}{2} \frac{\mathbf{s}}{\bar{\sigma}} \tag{7}$$

where $\dot{\bar{\varepsilon}}^{\mathrm{p}}$ is the equivalent plastic strain rate, $\mathbf{s}$ is the deviatoric stress tensor

$$\mathbf{s} = \boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma})\mathbf{I} \tag{8}$$

and $\bar{\sigma}$ is the von Mises effective stress

$$\bar{\sigma} = \sqrt{\frac{3}{2}\mathbf{s}:\mathbf{s}} \tag{9}$$

In the conventional rate-independent J_2 plasticity model, the yield criterion is met when $\bar{\sigma}$ equals the yield stress σ_{y}, which is a function of the equivalent plastic strain $\bar{\varepsilon}^{\mathrm{p}} = \int_0^t \dot{\bar{\varepsilon}}^{\mathrm{p}} dt'$ at time t. To study the quasi-static response of a rate-independent material, the time t has no physical meaning and represents a loading sequence. In the present hybrid FEM-NN framework, the yield criterion is met when $\bar{\sigma}$ equals the yield stress given by a NN, denoted as $\sigma_{\mathrm{y}}^{\mathrm{NN}}$. Note that $\sigma_{\mathrm{y}}^{\mathrm{NN}}$

depends on $\bar{\varepsilon}^{\mathrm{p}}$ without an explicit function form and is determined by training a NN. When $\bar{\sigma}$ is smaller than the yield stress, the material behaves elastically and produces zero plastic strain rate.

2.3 Integration procedure

To solve a nonlinear problem under an applied load, we break the FEM simulation into a total of *N* load increments and obtain the equilibrium solution via time integration at the end of each load increment. It usually takes several iterations to find an acceptable solution of a load increment by Newton's method. Here we introduce a smoothing scheme of elastic-plastic transition which can eliminate the need for iterations and also enhance the stability of NN training. The sum of all of the incremental responses gives the integration solution for the applied load.

At the load increment *n*, the stress-strain response at each material point is determined using the standard radial return algorithm [32]. That is, the elastic predictor $\boldsymbol{\sigma}^{\mathrm{pr}}$ is calculated by assuming the strain increment $\Delta\boldsymbol{\varepsilon}$ is purely elastic,

$$\boldsymbol{\sigma}^{\mathrm{pr}} = \boldsymbol{\sigma}_n + \lambda \mathrm{tr}\left(\Delta\boldsymbol{\varepsilon}\right) + 2\mu\Delta\boldsymbol{\varepsilon} \tag{10}$$

where $\boldsymbol{\sigma}_n$ is the stress tensor at the increment *n*. Then the deviatoric part of $\boldsymbol{\sigma}^{\mathrm{pr}}$ is calculated as $\mathbf{s}^{\mathrm{pr}} = \boldsymbol{\sigma}^{\mathrm{pr}} - \frac{1}{3}\mathrm{tr}\left(\boldsymbol{\sigma}^{\mathrm{pr}}\right)\mathbf{I}$ and the corresponding effective stress as $\bar{\sigma}^{\mathrm{pr}} = \sqrt{\frac{3}{2}\mathbf{s}^{\mathrm{pr}} : \mathbf{s}^{\mathrm{pr}}}$. If $\bar{\sigma}^{\mathrm{pr}} < \sigma_{\mathrm{y}}\left(\bar{\varepsilon}_n^{\mathrm{p}}\right)$, the strain increment is indeed purely elastic, such that the plastic strain increment $\Delta\bar{\varepsilon}^{\mathrm{p}} = 0$ and the stress tensor at the end of this load increment is simply $\boldsymbol{\sigma}^{\mathrm{pr}}$. Otherwise, the backward Euler method is used to determine $\Delta\bar{\varepsilon}^{\mathrm{p}}$ via the following nonlinear equation [30],

$$\bar{\sigma}^{\mathrm{pr}} - 3\mu\Delta\bar{\varepsilon}^{\mathrm{p}} = \sigma_{\mathrm{y}}\left(\bar{\varepsilon}_n^{\mathrm{p}} + \Delta\bar{\varepsilon}^{\mathrm{p}}\right) \tag{11}$$

Since the elastic-plastic transition often involves an abrupt change of the slope in the stress-strain curve, this may cause a failure in the backpropagation during training of $\sigma_{\mathrm{y}}^{\mathrm{NN}}$. To resolve this issue, we smoothen the elastic-plastic transition by using the sigmoid function $S\left(x\right) = 1/\left(1+e^{-x}\right)$, a widely used activation function in NN studies [33]. Namely, Eq. (11) is modified as

$$\bar{\sigma}^{\mathrm{pr}} - 3\mu\Delta\bar{\varepsilon}^{\mathrm{p}} = f\left(\bar{\sigma}^{\mathrm{pr}}, \sigma_{\mathrm{y}}\right)\sigma_{\mathrm{y}}\left(\bar{\varepsilon}_n^{\mathrm{p}} + \Delta\bar{\varepsilon}^{\mathrm{p}}\right) \tag{12}$$

where the indicator function f is defined as $f\left(\bar{\sigma}^{\mathrm{pr}}, \sigma_{\mathrm{y}}\right) \equiv S\left(k\left(\bar{\sigma}^{\mathrm{pr}} - \sigma_{\mathrm{y}}\right)\right)$; when k is sufficiently large, this indicator function gives a smooth transition between the elastic ($f = 0$) and plastic ($f = 1$) response. To the first order of $\Delta\bar{\varepsilon}^{\mathrm{p}}$, we expand $\sigma_{\mathrm{y}}^{\mathrm{NN}}\left(\bar{\varepsilon}_n^{\mathrm{p}} + \Delta\bar{\varepsilon}^{\mathrm{p}}\right) \approx \sigma_{\mathrm{y}}\left(\bar{\varepsilon}_n^{\mathrm{p}}\right) + H\Delta\bar{\varepsilon}^{\mathrm{p}}$, where H is the hardening modulus $H = \dfrac{\partial\sigma_{\mathrm{y}}}{\partial\bar{\varepsilon}^{\mathrm{p}}}$ calculated by automatic differentiation in *FEniCS*. Plugging this expansion into Eq. (12), we obtain the approximate equivalent plastic strain increment

$$\Delta\bar{\varepsilon}^{\mathrm{p}} \approx f\left(\bar{\sigma}^{\mathrm{pr}}, \sigma_{\mathrm{y}}\right)\frac{\bar{\sigma}^{\mathrm{pr}} - \sigma_{\mathrm{y}}\left(\bar{\varepsilon}_n^{\mathrm{p}}\right)}{3\mu + H} \tag{13}$$

While solving $\Delta\bar{\varepsilon}^{\mathrm{p}}$ through Eq. (11) requires numerical iterations in a load increment, Eq. (13) is an explicit solution of $\Delta\bar{\varepsilon}^{\mathrm{p}}$ and thus eliminates the need for iterations. The above integration procedure is implemented in *FEniCS* [26]. In Eq. (13), the yield stress $\sigma_{\mathrm{y}}\left(\bar{\varepsilon}^{\mathrm{p}}\right)$ can be either a prescribed function, denoted as $\sigma_{\mathrm{y}}^{\mathrm{ref}}$, to generate the ground truth data or a NN, denoted as $\sigma_{\mathrm{y}}^{\mathrm{NN}}$, to train the constitutive relations of plasticity. The NN representation of constitutive relations is illustrated in Fig. 1, where the NN constitutive relation in the left module of Fig. 1a provides the output of yield stresses based on the input of accumulated plastic strains.

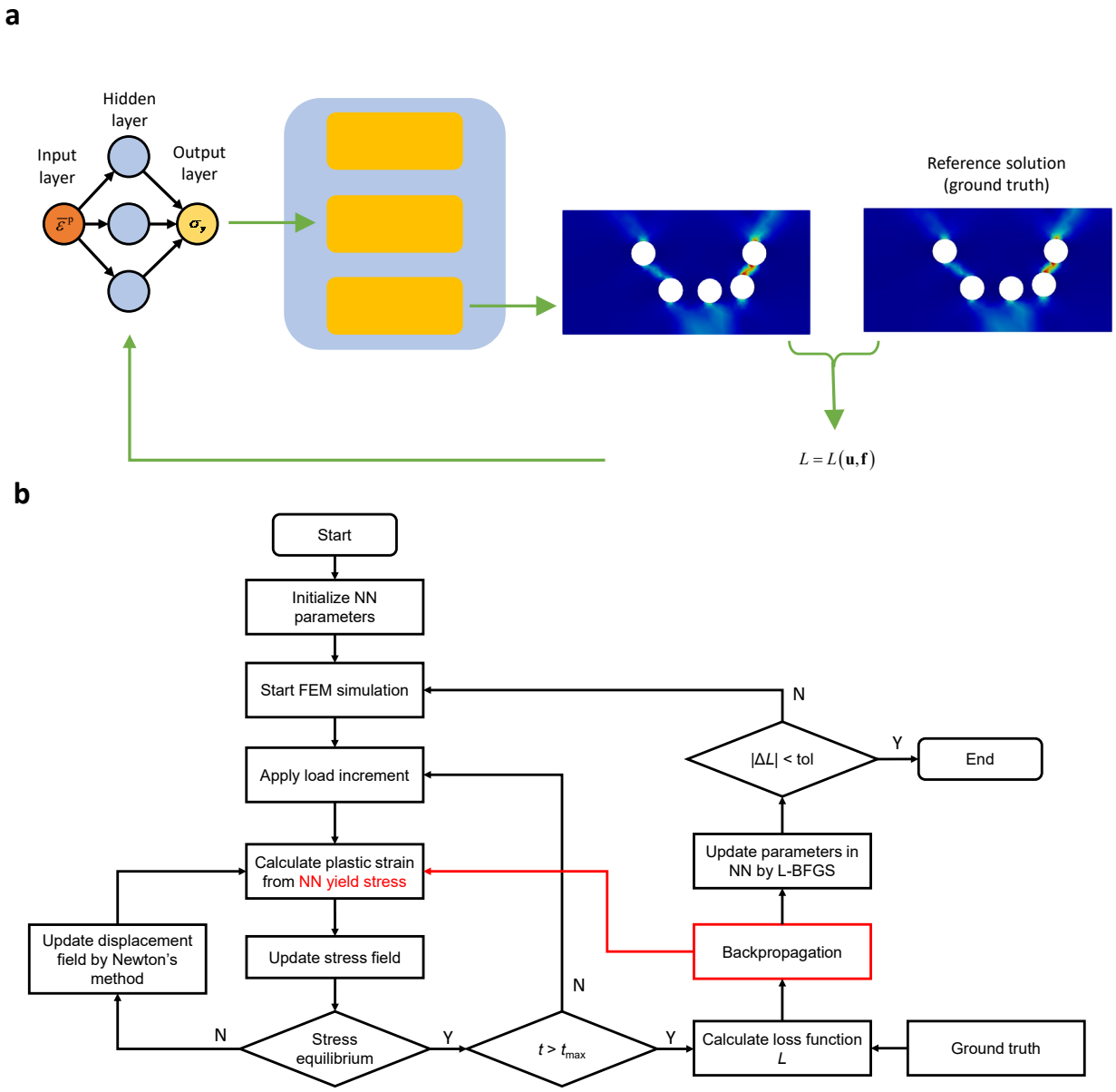

Fig. 1. Illustration of the hybrid FEM-NN framework for learning constitutive relations based on full-field data. (a) Summary of FEM-NN training iterations. Each iteration involves the generation/update of the NN; solving a FEM problem for obtaining the displacement field of a deformed sample using the NN-based plasticity relation under training; computing the cost function and its gradient for backpropagation toward minimizing the discrepancy between the FEM-NN simulation results with the ground truth. (b) Flowchart of the hybrid FEM-NN algorithm. Backpropagation to update the NN yield stress is highlighted in red, as this step requires a key operation of reverse mode algorithmic differentiation.

2.4 Training NNs from full-field data

The NN-based constitutive relation of plasticity in Section 2.3 can be trained using the experimentally measurable full-field data, such as the non-uniform displacement of a tensile sample containing geometrical heterogeneities, together with the sample-level force-displacement data. In this work, we consider a thin-plate sample containing five randomly positioned circular holes. The elastic and plastic properties of the material are homogeneous. We generate the ground truth from a surrogate computational model instead of experimental measurements. That is, the material in this surrogate model obeys the J_2 plasticity with a specified yield stress function $\sigma_{\mathrm{y}}^{\mathrm{ref}}\left(\bar{\varepsilon}^{\mathrm{p}}\right)$, and the corresponding displacement field in the plate is solved by implementing the integration procedure in section 2.3 in *FEniCS*. The obtained ground truth data include the in-plane displacement field $\mathbf{u}_n^{\mathrm{ref}}$ and the sample-level force $\mathbf{f}_n^{\mathrm{ref}}$ versus displacement $\mathbf{u}_n^{\mathrm{ref}}$ for all the load increments $n=1...N$. During each training iteration (see Fig. 1a), we obtain the hybrid FEM-NN solution of the plate using the $\sigma_{\mathrm{y}}^{\mathrm{NN}}$ under training. The obtained data include the displacement field $\mathbf{u}_n^{\mathrm{NN}}$ and the sample-level force $\mathbf{f}_n^{\mathrm{NN}}$ versus displacement $\mathbf{u}_n^{\mathrm{NN}}$ for all the load increments $n=1...N$. Such training iterations aim to minimize the loss function

$$L=\sum_{n=1}^{N}\left[\lambda_{\mathrm{u}}\frac{1}{V}\int_V\left\|\frac{\mathbf{u}_n^{\mathrm{ref}}-\mathbf{u}_n^{\mathrm{NN}}}{u_x^{\max}}\right\|^2 dV+\lambda_{\mathrm{t}}\left\|\frac{\mathbf{f}_n^{\mathrm{ref}}-\mathbf{f}_n^{\mathrm{NN}}}{f_x^{\max}}\right\|^2\right] \tag{14}$$

where $\|\cdot\|$ represent the L2 vector norm; $u_x^{\max}$ and $f_x^{\max}$ are the maximum displacement and force in the x-direction from the ground truth, respectively; λ_{u} and λ_{t} are the weights used to balance the relative contributions to the loss from discrepancies of the displacement field and external load between the hybrid FEM-NN result and the ground truth.

The detailed FEM-NN flowchart is described in Fig. 1b. Note that in order to update the NN weights and biases during each training iteration, the gradients of the loss with respect to these weights and biases are needed. Since the loss function is indirectly related to the NN weights and biases through the FEM solution, we employ reverse mode algorithmic differentiation to calculate the gradients by the adjoint of an expansion through the chain rule. It is challenging to obtain the automated adjoint for backpropagation. This operation is enabled by *dolfin-adjoint* [25, 27] and

highlighted in red in Fig. 1b. After the loss function and its gradient are evaluated, the limited-memory BFGS algorithm [28] in the open-source Python library *SciPy* [34] is invoked to minimize the loss and update the NN weights and biases. Hence, each training iteration (Fig. 1b) involves an FEM simulation using the NN-based plasticity relation under training, evaluation of the loss and its gradients, backpropagation, update of NN weights and biases. Such iterations are repeated until the convergence tolerance for the loss function is met.

The dimension of the simulated thin plate in Fig. 2a is 100 cm × 50 cm × 1 cm, and the radii of the circular holes are 5 cm. A 3D mesh consisting of 4,155 four-node tetrahedral C3D4 elements is generated using an Abaqus script [30]. The Abaqus mesh is converted to an XDMF file by *meshio* [35] to feed into *FEniCS*. The sample is subjected to three symmetrical boundary conditions on the surfaces of $x = 0$, $y = 0$ and $z = 0$. A constant velocity $v = 0.01$ cm/s is applied on the right side of the sample ($x = 100$ cm) for 20 s, giving a total sample-level tensile strain of 0.2%. The FEM simulations are solved with the time step of 1.0 s. As noted earlier, we focus on the quasi-static response of a rate-independent material, so that the loading time only represents a loading sequence instead of a physical time. The elastic properties are taken as follows: Young's modulus $E = 200$ GPa and Poisson's ratio $\nu = 0.3$. The ground truth data are generated for a nonlinear strain hardening material characterized by the yield stress function of $\sigma_{\mathrm{y}}^{\mathrm{ref}} = 100 + 50\tanh\left(2000\bar{\varepsilon}^{\mathrm{p}}\right)$ MPa . When the corresponding NN-based yield stress $\sigma_{\mathrm{y}}^{\mathrm{NN}}$ is trained, we scale the NN input and output to improve the stability and efficiency of training,

$$\sigma_{\mathrm{y}}^{\mathrm{NN}}\left(\bar{\varepsilon}^{\mathrm{p}}\right)/\sigma_0 = \left|\mathrm{NN}\left(\kappa\bar{\varepsilon}^{\mathrm{p}}\right)\right| \tag{15}$$

where σ_0 (= 100 MPa) is the initial guess of the yield stress and the dimensionless parameter κ (= 100) scales $\bar{\varepsilon}^{\mathrm{p}}$ in the activation functions (see below).

The NN used in this work consists of one hidden layer with three neurons. The activation functions in all the neurons are chosen to be the following exponential linear unit (ELU) function [36].

$$\mathrm{ELU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \exp(x) - 1 & \text{if } x < 0 \end{cases} \tag{16}$$

The output of this single hidden layer NN can be expressed as

$$\mathrm{NN}(x) = \sum_{i=1}^{3} w_{i1}\mathrm{ELU}\left(w_{i0}x + b_{i0}\right) + b_1 \tag{17}$$

where w_{i1} and w_{i0} represent the weights in the three neurons (i = 1…3), and b_{i0} and b_1 represent the corresponding biases. The initial weights follow a Gaussian distribution with mean 0 and variance 1 and the initial biases are 0. The weights in Eq. (14) are taken as $\lambda_u = 1$ and $\lambda_t = 1$ such that the displacement and force have equal importance in the loss function. The convergence tolerance for L-BFGS optimization is 1×10^{-10}. In addition to the focused study of a nonlinear strain hardening material, both the perfect plastic and linear strain hardening materials are studied to demonstrate the general applicability of the hybrid FEM-NN approach for learning constitutive relations of plasticity.

3. Results and discussion

In this section, we focus on the hybrid FEM-NN results for a nonlinear strain hardening material characterized by the yield stress function $\sigma_y^{\mathrm{ref}} = 100 + 50\tanh\left(2000\bar{\varepsilon}^{\mathrm{p}}\right)$ MPa . The FEM-NN results are compared with the FEM solutions based on σ_y^{ref}, the latter of which are referred to the reference results representing the ground truth. Fig. 2a shows the finite element mesh of the thin plate. During the training iterations of σ_y^{NN}, both the loss and gradient magnitude are reduced significantly after 100 iterations (Fig. 2b). Given the relatively small NN used, rapid convergence is achieved by only 195 L-BFGS iterations. Using the trained NN, we calculate the FEM-NN-predicted response of sample-level force versus displacement, which agrees closely with the reference result (Fig. 2c). The mean absolute percentage error (MAPE) is 0.083%. We also use the trained NN to simulate the tensile stress-strain response by a single element that agrees closely with the reference result (Fig. 2d). The MAPE is 0.375%. While the full-field data for NN training are taken from the thin plate loaded to the maximum sample-level strain of 0.2%, the local strains near the holes are substantially larger and thus enable the effective training of σ_y^{NN} through many large stress-strain pairs. Hence, a close agreement between the predicted and reference results is

achieved for the material stress-strain response when the yield stress becomes saturated, i.e., at the strain up to 0.5%. The consistently small MAPEs in both the predicted and reference results indicate no significant overfitting in our FEM-NN results.

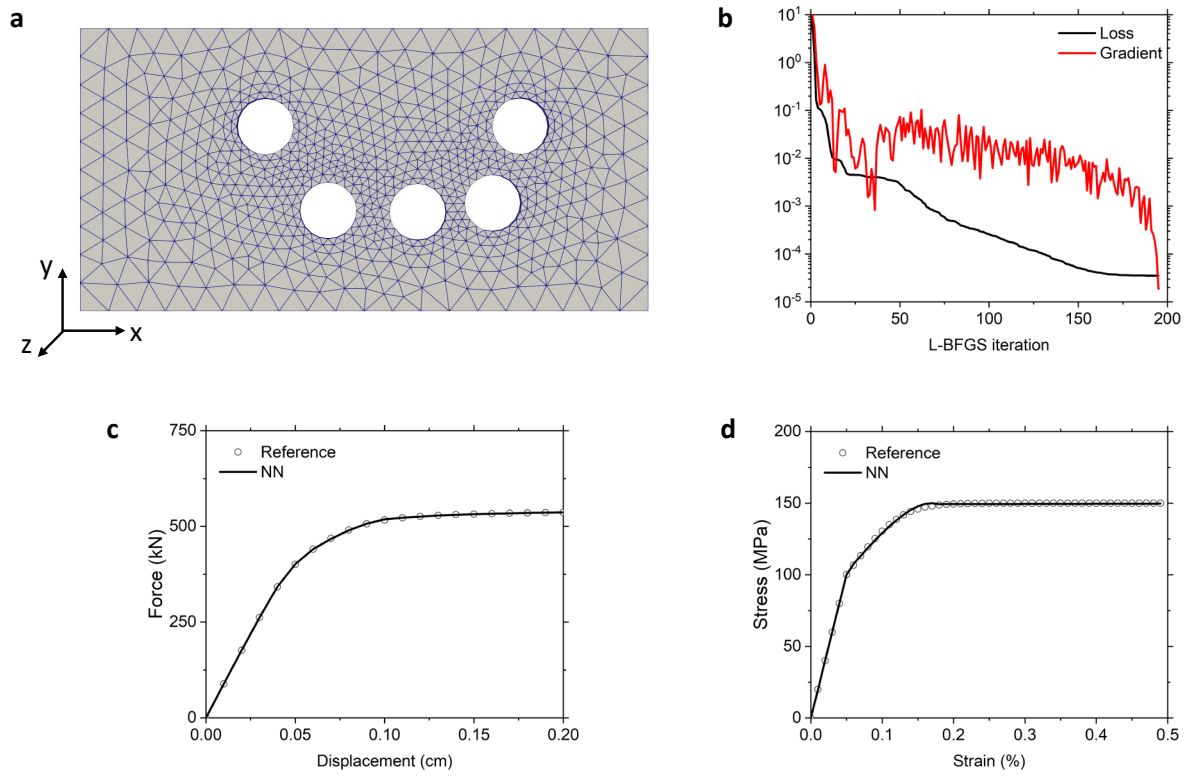

Fig. 2. FEM-NN results for learning the constitutive relation of a nonlinear strain-hardened material. (a) FEM setup of a thin plate containing randomly distributed holes. (b) Loss function and its gradient magnitude as functions of the number of training iterations by the L-BFGS algorithm. (c) Comparison between the reference and FEM-NN-predicted results of sample-level force-displacement response. (d) Comparison between the reference and FEM-NN-predicted results of tensile stress-strain response.

Although the displacement field is directly observable from experiments, its spatial derivatives give the strain field that provides more physically transparent information on the deformation distribution. In Fig. 3, we compare the FEM-NN-predicted and reference strain contour plots in the plate at the maximum sample-level strain of 0.2%. All strain contours are plotted using the open-source package *ParaView* [37]. We focus on the comparison of in-plane normal strain components of ε_{xx} and ε_{yy}. It is seen from Fig. 3a that a strain-localized band arises in between the two neighboring holes close to the right end of the plate, giving about ten times higher $\varepsilon_{xx}^{\text{ref}}$

values than the sample-average tensile strain of 0.2%. Fig. 3b shows a similar strain localized band, where $\varepsilon_{yy}^{\text{ref}}$ is elevated but lower than $\varepsilon_{xx}^{\text{ref}}$. This is because $\varepsilon_{yy}^{\text{ref}}$ mainly results from the Poisson's effect associated with $\varepsilon_{xx}^{\text{ref}}$ and thus give values smaller than $\varepsilon_{xx}^{\text{ref}}$, while $\varepsilon_{xx}^{\text{ref}}$ reflects a direct response from the applied tensile load along the *x*-direction. More importantly, the FEM-NN-predicted strain contour plots in Figs. 3c and 3d are highly consistent with the reference results in Figs. 3a and 3b, respectively. In addition, the differences of $\varepsilon_{xx}^{\text{NN}} - \varepsilon_{xx}^{\text{ref}}$ and $\varepsilon_{yy}^{\text{NN}} - \varepsilon_{yy}^{\text{ref}}$ are plotted in Figs. 3e and 3f, respectively. Most areas have differences close to zero (green), while some minor differences are observed in strain-localized bands. Note that the maximum strain differences are one order of magnitude smaller than the corresponding reference strains, showing the accuracy of FEM-NN predictions.

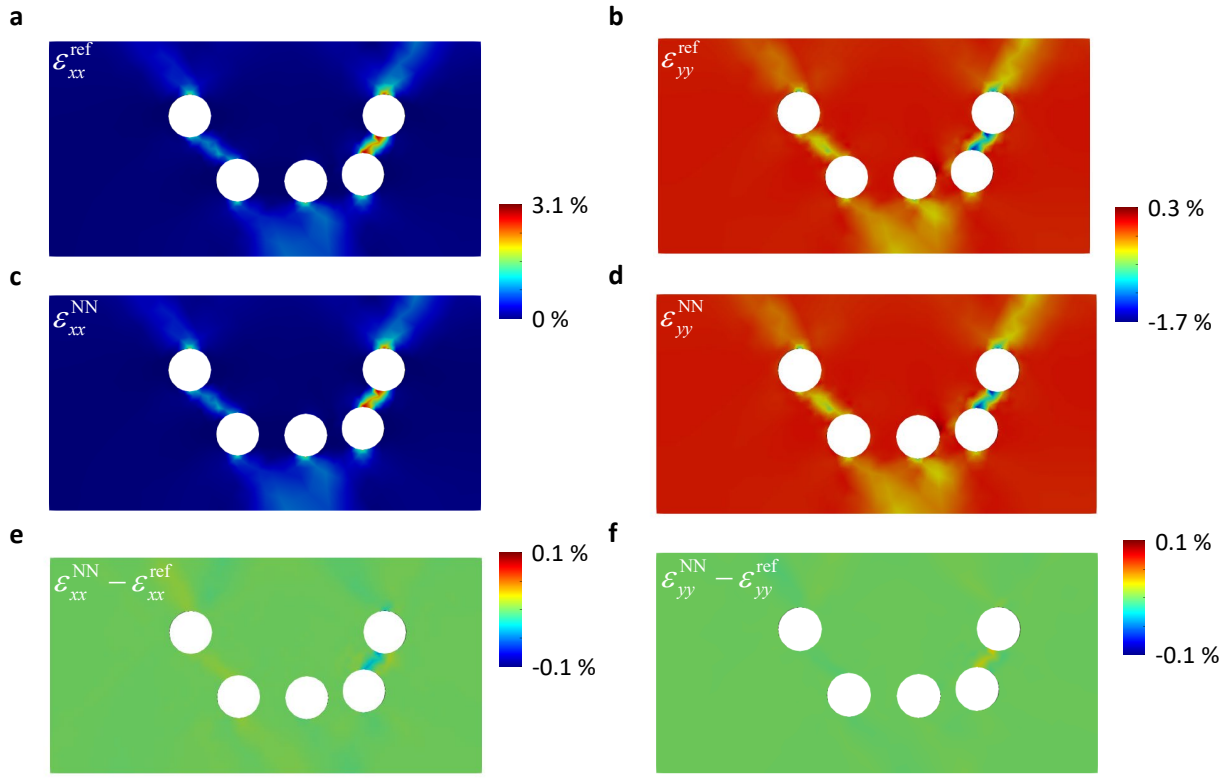

Fig. 3. Comparison between the reference and FEM-NN-predicted strain fields. (a) Contour plot of the reference strain $\varepsilon_{xx}^{\text{ref}}$ along the horizontal direction of tensile loading. (b) Reference strain $\varepsilon_{yy}^{\text{ref}}$ along the transverse direction. (c) Predicted strain $\varepsilon_{xx}^{\text{NN}}$. (d) Predicted strain $\varepsilon_{yy}^{\text{NN}}$. (e) Difference between (a) and (c) $\varepsilon_{xx}^{\text{NN}} - \varepsilon_{xx}^{\text{ref}}$. (f) Difference between (b) and (d) $\varepsilon_{yy}^{\text{NN}} - \varepsilon_{yy}^{\text{ref}}$.

We test the trained NN using a thin plate not included in the training set. As shown in Fig. 4a, this thin plate has the same size, but a different hole arrangement compared with the plate (Fig. 2a) used for NN training. The predicted sample-level force-displacement curve in Fig. 4b closely matches the reference result, with the MAPE of 0.055%. This test shows the high accuracy and transferability of the trained NN from the hybrid FEM-NN approach.

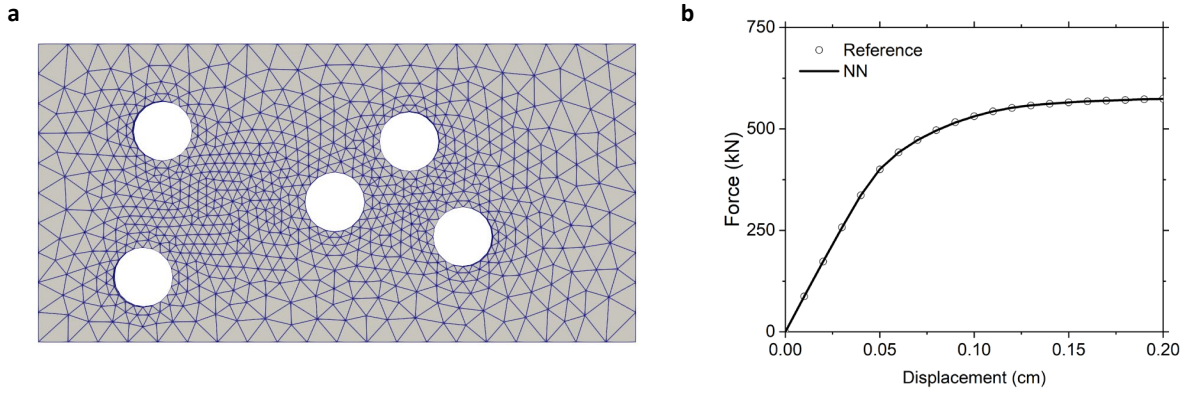

Fig. 4. FEM-NN results for a thin-plate geometry that is not included in the training set. (a) FEM mesh in the plate with hole arrangement different from that in Fig. 2(a). (b) Comparison between the reference and FEM-NN-predicted results of sample-level force-displacement response.

We also test the general applicability of the hybrid FEM-NN approach for learning different constitutive relations of plasticity. To this end, we train the NN-based plasticity relations using the ground truths given by other strain hardening models. With the same thin plate in Fig. 2a, the ground truths are generated by a perfect plasticity model with $\sigma_{\mathrm{y}}^{\mathrm{ref}} = 150$ MPa and a linear hardening model with $\sigma_{\mathrm{y}}^{\mathrm{ref}}(\bar{\varepsilon}^{\mathrm{p}}) = 100 + 50000\bar{\varepsilon}^{\mathrm{p}}$ MPa. After the training of respective $\sigma_{\mathrm{y}}^{\mathrm{NN}}$, the FEM-NN predictions are validated by the simulations of tensile stress-strain response through a single element, are shown in Fig. 5. The FEM-NN-predicted stress-strain curves are consistent with the ground truths; the MAPEs are 0.029% and 0.176% for perfect plastic and linear strain

hardening materials, respectively. These tests demonstrate the high accuracy and general applicability of learning the constitutive relations of plasticity by the hybrid FEM-NN approach.

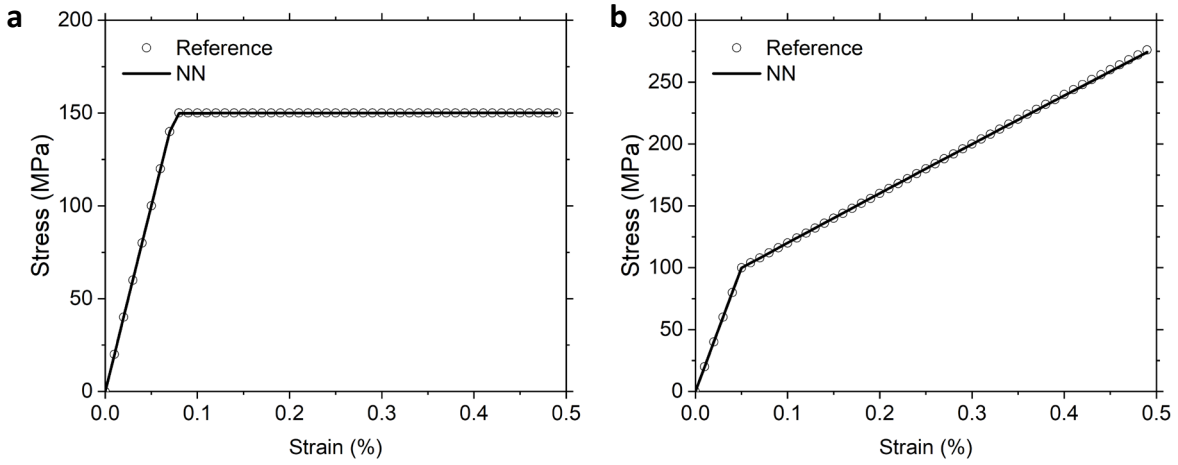

Fig. 5. Comparison between the reference and FEM-NN-predicted results of tensile stress-strain responses for (a) a perfectly plastic material, and (b) a linearly strain-hardened material.

The present FEM-NN framework shows excellent capability and flexibility for learning the constitutive relations of rate-independent J_2 plasticity. The hybrid FEM-NN training of constitutive relations utilizes both the full-field and sample-level data, while the conventional training only relies on the latter. We focus on learning the constitutive relations that depend on accumulated plastic strain in this work, and are working to extend the FEM-NN framework for learning more complex constitutive relations dependent on temperature, strain rate, composition, etc. The FEM-NN learning of complex constitutive relations will be reported in a future paper. We envision the hybrid FEM-NN approach can be combined with high throughput experiments to accelerate the material design and selection [38], for example, when exploring the large composition space of high-entropy alloys. This integrated approach may greatly accelerate the screening of alloy compositions for further in-depth investigation.

4. Conclusion

We have developed a hybrid FEM-NN framework for learning constitutive relations using the full-field data such as a displacement field, in conjunction with the sample-level data such as a force-displacement curve. The non-uniform displacement field in a deformed sample is used for training NNs. It has the advantage of providing many different stress-strain paths at different locations in the sample by a single test, thereby enabling the highly efficient training of NNs. Our studies of the model materials characterized by rate-independent J_2 plasticity demonstrate that the FEM-NN framework can learn their constitutive relations accurately and efficiently. The open-source FEM framework *FEniCS* is a general and powerful platform that can greatly facilitate the future development of the FEM-NN approach. Furthermore, the NN is designed to be compatible with the displacement-based FEM framework, such that the trained NN constitutive relations can be transferred to other FEM programs without additional modification. Broadly, the integration of the FEM-NN approach and full-field data enables the effective constitutive modeling of materials through big data.

Declaration of competing interest

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

Acknowledgment

Y.Z., Q.J.L. and J.L. are supported by the Laboratory Directed Research & Development Program at Idaho National Laboratory under the Department of Energy (DOE) Idaho Operations Office (an agency of the U.S. Government) Contract DE-AC07-05ID145142, and also thank Geoffrey L. Beausoleil II for the funding acquisition. T.Z. is supported by the National Science Foundation (DMR-1810720). We thank Mahmut N. Cinbiz for discussions of the initial idea.

References

[1] R. Hill, J.R. Rice, Constitutive analysis of elastic-plastic crystals at arbitrary strain, Journal of the Mechanics and Physics of Solids 20(6) (1972) 401-413.

[2] R. Hill, The mathematical theory of plasticity, Oxford University Press 1998.
[3] A. Argon, Strengthening mechanisms in crystal plasticity, Oxford University Press 2008.
[4] L. Lu, et al., Extraction of mechanical properties of materials through deep learning from instrumented indentation, Proceedings of the National Academy of Sciences of the U.S.A. 117(13) (2020) 7052.
[5] X. Meng, Z. Li, D. Zhang, G.E. Karniadakis, PPINN: Parareal physics-informed neural network for time-dependent PDEs, Computer Methods in Applied Mechanics and Engineering 370 (2020) 113250.
[6] X. Liu, et al., Knowledge extraction and transfer in data-driven fracture mechanics, Proceedings of the National Academy of Sciences 118(23) (2021).
[7] S. Ye, et al., Deep neural network method for predicting the mechanical properties of composites, Applied Physics Letters 115(16) (2019) 161901.
[8] S. Ye, W.Z. Huang, M. Li, X.Q. Feng, Deep learning method for determining the surface elastic moduli of microstructured solids, Extreme Mechanics Letters 44 (2021) 101226.
[9] T. Furukawa, G. Yagawa, Implicit constitutive modelling for viscoplasticity using neural networks, International Journal for Numerical Methods in Engineering 43(2) (1998) 195-219.
[10] M.S. Al-Haik, M.Y. Hussaini, H. Garmestani, Prediction of nonlinear viscoelastic behavior of polymeric composites using an artificial neural network, International Journal of Plasticity 22(7) (2006) 1367-1392.
[11] H. Yang, et al., Exploring elastoplastic constitutive law of microstructured materials through artificial neural network—a mechanistic-based data-driven approach, Journal of Applied Mechanics 87(9) (2020).
[12] X. Li, C.C. Roth, D. Mohr, Machine-learning based temperature- and rate-dependent plasticity model: Application to analysis of fracture experiments on DP steel, International Journal of Plasticity 118 (2019) 320-344.
[13] K.S. Pandya, C.C. Roth, D. Mohr, Strain rate and temperature dependent fracture of aluminum alloy 7075: Experiments and neural network modeling, International Journal of Plasticity 135 (2020) 102788.
[14] J. Wen, Q. Zou, Y. Wei, Physics-driven machine learning model on temperature and time-dependent deformation in lithium metal and its finite element implementation, Journal of the Mechanics and Physics of Solids 153 (2021) 104481.
[15] M.B. Gorji, et al., On the potential of recurrent neural networks for modeling path dependent plasticity, Journal of the Mechanics and Physics of Solids 143 (2020) 103972.
[16] T. Tancogne-Dejean, M.B. Gorji, J. Zhu, D. Mohr, Recurrent neural network modeling of the large deformation of lithium-ion battery cells, International Journal of Plasticity 146 (2021) 103072.
[17] L. Wu, V.D. Nguyen, N.G. Kilingar, L. Noels, A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths, Computer Methods in Applied Mechanics and Engineering 369 (2020) 113234.
[18] A. Zhang, D. Mohr, Using neural networks to represent von Mises plasticity with isotropic hardening, International Journal of Plasticity 132 (2020) 102732.
[19] D.Z. Huang, K. Xu, C. Farhat, E. Darve, Learning constitutive relations from indirect observations using deep neural networks, Journal of Computational Physics 416 (2020) 109491.
[20] K. Xu, D.Z. Huang, E. Darve, Learning constitutive relations using symmetric positive definite neural networks, Journal of Computational Physics 428 (2021) 110072.
[21] N.N. Vlassis, W. Sun, Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening, Computer Methods in Applied Mechanics and Engineering 377 (2021) 113695.
[22] U. Ali, et al., Application of artificial neural networks in micromechanics for polycrystalline metals, International Journal of Plasticity 120 (2019) 205-219.
[23] E.P. George, D. Raabe, R.O. Ritchie, High-entropy alloys, Nature Reviews Materials 4(8) (2019) 515-534.

[24] Q.Q. Ding, et al., Tuning element distribution, structure and properties by composition in high-entropy alloys, Nature 574(7777) (2019) 223-227.
[25] S.K. Mitusch, S.W. Funke, M. Kuchta, Hybrid FEM-NN models: Combining artificial neural networks with the finite element method, Journal of Computational Physics 446 (2021) 110651.
[26] M. Alnæs, et al., The FEniCS project version 1.5, Archive of Numerical Software 3(100) (2015).
[27] S.K. Mitusch, S.W. Funke, J.S. Dokken, dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake, Journal of Open Source Software 4(38) (2019) 1292.
[28] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Mathematical Programming 45(1) (1989) 503-528.
[29] T.C. Chu, W.F. Ranson, M.A. Sutton, Applications of digital-image-correlation techniques to experimental mechanics, Experimental Mechanics 25(3) (1985) 232-244.
[30] Abaqus 6.13 user's manual, Dassault Systems, Providence, RI (2013).
[31] M.S. Alnæs, et al., Unified form language: A domain-specific language for weak formulations of partial differential equations, ACM Transactions on Mathematical Software (TOMS) 40(2) (2014) 1-37.
[32] J.C. Simo, R.L. Taylor, Consistent tangent operators for rate-independent elastoplasticity, Computer Methods in Applied Mechanics and Engineering 48(1) (1985) 101-118.
[33] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press 2016.
[34] P. Virtanen, et al., SciPy 1.0: fundamental algorithms for scientific computing in Python, Nature Methods 17(3) (2020) 261-272.
[35] N. Schlömer, meshio: Tools for mesh files, https://github.com/nschloe/meshio.
[36] D.A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus), arXiv preprint arXiv:1511.07289 (2015).
[37] U. Ayachit, The ParaView guide: a parallel visualization application, Kitware, Inc.2015.
[38] D. Morgan, et al., Machine learning in nuclear materials research, Current Opinion in Solid State & Materials Science (2021) Accepted manuscript.