

A neural kernel method for capturing multiscale high-dimensional micromorphic plasticity of materials with internal structures

Zeyu Xiong, Mian Xiao, Nikolaos Vlassis, WaiChing Sun*

Department of Civil Engineering and Engineering Mechanics, Columbia University, 614 SW Mudd, Mail Code: 4709, New York, NY 10027, United States of America

Received 19 March 2023; received in revised form 11 July 2023; accepted 25 July 2023

Available online 19 August 2023

Abstract

This paper introduces a neural kernel method to generate machine learning plasticity models for micropolar and micromorphic materials that lack material symmetry and have internal structures. Since these complex materials often require higher-dimensional parametric space to be precisely characterized, we introduce a representation learning step where we first learn a feature vector space isomorphic to a finite-dimensional subspace of the original parametric function space from the augmented labeled data expanded from the narrow band of the yield data. This approach simplifies the data augmentation step and enables us to constitute the high-dimensional yield surface in a feature space spanned by the feature kernels. In the numerical examples, we first verified the implementations with data generated from known models, then tested the capacity of the models to discover feature spaces from meso-scale simulation data generated from representative elementary volume (RVE) of heterogeneous materials with internal structures. The neural kernel plasticity model and other alternative machine learning approaches are compared in a computational homogenization problem for layered geomaterials. The results indicate that the neural kernel feature space may lead to more robust forward predictions against sparse and high-dimensional data.

© 2023 Elsevier B.V. All rights reserved.

Keywords: Neural kernel; Micropolar continua; Micromorphic continua; Level set plasticity

1. Introduction

When deriving models to predict path-dependent responses of materials that lack material symmetry or exhibit complex size-dependent behaviors, the smallest number of variables (e.g., stress measures, stress invariants, internal variables) required to replicate constitutive responses increases, and the dimension of the parametric space in which the model is formulated also becomes higher. This increase in variables often leads to the increase of material parameters due to the need for additional support to control the geometry in the high-dimensional space.

As such, modelers must decide a trade-off between simplicity and sophistication [1]. Increasing the number of material parameters is often undesirable or only used as the last measure to capture the phenomenology precisely [2]. This preference for simpler models is not limited to constitutive theories of solids and has a long history in science (p.398, Newton [3]) and philosophy [4–6]. In the early ages of the development of plasticity theories, experimental data were relatively limited in quantities and lacked precision afforded by the state-of-the-art

* Corresponding author.

E-mail address: wsun@columbia.edu (W. Sun).

instruments [1,7,8]. Hence, assumptions on material symmetry and limitations on the number of variables used to describe the deformation mechanisms become common strategies that maximize what [1] described as the trade-off between simplicity and sophistication.

1.1. Previous approaches for modeling low-symmetry, higher-order, and multiphysics coupling plasticities

Increasing the precision of a material model may unavoidably increase the least number of material parameters necessary to describe the material behaviors. As such, many simpler models that involve more hypotheses often become the blueprint of more elaborated models of higher dimensions. This approach is often adopted by plasticity models in, for instance, the following scenarios.

1. There is a need to capture additional causal mechanisms that are not describable with the current existing variables [9]. For instance, the anisotropy and non-coaxiality induced by fabric evolution require incorporating the fabric tensors into the constitutive laws [10–12].
2. There is a need to enrich the description of a model such that the model can be further generalized for broader applications. For instance, the Drucker–Prager model can be viewed as a generalization of the Von Mises model by introducing the dependence of the yielding behaviors with respect to the pressure (position of the hydrostatic axis in the principal stress space). de Borst [13] introduces micropolar constitutive models for geo-materials by introducing the additional couple stress terms and length scale parameter in a Drucker–Prager model. Multi-physics constitutive models, such as [14], also adopt an extension strategy where a pure solid mechanics model is enhanced by introducing additional variables such as temperature, degree of saturation, and volume fraction of void that are necessary to capture the multiphysical coupling [14–18]).
3. Finally, there have also been cases in which a model is amended to circumvent known limitations. A classic example is the usage of micropolar theory to circumvent the pathological mesh dependence of plasticity models in the softening regimes. (e.g. [19–21]).

Nevertheless, this incremental strategy to amend models exhibits disadvantages. First of all, obtaining plasticity data from either representative elementary volume simulations or experiments could be costly. As such, the lack of foresight of the model dimensions may lead to a biased data acquisition strategy where data might not distribute well enough to calibrate and test the learned model. For models expressed in higher-dimensional space (e.g., anisotropic elasticity model that requires not just the strain invariants but also the principal directions, micropolar and micromorphic models that require higher-order terms to fully describe the kinematics), data may appear to be sparser and hence requires more data points to characterize the behaviors of comparable complexity [22].

1.2. Previous efforts in machine learning plasticity modeling

Machine learning approaches with advanced architecture, such as the recurrent neural networks for sequential learning [23–26], 1D convolutional neural network [27], and transformer with attention mechanisms [28], can be, in theory, trained to replicate these high-dimensional constitutive responses. However, the vanishing/exploding gradients, the increased computational cost, and the increased demand for data can all become the bottleneck of these approaches. Furthermore, the higher dimensionality of the model also makes the learned model more vulnerable to overfitting (incapable of generalized prediction) [29] and the results more difficult to interpret properly [27,30,31].

On the other hand, there are multiple attempts to represent the plastic yield surfaces via parametrized surface or implicit functions. Vlassis and Sun [27], for instance, introduce the usage of the signed distance function to implicitly represent the yield surface in the stress-internal-variable space. The additional signed distance property enables the plastic flow to be a unit gradient which simplifies the calculation of the plastic multiplier, and allows one to incorporate the plastic flow direction into the Sobolev training of yield function. The resultant yield function is then parametrized via an MLP architecture. Meanwhile, Coombs et al. [32], Coombs and Motlagh [33,34] leverage the flexibility afforded by non-uniformed rational B-splines (NURBs) to parametrize yield function. Xiao and Sun [35], on the other hand, represents the yield surface as a manifold and train neural networks to parametrize an atlas of coordinate charts to form three-variable yield surface of complex shapes.

While the component/modular-based learning approach [36,37] enables one to reformulate the plasticity learning problem without enforcing the memory effect through the architecture, the challenges of training and interpreting

those models remain. Another less explored route to model plasticity is to use the kernel method in which a nonlinear transformation maps the material data in a higher-dimensional feature space to make it easier to perform clustering (e.g., unsupervised learning) or insert hyperplane (e.g., classification). The advantage of the kernel method is that it is generally more robust (due to the non-parametric nature and the implicit feature mapping without any explicit feature engineering) and less sensitive to outliers [38]. More importantly, the construction of a higher-dimensional feature space also makes it possible to interpret the relations between the features and the learned plasticity models. However, a key technical barrier of the classical kernel method is that the performance of the method is highly sensitive to the specific kernel function (e.g., polynomials, radial basis function) used to generate the feature space. As different data sets may require different kernel functions to achieve good performance, selecting the kernel function becomes a time-consuming trial-and-error process.

1.3. Neural kernel method for high-dimensional plasticity

This paper introduces a neural kernel approach to generate micropolar and micromorphic plasticity models. Here we want to leverage the robustness and interpretability of the kernel method afforded by the feature space while **generating a data-dependent kernel** tailored to our specific need to capture the high-dimensional constitutive responses of the materials with complex internal structures. This treatment provides us with a unified data-driven approach to recognize the pattern of the data (through the data-dependent kernel), regardless of the data dimensions. Combined with a simple kernel ridge regression, we may generate a yield function of arbitrary input space dimensions without explicitly handcrafting the feature space. This trait is input for us to automate the process of generating yield surface from data of dimensions higher than 3, especially when data is sparse.

1.4. Organization of the rest of the paper

The organization of the rest of the paper is as follows. For completeness, we first review the plasticity theory of micropolar and micromorphic materials and the related Hill–Mandel lemma necessary for generating the multiscale data for computational homogenization. This review explains the difficulties in formulating yield surfaces in a high-dimensional stress space (Section 2). We then introduce the neural kernel method formulated for the high-dimensional space. In particular, we explain both the theory of the neural kernel method, as well as the strategy we adopted to train both the data-dependent neural network kernels and obtain the coefficients used to interpolate the yield function in the high-dimensional feature space. The return mapping algorithm that adopts the neural kernel yield function is also included (Section 3). This is followed by a collection of representative numerical examples in which we provide verifications and demonstrate how the proposed approach can be applied to complex data obtained from direct numerical simulations of layered pressure-sensitive materials (Section 4). We then summarize our major findings in Section 5. Additional numerical tests and the detailed procedure necessary for third-party inspection and validations are provided in the [Appendix](#).

2. Constitutive framework for higher-order continua

For completeness, we review the theory of micropolar and micromorphic continua (i.e., higher-order continua). Cosserat and Cosserat [39] is credited with introducing the first high-order continuum theory in which the concept of micro-rotation is introduced to describe the effects of internal structures on the constitutive responses. Different variations, for instance, the couple stress theory [40–43], which derives energy density as a function of both strain and the curl of strain. The generalized micromorphic continuum theory, which introduces the concepts of micro-deformation and the corresponding energy-conjugate stress measures as a generalization of the kinematics, have been introduced in the 1960s [44–47]. The extension of the higher-order continua theory to the finite deformation range has been formulated by Toupin [41] where an action density is derived such that it is invariant under the group of Euclidean displacements in a Hamiltonian mechanics framework. Mühlhaus and Vardoulakis [48] and Peerlings et al. [49] further extended the higher-order theories for elastoplasticity problems and examined the regularization effect of higher-order continuum theories. Steinmann [50] extends the multiplicative kinematics theory to formulate micropolar elastoplasticity in the geometric nonlinear regime. Recently, the multiscale micropolar [51] and micromorphic [52] constitutive modeling has been derived in the finite deformation regime, while [53] have introduced a linear relaxed version of micromorphic models to capture the wave propagation in meta-materials.

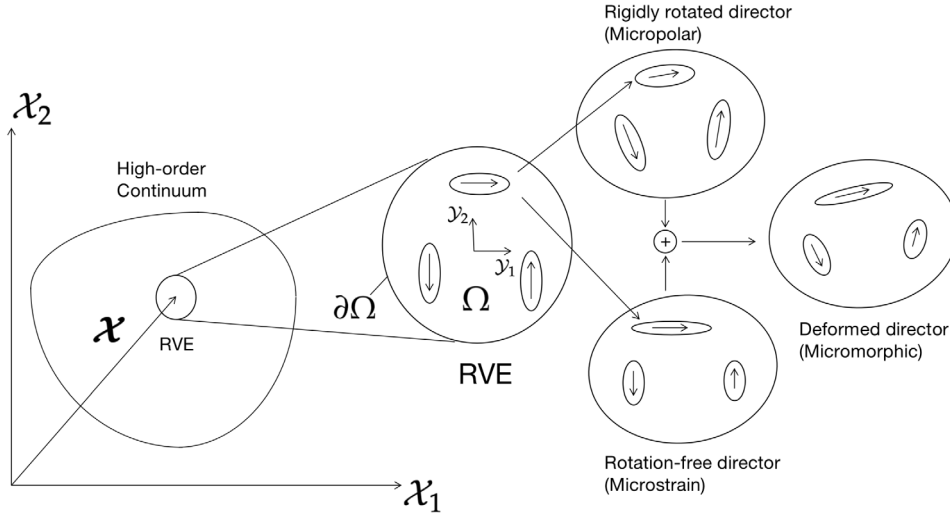


Fig. 1. A microscopic representative volume element (RVE) is attached to the macroscopic material point with position vector $\mathbf{x}$; the local coordinate of the RVE is denoted by $\mathbf{y}$. The directors in the RVE describe the internal micro-structure (e.g., voids or inclusions), and the deformed configuration of the directors reflects the distortion of the internal structure.

A comprehensive review of connections between the higher-order and non-local continuum theories can be found in [54].

For simplicity, we restrict our learned models to be within the infinitesimal deformation regime. We then introduce machine learning to generate three classes of plasticity models based on Cauchy, micropolar, and micromorphic continuum theories. The micro-deformation χ_{ij} describes the configuration of the directors, i.e., the internal micro-structures (e.g., voids and inclusions) contained in the material point sampled at an arbitrary position $\mathbf{x}$, as shown in Fig. 1. The material point of an effective medium is also called the representative volume element (RVE). For a Cauchy continuum, the directors are much smaller than the RVE, so the kinematic configuration of the directors χ_{ij} mechanically affects the RVE much less than the strain tensor,

$$\varepsilon_{ij} = (u_{i,j} + u_{j,i})/2 \quad (1)$$

such that the χ_{ij} can be neglected. In comparison, the micromorphic theory considers that the size effect of the deformable directors is not negligible. As such χ_{ij} must be considered to describe the distortion of the RVE internal structure. Micropolar continua is a sub-class of micromorphic continua in which the directors can be assumed to be rigid (e.g., rigid inclusions) such that

$$\chi_{ij} = -\epsilon_{ijk}\theta_k, \quad (2)$$

where θ_k describes the rigid rotation and ϵ_{ijk} is the Levi-Civita permutation symbol. Based on the kinematic configuration of higher-order continua described by u_i and χ_{ij} , the boundary value problems can be solved given the constitutive laws of micropolar and micromorphic continua, which are reviewed in Section 2.1, followed by the RVE homogenization scheme that models the constitutive law by multiscale simulation (Section 2.2) and data-driven approaches to learn the constitutive law (Section 2.3).

2.1. Boundary value problems

To solve the boundary value problems (BVP) of higher-order continua for the kinematic configuration u_i and χ_{ij} , we need to consider the kinematic relation, balance of linear and angular momentum, and constitutive law as summarized in the upper half of Table 1 [55]. The kinematic relation defines the strain tensor as the difference between the displacement gradient $u_{i,j}$ and the micro-deformation χ_{ij} or micro-rotation $-\epsilon_{ijk}\theta_k$; the gradient of micro-deformation G_{ijk} or the gradient micro-rotation κ_{ij} (i.e., curvature tensor) are also introduced to describe the higher-order deformation. The balance law of linear and angular momentum states the relationship between the

Table 1

The summary of BVP components of the micropolar and micromorphic materials.

	Micropolar	Micromorphic
Kinematic relation	$\varepsilon_{ij} = u_{i,j} + \epsilon_{ijk}\theta_k, \kappa_{ij} = \theta_{i,j}$	$\varepsilon_{ij} = u_{i,j} - \chi_{ij}, G_{ijk} = \chi_{ij,k}$
Balance of linear momentum	$\sigma_{ji,j} = 0$	$\sigma_{ji,j} = 0$
Balance of angular momentum	$\frac{1}{2}\epsilon_{ijk}\sigma_{ji} - m_{jk,j} = 0$	$\sigma_{ji} + \zeta_{ijk,k} = 0$
Constitutive law	$(\varepsilon_{ij}, \kappa_{ij}) \rightarrow (\sigma_{ji}, m_{ji})$	$(\varepsilon_{ij}, G_{ijk}) \rightarrow (\sigma_{ji}, \zeta_{ijk})$
Elastoplastic deformation	$\varepsilon_{ij} = \varepsilon_{ij}^e + \varepsilon_{ij}^p, \kappa_{ij} = \kappa_{ij}^e + \kappa_{ij}^p$	$\varepsilon_{ij} = \varepsilon_{ij}^e + \varepsilon_{ij}^p, G_{ijk} = G_{ijk}^e + G_{ijk}^p$
Elasticity	$\sigma_{ji} = \frac{\partial W(\varepsilon_{ij}^e, \kappa_{ij}^e)}{\partial \varepsilon_{ij}^e}, m_{ji} = \frac{\partial W(\varepsilon_{ij}^e, \kappa_{ij}^e)}{\partial \kappa_{ij}^e}$	$\sigma_{ji} = \frac{\partial W(\varepsilon_{ij}^e, G_{ijk}^e)}{\partial \varepsilon_{ij}^e}, \zeta_{ijk} = \frac{\partial W(\varepsilon_{ij}^e, G_{ijk}^e)}{\partial G_{ijk}^e}$
Yield function	$f(\sigma_{ji}, m_{ji}) \leq 0$	$f(\sigma_{ji}, \zeta_{ijk}) \leq 0$
KKT condition	$\dot{\Lambda} f(\sigma_{ji}, m_{ji}) = 0 \ (\dot{\Lambda} \geq 0, f \leq 0)$	$\dot{\Lambda} f(\sigma_{ji}, \zeta_{ijk}) = 0 \ (\dot{\Lambda} \geq 0, f \leq 0)$
Flow rule	$\dot{\varepsilon}_{ij}^p = \dot{\Lambda} \frac{\partial f}{\partial \sigma_{ji}}, \dot{\kappa}_{ij}^p = \dot{\Lambda} \frac{\partial f}{\partial m_{ji}}$	$\dot{\varepsilon}_{ij}^p = \dot{\Lambda} \frac{\partial f}{\partial \sigma_{ji}}, \dot{G}_{ijk}^p = \dot{\Lambda} \frac{\partial f}{\partial \zeta_{ijk}}$

Cauchy stress σ_{ji} and higher-order couple stress m_{ji} or generalized stress ζ_{ijk} , which are computed given ε_{ij} , κ_{ij} , and G_{ijk} based on the constitutive law.

To complete the solution of BVP, the elastoplastic constitutive law in the lower half of Table 1 needs to be found to model the relation between the kinematic modes and the higher-order stresses, i.e., between ε_{ij} and σ_{ji} , between κ_{ij} and m_{ji} , and between G_{ijk} and ζ_{ijk} . The constitutive law consists of the elasticity model, yield function, KKT condition, and plastic flow rule, as summarized in Table 1. The elasticity model finds the elastic energy functional W such that the elastic stresses are work conjugates of the kinematic modes. The core component of the plasticity model [56] is the yield function f that defines the onset of plastic yielding at $f = 0$. The elastic region in the stress space requires $f < 0$. When the stresses touch the yield surface, then $f = 0$. The permanent plastic deformation grows at the rate of $\dot{\Lambda}$ and in the direction of gradients of f according to the associative flow rule, where Λ is called the plastic multiplier. The KKT condition shown in Table 1 implies that if the yield surface is not touched, equivalent to $f < 0$, then $\dot{\Lambda} = 0$ and no permanent plastic deformation is growing.

The micromorphic elastoplastic constitutive law can then be derived as the return mapping algorithm, i.e., given current elastic deformation ($\varepsilon_{ij}^e(t)$, $G_{ijk}^e(t)$) and incremental deformation ($\Delta\varepsilon_{ij}$, ΔG_{ijk}), finding the stresses at the next time step ($\sigma_{ji}(t + \Delta t)$, $\zeta_{ijk}(t + \Delta t)$).

The first step is to find the trial stresses (σ_{ji}^{tr} , ζ_{ijk}^{tr}), assuming the incremental deformation is elastic, which is equivalent to $f(\sigma_{ji}^{tr}, \zeta_{ijk}^{tr}) < 0$ as shown in Eq. (3).

$$\begin{bmatrix} \sigma_{ji}(t + \Delta t) \\ \zeta_{ijk}(t + \Delta t) \end{bmatrix} = \begin{bmatrix} \sigma_{ji}^{tr} \\ \zeta_{ijk}^{tr} \end{bmatrix} = \begin{bmatrix} \frac{\partial W(\varepsilon_{ij}^e(t) + \Delta\varepsilon_{ij}, G_{ijk}^e(t) + \Delta G_{ijk})}{\partial \varepsilon_{ij}^e} \\ \frac{\partial W(\varepsilon_{ij}^e(t) + \Delta\varepsilon_{ij}, G_{ijk}^e(t) + \Delta G_{ijk})}{\partial G_{ijk}^e} \end{bmatrix} \text{ if } f(\sigma_{ji}^{tr}, \zeta_{ijk}^{tr}) < 0 \quad (3)$$

If the incremental deformation is not elastic, i.e., $f(\sigma_{ji}^{tr}, \zeta_{ijk}^{tr}) \geq 0$, then a correction of the trial stresses should be made to ensure $f(\sigma_{ji}(t + \Delta t), \zeta_{ijk}(t + \Delta t)) = 0$, and the final stresses can be found based on the associative plastic flow rule as shown in Eq. (5). The elasticity tensors are defined as,

$$\mathbb{C}_{ijmn}^{e, \sigma-\varepsilon} = \frac{\partial^2 W}{\partial \varepsilon_{ij} \partial \varepsilon_{mn}}, \mathbb{C}_{ijmnl}^{e, \sigma-G} = \frac{\partial^2 W}{\partial \varepsilon_{ij} \partial G_{mnl}}, \mathbb{C}_{ijkmn}^{e, \zeta-\varepsilon} = \frac{\partial^2 W}{\partial \varepsilon_{mn} \partial G_{ijk}}, \text{ and } \mathbb{C}_{ijkmnl}^{e, \zeta-G} = \frac{\partial^2 W}{\partial G_{ijk} \partial G_{mnl}}. \quad (4)$$

The incremental stress update expressed in Voigt notation reads,

$$\begin{bmatrix} \sigma_{ji}(t + \Delta t) \\ \zeta_{ijk}(t + \Delta t) \end{bmatrix} = \begin{bmatrix} \sigma_{ji}^{tr} \\ \zeta_{ijk}^{tr} \end{bmatrix} - \begin{bmatrix} \mathbb{C}_{ijmn}^{e, \sigma-\varepsilon} & \mathbb{C}_{ijmnl}^{e, \sigma-G} \\ \mathbb{C}_{ijkmn}^{e, \zeta-\varepsilon} & \mathbb{C}_{ijkmnl}^{e, \zeta-G} \end{bmatrix} \begin{bmatrix} \Delta \varepsilon_{mn}^p \\ \Delta G_{mnl}^p \end{bmatrix} = \begin{bmatrix} \sigma_{ji}^{tr} \\ \zeta_{ijk}^{tr} \end{bmatrix} - \Delta \Lambda \begin{bmatrix} \mathbb{C}_{ijmn}^{e, \sigma-\varepsilon} & \mathbb{C}_{ijmnl}^{e, \sigma-G} \\ \mathbb{C}_{ijkmn}^{e, \zeta-\varepsilon} & \mathbb{C}_{ijkmnl}^{e, \zeta-G} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial \sigma_{nn}} \\ \frac{\partial f}{\partial \zeta_{mnl}} \end{bmatrix}. \quad (5)$$

The two equations above summarize the return mapping algorithm for micromorphic continua. The micropolar continua is a special case of the micromorphic continua where the micro-deformation is restricted to be rotational only. As such, the return mapping algorithm can be implemented by re-expressing the micro-rotation gradient κ_{ij} and the couple stress m_{ji} in terms of the micro-deformation gradient G_{ijk} and generalized stress ζ_{ijk} , i.e.,

$$(G_{ijk}, \zeta_{ijk}) = (-\epsilon_{ijl}\kappa_{lk}, -\frac{1}{2}\epsilon_{ijl}m_{lk}). \quad (6)$$

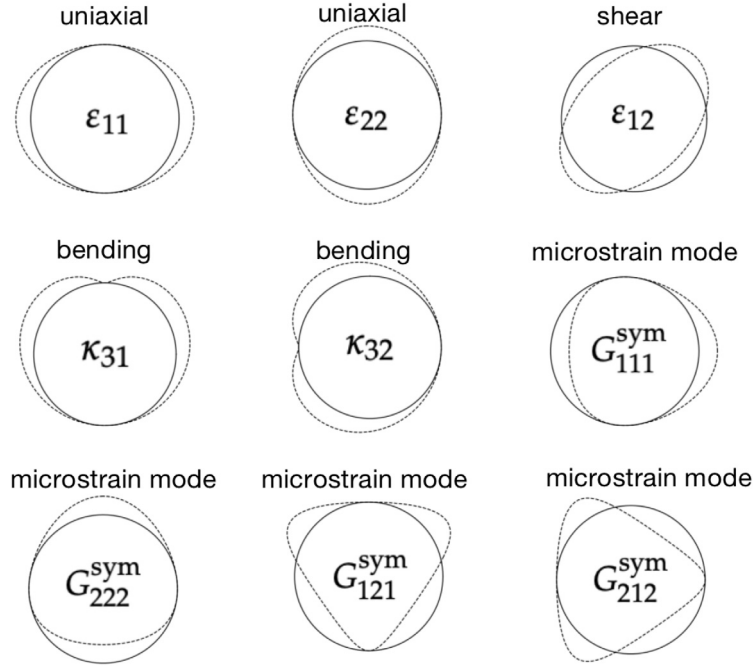


Fig. 2. The characteristic micromorphic kinematic modes.

The micromorphic return mapping algorithm then return the constitutive updates and the gradient of micro-rotation and couple stress can then be recovered via

$$(\kappa_{ij}, m_{ji}) = (-\frac{1}{2}\epsilon_{imn}G_{mnj}, -\epsilon_{imn}\zeta_{mnj}). \quad (7)$$

2.2. RVE homogenization based on Hill–Mandel’s condition

The constitutive relation of the material point at position $\mathbf{x}$, shown in Fig. 1, is obtained by studying the RVE domain Ω with the local coordinate $\mathcal{Y}$ as shown in Fig. 1. The kinematic modes denoted as $\bar{\epsilon}_{ij}$, $\bar{\kappa}_{ij}$, and $\bar{G}_{ijk}$ are prescribed by deforming the RVE with the displacement boundary condition shown in Eq. (8) and Fig. 2 [51,52], where the boundary condition is the linear combination of the prescribed kinematic modes. This boundary condition is admissible, as proven by Hill’s Lemma in Appendix A.

$$u_i = \begin{cases} \bar{\epsilon}_{ij}\mathcal{Y}_j + \frac{1}{2}\bar{G}_{ijk}\mathcal{Y}_j\mathcal{Y}_k & \text{on } \partial\Omega & \text{for micromorphic continua} \\ \bar{\epsilon}_{ij}\mathcal{Y}_j - \frac{1}{2}\epsilon_{ijl}\bar{\kappa}_{lk}\mathcal{Y}_j\mathcal{Y}_k & \text{on } \partial\Omega & \text{for micropolar continua} \\ \bar{\epsilon}_{ij}\mathcal{Y}_j & \text{on } \partial\Omega & \text{for Cauchy continua} \end{cases} \quad (8)$$

The local boundary value problem is then solved given the prescribed boundary condition to solve for the local displacement and stress field. The homogenized stress and generalized stress [52] can be computed as Eq. (9), which satisfies the Hill–Mandel’s condition as proven in Appendix A.

$$\begin{cases} \bar{\zeta}_{ijk} = \frac{1}{V} \int_{\partial\Omega} \frac{1}{2}n_l\sigma_{li}\mathcal{Y}_j\mathcal{Y}_k dS = \frac{1}{V} \int_{\partial\Omega} \frac{1}{2}(\sigma_{ji}\mathcal{Y}_k + \sigma_{ki}\mathcal{Y}_j)dV & \text{for micromorphic continua} \\ \bar{m}_{ji} = -\epsilon_{imn}\bar{\zeta}_{mnj} & \text{for micropolar continua} \\ \bar{\sigma}_{ji} = \frac{1}{V} \int_{\partial\Omega} n_l\sigma_{li}\mathcal{Y}_j dS = \frac{1}{V} \int_{\Omega} \sigma_{ji}dV & \text{for all continua} \end{cases} \quad (9)$$

2.3. Supervised learning tasks for neural kernel plasticity

The elastoplastic constitutive law can be sufficiently modeled by machine learning by training the elastic energy functional and the plastic yield function *independently*. The elastic energy functional, $W(\epsilon_{ij}^e, \kappa_{ij}^e)$ or $W(\epsilon_{ij}^e, G_{ijk}^e)$,

can be simply modeled by a neural network consisting of multi-layer perceptrons (MLP), with the strain measures as the input and the elastic stored energy as output as the output. In our implementation, we adopt the Voigt vectorized notation used in [57] to train the neural network elasticity models. For brevity, the supervised learning of elasticity for micromorphic continua will not be discussed in great detail. Interested readers may refer to, for instance, Vlassis and Sun [27]. On the other hand, the supervised learning for the yielding function and the corresponding hardening laws are formulated in the next section.

Remark 1 (Neural Network Architecture). The architecture of the MLP, which we adopted in this study, is shown in Eq. (10). The 3-layer architecture is composed of neurons equipped with Exponential Linear Unit (ELU) activation function. The activation function of the output layer A_3 can be ELU or identity map, depending on how the MLP is used: A_3 should be an identity map mostly except when the MLP is used to construct the neural kernel (NK) architecture.

$$f_{NN}(\mathbf{x}) = \mathbf{h}_3 \circ \mathbf{h}_2 \circ \mathbf{h}_1(\mathbf{x}), \text{ where } \begin{cases} \mathbf{h}_1(\mathbf{x}) = ELU(\mathbf{W}_1 \cdot \mathbf{x} + \mathbf{b}_1) \\ \mathbf{h}_2(\mathbf{h}_1) = ELU(\mathbf{W}_2 \cdot \mathbf{h}_1 + \mathbf{b}_2) \\ \mathbf{h}_3(\mathbf{h}_2) = A_3(\mathbf{W}_3 \cdot \mathbf{h}_2 + \mathbf{b}_3) \end{cases} \text{ and } ELU(x) = \begin{cases} x & \text{if } x \geq 0 \\ e^{-x} - 1 & \text{if } x < 0 \end{cases} \quad (10)$$

This neural network design is used for both the elastic stored energy functional as well as the yield surface because the derivatives of ELU are sufficiently differentiable. This smoothness may improve the robustness of the optimization process and alleviates the vanishing and exploding gradient problems.

3. Yield surface reconstruction via neural kernel (NK) method

This section describes (1) how to use the neural kernel (NK) method to reconstruct the yield surface for micromorphic continua and (2) provides the implementation details necessary to incorporate the learned model into a return mapping algorithm. Here, we represent the yield surface in a multi-dimensional parametric space via an implicit scalar signed distance level set function $f(\mathbf{x})$, such that the yield surface geometry is recovered at $f(\mathbf{x}) = 0$ where $\mathbf{x}$ stores the higher-order stress components of the implicit yield function and the internal variables.

The general framework of our NK method follows the work of Williams et al. [38] on Neural Kernel Field (NKF), while we adopt the kernel function architecture as deep neural networks for the generalizability to arbitrary dimensions. A general workflow of this framework is presented in Fig. 3 containing three major steps:

1. Generate the labeled narrow band level set data given the yield stress point $\mathbf{x}$ and plastic flow direction $\mathbf{n}$ as discussed in Section 3.1,
2. Train the kernel coefficients for the kernel function associated with the NN-based feature map ϕ_θ , after which a two-step training may be needed for the NN weight and bias θ to ensure that the sign of the yield function is correct, as shown in Algorithm 1, and
3. Predict the level set yield function $f_\theta(\mathbf{x})$ as a linear combination of the basis kernel functions and locate the surface at $f_\theta(\mathbf{x}) = 0$ (see Section 3.2).

The workflow shown in Fig. 3 presents the 2D and 3D views of the surface for the readability purpose, but the NK model is able to reconstruct a much higher-dimensional yield surface. After training the NK-based yield function, the elastoplastic constitutive law is reproduced by the return mapping algorithm presented in Section 3.3 and Algorithm 2.

3.1. Data processing

The raw data set for surface reconstruction consists of point coordinates sampled from the ground-truth surface and the corresponding normal vectors, such that the data set is described in the form of $S = \{(\mathbf{x}_i, \mathbf{n}_i) \in \mathbb{R}^d \times \mathbb{R}^d\}$, where $\mathbf{x}_i$ are point coordinates and $\mathbf{n}_i$ are the surface outward unit normal vector at $\mathbf{x}_i$; in the context of the plastic yield function, $\mathbf{x}_i$ are yield stress points and $\mathbf{n}_i$ are the plastic flow direction. For supervision purposes, we create two labeled datasets following the concept of the narrow band level set [58]. The first data set D is generated by

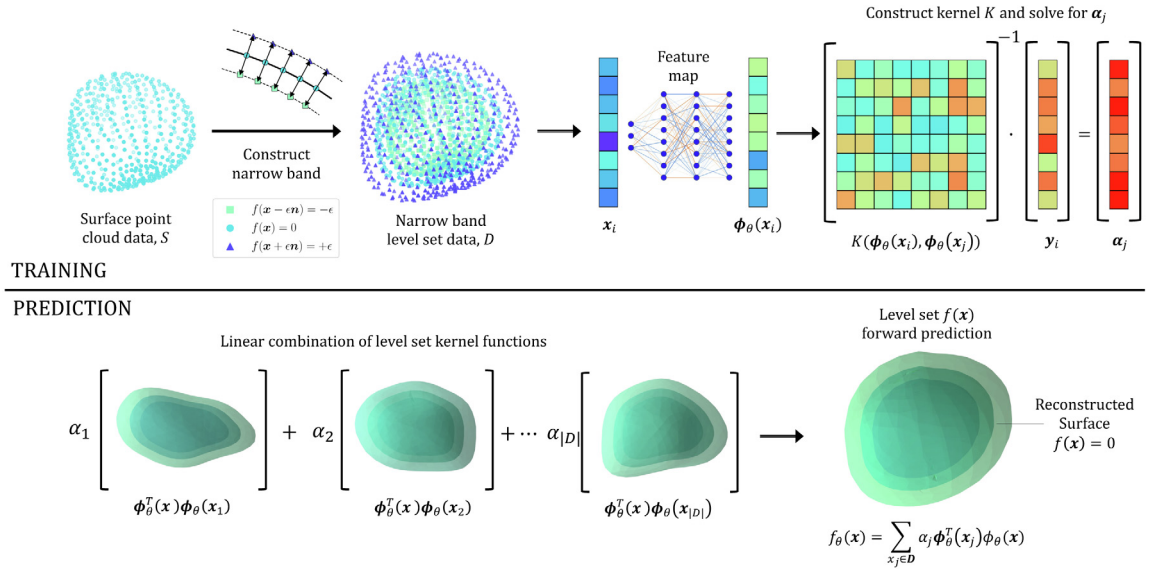


Fig. 3. The workflow of training the neural kernel function with the yield surface data consists of 3 steps: generating the labeled narrow band level set data, training the kernel coefficients for the kernel function, and finally forward predicting the level set yield function as a linear combination of the basis kernel functions.

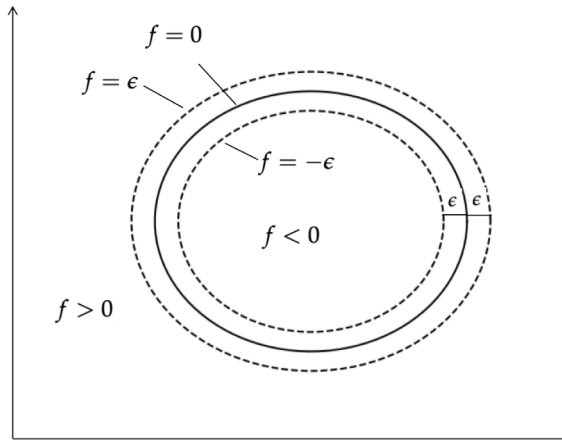


Fig. 4. Sketch of a narrow band level set yield function. The idea is to locally control the gradient around the boundary at $f = 0$ and globally control the sign of f on both sides of the boundary.

perturbing the spatial coordinate of surface points in the normal direction, and labeling them by the distance to the true surface as follows:

$$D = \{(x_i, 0) | (x_i, n_i) \in S\} \cup \{(x_i + \epsilon n_i, \epsilon) | (x_i, n_i) \in S\} \cup \{(x_i - \epsilon n_i, -\epsilon) | (x_i, n_i) \in S\} \quad (11)$$

where ϵ is a small number indicating the distance of perturbation for the surface points. The first data set controls the trained level set function to be zeros at the surface, and the gradient of the function is equal to the unit normal as shown in Fig. 4. In our numerical examples, the data are centered and scaled such that the average norm of each data point becomes one, and ϵ is tuned as a hyperparameter between 0.01 and 0.1.

The second set of labeled data V is needed to ensure the occupancy condition [59], i.e., the function should be negative inside the surface and positive outside the surface, and there should not be additional holes in the region enclosed by the surface, as shown in Fig. 4. V is sampled in the d -dimensional Euclidean space excluding the

points on the surface, where $y^{vol} = 1$ if $\mathbf{x}^{vol}$ is a point outside the surface, and $y^{vol} = -1$ otherwise.

$$V = \{(\mathbf{x}^{vol}, y^{vol}) \in \mathbb{R}^d \times \{-1, 1\} | \mathbf{x}^{vol} \notin \{\mathbf{x} | (\mathbf{x}, \mathbf{n}) \in S\}\} \quad (12)$$

3.2. Training scheme of NK method

This subsection describes the proposed NK method for surface reconstruction in detail. Before we go through the general framework of NK, we present a brief overview of classical kernel methods to facilitate further demonstrations of the NK method. Classical kernel methods for regression problems adopt a pre-defined kernel function with a set of trainable kernel coefficients α_i to approximate the function $y(\mathbf{x})$ given a labeled dataset $D = \{(\mathbf{x}_i, y_i)\}$ as follows:

$$\hat{y}(\mathbf{x}) = \sum_{(\mathbf{x}_j, y_j) \in D} \alpha_j \text{Ker}(\mathbf{x}, \mathbf{x}_j) \quad (13)$$

where the hat over $\hat{y}$ indicates an approximation of function y . The kernel coefficients α_j are trained by directly solving the following system of linear equations:

$$(\text{Ker}(\mathbf{x}_i, \mathbf{x}_j) + \lambda \delta_{ij}) \alpha_j = y_i, \quad (\mathbf{x}_i, y_i) \in D \quad (14)$$

where λ is the tunable hyperparameter used for regularization and data denoising.

It is theoretically proven that the conventional kernel method does not accurately make predictions of unseen features if the spatial dimension d is much smaller than the number of data $|D|$ [60]. In this sense, the neural network is adopted in kernel methods in order to increase the representation power of this machine learning model, where a trainable NN ϕ_θ is introduced to create a map from a d -dimensional input space to an h -dimensional feature space. The kernel regression is then implemented in the feature space enforcing h and $|D|$ in the same order of magnitude. The ϕ_θ implemented in [38] follows an architecture similar to C-OccNet [59] and is not applicable in higher-dimensional space. For generalizability to higher-dimensional inputs, we establish ϕ_θ as a multi-layer perceptron (MLP).

Instead of the kernel map of the original feature vector $\mathbf{x}$ shown in Eq. (13), the arguments of the kernel function are replaced by the feature map $\phi_\theta(\mathbf{x})$. As a result, we introduce the MLP weight θ in addition to coefficients α_j as the trainable parameters, and the architecture of the neural kernel $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}$ is described as follows:

$$\hat{y} = f_\theta(\mathbf{x}) = \sum_{\mathbf{x}_j \in D} \alpha_j K_{NS}(\phi_\theta(\mathbf{x}_j), \phi_\theta(\mathbf{x})) \quad (15)$$

where $K_{NS}(\mathbf{x}, \mathbf{y}) = \psi^T(\mathbf{x})\psi(\mathbf{y})$ is the neural spline kernel [61] induced from a single-layer non-trainable neural network ψ . In this paper, K_{NS} is replaced by $K'(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$ such that the neural spline kernel is integrated into ϕ_θ as an additional layer; which results in the following simplified NK architecture:

$$\hat{y} = f_\theta(\mathbf{x}) = \sum_{\mathbf{x}_j \in D} \alpha_j \phi_\theta^T(\mathbf{x}_j) \phi_\theta(\mathbf{x}) \quad (16)$$

We next present the loss function $L(\alpha_j, \theta)$ adopted as follows:

$$L(\alpha_j, \theta) = \sum_{(\mathbf{x}_i, y_i) \in D} (y_i - \hat{y}(\mathbf{x}_i))^2 + \gamma \sum_{(\mathbf{x}_i^{vol}, y_i^{vol}) \in V} \left(y_i^{vol} - \tanh\left(\frac{\hat{y}(\mathbf{x}_i^{vol})}{\epsilon}\right) \right)^2, \quad \text{where } \alpha_j^*, \theta^* = \underset{(\alpha_j, \theta)}{\operatorname{argmin}} L(\alpha_j, \theta). \quad (17)$$

where γ is a tunable hyperparameter. By minimizing this loss function, we achieve two goals in recovering the correct surface geometry: (1) constrain the value of f_θ to zero and the gradient of f_θ to fit the actual surface normal direction, which is satisfied as the first term goes to zero; and (2) enforce the occupancy condition so that f_θ predicts the correct sign inside and outside the yield surface, which is satisfied as the second term goes to zero. Notice that the second goal characterizes a binary classification problem, but the conventionally used binary cross entropy (BCE) loss is not included in the loss function, which is because the usage of logarithm becomes problematic with negative values in this case.

In the training process, α_j and θ are updated in an asynchronous fashion: α_j are trained by directly solving Eq. (14), but θ is updated using gradient-based optimization given fixed α_j and $\partial L/\partial \theta$. We summarize the training routine in Algorithm 1.

Algorithm 1 The NK training routine for optimizing trainable parameters α and θ

Require: Training data set D , occupancy data set V , learning rate η , hyperparameters λ and ϵ .

1. Setup the feature map neural network $\phi_\theta(\mathbf{x})$ with the trainable parameters θ .
 Define $\phi_\theta(\mathbf{x}) = ELU(\mathbf{W}_3 \cdot ELU(\mathbf{W}_2 \cdot ELU(\mathbf{W}_1 \cdot \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3)$.
 Initial $\theta = \{\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2, \mathbf{W}_3, \mathbf{b}_3\}$ with pytorch default initializer.
 2. Given pairs of feature vectors $\mathbf{x}_i, \mathbf{x}_j$ from D , compute the kernel matrix K_{ij}
 Assemble $K_{ij} = \phi_\theta^T(\mathbf{x}_i)\phi_\theta(\mathbf{x}_j)$ for $(\mathbf{x}_i, y_i), (\mathbf{x}_j, y_j) \in D$
 3. Solve the linear kernel equations for the kernel coefficients α_j .
 Solve $(K_{ij} + \lambda \delta_{ij})\alpha_j = y_i$ for $(\mathbf{x}_i, y_i) \in D$.
 4. Given the level set yield function $f_\theta(\mathbf{x})$, compute the loss function $L(\alpha_j, \theta)$ and its gradient $\partial L/\partial \theta$.
 Compute $f_\theta(\mathbf{x}) = \sum_{(\mathbf{x}_i, y_i) \in D} \alpha_j \phi_\theta^T(\mathbf{x}_j)\phi_\theta(\mathbf{x})$.
 Predict $\hat{y}_i = f_\theta(\mathbf{x}_i)$ and $\hat{y}_i^{vol} = f_\theta(\mathbf{x}_i^{vol})$ for $(\mathbf{x}_i, y_i) \in D$ and $(\mathbf{x}_i^{vol}, y_i^{vol}) \in V$.
 Compute $L(\alpha_j, \theta) = \sum_{(\mathbf{x}_i, y_i) \in D} (y_i - \hat{y}_i)^2 + \gamma \sum_{(\mathbf{x}_i^{vol}, y_i^{vol}) \in V} \left(y_i^{vol} - \tanh\left(\frac{\hat{y}_i^{vol}}{\epsilon}\right) \right)^2$.
 Differentiate $L(\alpha_j, \theta)$ for $\partial L/\partial \theta$ with `loss.backward()`.
 5. Update θ given η and $\partial L/\partial \theta$ using gradient-based optimizer like ADAM.
 Run `torch.optim.Adam( $\theta, \eta$ ).step()`.
 6. Repeat Steps 2-5 until the loss function converges and output $f_\theta(\mathbf{x})$.
-

Remark 2 (Hyperparameter Tuning).

1. $\gamma < \epsilon < r_{min}$ can be a set of good hyperparameters in Eqs. (11) and (17), where r_{min} is found by first centering the data, i.e. translating the data such that the centroid goes to the origin, and then computing the distance from the closest data point to the origin.
2. The output dimension h should be large enough, which is equivalent to increasing the dimension of the basis level set function shown in Fig. 3. Ideally, h should be comparable with $|D|$.
3. Increasing λ would also improve the convergence of the loss function when the data are noisy. In both examples of this paper, $\lambda = 0.01$ is used.

In the Sobolev training technique with deep learning, we consider the influence of the derivative of the neural network function with respect to the network input in the loss function, such that we enforce the prediction accuracy for the neural network derivative with respect to its input. For the global loss function, we directly supply the original loss in Eq. (17) with the MSE between the groundtruth and predicted surface normal directions:

$$L_{Sob}(\alpha, \theta) = \sum_{(\mathbf{x}_i, y_i) \in D} (y_i - \hat{y}_i)^2 + \gamma \sum_{(\mathbf{x}_i^{vol}, y_i^{vol}) \in V} \left(y_i^{vol} - \tanh\left(\frac{\hat{y}_i^{vol}}{\epsilon}\right) \right)^2 + \gamma' \sum_{(\mathbf{x}_i, y_i) \in D} \left\| \mathbf{n}_i - \frac{\partial \hat{y}_i(\mathbf{x}_i)}{\partial \mathbf{x}_i} \right\|^2 \quad (18)$$

where L_{Sob} is the Sobolev loss function we adopt, γ' is a hyperparameter controlling the influence of derivative term in the global loss. We will compare NK with the MLP-based level set (MLP-LS) method, which is documented in [27]; the details of the MLP-LS method are provided in Appendix C.

3.3. Return mapping algorithm

The return mapping algorithm for micromorphic materials is presented in Algorithm 2, which follows the return mapping theory shown in Eqs. (3) and (5). We further assume that the yield surface is able to evolve, and such hardening process is governed by an internal variable Λ , i.e., the magnitude of the cumulative plastic strain (plastic multiplier), such that $f_\theta(\mathbf{x}, \Lambda)$ represents a family of yield functions evolving with Λ . For micropolar

materials, convert the (κ_{ij}, m_{ji}) into $(G_{ijk}, \zeta_{ijk}) = (-\epsilon_{ijl}\kappa_{lk}, -\frac{1}{2}\epsilon_{ijl}m_{lk})$ before Algorithm 2 and convert back by $(\kappa_{ij}, m_{ji}) = (-\frac{1}{2}\epsilon_{imn}G_{mnj}, -\epsilon_{imn}\zeta_{mnj})$ after the return mapping. The notation $\overset{3}{\mathbf{G}}$ and $\overset{3}{\boldsymbol{\zeta}}$ are used to represent the 3rd-order gradient of the micro-deformation tensor and micromorphic generalized stress tensor respectively.

The implementation of the return mapping algorithm requires the elastic energy functional $W(\boldsymbol{\epsilon}, \overset{3}{\mathbf{G}})$, the yield function $f_\theta(\mathbf{x}, \Lambda)$, where $\mathbf{x}$ is the vector consisting of the components of $\boldsymbol{\sigma}$ and $\overset{3}{\boldsymbol{\zeta}}$; given the pre-trained plastic yield function, the plastic flow direction $\nabla_{\mathbf{x}} f_\theta(\mathbf{x}, \Lambda)$ can be derived by pytorch automatic differentiation and $\nabla_\Lambda f_\theta(\mathbf{x}, \Lambda) = 0$ in the case of perfect plasticity. Unless otherwise stated, $f_\theta(\mathbf{x}, \Lambda)$ is derived from the pretrained machine learning model, either from NK or MLP-LS, and $W(\boldsymbol{\epsilon}, \overset{3}{\mathbf{G}})$ comes from the Sobolev training of elastic energy functional [62].

Given all the necessary ingredients, Algorithm 2 first computes the elastic trial Cauchy and higher-order stresses following Eq. (3). If yielding is detected by $f_\theta(\mathbf{x}^{tr}, \Lambda) \leq 0$, a system of nonlinear equations following Eq. (5) would be solved to find the vectorized stresses $\mathbf{x}$ and the increment in the plastic multiplier $\Delta\Lambda$. Otherwise, the trial states are directly output.

Algorithm 2 Return mapping algorithm for higher-order continuum.

Require: Current internal variable Λ , elastic strain $\boldsymbol{\epsilon}_n^e$, elastic gradient of micro-deformation $\overset{3}{\mathbf{G}}_n^e$, increments of deformation $\Delta\boldsymbol{\epsilon}$, $\Delta\overset{3}{\mathbf{G}}$, yield function $f_\theta(\mathbf{x}, \Lambda)$, and elastic energy functional $W(\boldsymbol{\epsilon}, \overset{3}{\mathbf{G}})$.

Outputs: Higher-order stresses $\boldsymbol{\sigma}_{n+1}$, $\overset{3}{\boldsymbol{\zeta}}_{n+1}$, and plastic multiplier Λ at the next time step.

1. Compute trial elastic strain, trial elastic stress, and elasticity tensor.

$$\begin{aligned} \text{Compute } \begin{bmatrix} \boldsymbol{\epsilon}_{n+1}^{e, tr} \\ \overset{3}{\mathbf{G}}_{n+1}^{e, tr} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\epsilon}_n^e + \Delta\boldsymbol{\epsilon} \\ \overset{3}{\mathbf{G}}_n^e + \Delta\overset{3}{\mathbf{G}} \end{bmatrix}. \\ \text{Compute } \begin{bmatrix} \boldsymbol{\sigma}_{n+1}^{tr} \\ \overset{3}{\boldsymbol{\zeta}}_{n+1}^{tr} \end{bmatrix} &= \begin{bmatrix} \partial W / \partial \boldsymbol{\epsilon} \\ \partial W / \partial \overset{3}{\mathbf{G}} \end{bmatrix} \text{ at } (\boldsymbol{\epsilon}_{n+1}^{e, tr}, \overset{3}{\mathbf{G}}_{n+1}^{e, tr}). \\ \text{Vectorize } \mathbf{x}^{tr} &= \begin{bmatrix} \boldsymbol{\sigma}_{n+1}^{tr} \\ \overset{3}{\boldsymbol{\zeta}}_{n+1}^{tr} \end{bmatrix}. \\ \text{Matrixize } \mathbb{C}^e &= \begin{bmatrix} \partial^2 W / \partial \boldsymbol{\epsilon}^2 & \partial^2 W / \partial \boldsymbol{\epsilon} \partial \overset{3}{\mathbf{G}} \\ \partial^2 W / \partial \boldsymbol{\epsilon} \partial \overset{3}{\mathbf{G}} & \partial^2 W / \partial \overset{3}{\mathbf{G}}^2 \end{bmatrix} \text{ at } (\boldsymbol{\epsilon}_{n+1}^{e, tr}, \overset{3}{\mathbf{G}}_{n+1}^{e, tr}). \end{aligned}$$

2. Check the yield condition and perform return mapping if yield is detected.

if $f_\theta(\mathbf{x}^{tr}, \Lambda) \leq 0$ **then**

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{tr}, \quad \overset{3}{\boldsymbol{\zeta}}_{n+1} = \overset{3}{\boldsymbol{\zeta}}_{n+1}^{tr}$$

else

$$\text{Solve for } \mathbf{x} \text{ and } \Delta\Lambda, \text{ such that } \begin{bmatrix} \mathbf{x} - \mathbf{x}^{tr} + \mathbb{C}^e \Delta\Lambda \nabla_{\mathbf{x}} f_\theta(\mathbf{x}, \Lambda + \Delta\Lambda) \\ f_\theta(\mathbf{x}, \Lambda + \Delta\Lambda) \end{bmatrix} = \mathbf{0}.$$

$$\text{Extract } \boldsymbol{\sigma}_{n+1} \text{ and } \overset{3}{\boldsymbol{\zeta}}_{n+1} \text{ from } \mathbf{x} = \begin{bmatrix} \boldsymbol{\sigma}_{n+1} \\ \overset{3}{\boldsymbol{\zeta}}_{n+1} \end{bmatrix}, \text{ update } \Lambda = \Lambda + \Delta\Lambda.$$

Remark 3 (*Stress Integration for Non-Convex Yield Surfaces*). As pointed out by Lin and Bažant [63], there do exist non-convex yield surfaces in the literature, such as the Argyris yield surface (cf. Argyris et al. [64]) and the Barcelona Basic Model (cf. Sheng et al. [65]), which purposely introduce non-convexity for the sake of matching the phenomenological responses observed from experiments. As we did not enforce the convexity of the yield function explicitly, the resultant yield functions (see Figs. 11 and 14) are found to have concave regions. A robust implicit

stress integration may require specific treatment to find the first intersection between the non-convex yield function and an elastic trial stress point (cf. Pedroso et al. [66] and Sheng et al. [65]). In our case, we follow the treatment in Glüge and Bucci [67], in which an incremental trial step smaller than the radius of the curvature of the yield surface is used such that the return mapping algorithm may yield a unique corrected state.

4. Numerical experiments

In this section, two examples of yield surface reconstruction with MLP-LS and NK are presented, each followed by a performance evaluation. In the first example, both methods are verified by the analytical micropolar J2 yield surface [68], followed by a short case study comparing the performance of both methods given limited or missing data. In the second example, both methods are validated by a direct numerical simulation (DNS) data upscaled from finite element simulations. For brevity, we only present the comparisons of yield surfaces obtained from MLP-LS and MK methods here. The results obtained from other alternative approaches (e.g., Gaussian Kernel, and NURBS) are presented in the [Appendix](#).

4.1. Two-dimensional micropolar J2 plasticity model

The first example applies the MLP-LS and NK methods to reconstruct the micropolar J2 yield surface [68] with a data set inferred from the yield function in Eq. (19). This micropolar J2 plasticity model can be viewed as a generalized version of the classical J2 plasticity, where the additional terms with respect to the couple stress tensor $\mathbf{m}$ and micropolar length scale l are introduced to capture the size effect. The spatial dimension is reduced to 2D assuming plane stress, i.e. $\sigma_{33} = 0$, such that only the components σ_{11} , σ_{22} , σ_{12} , m_{13} , and m_{23} are non-zero in the yield function, where the mean stress $p = (\sigma_{11} + \sigma_{22})/3$ and the deviatoric stress $\mathbf{s} = \boldsymbol{\sigma} - p\mathbf{I}$.

$$f(\boldsymbol{\sigma}, \mathbf{m}) = \sqrt{\mathbf{s} : \mathbf{s} + \frac{\mathbf{m} : \mathbf{m}}{l^2}} - \sqrt{\frac{2}{3}}Y \quad (19)$$

where Y indicates the yield stress. The numerical specimen used for verification follows linear micropolar elasticity [68,69],

$$\boldsymbol{\sigma} = K \operatorname{tr}(\boldsymbol{\varepsilon})\mathbf{I} + 2\mu\boldsymbol{\varepsilon}^{dev} + 2\mu_c\boldsymbol{\varepsilon}^{skw}, \quad (20)$$

with bulk modulus $K = \frac{100}{3}$ MPa, shear modulus $\mu = 50$ MPa, coupled shear modulus $\mu_c = 0$, and yield stress $Y = \sqrt{\frac{2}{3}}$ MPa. Due to the micropolar effect, the strain tensor is divided into the hydrostatic part $\operatorname{tr}(\boldsymbol{\varepsilon})\mathbf{I}$, deviatoric part $\boldsymbol{\varepsilon}^{dev}$ and the skew-symmetric part $\boldsymbol{\varepsilon}^{skw}$.

The performance of the two methods is examined by visualizing cross sections of the yield surface and the results of the return mapping algorithm. According to the test results, both methods are able to detect the yield point on the yield surface accurately and produce correct higher-order stress curves when the trained yield function is integrated into a return mapping algorithm. An additional case study with limited and missing data is conducted to evaluate the performance of both methods given low-quality data.

4.1.1. Yield surface reconstruction and return mapping

The yield data points generated by Algorithm 3 in [Appendix B](#) are used to train both the MLP-LS and the NK models. The MLP-LS model, constructed by a 3-layer MLP shown in Eq. (10) function and 64 neurons in each hidden layer, is trained with the batch size of 1000 for 500 epochs. The NK model, whose feature map $\boldsymbol{\phi}_\theta$ is constructed by a 3-layer MLP architecture with 64 neurons in each hidden and output layer, is trained for 500 epochs. The training and validation loss histories are shown in [Fig. 5](#), where the validation loss can be less than the training loss because the training loss function has an additional term controlling the sign of the yield function, as shown in Eq. (17).

The smooth yield surface is reconstructed by the MLP-LS and NK. The cross sections on $\sigma_{11} - \sigma_{22}$, $\sigma_{11} - \sigma_{12}$, and $m_{13} - m_{23}$ planes are shown in [Fig. 6](#), where both methods accurately reconstruct a yield surface that respects the ground truth yield points. Therefore, we consider that both methods are generalizable to higher-dimensional yield surfaces given a set of sufficient and well-distributed data.

In addition to the yield surface reconstruction, the return mapping algorithm 2 is implemented to verify both MLP-LS and NK methods. The material is loaded elastoplastically in a single kinematic mode, i.e. one of the

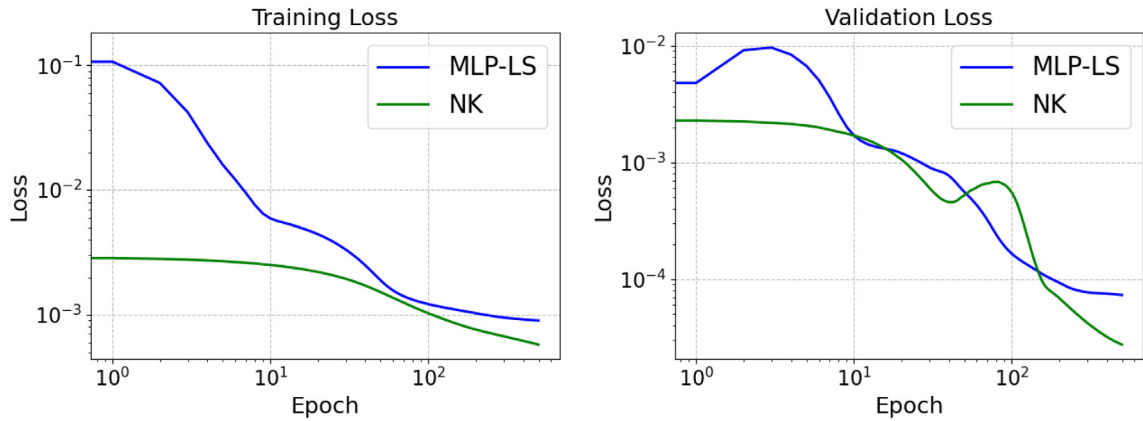


Fig. 5. Training and validation losses when learning the analytical plastic yield function.

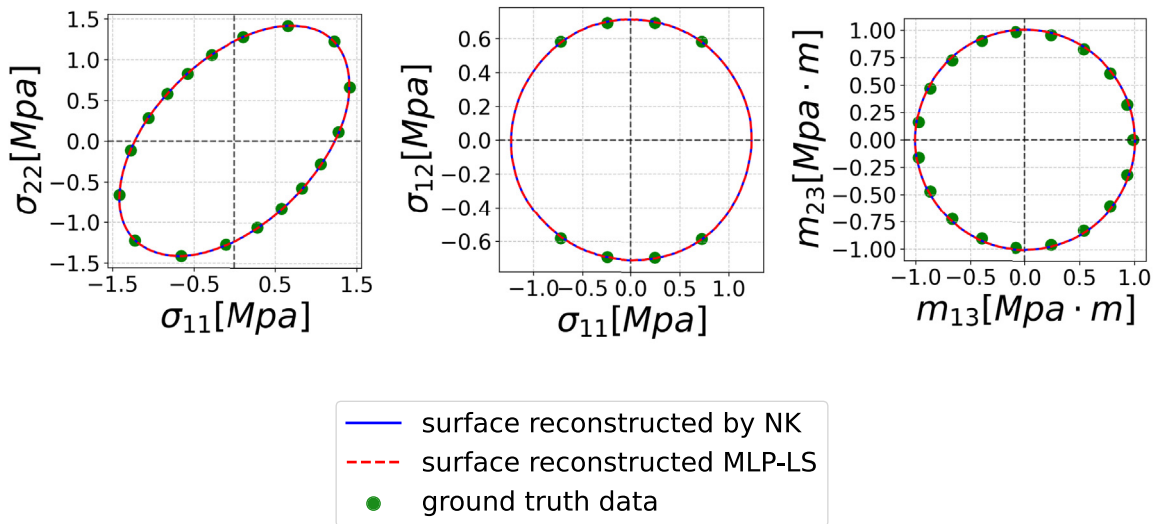


Fig. 6. Cross sections of the analytical yield surface.

uniaxial tension, shear, and bending modes, and then unloaded elastically. The test results of the return mapping, as shown in Fig. 7, indicate that both methods can be integrated into the return mapping algorithm and are able to produce valid stress curves.

4.1.2. Case study with limited and missing data

It has been shown that given a set of sufficient and well-distributed yield point data, both MLP-LS and NK methods are able to reconstruct the yield surface accurately, as shown in Fig. 6. However, when the sufficient and well-distributed data are not available, the performance of the two methods is of interest. Therefore, two sets of non-ideal data are generated from the training set to test the performance of the two methods. The first data set, called the set of limited data, is sampled from the training set with a limited size via `numpy.random.choice()`, such that the data set is as well-distributed as the training set but has a much smaller data size. The other data set, called the set of missing data, is generated by removing a cluster of yield point data from the training set, such that the data set has a missing data patch and is considered poorly distributed.

In this case study, the limited data set with 800 data is sampled from the training set, and the missing data is generated by removing a cluster of data as shown in Fig. 8 (MIDDLE), which contains 13 665 data after the data

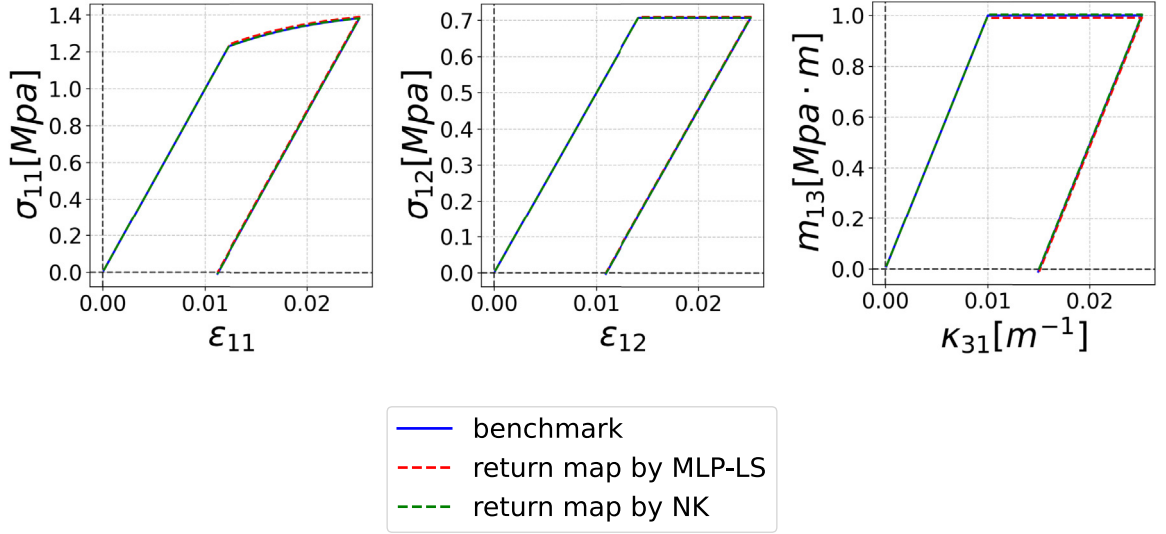


Fig. 7. Return mapping results of uniaxial (LEFT), shear(MIDDLE), and bending (RIGHT) tests verify both MLP-LS and NK methods.

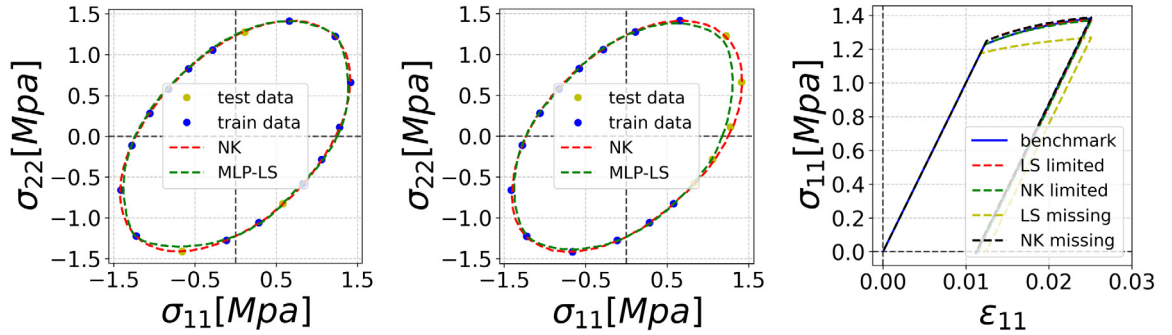


Fig. 8. Surface reconstructed given limited data (LEFT), missing data (MIDDLE), and the return mapping results of uniaxial tension test (RIGHT).

cluster is removed. It is observed from Fig. 8 that given limited data that follows the same distribution with the test data, the yield surface can still be reconstructed accurately by both methods, shown in Fig. 8 (LEFT); however, when the missing data set is used for training, it is observed in Fig. 8 (MIDDLE and RIGHT) that NK produces a significantly more accurate prediction of the test data and stress history than MLP-LS, which reflects the robustness against the missing data.

4.2. Multiscale homogenization for anisotropic plasticity of layered clay

The second example presents the application of the neural kernel method to reconstruct micropolar and micromorphic yield surfaces upscaled from direct numerical simulations. We select an idealized microstructure commonly used to represent shale, i.e. a layered material consisting of hard and soft constituents [70–72]. Readers interested in previous mathematical and neural network modeling efforts of layered geomaterials may refer to Semnani et al. [73], Zhao et al. [74], Borja et al. [75] and Xiao and Sun [35]. To the best knowledge of the authors, there has not yet been any attempt to model the micropolar and micromorphic plasticity of shale via deep learning.

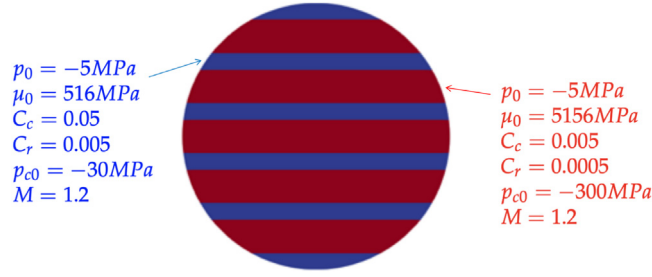


Fig. 9. The internal structure of the RVE consists of layers of clay materials, with the material parameters labeled.

4.2.1. Data preparation

The RVE data is obtained from finite element simulations on a domain consisting of layered clay materials composed of a hard and a soft constituent with intact interfaces (see Fig. 9). These layer constituents are assumed to be Cauchy continua with elasto-plastic behaviors characterized by the classical Cam-Clay model but with different material parameters. Since the constituents are assumed to exhibit no higher-order effects, the higher-order effect of the effective medium is stemmed from the spatial heterogeneity of the micro-structures. The elastic response is captured by the following elasticity energy function

$$\psi^e(\boldsymbol{\varepsilon}^e) = -p_0 C_r \exp\left(-\frac{\varepsilon_v^e}{C_r}\right) + \frac{3}{2} \mu_0 (\varepsilon_s^e)^2, \quad \varepsilon_v^e = \text{tr}(\boldsymbol{\varepsilon}^e), \quad \varepsilon_s^e = \|\boldsymbol{\varepsilon}^e - \frac{1}{3} \varepsilon_v^e \mathbf{1}\| \quad (21)$$

where p_0 is the initial pressure, ε_{v0}^e is the initial volumetric deformation, μ_0 indicates a constant shear modulus, and C_r is the elastic re-compression ratio. The yield function with the hardening law is captured by,

$$f(\boldsymbol{\sigma}) = \frac{q^2}{M^2} + p(p - p_c), \quad p = \frac{\text{tr}(\boldsymbol{\sigma})}{3}, \quad q = \sqrt{\frac{3}{2}} \|\boldsymbol{\sigma} - p \mathbf{1}\| \quad (22)$$

where p_c the preconsolidation pressure, M is the slope of the critical state line. The hardening law that governs the evolution of p_c is expressed as follows:

$$p_c = p_{c0} \exp(\varepsilon_v^p / (C_c - C_r)), \quad \varepsilon_v^p = \text{tr}(\boldsymbol{\varepsilon}^p) \quad (23)$$

where C_c is the plastic compression index and p_{c0} indicates the initial preconsolidation pressure.

This data set is generated from the same finite element domain used in [35]. The boundary value problem that generates the data set is solved also by the same finite element solver (cf. Xiao and Sun [35]) that employs the deal.ii library [76]. To capture the high-order constitutive behaviors, data are collected by applying the admissible boundary conditions, solving local BVP, and homogenizing the higher-order stresses according to Eqs. (8) and (9) in Section 2.

The DNS yield data are first sampled by loading the RVE at evenly parametrized deformation rate by Algorithm 4 in Appendix B, and the yield points are recorded when permanent deformation is detected.

4.2.2. Reconstruction of yield surfaces

We present the yield surfaces in a higher-dimensional stress space. The results are evaluated by comparing the accuracy and robustness of the yielding and hardening behaviors from unseen stress paths predicted by the NK and MLP-LS (which serves as the benchmark model).

To present a yield surface in a higher-dimensional stress space, the cross sections of the yield surfaces are presented by projecting the surface onto the stress planes defined by the combinations of two stress components. Two cases are studied: micropolar and micromorphic yield surfaces in higher-dimensional stress spaces. The results show that both MLP-LS and NK are able to capture the complex features of the higher-dimensional yield surface where data is sufficient. However, when data is sparsely distributed at some parts of the yield surface, NK outperforms MLP-LS in terms of extrapolating the unseen data.

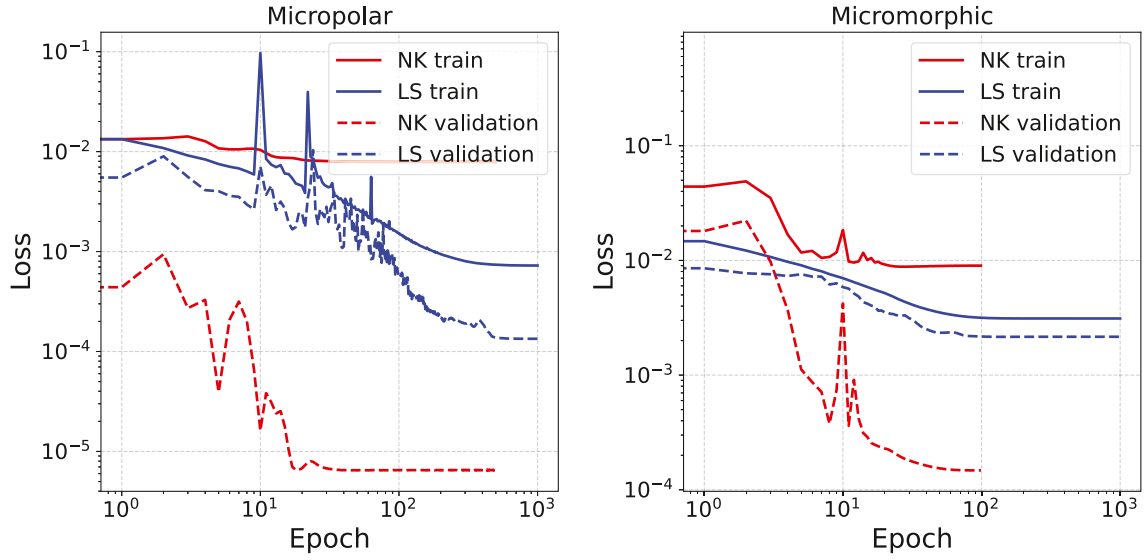


Fig. 10. Training and validation losses of different combinations of yield surfaces and models. The micropolar and micromorphic yield surfaces can be learned by either NK or MLP-LS (LS).

4.2.3. Hyperparameters

Different sets of hyperparameters are used to learn the two yield surfaces with different dimensions. The micropolar yield function consists of 5 stress components, learned by the deep architecture with an input layer of 5 neurons. The MLP architecture consists of two dense hidden layers of 256 neurons and an output layer of 1 neuron, while the NK architecture consists of three dense layers of 512 neurons. The MLP-LS is trained with the batch size of 100 and the learning rate of 0.01 for 1000 epochs, while NK is trained for 500 epochs with the batch size of 200 and the learning rate of 0.0001, with the other hyperparameters being $\gamma = 0.1$, $\epsilon = 0.02$, $\lambda = 0.01$. In the micromorphic case, the yield function consists of 9 stress components, such that the input layer with 9 neurons is used. The MLP with the same hidden dimension is trained with the batch size of 100 and the learning rate of 0.005 for 1000 epochs, while the NK architecture that consists of three dense layers of 512, 512, and 1024 neurons is trained for 100 epochs with the batch size of 500, learning rate of 0.0002, with the other hyperparameters being $\gamma = 0.1$, $\epsilon = 0.05$, $\lambda = 0.01$.

4.2.4. Training of neural kernels

The training and validation loss histories of different combinations of yield surfaces and models are shown in Fig. 10. The training loss is generally higher than the validation loss due to the additional term in the training loss function Eq. (17) that controls the sign of the yield function.

The strikes in the loss history are probably because of the mini-batch gradient-descent optimization of the loss function, where the loss is not guaranteed to be consistently decreasing, and the gradient evaluated on some data batches may be very large to create the instability of the loss history. In general, the NK has a higher training loss but a lower validation loss than MLP-LS which is more likely to overfit the data. The validation loss is defined by $|f_{\theta}(\mathbf{x}_i)|$ given the test data $\mathbf{x}_i$, which does not reflect the accuracy of prediction defined by $\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|$, where $\hat{\mathbf{x}}_i$ is the predicted yield point ($f_{\theta}(\hat{\mathbf{x}}_i) = 0$), but the convergence of the validation loss reflects a decent accuracy of prediction.

4.2.5. Micropolar yield surfaces

Since the yield surface for higher-order continua depends on more than 3 variables, it is not feasible to fully visualize the learned yield surface geometrically in a three-dimensional space. As such, we projected the high-dimensional yield surface onto 2D stress planes (where the rest of the stress components and internal variables are fixed) to demonstrate the geometrical features of the yield surfaces in the high-dimensional models. We then further

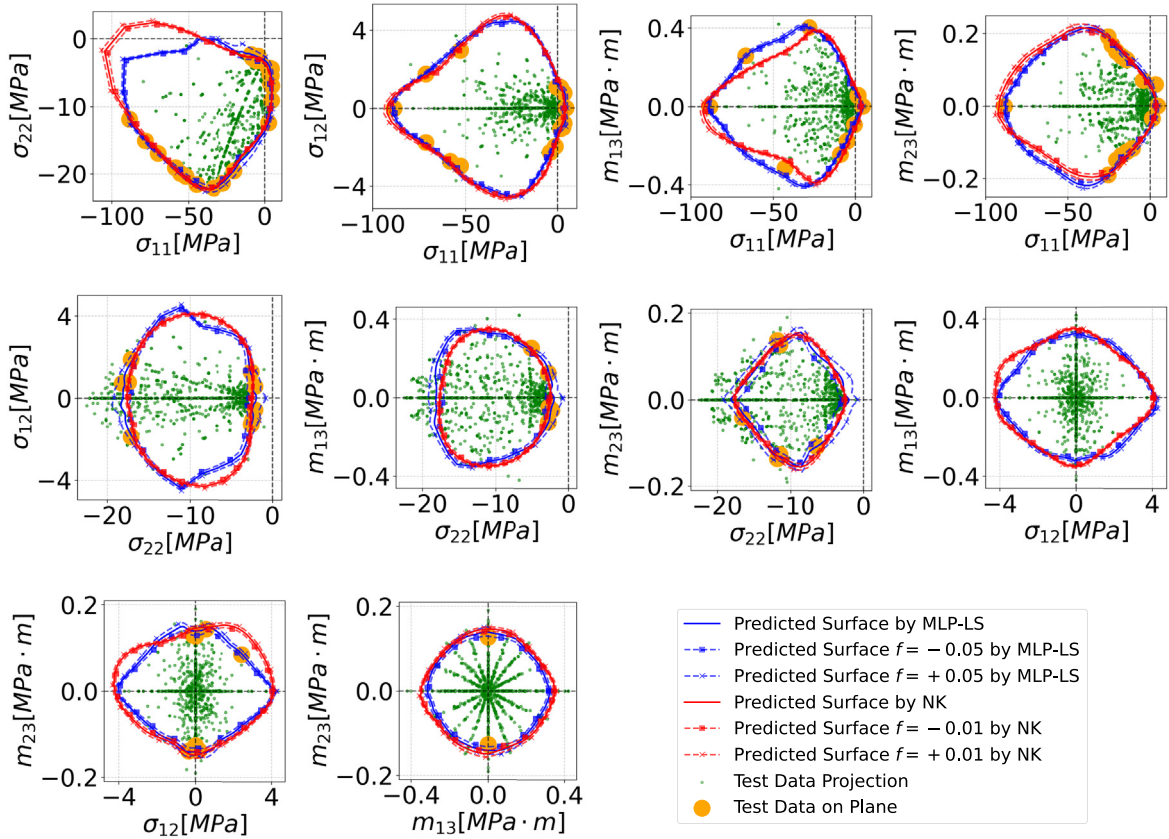


Fig. 11. Cross sections of the micropolar yield surface reconstructed by MLP-LS and NK.

examine the results via 2D prediction vs. ground-truth plots, samples of stress paths, and stress–strain curves for individual stress/strain components.

Micropolar yield surfaces projected onto 2D stress planes. The micropolar yield surface is visualized by the cross sections projected on 10 stress planes of different combinations of stress components, as shown in Fig. 11; the ground truth yield data are projected to the 10 stress planes as well and compared with the yield points predicted by NK and MLP-LS. We observe that both NK and MLP-LS are able to capture the complex features of the higher-dimensional yield data. Furthermore, The yield surfaces parametrized by both approaches are **geometrically similar** in most regions, except for a few locations where the training data is sparse, e.g., the top left corner of the yield surface projected to the first stress plane in Fig. 11. This convergence of the learned function indicates that the data is sufficiently abundant to constrain the geometry of the yield surfaces. As such, the feature extraction enabled by the data-dependent kernel of NK does not lead to significant difference in the data-rich regions.

To have a better understanding of the difference between the two predicted surfaces, we inspect the yield surface projected on the first stress plane and generate two additional unseen data as shown in Fig. 12. It is observed that although MLP-LS fits the training data more accurately, NK outperforms MLP-LS when extrapolating the missing data. The same phenomenon is observed in Fig. 8 in the previous example. The reason why NK is better at extrapolating is probably that the function space of the NK model is spanned by a controlled number of basis kernel functions as shown in Fig. 3, such that the pattern of the yield surface is recognized by the linear Galerkin projections of the data to the finite-dimensional function space; the yield surface can then be extrapolated more reasonably based on the learned pattern.

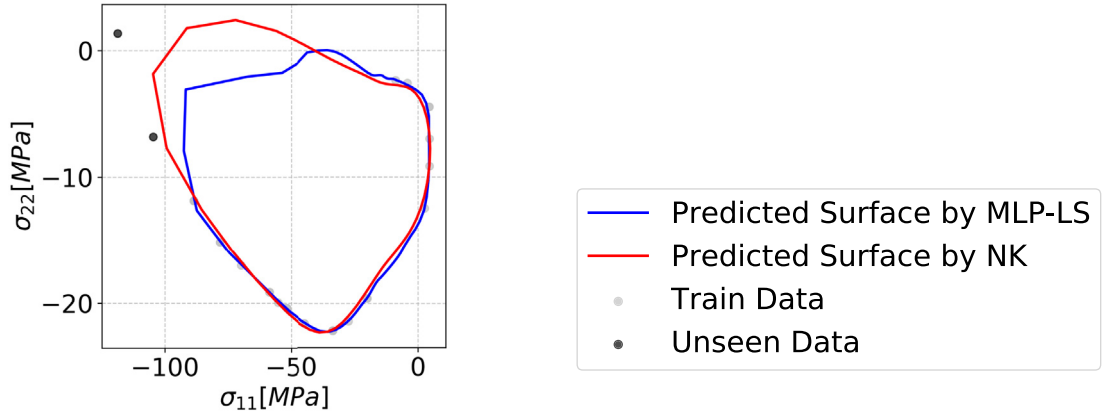


Fig. 12. Considering the $\sigma_{11} - \sigma_{22}$ plane, although MLP-LS is able to fit the training data more accurately, NK is able to better extrapolate the surface with missing data and fit the unseen data more accurately.

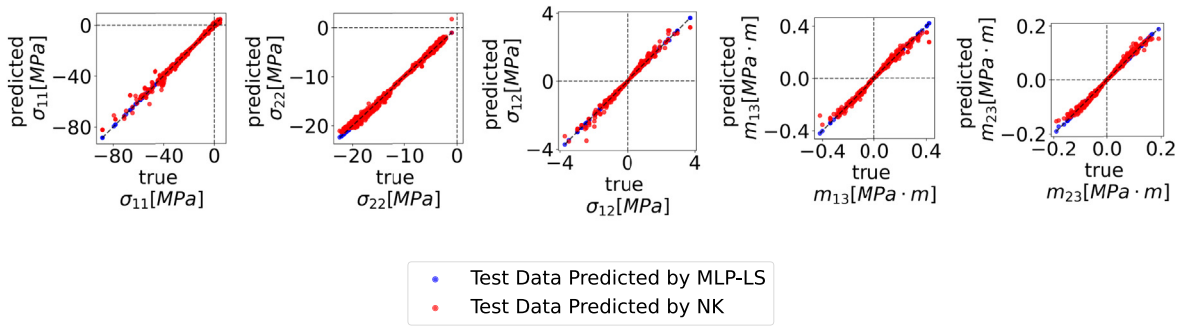


Fig. 13. Micropolar test data compared component-by-component with predictions by MLP-LS and NK.

Accuracy on unseen data. The unseen test yield data (sampled within the identical distribution of the train data) are compared with the yield point predictions component-wisely in Fig. 13. In this Prediction vs. Ground Truth figure, a perfect result would be a straight line with a 45-degree inclination angle. In our case, the micropolar yield surface generated by MK and MLP-LS both demonstrate sufficient accuracy, with the MLP-LS performing slightly better. This performance difference could be attributed by the different ways NK and MLP-LS parametrize the yield surface. In the NK case, the inductive bias obtained from the training is generated via the finite-dimensional space spanned by the data-dependent basis kernels which is used to express the yield surface. In the MLP-LS case, the yield surface is directly parametrized by the neural network weights. This setting is less restrictive than the kernel approach where the learned yield function must be an element of the space spanned by the kernel basis. As such, assuming that the data is sufficiently populated in this micropolar experiment, the performance gain of the MLP-LS might have been attributed to the higher expressivity of the MLP neural network [77].

4.2.6. Micromorphic yield surfaces

Compared to the micropolar case, the dimension of the micromorphic yield surface is higher. With 9 stress components and an internal variable as the input, the yield surface would require significantly more data to populate the parametric space. Theoretically, the number of data needed to produce similar performance increases exponentially with the dimension [78], which means if 10^3 is needed to reconstruct the micropolar yield surface accurately, then approximately 10^6 data is needed in the micromorphic case in order to reach the similar performance. This data sparsity manifested by the high dimensionality might contribute to the fact that there are so few attempts to hand-craft yield surfaces for micromorphic continua and most of them are simply an extension of the Cauchy continuum counterpart. As such, many of the micromorphic and micropolar simulations are often

related to FEM^2 multiscale approach that is computationally expensive [79,80]. Since it is increasingly unrealistic to always expect that we will have sufficient data when we increase the dimensionality of the model to incorporate the micromorphic effect, the robustness of the learning algorithms **in the sparse data regime** becomes critical.

In this micromorphic example, 16 904 data points (split into 13 500 training data and 3404 test data) are used to train and test the model. In other words, we have only increased the training data by 2.25 times (cf. [Appendix B.2](#)) whereas the data demand is expected to be increased by 3 orders. This sparsity of data is intentional, as we would like to investigate whether the feature space generated from the data-dependent kernel may enable us to generate a more robust inductive bias for extrapolation when the data is limited.

Micromorphic yield surfaces projected onto 2D stress planes. We first examine the NK and MLP-LS yield surfaces by projecting them onto 36 different stress planes. By comparing [Fig. 14](#) with [Fig. 11](#), this micromorphic data is distributed in a much sparser manner due to the increased dimensionality. In many of the planes shown in [Fig. 14](#), there is not even one single data point on the same plane (the orange point(s)). The sparsity showcased in these 2D planes indicates that the inductive bias inferred from the rest of the data becomes the only dominant factor dictating the prediction accuracy in those data-missing regions.

At the location regions far away from the DNS data (see $\sigma_{22} - \zeta_{111}$, $m_{23} - \zeta_{212}^{sym}$, and $\zeta_{111} - \zeta_{212}^{sym}$ planes.), MLP-LS method tends to generate yield surfaces of more complex shapes that vary significantly among different 2D stress planes. On the other hand, the NK yield functions (see [Fig. 14](#)) have significantly less concave regions and generally maintain a convex shape. The yield surface projected on different stress planes also exhibits more consistent geometrical patterns than those obtained from MLP-LS. This difference in the resultant yield functions is attributed to the sparsity of the data, which makes the learned function depend more significantly on the hypothesis sets employed by the NK and MLP-LS models (cf. Mohri et al. [78]).

Remarkably, in this data-limited region, the NK model seems to be capable of exploiting the structure and similarity of the data in the feature space. This exploitation on the structure and similarity of data seems to be helpful in preventing overfitting (and hence the less complex yield surface) as well as enabling the learned model to be consistent with the underlying physical laws obeyed by the data. In particular, while both the NK and MLP-LS algorithms are subjected to the DNS data set compatible to the thermodynamics principles, only the NK model yields a convex yield surface compatible with the thermodynamics constraint where data are sparse. This result is particularly interesting, because the convexity of the yield function has not been explicitly enforced via loss function or specific neural network architecture design.

Accuracy on unseen data. As shown in [Fig. 15](#), both NK and MLP-LS are able to predict the micromorphic test data with reasonable accuracy in the unseen data. However, due to the high dimensionality and sparsity of the data, the similar accuracy in predicting a limited set of test data does not necessarily imply the similarity in the geometry of the learned yield function.

As such, the test data sampled within the same distribution of the training may not be sufficient in painting a complete picture of the prediction performance. Instead, an adversarial sampling on regions distant from the training data may provide more useful insight on the robustness of the model, as shown in numerical examples demonstrated in [Figs. 8](#) and [12](#).

4.2.7. Validation exercise for constitutive responses along unseen loading paths

This subsection shows the hardening process by the evolution of the yield surface and validates the NK return mapping algorithm (Algorithm 2) by the benchmark DNS stress history. To realistically reproduce the stress curve, the hardening process is modeled by introducing the magnitude of cumulative plastic strain as the internal variable, equivalent to an additional dimension of the higher-order yield surface; the initial yield surface expands as the internal variable increases. To further validate the constitutive modeling with hardening, the benchmark DNS stress history as the ground truth is first generated by controlling the deformation rate and solving the local BVP with the prescribed time-dependent boundary conditions. Another stress history is then produced by the NK return mapping algorithm and validated by the DNS benchmark. Given that the micropolar and micromorphic data could be too sparse, the DNS simulations that generate the data are run sequentially with the machine learning step described in Section 3. This adaptive strategy enables us to sample additional stress data at the locations where local support of the NK-based yield function is needed to amend the neural kernels. Note that a more rigorous active sampling strategy could potentially be derived via deep reinforcement learning, as shown in [81,82]. The rational design of

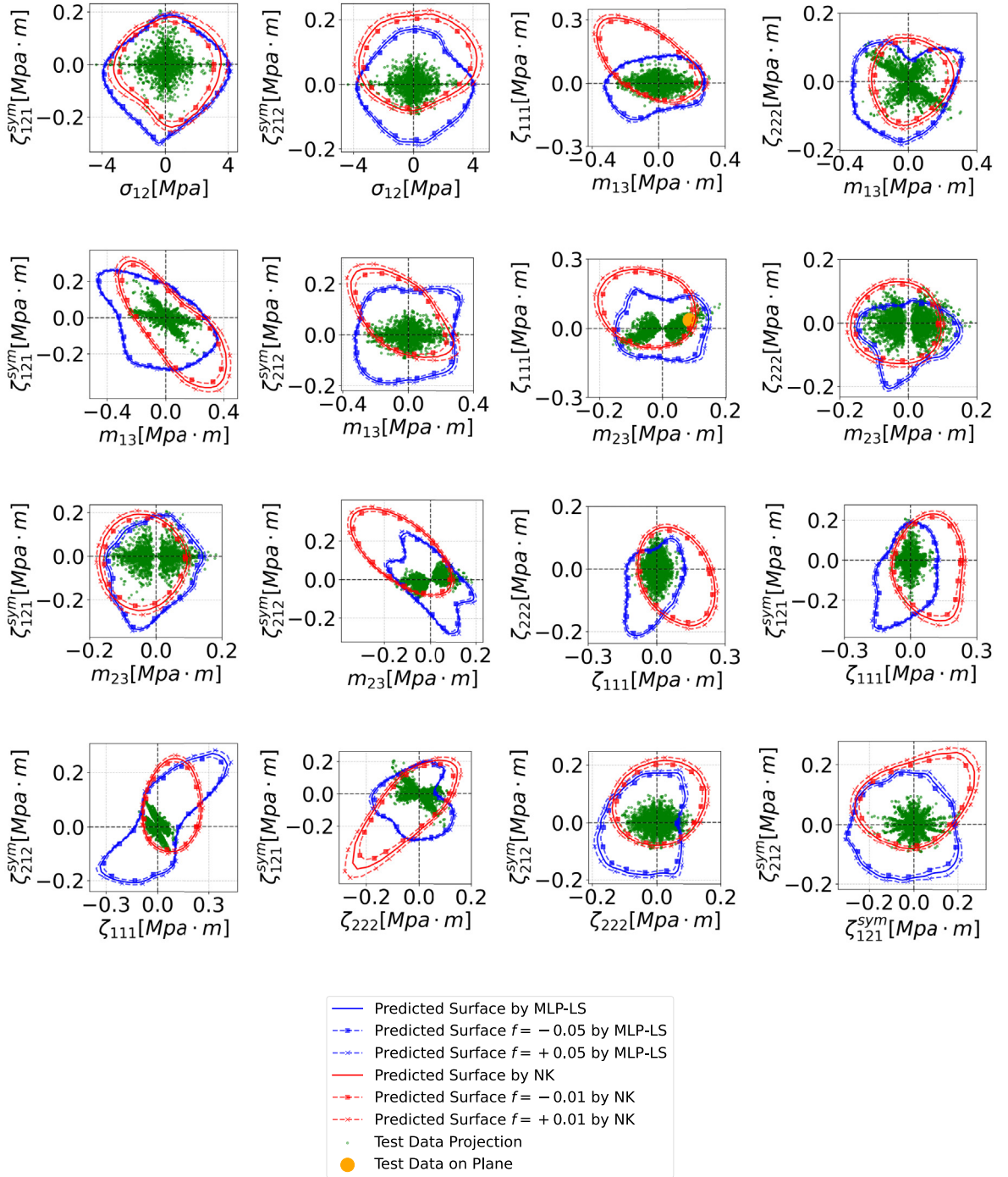


Fig. 14. Cross sections of the micromorphic yield surface reconstructed by MLP-LS and NK. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

experiments or sampling of data via deep reinforcement learning may potentially improve the robustness of the learned yield surface, but is out of the scope of this study. Two return mapping examples are presented, one with the micropolar and the other with the micromorphic constitutive model.

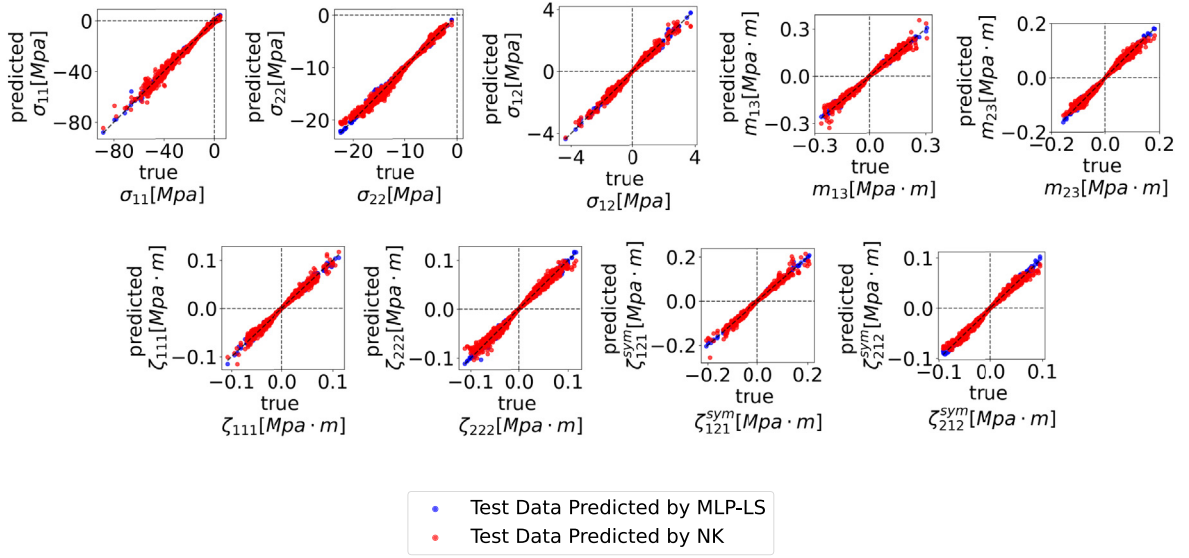


Fig. 15. Micromorphic test data compared component-by-component with predictions by MLP-LS and NK.

As shown in Algorithm 2, the implementation of the NK-enabled return mapping algorithm requires two ingredients: the elastic energy functional and the plastic yield function that evolves with an internal variable. The elastic energy functional with respect to strain and higher-order kinematic modes is first trained by Sobolev training with the loss function of Eq. (24), where the predicted elastic energy and its gradients are both constrained (cf. [27]). To simulate the hardening process upon yielding, the yield function is expressed as a function of both the individual stress components and internal variable Λ is trained via the NK Algorithm 1 where Λ is the magnitude of cumulative plastic strain.

$$L = \begin{cases} \sum_i |W_i - \hat{W}_i|^2 + \sum_i \|\sigma_i - \frac{\partial \hat{W}_i}{\partial \epsilon}\|^2 + \sum_i \|m_i - \frac{\partial \hat{W}_i}{\partial \kappa}\|^2 & \text{in micropolar case} \\ \sum_i |W_i - \hat{W}_i|^2 + \sum_i \|\sigma_i - \frac{\partial \hat{W}_i}{\partial \epsilon}\|^2 + \sum_i \|\xi_i^3 - \frac{\partial \hat{W}_i}{\partial \mathbf{G}}\|^2 & \text{in micromorphic case} \end{cases} \quad (24)$$

The micropolar return mapping example is first presented to introduce a test case with the microstrain being neglected. The deformation rate is controlled such that $\dot{\epsilon}_{11} = \dot{\epsilon}_{22} = -6e - 7$, $\dot{\kappa}_{32} = 1.9e - 4$ for $t < 200$ and the opposite deformation rate is applied for $t > 200$, where t is the pseudo time. The stress history is then simulated from DNS under the elastoplastic loading and elastic unloading with the prescribed boundary condition in Eq. (8); all the stress and couple stress components are homogenized. The stress history is projected onto the stress plane of the combination of each stress component and m_{23} , which is the component conjugate to the main kinematic mode κ_{32} . The yield surfaces with different internal variables are reconstructed by NK and projected to the same stress planes, as shown in Fig. 16. It is observed that as the stress propagates, the internal variable increases, and the yield surface evolves such that the elastic region expands.

In addition to the DNS stress history as the ground truth, another stress history from the return mapping algorithm of NK is also simulated by prescribing the same deformation history. The material is first loaded elastically such that the stress increments follow the elastic energy functional $W(\epsilon, \kappa)$; when the yield is detected, the return mapping algorithm is initiated and the internal variable is incremented to model the hardening process; finally, the elastic unloading is applied, and the stress decreases with the curve showing the permanent plastic deformation. Since some stress components remain zero as shown in Fig. 16, only the nonzero components are presented in Fig. 17, and the consistency in the component-wise comparison validates the micropolar NK return mapping.

One interesting finding observed from Fig. 17 is that the bending kinematic mode can affect the Cauchy stress. Most of the handwritten models are chiral, i.e. higher-order kinematic modes and stresses being independent of the Cauchy stress and strain [51] to avoid modeling the complex coupling effect of the non-chiral material. The geomaterial we are studying is a non-chiral material according to the stress pattern shown in Fig. 18, because under

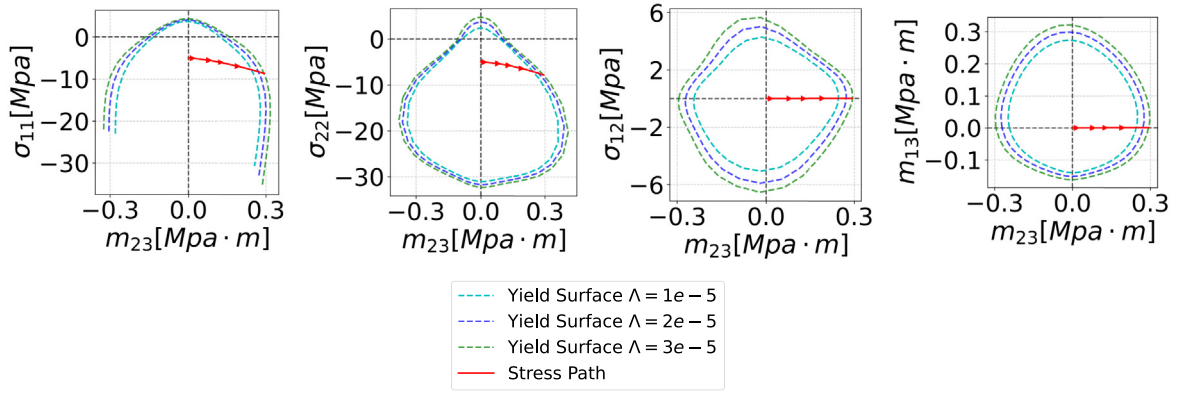


Fig. 16. Micropolar yield surface evolution with stress path projected to the stress planes consisting of different micropolar stress component combined with m_{23} , where m_{23} is the stress components conjugate to the micropolar kinematic deformation κ_{32} .

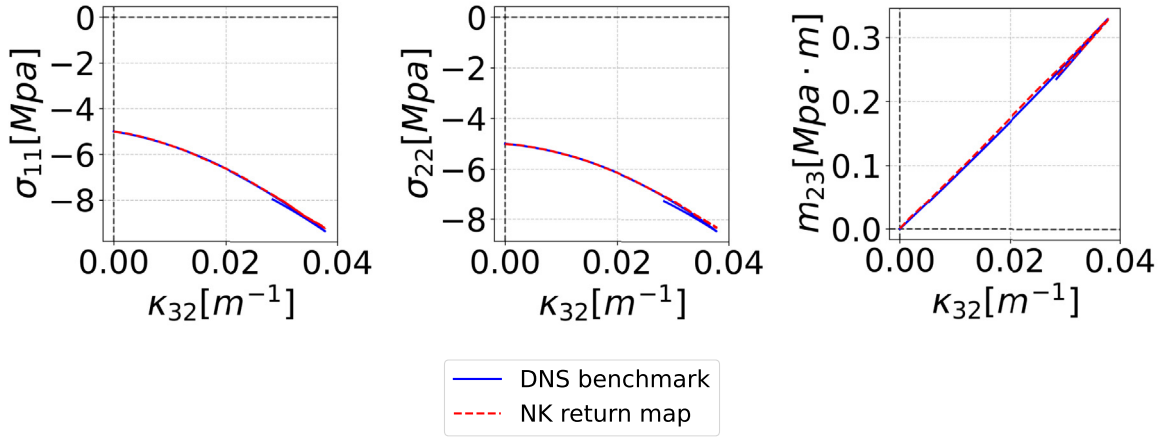


Fig. 17. The history of micropolar stress components is accurately predicted given the history of the kinematic deformation κ_{32} .

the bending mode, the tension part starts to yield such that the stress stops increasing, while the compression part remains elastic with the normal stresses decreasing. As a result, the homogenized σ_{11} and σ_{22} decrease under the higher-order modes, and our NK method is able to capture this non-chiral effect.

In the micromorphic case, the stress history involves more components due to the additional kinematic modes. The deformation modes are controlled by the rate of $\dot{\epsilon}_{11} = \dot{\epsilon}_{22} = -6e - 7$, $\dot{G}_{222} = 3.5e - 4$ for $t < 200$ and the opposite rate is applied for $t > 200$. The stress history is then simulated from DNS under the elastoplastic loading and elastic unloading with the prescribed boundary condition in Eq. (8); all the stress and couple stress components are homogenized. The stress history is projected onto the stress plane of the combination of each stress component and ζ_{222} . The yield surfaces evolving with the internal variable are reconstructed by NK and projected to the same stress planes, as shown in Fig. 19.

Similar to the micropolar case, the stress history simulated by DNS and predicted by NK return mapping under elastoplastic loading and elastic unloading, with more kinematic modes involved. The non-zero components of the

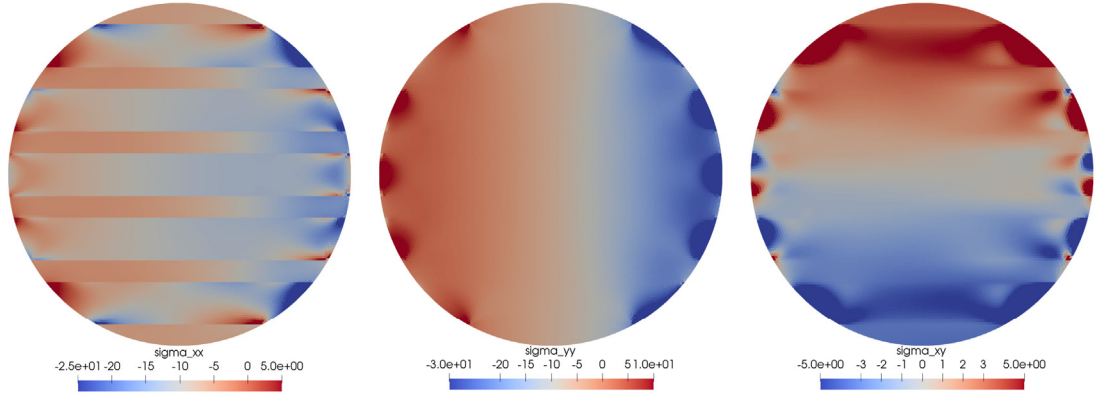


Fig. 18. Patterns of normal stress σ_{11} , σ_{22} , and σ_{12} indicate that the geo-material is nonchiral.

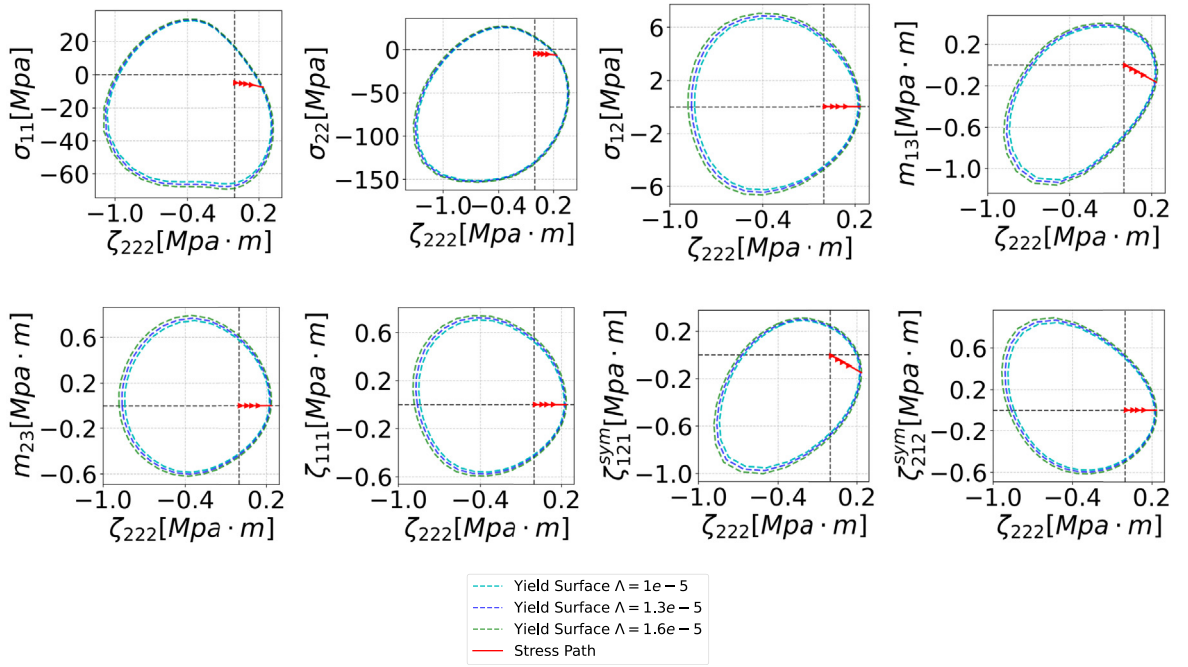


Fig. 19. Micromorphic yield surface evolution with stress path projected to the stress planes consisting of different micromorphic stress components combined with ζ_{222} , where ζ_{222} is the stress component conjugate to the micromorphic kinematic deformation G_{222} .

stress history are compared and found to be consistent as shown in Fig. 20, which validates the micromorphic NK return mapping algorithm and shows the robustness of the NK method in modeling higher-order plasticity problems.

5. Conclusions

This paper presents the NK and MLP-LS method to reconstruct the higher-order yield surface in order to model the path-dependent constitutive law of a material with a complex micro-structure, e.g., a layered geo-material. The DNS constitutive data are first collected by solving local BVPs over the RVE domain, and then used to train the MLP-based elastic energy functional and NK-based narrow band yield function, which reproduce the path-dependent constitutive relation via return mapping algorithm.

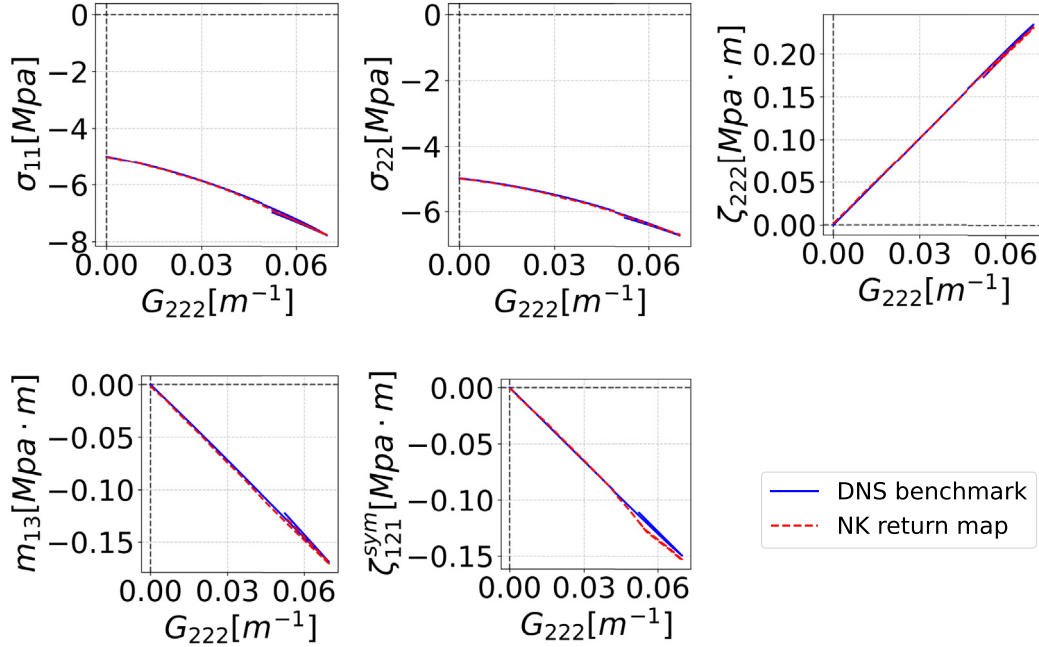


Fig. 20. The history of micromorphic stress components is accurately predicted given the history of the kinematic deformation G_{222} .

Two examples are presented to evaluate the performance of NK compared to MLP-LS. In the first example, the NK and MLP-LS are both able to reproduce a simple analytical plasticity yield model, verified by the yield surface and path-dependent stress curves. However, in the following case study, NK significantly outperforms MLP-LS in the accuracy of the yield surface and the stress curves reproduction, given the poorly distributed data with missing patches. In the second example, the two methods are validated by the micropolar and micromorphic DNS constitutive data of a layered geo-material. It is observed from the micropolar yield surface that NK outperforms MLP-LS in extrapolating unseen data, which is also observed in the first example. We consider the reason is that NK learns the pattern of the higher-dimensional surface via projecting the data onto a finite-dimensional kernel function space and extrapolates the data from the learned pattern. The micropolar return mapping results show that the material is non-chiral (i.e. Cauchy and higher-order stresses are coupled), which is also observed from the distortion of the internal RVE structure; the NK method is able to model the non-chiral effect that is hard to reflect by the handwritten models. The micromorphic results of yield surface reconstruction and return mapping further show that the NK method is generalizable to a much higher-dimensional stress space and decently reproduces the path-dependent constitutive responses given limited and missing 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.

Data availability

Data will be made available on request

Acknowledgments

The authors would like to thank the three anonymous reviewers for the constructive feedback and suggestions. WCS and NNV are primarily supported by the Department of Energy, United States DE-NA0003962. The rest of

the authors are supported by the National Science Foundation, United States under grant contracts CMMI-1846875 and the Dynamic Materials and Interactions Program from the Air Force Office of Scientific Research under grant contracts FA9550-21-1-0391 with the major equipment supported by FA9550-21-1-0027, and the MURI, United States of America Grant No. FA9550-19-1-0318. These supports are gratefully acknowledged. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the sponsors, including the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the sponsors, including the Army Research Laboratory, United States or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Appendix A. Some mathematical background of higher-order continua

This section provides the mathematical derivation of the theory in Section 2. We start with the proof of Hill's Lemma of micromorphic, micropolar, and Cauchy materials, which is used to derive the admissible boundary condition for the higher-order continua. Based on the admissible boundary condition, the stress homogenization scheme satisfying Hill–Mandel's condition is derived.

A.1. Proof of Hill's lemma

The higher-order continuum is studied on an RVE domain Ω with the local coordinate $\mathbf{y}$ as shown in Fig. 1. Hill's Lemma of the micromorphic continuum can be derived as Eq. (25), where the notation $\bar{\cdot} = \frac{1}{V} \int_{\Omega} \cdot dV$.

$$\begin{aligned} \overline{\sigma_{ji}\varepsilon_{ij}} - \bar{\sigma}_{ji}\bar{\varepsilon}_{ij} + \overline{\zeta_{ijk}G_{ijk}} - \bar{\zeta}_{ijk}\bar{G}_{ijk} &= \frac{1}{V} \int_{\partial\Omega} (n_k\sigma_{ki} - n_k\bar{\sigma}_{ki})(u_i - \bar{u}_{i,j}\mathcal{Y}_j - \frac{1}{2}\bar{G}_{ijl}\mathcal{Y}_j\mathcal{Y}_l) dS \\ &\quad + \frac{1}{V} \int_{\partial\Omega} (n_k\zeta_{ijk} - n_k\bar{\zeta}_{ijk})(\chi_{ij} - \bar{G}_{ijl}\mathcal{Y}_l) dS + \bar{\sigma}_{ji}\bar{\chi}_{ij} \end{aligned} \quad (25)$$

The Hill's Lemma shown in Eq. (25) contains two terms of surface integral. To prove the Hill's Lemma, the two terms are expanded separately by the divergence theorem. The first term of Eq. (25), named “part 1”, is the surface integral regarding Cauchy stress. The “part 1” surface integral can be expanded into two volume integrals by the divergence theorem and product rule, with the first volume integral being zero, given $\sigma_{ki,k} = \bar{\sigma}_{ki,k} = 0$ as shown in Eq. (26).

$$\begin{aligned} \text{part 1} &:= \frac{1}{V} \int_{\partial\Omega} (n_k\sigma_{ki} - n_k\bar{\sigma}_{ki})(u_i - \bar{u}_{i,j}\mathcal{Y}_j - \frac{1}{2}\bar{G}_{ijl}\mathcal{Y}_j\mathcal{Y}_l) dS \\ &= \frac{1}{V} \int_{\Omega} (\sigma_{ki,k} - \bar{\sigma}_{ki,k})(u_i - \bar{u}_{i,j}\mathcal{Y}_j - \frac{1}{2}\bar{G}_{ijl}\mathcal{Y}_j\mathcal{Y}_l) dV \\ &\quad + \frac{1}{V} \int_{\Omega} (\sigma_{ki} - \bar{\sigma}_{ki})(u_i - \bar{u}_{i,j}\mathcal{Y}_j - \frac{1}{2}\bar{G}_{ijl}\mathcal{Y}_j\mathcal{Y}_l)_{,k} dV \\ &= \frac{1}{V} \int_{\Omega} (\sigma_{ki} - \bar{\sigma}_{ki})(u_i - \bar{u}_{i,j}\mathcal{Y}_j - \frac{1}{2}\bar{G}_{ijl}\mathcal{Y}_j\mathcal{Y}_l)_{,k} dV \end{aligned} \quad (26)$$

The “part 1” can be further expanded based on Eq. (26), and some terms can be canceled given $\int_{\Omega} (\sigma_{ki} - \bar{\sigma}_{ki}) dV = 0$ and $\bar{G}_{ijk} = \bar{G}_{ikj}$, such that the results can be simplified as shown in Eq. (27).

$$\begin{aligned} \text{part 1} &= \frac{1}{V} \int_{\Omega} (\sigma_{ki} - \bar{\sigma}_{ki})(u_i - \bar{u}_{i,j}\mathcal{Y}_j - \frac{1}{2}\bar{G}_{ijl}\mathcal{Y}_j\mathcal{Y}_l)_{,k} dV \\ &= \frac{1}{V} \int_{\Omega} (\sigma_{ki} - \bar{\sigma}_{ki})u_{i,k} dV - \frac{1}{V} \int_{\Omega} (\sigma_{ki} - \bar{\sigma}_{ki})dV(\bar{u}_{i,j}\delta_{jk}) - \frac{1}{V} \int_{\Omega} (\sigma_{ki} - \bar{\sigma}_{ki})(\frac{1}{2}\bar{G}_{ijl}\mathcal{Y}_j\mathcal{Y}_l)_{,k} dV \\ &= \overline{\sigma_{ki}u_{i,k}} - \bar{\sigma}_{ki}\bar{u}_{i,k} - \frac{1}{2}\bar{G}_{ijl}\frac{1}{V} \int_{\Omega} (\sigma_{ki} - \bar{\sigma}_{ki})(\delta_{kj}\mathcal{Y}_l + \delta_{lj}\mathcal{Y}_k) dV \quad (\text{given } \int_{\Omega} (\sigma_{ki} - \bar{\sigma}_{ki}) dV = 0) \\ &= \overline{\sigma_{ji}u_{i,j}} - \bar{\sigma}_{ji}\bar{u}_{i,j} - \bar{G}_{ijk}\frac{1}{V} \int_{\Omega} \sigma_{ji}\mathcal{Y}_k dV \quad (\text{given } \bar{G}_{ijk} = \bar{G}_{ikj}) \end{aligned} \quad (27)$$

The second term of Eq. (25), named “part 2”, is the surface integral involving the micromorphic generalized stress. The “part 2” surface integral can be expanded by the divergence theorem and simplified similarly to the previous derivation.

$$\begin{aligned}
 \text{part 2} &:= \frac{1}{V} \int_{\partial\Omega} (n_k \zeta_{ijk} - n_k \bar{\zeta}_{ijk})(\chi_{ij} - \bar{G}_{ijl} \mathcal{Y}_l) dS \\
 &= \frac{1}{V} \int_{\partial\Omega} n_k \zeta_{ijk} \chi_{ij} dS + \frac{1}{V} \int_{\partial\Omega} n_k \bar{\zeta}_{ijk} \bar{G}_{ijl} \mathcal{Y}_l dS - \frac{1}{V} \int_{\partial\Omega} n_k \bar{\zeta}_{ijk} \chi_{ij} dS - \frac{1}{V} \int_{\partial\Omega} n_k \zeta_{ijk} \bar{G}_{ijl} \mathcal{Y}_l dS \\
 &= \frac{1}{V} \int_{\Omega} (\zeta_{ijk,k} \chi_{ij} + \zeta_{ijk} G_{ijk}) dV + \bar{\zeta}_{ijk} \bar{G}_{ijk} - \bar{\zeta}_{ijk} \bar{G}_{ijk} - \frac{1}{V} \int_{\Omega} \zeta_{ijk,k} \bar{G}_{ijl} \mathcal{Y}_l dV - \bar{\zeta}_{ijk} \bar{G}_{ijk} \\
 &= \overline{\zeta_{ijk} G_{ijk}} - \bar{\zeta}_{ijk} \bar{G}_{ijk} + \frac{1}{V} \int_{\Omega} \zeta_{ijk,k} (\chi_{ij} - \bar{G}_{ijl} \mathcal{Y}_l) dV
 \end{aligned} \tag{28}$$

Adding the equations Eqs. (27) and (28), the Hill’s lemma Eq. (25) can be proven. Given the balance equation $\sigma_{ji} + \zeta_{ijk,k} = 0$, all additional terms can be canceled other than the final result consisting of 5 terms of volume integrals, as shown in Eq. (29). This concludes the proof of Eq. (25).

$$\begin{aligned}
 &\frac{1}{V} \int_{\partial\Omega} (n_k \sigma_{ki} - n_k \bar{\sigma}_{ki})(u_i - \bar{u}_{i,j} \mathcal{Y}_j - \frac{1}{2} \bar{G}_{ijl} \mathcal{Y}_j \mathcal{Y}_l) dS + \frac{1}{V} \int_{\partial\Omega} (n_k \zeta_{ijk} - n_k \bar{\zeta}_{ijk})(\chi_{ij} - \bar{G}_{ijl} \mathcal{Y}_l) dS \\
 &= \overline{\sigma_{ji} u_{i,j}} - \bar{\sigma}_{ji} \bar{u}_{i,j} - \bar{G}_{ijk} \frac{1}{V} \int_{\Omega} \sigma_{ji} \mathcal{Y}_k dV + \overline{\zeta_{ijk} G_{ijk}} - \bar{\zeta}_{ijk} \bar{G}_{ijk} + \frac{1}{V} \int_{\Omega} \zeta_{ijk,k} (\chi_{ij} - \bar{G}_{ijl} \mathcal{Y}_l) dV \\
 &= \overline{\sigma_{ji} \varepsilon_{ij}} - \bar{\sigma}_{ji} \bar{\varepsilon}_{ij} + \overline{\zeta_{ijk} G_{ijk}} - \bar{\zeta}_{ijk} \bar{G}_{ijk} - \bar{\sigma}_{ji} \bar{\chi}_{ij} + \frac{1}{V} \int_{\Omega} (\sigma_{ji} + \zeta_{ijk,k})(\chi_{ij} - \bar{G}_{ijl} \mathcal{Y}_l) dV \\
 &= \overline{\sigma_{ji} \varepsilon_{ij}} - \bar{\sigma}_{ji} \bar{\varepsilon}_{ij} + \overline{\zeta_{ijk} G_{ijk}} - \bar{\zeta}_{ijk} \bar{G}_{ijk} - \bar{\sigma}_{ji} \bar{\chi}_{ij}
 \end{aligned} \tag{29}$$

A.2. Hill’s lemma for micropolar and Cauchy continua

In the case where the micropolar continuum is considered, the generalized stress is replaced by the couple stress $m_{ji} = -\epsilon_{imn} \zeta_{mnj}$, and the micro-deformation is replaced by the micro-rotation $\theta_i = -\frac{1}{2} \epsilon_{ijk} \chi_{jk}$, where ϵ_{ijk} is the Levi-Civita permutation symbol. As a result, the curvature tensor becomes $\kappa_{ij} = \theta_{i,j} = -\frac{1}{2} \epsilon_{imn} G_{mnj}$, and Hill’s lemma can be reduced to Eq. (30).

$$\begin{aligned}
 \overline{\sigma_{ji} \varepsilon_{ij}} - \bar{\sigma}_{ji} \bar{\varepsilon}_{ij} + \overline{m_{ji} \kappa_{ij}} - \bar{m}_{ji} \bar{\kappa}_{ij} &= \frac{1}{V} \int_{\partial\Omega} (n_k \sigma_{ki} - n_k \bar{\sigma}_{ki})(u_i - \bar{u}_{i,j} \mathcal{Y}_j + \epsilon_{ijk} \bar{\kappa}_{kl} \mathcal{Y}_j \mathcal{Y}_l) dS \\
 &\quad + \frac{1}{V} \int_{\partial\Omega} (n_k m_{ki} - n_k \bar{m}_{ki})(\theta_i - \bar{\kappa}_{ij} \mathcal{Y}_j) dS - \epsilon_{ijk} \bar{\sigma}_{ji} \bar{\theta}_k
 \end{aligned} \tag{30}$$

For the Cauchy continuum, the Hill’s Lemma is further reduced to Eq. (31) by neglecting the micro-deformation and the higher-order generalized stress.

$$\overline{\sigma_{ji} \varepsilon_{ij}} - \bar{\sigma}_{ji} \bar{\varepsilon}_{ij} = \frac{1}{V} \int_{\partial\Omega} (n_k \sigma_{ki} - n_k \bar{\sigma}_{ki})(u_i - \bar{u}_{i,j} \mathcal{Y}_j) dS \tag{31}$$

A.3. Admissible RVE boundary conditions

The admissible boundary condition of the RVE can be derived from the Hill’s Lemma, which should satisfy the Hill–Mandel’s condition in the form of Eq. (32), such that the left-hand side of Eq. (25) vanishes.

$$\overline{\sigma_{ji} \varepsilon_{ij}} - \bar{\sigma}_{ji} \bar{\varepsilon}_{ij} + \overline{\zeta_{ijk} G_{ijk}} - \bar{\zeta}_{ijk} \bar{G}_{ijk} = 0 \tag{32}$$

The Hill–Mandel’s condition is satisfied if the 3 terms of the right-hand side of Eq. (25), including two surface integrals and one volume integral, become zero, as shown in (33).

$$\begin{cases} \frac{1}{V} \int_{\partial\Omega} (n_k \sigma_{ki} - n_k \bar{\sigma}_{ki})(u_i - \bar{u}_{i,j} \mathcal{Y}_j - \frac{1}{2} \bar{G}_{ijl} \mathcal{Y}_j \mathcal{Y}_l) dS = 0 \\ \frac{1}{V} \int_{\partial\Omega} (n_k \zeta_{ijk} - n_k \bar{\zeta}_{ijk})(\chi_{ij} - \bar{G}_{ijl} \mathcal{Y}_l) dS = 0 \\ \bar{\sigma}_{ji} \bar{\chi}_{ij} = 0 \end{cases} \tag{33}$$

One admissible boundary condition that satisfies the Hill–Mandel’s condition can be derived based on the macroscopic strain tensor $\bar{\epsilon}_{ij}$ and the gradient of micro-deformation $\bar{G}_{ijk}$ [51,52]. In the case where the base materials of RVE are Cauchy continuum (i.e. Cauchy to higher-order upscaling), $\chi_{ij} = 0$ internally and $\bar{\chi}_{ij} = 0$, such that $\bar{u}_{i,j} = \bar{\epsilon}_{ij}$, the boundary condition shown in Eq. (34) can be derived, which satisfies Hill–Mandel’s condition by satisfying Eq. (33).

$$\begin{cases} u_i = \bar{\epsilon}_{ij}\mathcal{Y}_j + \frac{1}{2}\bar{G}_{ijk}\mathcal{Y}_j\mathcal{Y}_k & \text{on } \partial\Omega \\ \chi_{ij} = \bar{G}_{ijk}\mathcal{Y}_k & \text{on } \partial\Omega \end{cases} \quad (34)$$

The 3rd-order tensor G_{ijk} is further split into the symmetric part $G_{ijk}^{\text{sym}} = (G_{ijk} + G_{jik})/2$ and the skew-symmetric part $G_{ijk}^{\text{skw}} = (G_{ijk} - G_{jik})/2$, such that the higher order modes can be separated into micropolar bending modes and microstrain modes. The micropolar behavior of the material can be studied by prescribing the bending modes $\kappa_{ij} = \theta_{i,j} = -\frac{1}{2}\epsilon_{imn}G_{mnj} = -\frac{1}{2}\epsilon_{imn}G_{mnj}^{\text{skw}}$. As a results, all the characteristic kinematic modes in 2D case are presented in Fig. 2, such that the prescribed boundary condition is the linear combination of those characteristic modes.

In the special case when the micropolar or Cauchy continuum is considered, the boundary condition is reduced to the form of Eq. (35), where $\bar{G}_{ijk}$ becomes $\epsilon_{ijl}\bar{\kappa}_{lk}$ for micropolar continuum and becomes zero for Cauchy continuum.

$$\begin{cases} u_i = \bar{\epsilon}_{ij}\mathcal{Y}_j - \frac{1}{2}\epsilon_{ijl}\bar{\kappa}_{lk}\mathcal{Y}_j\mathcal{Y}_k, & \theta_i = \bar{\kappa}_{ij}\mathcal{Y}_j & \text{on } \partial\Omega & \text{for micropolar continuum} \\ u_i = \bar{\epsilon}_{ij}\mathcal{Y}_j & & \text{on } \partial\Omega & \text{for Cauchy continuum} \end{cases} \quad (35)$$

A.4. Homogenization based on Hill–Mandel’s condition

The homogenization of stress and generalized stress should still satisfy Hill–Mandel’s condition in the case of Cauchy to micromorphic homogenization. In the case of Cauchy to higher-order upscaling, the $\bar{\zeta}_{ijk}G_{ijk}$ term in Eq. (32) vanishes because ζ_{ijk} vanishes in Cauchy continua and G_{ijk} is redefined as $G_{ijk} = u_{i,jk}$. The Hill–Mandel’s condition can be rewritten in the form of Eq. (36).

$$\overline{\sigma_{ji}\bar{\epsilon}_{ij}} = \bar{\sigma}_{ji}\bar{\epsilon}_{ij} + \bar{\zeta}_{ijk}\bar{G}_{ijk} \quad (36)$$

The left hand side of Eq. (36) can be rewritten into the surface integral terms by divergence theorem as shown in Eq. (37).

$$\overline{\sigma_{ji}\bar{\epsilon}_{ij}} = \frac{1}{V} \int_{\Omega} \sigma_{ji}u_{i,j}dV = \frac{1}{V} \int_{\partial\Omega} n_l\sigma_{li}u_idS \quad (37)$$

Based on the admissible boundary condition shown in Eq. (34), Eq. (37) is rewritten into the form with respect to $\bar{\epsilon}_{ij}$ and $\bar{G}_{ijk}$ as shown in Eq. (38).

$$\overline{\sigma_{ji}\bar{\epsilon}_{ij}} = \frac{1}{V} \int_{\partial\Omega} n_l\sigma_{li}u_idS = \frac{1}{V} \int_{\partial\Omega} n_l\sigma_{li}\mathcal{Y}_j dS \cdot \bar{\epsilon}_{ij} + \frac{1}{V} \int_{\partial\Omega} \frac{1}{2}n_l\sigma_{li}\mathcal{Y}_j\mathcal{Y}_k dS \cdot \bar{G}_{ijk} \quad (38)$$

Subtracting equation Eq. (38) from Eq. (36), the homogenized stress $\bar{\sigma}_{ji}$ and generalized stress $\bar{\zeta}_{ijk}$ can be derived in Eq. (39).

$$\begin{aligned} (\bar{\sigma}_{ji} - \frac{1}{V} \int_{\partial\Omega} n_l\sigma_{li}\mathcal{Y}_j dS) \cdot \bar{\epsilon}_{ij} + (\bar{\zeta}_{ijk} - \frac{1}{V} \int_{\partial\Omega} \frac{1}{2}n_l\sigma_{li}\mathcal{Y}_j\mathcal{Y}_k dS) \cdot \bar{G}_{ijk} &= 0 \\ \Rightarrow \begin{cases} \bar{\sigma}_{ji} = \frac{1}{V} \int_{\partial\Omega} n_l\sigma_{li}\mathcal{Y}_j dS = \frac{1}{V} \int_{\Omega} \sigma_{ji}dV \\ \bar{\zeta}_{ijk} = \frac{1}{V} \int_{\partial\Omega} \frac{1}{2}n_l\sigma_{li}\mathcal{Y}_j\mathcal{Y}_k dS = \frac{1}{V} \int_{\Omega} \frac{1}{2}(\sigma_{ji}\mathcal{Y}_k + \sigma_{ki}\mathcal{Y}_j)dV \end{cases} \end{aligned} \quad (39)$$

For micromorphic continuum, $\bar{\sigma}_{ij}$ is the homogenized Cauchy stress conjugate to $\bar{\epsilon}_{ij}$ and $\bar{\zeta}_{ijk}$ is the generalized stress conjugate to $\bar{G}_{ijk}$ as computed in Eq. (39). For Cauchy continuum, only $\bar{\sigma}_{ij}$ is effective as G_{ijk} is neglected. For micropolar continuum, the coupled stress $\bar{m}_{ij}$ is derived from the skew-symmetric part of $\bar{\zeta}_{ijk}$, such that $\bar{m}_{ij} = -\epsilon_{jmn}\bar{\zeta}_{mni}$ is conjugate to the curvature tensor $\bar{\kappa}_{ij} = -\frac{1}{2}\epsilon_{imn}\bar{G}_{mnj}$. In the special case where only the symmetric part of $\bar{\zeta}_{ijk}$ is considered, $\bar{\zeta}_{ijk}^{\text{sym}} = \frac{1}{2}(\bar{\zeta}_{ijk} + \bar{\zeta}_{jik})$ is homogenized [83] to represent the microstrain generalized stress.

Appendix B. Data generation

B.1. Analytical data generation

The data are generated based on the yield function shown in Eq. (19) with the yield strength $Y = \sqrt{3/2}$ and the length scale $l = 1$. The data are generated by first determining the loading direction $(\hat{\sigma}_{11}, \hat{\sigma}_{22}, \hat{\sigma}_{12}, \hat{m}_{13}/l, \hat{m}_{23}/l)$ in terms of the 4 parameters $(\theta_0, \theta_1, \theta_2, \theta)$; the magnitude of the yield stress is then found given the loading direction as shown in Algorithm 3. As a result, the point cloud data set of size 20 000 is sampled from the yield surface, which is then split into a training set of 15 000 data and a test set of 5000 data.

Algorithm 3 Data sampling from micropolar J2 plasticity model.

Require: Yield function $f(\sigma, m)$

1. Sample the parametric variables with equal spacing (linspace).

Iterate through $\theta_1 = \text{numpy.linspace}(\pi/4, 5\pi/4, 10)$ and

$\theta = \theta_0 = \theta_2 = \text{numpy.linspace}(0, \pi, 10)$.

2. Define the stress and couple stress directions in terms of parametric variables.

Define $(\hat{\sigma}_1, \hat{\sigma}_2, \hat{m}_{13}/l, \hat{m}_{23}/l) = (\cos \theta_0 \cos \theta_1, \cos \theta_0 \sin \theta_1, \sin \theta_0 \cos \theta_2, \sin \theta_0 \sin \theta_2)$.

Compute directional tensors $\hat{\sigma} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \hat{\sigma}_1 & 0 \\ 0 & \hat{\sigma}_2 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$, $\hat{m} = \begin{bmatrix} \hat{m}_{13} \\ \hat{m}_{23} \end{bmatrix}$

3. Determine and record the yield point.

Compute the yield point $(\sigma_Y, m_Y) = \frac{(\hat{\sigma}, \hat{m})}{f(\hat{\sigma}, \hat{m})/(\sqrt{2/3}Y)+1}$

B.2. DNS data generation and sampling

To generate nonpolar, micropolar, and micromorphic yield data, the strain rate and generalized strain rate are controlled to prescribe the displacement boundary condition; the RVE is loaded at the prescribed rate of deformation until yield is detected, when the stress is recorded as a point on the yield surface. The direction is the strain and generalized rates are parametrized $d - 1$ variables, where d is the dimension of the stress space ($d = 3, 5, 9$ in nonpolar, micropolar, and micromorphic cases respectively). The loading direction is first determined by the parametric variables and the strain rate is tuned in magnitude to make sure the RVE yield in 300 strain increments. The details of the data generation are described in the following steps.

After the DNS simulation, 3185 raw data are generated. The data set is then expanded by utilizing the symmetry of the RVE (e.g. adding $(\sigma_{11}, \sigma_{22}, -\sigma_{12})$ if $(\sigma_{11}, \sigma_{22}, \sigma_{12})$ is in the data set in nonpolar case), and then reduced by removing the outliers and redundant data in the data set, there are finally 214 data (split into 200 training data and 14 test data) in nonpolar case, 7048 data (split into 6000 training data and 1048 test data) in micropolar case, and 16 904 data (split into 13 500 training data and 3404 test data) in micromorphic case.

Appendix C. Benchmark against alternative approaches for yield surface reconstruction

For completeness, we briefly introduce three other alternative surface reconstruction approaches, which we will use to benchmark the performance against the proposed neural kernel method. The first approach is the level set plasticity [27], which fits the yield function as a signed distance function $f(\mathbf{x})$ parametrized by deep neural networks (Appendix C.1). The second approach is the classical kernel method with a pre-determined Gauss Process (GP) kernel (Appendix C.2). The last one is the Non-uniform Rational Basis Spline (NURBS) method, which fits the surface with a set of B-spline basis functions (Appendix C.3). The benchmark results are reported in Section (Appendix C.4).

C.1. MLP-based level-set method (MLP-LS)

The level set function, also called signed distance function, is defined as the shortest signed distance of a point to a given surface satisfying Eq. (40), where the sign is positive if the point is outside the surface and negative otherwise. For example, for J2 plasticity, the yield surface is a circle on the π plane, and the yield function is a

Algorithm 4 Data sampling from DNS simulation.**Require:** case $\in$ {"nonpolar", "micropolar", "micromorphic"}

1. Sample the parametric variables with equal spacing (linspace).

if case == "nonpolar" **then**

$$\theta = \text{numpy.linspace}(\frac{\pi}{8}, \frac{\pi}{2}, 6) \text{ and } \theta_1 = \text{numpy.linspace}(\frac{3\pi}{4}, \pi, 10)$$

if case == "micropolar" **then**

$$\theta, \theta_0, \theta_2 = \text{numpy.linspace}(0, \frac{\pi}{2}, 5), \text{ and } \theta_1 = \text{numpy.linspace}(\frac{\pi}{2}, \frac{5\pi}{4}, 10)$$

if case == "micromorphic" **then**

$$\theta, \theta_0, \theta_2, \phi_0, \phi_1, \phi_2, \phi_3 = \text{numpy.linspace}(\frac{\pi}{12}, \frac{5\pi}{12}, 3), \theta_1 = \text{numpy.linspace}(\frac{3\pi}{4}, \pi, 4)$$

2. Define the deformation direction by the directional tensors $\hat{\mathbf{e}}$, $\hat{\mathbf{k}}$, and $\hat{\mathbf{G}}^{sym}$.**if** case == "nonpolar" **then**

$$(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2) = (\cos \theta_1, \sin \theta_1)$$

if case == "micropolar" **then**

$$(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{k}}_{13}, \hat{\mathbf{k}}_{23}) = (\cos \theta_0 \cos \theta_1, \cos \theta_0 \sin \theta_1, \sin \theta_0 \cos \theta_2, \sin \theta_0 \sin \theta_2)$$

if case == "micromorphic" **then**

$$\begin{bmatrix} \hat{\mathbf{e}}_1 & \hat{\mathbf{G}}_{111}^{sym} \\ \hat{\mathbf{e}}_2 & \hat{\mathbf{G}}_{222}^{sym} \\ \hat{\mathbf{k}}_{13} & \hat{\mathbf{G}}_{121}^{sym} \\ \hat{\mathbf{k}}_{23} & \hat{\mathbf{G}}_{212}^{sym} \end{bmatrix} = \begin{bmatrix} \cos \phi_0 \cos \theta_0 \cos \theta_1 & \sin \phi_0 \cos \phi_1 \cos \phi_2 \\ \cos \phi_0 \cos \theta_0 \sin \theta_1 & \sin \phi_0 \cos \phi_1 \sin \phi_2 \\ \cos \phi_0 \sin \theta_0 \cos \theta_2 & \sin \phi_0 \sin \phi_1 \cos \phi_3 \\ \cos \phi_0 \sin \theta_0 \sin \theta_2 & \sin \phi_0 \sin \phi_1 \sin \phi_3 \end{bmatrix}$$

$$\text{compute } \hat{\mathbf{e}} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \hat{\mathbf{e}}_1 & 0 \\ 0 & \hat{\mathbf{e}}_2 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

3. Tune the deformation rate based on $\hat{\mathbf{e}}$, $\hat{\mathbf{k}}$, and $\hat{\mathbf{G}}^{sym}$.**if** case == "nonpolar" **then**

$$(\dot{\mathbf{e}}_{11}, \dot{\mathbf{e}}_{22}, \dot{\mathbf{e}}_{12}) = 10^{-5} * (0.3 * \hat{\mathbf{e}}_{11}, 0.3 * \hat{\mathbf{e}}_{22}, 0.06 * \hat{\mathbf{e}}_{12})$$

if case == "micropolar" **then**

$$(\dot{\mathbf{e}}_{11}, \dot{\mathbf{e}}_{22}, \dot{\mathbf{e}}_{12}, \dot{\mathbf{k}}_{13}, \dot{\mathbf{k}}_{23}) = 10^{-5} * (0.2 * \hat{\mathbf{e}}_{11}, 0.2 * \hat{\mathbf{e}}_{22}, 0.06 * \hat{\mathbf{e}}_{12}, 2 * \hat{\mathbf{k}}_{13}, 2 * \hat{\mathbf{k}}_{23})$$

if case == "micromorphic" **then**

$$\begin{bmatrix} \dot{\mathbf{e}}_{11} & \dot{\mathbf{e}}_{22} & \dot{\mathbf{e}}_{12} \\ \dot{\mathbf{k}}_{13} & \dot{\mathbf{k}}_{23} & \dot{\mathbf{G}}_{111}^{sym} \\ \dot{\mathbf{G}}_{222}^{sym} & \dot{\mathbf{G}}_{121}^{sym} & \dot{\mathbf{G}}_{212}^{sym} \end{bmatrix} = 1.5(10^{-5}) * \begin{bmatrix} 0.2 * \hat{\mathbf{e}}_{11} & 0.2 * \hat{\mathbf{e}}_{22} & 0.06 * \hat{\mathbf{e}}_{12} \\ 2 * \hat{\mathbf{k}}_{13} & 2 * \hat{\mathbf{k}}_{23} & \hat{\mathbf{G}}_{111}^{sym} \\ \hat{\mathbf{G}}_{222}^{sym} & \hat{\mathbf{G}}_{121}^{sym} & \hat{\mathbf{G}}_{212}^{sym} \end{bmatrix}$$

cone, as shown in Fig. 21. The deep neural network with 2nd-order differentiable activation functions is then used to learn the scalar-valued level-set function.

$$\|\nabla_{\sigma''} f\| = 1 \text{ and } f = 0 \text{ on yield surface.} \quad (40)$$

C.2. Classical kernel method based on Gauss Process (GP)

The GP kernel method is exactly the same as the neural kernel method except that the GP kernel is in the form of Eq. (41) which is not trainable. Gauss process can be further used for statistical variance analysis and uncertainty

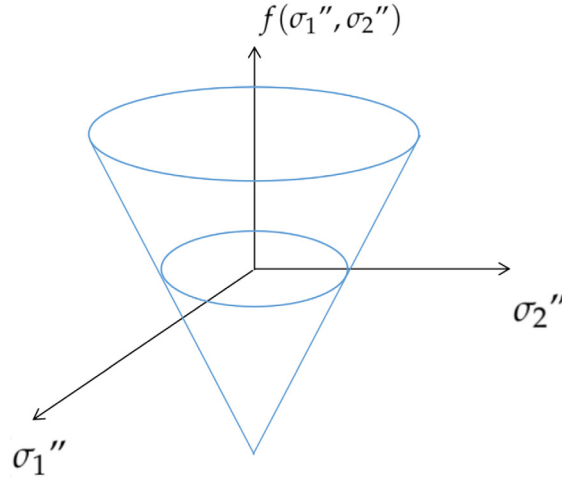


Fig. 21. Level set function of J2 plasticity.

quantification, but in our application, we only use the GP kernel to predict a deterministic yield surface; please refer to Williams and Rasmussen [84] for more mathematical background of GP.

$$Ker(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2L^2}\right) \quad (41)$$

C.3. Non-uniform Rational Basis Spline (NURBS)

The other alternative method is the NURBS-based plasticity [32] that approximates the yield function with basis spline functions, where the isotropic hardening can be modeled by the evolving control points with the internal variable [33], and the non-associative flow rule can be obtained via the gradient of the plastic potential extrapolated by Non-Uniform Rational Basic Splines (NURBS) [34].

Basis spline (B-spline) is one popular method for fitting curves or surfaces [85]. NURBS, as one kind of B-spline method, has been used to reconstruct the yield surface [32], where the NURBS model, with the control points selected in a particular way, fits the 3D ellipsoidal surface accurately. In this section, we generalize the NURBS model to surface fitting in arbitrary dimensional Euclidean spaces. We will introduce the adapted NURBS implementation from the following perspectives: (1) the basis functions, (2) parametrization of higher-dimensional yield surfaces, and (3) surface fitting with NURBS.

C.3.1. B-spline basis function in higher dimension

In general, we adopt the multi-variate B-spline basis functions as compositions of B-spline basis functions with a single variable. To this end, we first introduce the single-variate B-spline basis functions with the independent variable denoted as ϑ , which form a family of polynomials. These functions are defined piecewisely given a knot vector $\mathbf{t}$ that specifies a partition of interval $[0, 1]$: $0 = t_1 \leq t_2 \leq \dots \leq t_i \leq \dots \leq t_q = 1$, where q is the length of the knot vector. The single-variate basis functions $B(\cdot)$ of zero degree are shown to be piecewise constant functions as follows:

$$B_i^{(0)}(\vartheta) = \begin{cases} 1 & \text{if } t_i \leq \vartheta < t_{i+1}, \\ 0 & \text{else} \end{cases}, \quad 1 \leq i \leq q - 1 \quad (42)$$

where the superscript with bracket indicates the degree of polynomials and the subscript i is an index indicator within the basis family. We further express the single-variate basis function of higher degrees k recursively as follows:

$$B_i^{(k)}(\vartheta) = \frac{\vartheta - t_i}{t_{i+k} - t_i} B_i^{(k-1)}(\vartheta) + \frac{t_{i+k+1} - \vartheta}{t_{i+k+1} - t_{i+1}} B_{i+1}^{(k-1)}(\vartheta), \quad 1 \leq i \leq q - k - 1 \quad (43)$$

where we adopt $k = 3$, and the knot vector $\mathbf{t} = [0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1]$ in our implementation, which indicates that the length of the knot vector $q = 11$ and the number of basis function is $q - k - 1 = 7$. We denote the number of basis function members as $|B|$.

The composition of B_i (for simplicity the superscript (k) is omitted) that formulates multi-variate B-spline basis functions is then expressed as follows:

$$\begin{aligned} N_i(\vartheta) &= B_i(\vartheta) && \text{for single variable,} \\ N_K(\vartheta) &= \prod_{l=1}^d B_{i_l}(\vartheta_l), \text{ where } K = \sum_{l=1}^d i_l |B|^{d-l} && \text{for multiple variables.} \end{aligned} \quad (44)$$

C.3.2. Parametrization of higher-dimensional surface

The yield surface living in d -dimensional space describes a $d - 1$ -dimensional manifold and hence could be parametrized with $d - 1$ independent variables $\vartheta = (\vartheta_1, \vartheta_2, \dots, \vartheta_{d-1})$. We adopt the generalized spherical parametrization for describing the yield surface mathematically in Eq. (45), where the independent variables are angle indicators within the range $[0, 1]$ and there is a dependent variable $\rho(\vartheta)$ indicating the radial distance.

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \rho(\vartheta) \cos 2\pi \vartheta_1 \\ \rho(\vartheta) \sin 2\pi \vartheta_1 \end{bmatrix} && \text{for 1D curve} \\ \mathbf{x} &= \begin{bmatrix} \rho(\vartheta) \cos \pi \vartheta_1 \\ \rho(\vartheta) \sin \pi \vartheta_1 \cos 2\pi \vartheta_2 \\ \rho(\vartheta) \sin \pi \vartheta_1 \sin 2\pi \vartheta_2 \end{bmatrix} && \text{for 2D surface} \\ x_i &= \begin{cases} \rho(\vartheta)(\prod_{j=1}^{i-1} \sin \pi \vartheta_j) \cos \pi \vartheta_i & \text{if } i < d - 2 \\ \rho(\vartheta)(\prod_{j=1}^{d-2} \sin \pi \vartheta_j) \cos 2\pi \vartheta_i & \text{if } i = d - 2 \\ \rho(\vartheta)(\prod_{j=1}^{d-2} \sin \pi \vartheta_j) \sin 2\pi \vartheta_i & \text{if } i = d - 1 \end{cases} && \text{for higher dimensions} \end{aligned} \quad (45)$$

C.3.3. Fitting yield surfaces with NURBS

The point cloud data $\mathbf{x}^{(i)}$ sampled from the yield surface can be mapped into the labeled data $(\vartheta^{(i)}, \rho^{(i)})$ through the parametrization shown in Eq. (45). The NURBS model is described by Eq. (46), where $\mathbf{c} \in \mathbb{R}^{|B|^{d-1}}$ is an array of control points.

$$\rho(\vartheta) = \sum_{j=1}^{|B|^{d-1}} N_j(\vartheta) c_j = \mathbf{N}(\vartheta) \cdot \mathbf{c} \quad (46)$$

To fit the NURBS model with the data sampled from the yield surface, the control points $\mathbf{c}$ are found such that the difference between the prediction $\hat{\rho}^{(i)} = \mathbf{N}(\vartheta^{(i)}) \cdot \mathbf{c}$ and the label $\rho^{(i)}$ is minimized. Therefore, the control points are computed through the regularized least square method as shown in Eq. (47), where $[\rho]_i = \rho^{(i)}$ and $[\mathbf{N}(\vartheta)]_{ij} = N_j(\vartheta^{(i)})$.

$$\mathbf{c} = \underset{\mathbf{c}}{\operatorname{argmin}} \sum_i \|\rho^{(i)} - \mathbf{N}(\vartheta^{(i)}) \cdot \mathbf{c}\|^2 + \lambda \|\mathbf{c}\|^2 \iff \mathbf{c} = (\mathbf{N}^T(\vartheta) \mathbf{N}(\vartheta) + \lambda \mathbf{I})^{-1} \mathbf{N}^T(\vartheta) \rho. \quad (47)$$

C.4. Yield surface reconstructed via alternative approaches

To compare the performances of the approaches outlined in [Appendices C.1–C.3](#), we obtained the learned yield functions via these approaches for three data sets we used in this paper, i.e., the classical J2 von Mises yield function, a generalized J2 von Mises yield function with the additional couple stress terms, and the micropolar DNS data inferred from finite element simulations of the layered materials. The results are shown in [Fig. 22](#).

In the first case, the actual learned function only depends on the J2 stress and the data set is of low dimension. As such, the results in [Fig. 22\(a\)](#) indicate that all four approaches (neural kernel method (red), classical Gaussian kernel (black), level set MLP (green), and NURBS (yellow)) perform well in this simple case. In the second case, the Von Mises yield function is amended with a regularization term (see Eq. (19)). This dependence on coupled stress also breaks the symmetry of the force stress, and hence leads to the dimension of the parametric space to be increased from 3 to 5. This increase of dimensionality leads to the NURBS yield function failing, but the rest

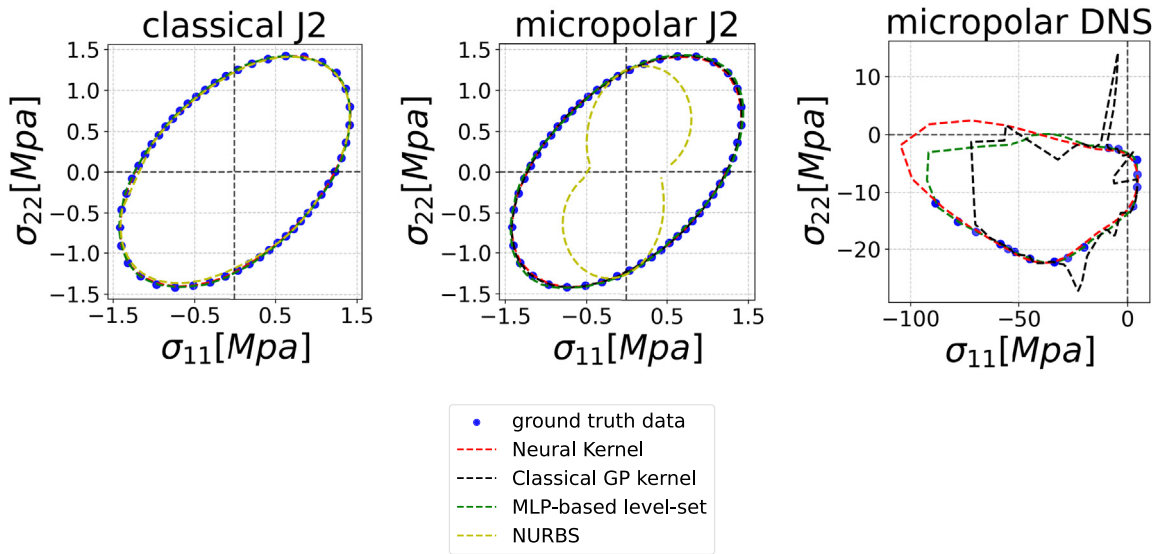


Fig. 22. The cross section of the classical J2 yield surface $f(\sigma_{11}, \sigma_{22}, \sigma_{12} = 0) = 0$ in 3D space (LEFT), the micropolar J2 yield surface $f(\sigma_{11}, \sigma_{22}, \sigma_{12} = 0, m_{13} = 0, m_{23} = 0) = 0$ in 5D space (MIDDLE), and the micropolar DNS yield surface (RIGHT). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the three approaches still perform well (see Fig. fig:nurbs(b)). In the last case, the DNS data set is used where only a portion of data is purposely missing (see Fig. 22(c)), the dimensionality of the data is identical to the second case, but the data exhibits lower symmetry. Visual inspections reveal that the neural kernel generates yield function with geometric features consistent with those of data. On the other hand, the MLP level set is capable to generate non-oscillatory yield functions, whereas both the NURBS and classical GP kernel may lead to spurious oscillation in the learned function.

Note that, in the last case where data are missing in one portion of the parametric space, the robustness of the learned function in the extrapolated regime depends strongly on the types of inductive biases of the learning algorithm [86]. Since the neural kernel method employs a data-dependent adaptive kernel, it is more expressive than the classical GP and the NURBS approach where the parametric space is spanned by a pre-determined set of bases.

References

- [1] Yannis F. Dafalias, Modelling cyclic plasticity: simplicity versus sophistication, *Mech. Eng. Mater.* 153178 (1984).
- [2] Kun Wang, WaiChing Sun, Simon Salager, SeonHong Na, Ghonwa Khaddour, Identifying material parameters for a micro-polar plasticity model via X-ray micro-computed tomographic (CT) images: lessons learned from the curve-fitting exercises, *Int. J. Multiscale Comput. Eng.* 14 (4) (2016).
- [3] Isaac Newton, *Philosophiæ naturalis principia mathematica* (mathematical principles of natural philosophy), Lond. (1687) (1687).
- [4] Willard van Orman Quine, On simple theories of a complex world, *Synthese* (1963) 103–106.
- [5] Don A. Howard, Marco Giovanelli, et al., Einstein's philosophy of science, *Stanford Ency. Philos.* (2019) 1–20.
- [6] Immanuel Kant, Critique of pure reason. 1781, *Mod. Class. Philos. Camb. MA: Houghton Mifflin* (1908) 370–456.
- [7] Ivo Herle, Gerd Gudehus, Determination of parameters of a hypoplastic constitutive model from properties of grain assemblies, *Mech. Cohesive-Frict. Mater.: Int. J. Exp. Modell. Comput. Mater. Struct.* 4 (5) (1999) 461–486.
- [8] Gerd Gudehus, Angelo Amorosi, Antonio Gens, Ivo Herle, Dimitrios Kolymbas, David Mašín, David Muir Wood, Andrzej Niemunis, Roberto Nova, Manuel Pastor, et al., The soilmodels. info project, *Int. J. Numer. Anal. Methods Geomech.* 32 (12) (2008) 1571–1572.
- [9] Xiao Sun, Bahador Bahmani, Nikolaos N Vlassis, WaiChing Sun, Yanxun Xu, Data-driven discovery of interpretable causal relations for deep learning material laws with uncertainty propagation, *Granul. Matter* 24 (2022) 1–32.
- [10] Yannis F. Dafalias, Majid T. Manzari, Simple plasticity sand model accounting for fabric change effects, *J. Eng. Mech.* 130 (6) (2004) 622–634.
- [11] Jidong Zhao, Daichao Sheng, M. Rouainia, Scott W. Sloan, Explicit stress integration of complex soil models, *Int. J. Numer. Anal. Methods Geomech.* 29 (12) (2005) 1209–1229.

- [12] Yannis F Dafalias, Majid T Manzari, Achilleas G Papadimitriou, SANICLAY: simple anisotropic clay plasticity model, *Int. J. Numer. Anal. Methods Geomech.* 30 (12) (2006) 1231–1257.
- [13] René de Borst, A generalisation of J2-flow theory for polar continua, *Comput. Methods Appl. Mech. Engrg.* 103 (3) (1993) 347–362.
- [14] S.J. Wheeler, D. Gallipoli, M. Karstunen, Comments on use of the Barcelona basic model for unsaturated soils, *Int. J. Numer. Anal. Methods Geomech.* 26 (15) (2002) 1561–1571.
- [15] SeonHong Na, WaiChing Sun, Computational thermo-hydro-mechanics for multiphase freezing and thawing porous media in the finite deformation range, *Comput. Methods Appl. Mech. Engrg.* 318 (2017) 667–700.
- [16] Qing Yin, Edward Andò, Gioacchino Viggiani, WaiChing Sun, Freezing-induced stiffness and strength anisotropy in freezing clayey soil: Theory, numerical modeling, and experimental validation, *Int. J. Numer. Anal. Methods Geomech.* 46 (11) (2022) 2087–2114.
- [17] Mahyar Malekzade Kebria, SeonHong Na, Fan Yu, An algorithmic framework for computational estimation of soil freezing characteristic curves, *Int. J. Numer. Anal. Methods Geomech.* 46 (8) (2022) 1544–1565.
- [18] Ran Ma, WaiChing Sun, Computational thermomechanics for crystalline rock. Part II: Chemo-damage-plasticity and healing in strongly anisotropic polycrystals, *Comput. Methods Appl. Mech. Engrg.* 369 (2020) 113184.
- [19] Majid T. Manzari, Application of micropolar plasticity to post failure analysis in geomechanics, *Int. J. Numer. Anal. Methods Geomech.* 28 (10) (2004) 1011–1032.
- [20] Jia Lin, Wei Wu, Ronaldo I. Borja, Micropolar hypoplasticity for persistent shear band in heterogeneous granular materials, *Comput. Methods Appl. Mech. Engrg.* 289 (2015) 24–43.
- [21] Andreas Dietsche, Paul Steinmann, Kaspar Willam, Micropolar elastoplasticity and its role in localization, *Int. J. Plast.* 9 (7) (1993) 813–831.
- [22] David L. Donoho, et al., High-dimensional data analysis: The curses and blessings of dimensionality, *AMS Math Chall. Lect.* 1 (2000) 32.
- [23] Maysam B Gorji, Mojtaba Mozaffar, Julian N Heidenreich, Jian Cao, Dirk Mohr, On the potential of recurrent neural networks for modeling path dependent plasticity, *J. Mech. Phys. Solids* 143 (2020) 103972.
- [24] Colin Bonatti, Bekim Berisha, Dirk Mohr, From CP-FFT to CP-RNN: Recurrent neural network surrogate model of crystal plasticity, *Int. J. Plast.* 158 (2022) 103430.
- [25] Yousef Heider, Kun Wang, WaiChing Sun, SO (3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials, *Comput. Methods Appl. Mech. Engrg.* 363 (2020) 112875.
- [26] Kun Wang, WaiChing Sun, A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning, *Comput. Methods Appl. Mech. Engrg.* 334 (2018) 337–380.
- [27] Nikolaos N. Vlassis, WaiChing Sun, Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening, *Comput. Methods Appl. Mech. Engrg.* 377 (2021) 113695.
- [28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need, *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [29] Robert Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Stat. Soc. Ser. B Stat. Methodol.* 58 (1) (1996) 267–288.
- [30] Xiaolong He, Jiun-Shyan Chen, Thermodynamically consistent machine-learned internal state variable approach for data-driven modeling of path-dependent materials, *Comput. Methods Appl. Mech. Engrg.* 402 (2022) 115348.
- [31] David Sussillo, Omri Barak, Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks, *Neural Comput.* 25 (3) (2013) 626–649.
- [32] William M. Coombs, Oscar A. Petit, Yousef Ghaffari Motlagh, NURBS plasticity: Yield surface representation and implicit stress integration for isotropic inelasticity, *Comput. Methods Appl. Mech. Engrg.* 304 (2016) 342–358.
- [33] William M. Coombs, Yousef Ghaffari Motlagh, NURBS plasticity: yield surface evolution and implicit stress integration for isotropic hardening, *Comput. Methods Appl. Mech. Engrg.* 324 (2017) 204–220.
- [34] William M. Coombs, Yousef Ghaffari Motlagh, NURBS plasticity: non-associated plastic flow, *Comput. Methods Appl. Mech. Engrg.* 336 (2018) 419–443.
- [35] Mian Xiao, WaiChing Sun, Geometric prior of multi-resolution yielding manifolds and the local closest point projection for nearly non-smooth plasticity, *Comput. Methods Appl. Mech. Engrg.* 400 (2022) 115469.
- [36] Nikolaos N. Vlassis, WaiChing Sun, Component-based machine learning paradigm for discovering rate-dependent and pressure-sensitive level-set plasticity models, *J. Appl. Mech.* 89 (2) (2022).
- [37] Jan Niklas Fuhg, Craig M Hamel, Kyle Johnson, Reese Jones, Nikolaos Bouklas, Modular machine learning-based elastoplasticity: Generalization in the context of limited data, *Comput. Methods Appl. Mech. Engrg.* 407 (2023) 115930.
- [38] Francis Williams, Zan Gojic, Sameh Khamis, Denis Zorin, Joan Bruna, Sanja Fidler, Or Litany, Neural fields as learnable kernels for 3d reconstruction, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18500–18510.
- [39] Eugène Maurice Pierre Cosserat, François Cosserat, Théorie Des Corps DÉFormables, A. Hermann et fils, 1909.
- [40] RA144512 Toupin, Elastic materials with couple-stresses, *Arch. Ration. Mech. Anal.* 11 (1) (1962) 385–414.
- [41] Richard A. Toupin, Theories of elasticity with couple-stress, *Arch. Ration. Mech. Anal.* 17 (2) (1964) 85–112.
- [42] R.D. Mindlin, HF144513 Tiersten, et al., Effects of couple-stresses in linear elasticity, *Arch. Ration. Mech. Anal.* 11 (1) (1962) 415–448.
- [43] Albert Edward Green, Ronald S. Rivlin, Multipolar continuum mechanics, *Arch. Ration. Mech. Anal.* 17 (1964) 113–147.
- [44] Raymond David Mindlin, Microstructure in linear elasticity (tech. rep.), Columbia Univ. New York Dept. Civ. Eng. Eng. Mech. (1963).
- [45] Raymond David Mindlin, Second gradient of strain and surface-tension in linear elasticity, *Int. J. Solids Struct.* 1 (4) (1965) 417–438.
- [46] Raymond David Mindlin, NN0166 Eshel, On first strain-gradient theories in linear elasticity, *Int. J. Solids Struct.* 4 (1) (1968) 109–124.

- [47] A. Cemal Eringen, Mechanics of micromorphic continua, in: *Mechanics of Generalized Continua: Proceedings of the IUTAM-Symposium on the Generalized Cosserat Continuum and the Continuum Theory of Dislocations with Applications*, Freudenstadt and Stuttgart (Germany) 1967, Springer, 1968, pp. 18–35.
- [48] Hans-Bernd Mühlhaus, Ioannis Vardoulakis, The thickness of shear bands in granular materials, *Geotechnique* 37 (3) (1987) 271–283.
- [49] Ron HJ Peerlings, Marc GD Geers, René de Borst, WAM1032 Brekelmans, A critical comparison of nonlocal and gradient-enhanced softening continua, *Int. J. Solids Struct.* 38 (44–45) (2001) 7723–7746.
- [50] Paul Steinmann, A micropolar theory of finite deformation and finite rotation multiplicative elastoplasticity, *Int. J. Solids Struct.* 31 (8) (1994) 1063–1084.
- [51] Ragnar Larsson, Stefan Diebels, A second-order homogenization procedure for multi-scale analysis based on micropolar kinematics, *Internat. J. Numer. Methods Engrg.* 69 (12) (2007) 2485–2512.
- [52] R. Jänicke, S. Diebels, Numerical homogenisation of micromorphic media, *Tech. Mech.-Eur. J. Eng. Mech.* 30 (4) (2010) 364–373.
- [53] Patrizio Neff, Ionel-Dumitrel Ghiba, Angela Madeo, Luca Placidi, Giuseppe Rosi, A unifying perspective: the relaxed linear micromorphic continuum, *Contin. Mech. Thermodyn.* 26 (2014) 639–681.
- [54] Zdenek P. Bazant, Milan Jirásek, Nonlocal integral formulations of plasticity and damage: survey of progress, *J. Eng. Mech.* 128 (11) (2002) 1119–1149.
- [55] Jörg Schröder, Mohammad Sarhil, Lisa Scheunemann, Patrizio Neff, Lagrange and h (curl, b) based finite element formulations for the relaxed micromorphic model, *Comput. Mech.* (2022) 1–25.
- [56] Ronaldo I. Borja, *Plasticity*, vol. 2, Springer, 2013.
- [57] Oliver Weeger, Numerical homogenization of second gradient, linear elastic constitutive models for cubic 3D beam-lattice metamaterials, *Int. J. Solids Struct.* 224 (2021) 111037.
- [58] Paul Rosenthal, Vladimir Molchanov, Lars Linsen, A Narrow Band Level Set Method for Surface Extraction from Unstructured Point-Based Volume Data, *Universitätsbibliothek Chemnitz*, 2011.
- [59] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, Andreas Geiger, Convolutional occupancy networks, in: *European Conference on Computer Vision*, Springer, 2020, pp. 523–540.
- [60] Alexander Rakhlin, Xiyu Zhai, Consistency of interpolation with Laplace kernels is a high-dimensional phenomenon, in: *Conference on Learning Theory*, PMLR, 2019, pp. 2595–2623.
- [61] Youngmin Cho, Lawrence Saul, Kernel methods for deep learning, *Adv. Neural Inf. Process. Syst.* 22 (2009).
- [62] Nikolaos N. Vlassis, Ran Ma, WaiChing Sun, Geometric deep learning for computational mechanics part i: Anisotropic hyperelasticity, *Comput. Methods Appl. Mech. Engrg.* 371 (2020) 113299.
- [63] Feng-Bao Lin, Zdeněk P. Bažant, Convexity of smooth yield surface of frictional material, *J. Eng. Mech.* 112 (11) (1986) 1259–1262.
- [64] JH Argyris, G Faust, J Szimmat, EP Warnke, KJ Willam, Recent developments in the finite element analysis of prestressed concrete reactor vessels, *Nucl. Eng. Des.* 28 (1) (1974) 42–75.
- [65] Daichao Sheng, Charles E. Augarde, Andrew J. Abbo, A fast algorithm for finding the first intersection with a non-convex yield surface, *Comput. Geotech.* 38 (4) (2011) 465–471.
- [66] Dorival M. Pedroso, Daichao Sheng, Scott W. Sloan, Stress update algorithm for elastoplastic models with nonconvex yield surfaces, *Internat. J. Numer. Methods Engrg.* 76 (13) (2008) 2029–2062.
- [67] Rainer Glüge, Sara Bucci, Does convexity of yield surfaces in plasticity have a physical significance? *Math. Mech. Solids* 23 (9) (2018) 1364–1373.
- [68] Paul Steinmann, Theory and numerics of ductile micropolar elastoplastic damage, *Internat. J. Numer. Methods Engrg.* 38 (4) (1995) 583–606.
- [69] Ioannis Vardoulakis, *Cosserat continuum mechanics*, *Lect. Notes Appl. Comput. Mech.* 87 (2019).
- [70] Clayton Deutsch, Calculating effective absolute permeability in sandstone/shale sequences, *SPE Format. Eval.* 4 (03) (1989) 343–348.
- [71] Hiroki Sone, Mark D. Zoback, Mechanical properties of shale-gas reservoir rocks—Part 2: Ductile creep, brittle strength, and their relation to the elastic modulus, *Geophysics* 78 (5) (2013) D393–D402.
- [72] SeonHong Na, WaiChing Sun, Mathew D Ingraham, Hongkyu Yoon, Effects of spatial heterogeneity and material anisotropy on the fracture pattern and macroscopic effective toughness of Mancos shale in Brazilian tests, *J. Geophys. Res.: Solid Earth* 122 (8) (2017) 6202–6230.
- [73] Shabnam J. Semnani, Joshua A. White, Ronaldo I. Borja, Thermoplasticity and strain localization in transversely isotropic materials based on anisotropic critical state plasticity, *Int. J. Numer. Anal. Methods Geomech.* 40 (18) (2016) 2423–2449.
- [74] Yang Zhao, Shabnam J. Semnani, Qing Yin, Ronaldo I. Borja, On the strength of transversely isotropic rocks, *Int. J. Numer. Anal. Methods Geomech.* 42 (16) (2018) 1917–1934.
- [75] Ronaldo I. Borja, Qing Yin, Yang Zhao, Cam-clay plasticity. Part IX: On the anisotropy, heterogeneity, and viscoplasticity of shale, *Comput. Methods Appl. Mech. Engrg.* 360 (2020) 112695.
- [76] Daniel Arndt, Wolfgang Bangerth, Bruno Blais, Thomas C Clevenger, Marc Fehling, Alexander V Grayver, Timo Heister, Luca Heltai, Martin Kronbichler, Matthias Maier, et al., The deal. II library, version 9.2, *J. Numer. Math.* 28 (3) (2020) 131–146.
- [77] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, Jascha Sohl-Dickstein, On the expressive power of deep neural networks, in: *International Conference on Machine Learning*, PMLR, 2017, pp. 2847–2854.
- [78] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar, *Foundations of Machine Learning*, MIT Press, 2018.
- [79] Marc G.D. Geers, Varvara G. Kouznetsova, WAM Brekelmans, Multi-scale computational homogenization: Trends and challenges, *J. Comput. Appl. Math.* 234 (7) (2010) 2175–2182.
- [80] Gerafl Hütter, On the micro-macro relation for the microdeformation in the homogenization towards micromorphic and micropolar continua, *J. Mech. Phys. Solids* 127 (2019) 62–79.

- [81] Kun Wang, WaiChing Sun, Qiang Du, A non-cooperative meta-modeling game for automated third-party calibrating, validating and falsifying constitutive laws with parallelized adversarial attacks, *Comput. Methods Appl. Mech. Engrg.* 373 (2021) 113514.
- [82] Ruben Villarreal, Nikolaos N Vlassis, Nhon N Phan, Tommie A Catanach, Reese E Jones, Nathaniel A Trask, Charlotte LB Kramer, WaiChing Sun, Design of experiments for the calibration of history-dependent models via deep reinforcement learning and an enhanced kalman filter, *Comput. Mech.* 72 (1) (2023) 95–124.
- [83] Raja Biswas, Leong Hien Poh, A micromorphic computational homogenization framework for heterogeneous materials, *J. Mech. Phys. Solids* 102 (2017) 187–208.
- [84] Christopher K.I. Williams, Carl Edward Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2, MIT press Cambridge, MA, 2006, (3).
- [85] Gary D. Knott, *Interpolating Cubic Splines*, vol. 18, Springer Science & Business Media, 2000.
- [86] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, Eric P Xing, Deep kernel learning, in: *Artificial Intelligence and Statistics*, PMLR, 2016, pp. 370–378.