CMES, vol.117, no.3, pp.309-342, 2018

Machine Learning Models of Plastic Flow Based on Representation Theory

R. E. Jones[1, *], J. A. Templeton[1], C. M. Sanders[1] and J. T. Ostien[1]

Abstract: We use machine learning (ML) to infer stress and plastic flow rules using data from representative polycrystalline simulations. In particular, we use so-called *deep* (multilayer) neural networks (NN) to represent the two response functions. The ML process does not choose appropriate inputs or outputs, rather it is trained on selected inputs and output. Likewise, its discrimination of features is crucially connected to the chosen input-output map. Hence, we draw upon classical constitutive modeling to select inputs and enforce well-accepted symmetries and other properties. In the context of the results of numerous simulations, we discuss the design, stability and accuracy of constitutive NNs trained on typical experimental data. With these developments, we enable rapid model building in real-time with experiments, and guide data collection and feature discovery.

Keywords: Machine learning, constitutive modeling, plasticity, invariance.

1 Introduction

Our effort to produce viable models of plasticity from trusted data draws upon traditional constitutive modeling theory and newly developed machine learning (ML) techniques.

The theory of constitutive function representation has a long history, going back to the beginnings of the Rational Mechanics movement. Much of the pioneering work was done by Rivlin, Pipkin, Smith, Spencer, Boehler, and co-workers [Spencer and Rivlin (1958a,b, 1962); Pipkin and Wineman (1963); Wineman and Pipkin (1964); Smith and Rivlin (1964); Smith (1965); Rivlin and Smith (1969); Spencer (1971, 1987); Boehler (1987)]. Later, Zheng contributed a notable monograph on the application of representation theory to anisotropy [Zheng (1994)]. Much of these results have been condensed in: Spencer's monograph [Spencer (1971)], Truesdell and Noll's treatise [Truesdell and Noll (2004)], Gurtin's text [Gurtin (1982)], and the recent book by Itskov [Itskov (2007)].

The application of machine learning (ML) to engineering dates back to at least the 1980's and covers a wide variety of problems. For instance, Adeli et al. [Adeli and Yeh (1989)] applied ML to the design of steel beams; Hajela et al. [Hajela and Berke (1991)] used a ML model as a surrogate for the exact response of structures to enable fast optimization; Cheu et al. [Cheu and Ritchie (1995)] applied ML to traffic m odeling; and Theocaris et al. [Theocaris and Panagiotopoulos (1993)] used it to model fracture behavior and identification. For further bibliography along these lines, see a review of neural network applications in civil engineering that appeared in 2001 [Adeli (2001)].

[1] Sandia National Laboratories, P.O. Box 969, Livermore, CA 94551, USA.
[*] Corresponding Author: R. E. Jones. Email: rjones@sandia.gov.

CMES. doi:10.31614/cmes.2018.04285

Research on applying ML to constitutive modeling dates back to roughly the same time period. In solid mechanics in particular, Ghaboussi et al. [Ghaboussi, Pecknold, Zhang et al. (1998)] applied a neural network (NN) to data from experiments of beam deflection. They created a model which acquired increasing fidelity as experiment progressed via hierarchical learning and adapting new hidden layers. Furukawa et al. [Furukawa and Yagawa (1998)] constructed an "implicit" model of linear viscoplasticity with a NN based on a state space formulation, where the NN provided the driving term for plastic evolution and the elastic response was assumed to be known. Notably, they expressed a need for variety in the training data.

More recently, a number of studies have appeared comparing NN plasticity models to other models calibrated on experimental data for specific materials. Lin et al. [Lin, Zhang and Zhong (2008)] built a NN model of the flow stress of low alloy steel based on only experimentally observable quantities. Bobbili et al. [Bobbili, Ramakrishna, Madhu et al. (2015)] constructed a NN model of high strain rate Hopkinson bar tests of 7017 aluminium alloy and compared it to a Johnson-Cook model. For T24 steel, Li et al. [Li, Wang, Wei et al. (2012)] compared a NN model to a modified Zerilli-Armstrong and strain-compensated Arrhenius-type model. They remarked on the opacity of the NN model and the need for extensive data. Desu et al. [Desu, Guntuku, Aditya et al. (2014)] made flow stress prediction of austenitic 304 stainless steel with support vector machine construct and compared it to a NN model. Asgharzadeh et al. [Asgharzadeh, Aval and Serajzadeh (2016)] modelled the flow stress behavior of AA5086 aluminum using a NN with two hidden layers. Also, in the realm of fluid mechanics, Ling et al. [Ling and Templeton (2015); Ling, Kurzawski and Templeton (2016)], Duraisamy et al. [Tracey, Duraisamy and Alonso (2015); Duraisamy, Zhang and Singh (2015)], and Milano et al. [Milano and Koumoutsakos (2002)] have been particularly active in applying machine learning techniques to model turbulence [Wang, Wu, Ling et al. (2017)]. Unlike traditional models based on physical mechanisms and intuition, these ML models are purely data-driven and phenomenological. Recently, mathematical analysis has been applied to understanding the training and response structure of NNs, which have traditionally been treated as black boxes. The work of Tishby and co-workers [Shwartz-Ziv and Tishby (2017)] and Koh et al. [Koh and Liang (2017))] is particularly illuminating and explores the trade-offs between information compression and prediction accuracy in the training process.

In the wider context of data-driven modeling, a number of recent developments [Alharbi and Kalidindi (2015); Kirchdoerfer and Ortiz (2016); Smith, Xiong, Yan et al. (2016); Versino, Tonda and Bronkhorst (2017); Bessa, Bostanabad, Liu et al. (2017)] are also noteworthy. Alharbi et al. [Alharbi and Kalidindi (2015)] constructed a database of Fourier transformed microstructural data and used this spectral information to drive the evolution of crystal plasticity simulation. Kirchdoerfer et al. [Kirchdoerfer and Ortiz (2016)] sought to subvert the traditional empirical model in the data-to-model-to-prediction chain and replace it with a penalization of the prediction response by its distance to closest experimental observation/data point. This approach of directly using a database is commendable (and avoided data interpolation which appears, for example, in Shaughnessy et al. [Shaughnessy and Jones (2016))]. The optimization was constrained by conservation principles like a Newtonian force balance and was applied to truss and elasticity problems. The authors explored the technique's robustness to noise and convergence. Versino et al. [Versino, Tonda and Bronkhorst (2017)] applied a genetic/evolutionary algorithm and a symbolic regression to model Taylor impact test data. The symbolic regression ML technique selects a best model composed of given analytic building-blocks and is especially attractive since the resulting tree structure leads to a physically intepretable model based on the physics embedded in the building-block sub-models. Lastly, Bessa et al. [Bessa, Bostanabad, Liu et al. (2017)] integrated design of experiments, simulation, and machine learning in materials discovery and design. It should be noted that the Materials Genome and similar material discovery and selection efforts [Jain, Ong, Hautier et al. (2013); Saal, Kirklin, Aykol et al. (2013); Raccuglia, Elbert, Adler et al. (2016)] are a deep and active field of research but this classification problem has minor bearing on the constitutive modeling task at hand.

In the vein of designing the architecture NN suit to specific tasks, the method we adopt and generalize, the Tensor Basis Neural Network (TBNN) [Ling, Jones and Templeton (2016)], is not simply a feed-forward, deep neural network. Unlike other NN mechanics models of the components of output quantities, such as stress, TBNN models have built-in invariance properties. The TBNN formulation shifts the basis for the unknown coefficient functions from the (arbitrary) Cartesian basis of the training data to an objective basis made up of powers of the selected inputs, as representation theory [Spencer (1971); Truesdell and Noll (2004)] suggests. This comes with the cost that the coefficient functions and basis are not linearly independent i.e. they must be trained simultaneously. This representation is akin to the Gaussian Approximation Potential (GAP) with the Smooth Overlap of Atomic Positions (SOAP) basis [Bartók and Csányi (2015)] that is gaining popularity in molecular dynamics, in that this machine learning constitutive function uses a spectral basis to preserve rotational and permutational invariance. It also has goals in common with image transforms that embed invariance properties [Khotanzad and Hong (1990); Lowe (1999)].

Motivated by the goal of achieving on-the-fly model construction, directed sampling/experiments, and discovery of features/trends in large datasets, in this work we show how classical constitutive modeling is needed to obtain viable ML models of constitutive behavior. In Section 2, we provide the fundamentals of representation and plasticity theories and connect them with our NN formulation of the components of plasticity, namely the stress and flow rules. In Section 3 , we discuss how the data to train the models is obtained, the specifics of the learning algorithm, and the time integration algorithm used to predict the plastic evolution. One of the data sets is obtained from the elastic-plastic response of an ensemble of oligo-crystalline aggregates, and so the resulting NN model can be considered a form of homogenization. The results of these developments are discussed in Section 4 and include comparisons of various model architectures. These comparisons are based on cross-validation errors and evaluations of stability and prediction accuracy in the context of adequate training. Finally, in Section 5, we discuss results and innovations, such as the generalized tensor basis architecture, the novel ways of embedding physical constraints in the formulation, and the exploration of data sufficiency, robustness, and stability in the context of commonly available data.

2 Theory

In this section we provide a concise overview of representation theory and how we apply it in the context of constitutive modeling by (artificial) neural networks (NNs). Specifically, we employ a generalization of the Tensor Basis Neural Network (TBNN) [Ling, Jones and Templeton (2016)] concept based on an understanding of classical representation theory. With it we construct models that represent the selected output as a function of inputs with complete generality and compact simplicity. This construction is distinct from the predominance of component-based NN constructions, for example those mentioned in the Introduction, in that basic symmetries, such as frame invariance are built into the representation and do not need to be learned.

2.1 Representation theory

Representation theorems for functions of tensors have a foundation in group theory [Olver (2000); Goodman and Wallach (1998, 2009); Sattinger and Weaver (2013)] with the connection being that symmetry is described as functional invariance under group action. In mechanics, the relevant invariance under group action are rotations (and translations) of the coordinate system, which is known as *material frame indifference*, *invariance under super-posed rigid body motions* or simply *objectivity*.[1] This is a fundamental and exact symmetry. Practical applications of representation theory to mechanics are given in Truesdell and Noll's monograph [Truesdell and Noll (2004)] and Gurtin's text [Gurtin (1982)] and address complete, irreducible representations of general functions of physical vector and tensor arguments. For example, the scalar function $f(\mathbf{A})$ of a (second order) tensor $\mathbf{A}$ is invariant if

$$f(\mathbf{A}) = f(\mathbf{G}\mathbf{A}\mathbf{G}^T) \, , \tag{1}$$

and a (second order) tensor-valued function $\mathbf{M}(\mathbf{A})$ is objective if

$$\mathbf{G}\mathbf{M}(\mathbf{A})\mathbf{G}^T = \mathbf{M}(\mathbf{G}\mathbf{A}\mathbf{G}^T) \, , \tag{2}$$

for every member $\mathbf{G}$ of the orthogonal group.

Underpinning the representations of f and $\mathbf{M}$ are a number of theorems. The *spectral theorem* states that any *symmetric* second order tensor $\mathbf{A}$ has spectral representation:

$$\mathbf{A} = \sum_{i=1}^{3} \lambda_i \mathbf{a}_i \otimes \mathbf{a}_i \, , \tag{3}$$

composed of its eigen-values $\{\lambda_i\}$ and eigen-vectors $\{\mathbf{a}_i\}$ where $i = 1, 2, 3$. Spectral representation makes powers of $\mathbf{A}$ take a simple form: $\mathbf{A}^n = \sum_i \lambda_i^n \mathbf{a}_i \otimes \mathbf{a}_i$, and in particular $\mathbf{A}^0 \equiv \mathbf{I}$. The equally important Cayley-Hamilton theorem states that the tensor $\mathbf{A}$ satisfies its characteristic equation:

$$\mathbf{A}^3 - \underbrace{(\lambda_1 + \lambda_2 + \lambda_3)}_{J_1 = \operatorname{tr} \mathbf{A}} \mathbf{A}^2 + \underbrace{(\lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1)}_{J_2 = \frac{1}{2}(\operatorname{tr}^2 \mathbf{A} - \operatorname{tr} \mathbf{A}^2)} \mathbf{A} - \underbrace{(\lambda_1\lambda_2\lambda_3)}_{J_3 = \det \mathbf{A}} \mathbf{I} = \mathbf{0} \, , \tag{4}$$

where $\{J_i\}$ are the *principal (scalar) invariants* of $\mathbf{A}$. The (generalized) Rivlin's identities [Rivlin (1955); Rivlin and Smith (1997)] provide similar relations for multiple tensors and their joint invariants.

[1] Frame indifference is a special case of the more general principle of covariance with changes of the metric tensor [Marsden and Hughes (1983)].

Scalars that respect Eq. (1), such as $\{J_i\}$, are called *scalar invariants* and are formed from polynomials or, more generally, functions of the eigenvalues of $\mathbf{A}$. Hence, $f(\mathbf{A})$ reduces to

$$f(\mathbf{A}) = f(\mathcal{I}) \tag{5}$$

where $\mathcal{I}$ is a set of scalar invariants of $\mathbf{A}$, and consequently f is also an invariant. A set of invariants $\mathcal{I}$ is considered *irreducible* if each of its elements cannot be represented in terms of others and conveys a sense of completeness and simplicity.[2] Since the eigenvalues $\{\lambda_i\}$ are costly to compute, typically traces such as $\{\operatorname{tr}\mathbf{A} = \sum_i \lambda_i, \operatorname{tr}\mathbf{A}^2 = \sum_i \lambda_i^2, \operatorname{tr}\mathbf{A}^3\lambda_i^3\}$ are employed as scalar invariants. Joint invariants of a functional basis for multiple arguments are formed with the help of Pascal's triangle.

For tensor-valued functions such as $\mathbf{M}(\mathbf{A})$ in Eq. (2), a power series representation

$$\mathbf{M}(\mathbf{A}) = \sum_{i=0}^{\infty} c_i(\mathcal{I})\mathbf{A}^i \tag{6}$$

provides a starting point for a more compact representation. The coefficient functions c_i are represented in terms of scalar invariants as in Eq. (5). This power series representation can then be reduced by application of the Cayley-Hamilton theorem (4), in the recursive form $\mathbf{A}^{j+3} = J_1\mathbf{A}^{j+2} - J_2\mathbf{A}^{j+1} + J_3\mathbf{A}^j$. The *transfer theorem* (as referred to by Gurtin [Gurtin (1982)]) states that isotropic functions such as $\mathbf{M}(\mathbf{A})$ inherit the eigenvalues of their arguments and implies the fact that these functions are co-linear with their arguments. Also Wang's lemma ($\mathbf{I}, \mathbf{A}, \mathbf{A}^2$ span the space of all tensors co-linear with $\mathbf{A}$) is a consequence of Eq. (3) and Eq. (4), and gives a sense of completeness of the representation:

$$\mathbf{M}(\mathbf{A}) = c_0(\mathcal{I})\mathbf{I} + c_1(\mathcal{I})\mathbf{A} + c_2(\mathcal{I})\mathbf{A}^2 \,. \tag{7}$$

Eq. (7) evokes the general representation for a symmetric tensor function of an arbitrary number of arguments in terms of a sum of scalar coefficient functions multiplying the corresponding elements of the tensor basis. The general methodology for constructing the functional basis to represent scalar functions is given in Rivlin et al. [Rivlin and Ericksen (1955)], and the corresponding methodology to construct tensor bases is developed in Wang [Wang (1969, 1970)].

Representation theory, like machine learning, does not determine the *appropriate* inputs and output for constitutive functions. In mechanics, there is a certain amount of fungibility to both. For instance, the (spatial) Cauchy stress can easily be transformed into the (referential) first Piola-Kirchhoff stress, and left and right Cauchy-Green stretch have same eigenvalues but different eigen-bases. Also, any of the Seth-Hill/Doyle-Ericksen strain family [Seth (1961); Hill (1968); Doyle and Ericksen (1956)] provide equivalent information on deformation, and any of the objective rates formed from Lie derivatives [Johnson and Bammann (1984); SzABo and Balla (1989); Haupt and Tsakmakis (1989, 1996)] provide equivalent measures of rate of deformation; however, some choices of arguments and output lead to greater simplicity in the representation than others.

Lastly, it is important to note that isotropic functions are not restricted to isotropic responses. The addition of a structure tensor characterizing the material symmetry to the

[2] In some sense, a complete set of invariants are coordinates on the manifold induced by symmetry constraints and hence are clearly not unique in their ability to coordinatize the manifold.

arguments allows isotropic function theory to be applied so that the joint invariants encode anisotropies [Smith and Rivlin (1957b,a); Spencer (1982); Zhang and Rychlewski (1990); Svendsen (1994); Zheng (1994)].

2.2 Plasticity models

Briefly, plasticity is an inelastic, history-dependent process due to dislocation motion or other dissipative phenomena. We assume the usual multiplicative decomposition [Lee (1969); Lubarda (2004)] of the total deformation gradient $\mathbf{F}$ into elastic (reversible) $\mathbf{F}_e$ and plastic (irreversible) $\mathbf{F}_p$ components:

$$\mathbf{F} = \mathbf{F}_e \mathbf{F}_p \tag{8}$$

is valid. As a consequence, the velocity gradient in the current configuration, $\mathbf{l} \equiv \dot{\mathbf{F}}\mathbf{F}^{-1}$, can be additively decomposed into elastic and plastic components :

$$\mathbf{l} = \dot{\mathbf{F}}_e \mathbf{F}_e^{-1} + \mathbf{F}_e \underbrace{\dot{\mathbf{F}}_p \mathbf{F}_p^{-1}}_{\mathbf{L}_p} \mathbf{F}_e^{-1} \,, \tag{9}$$

refer to Lubliner [Lubliner (2008)]. The assumption that $\mathbf{F}_p$ is pure stretch (no rotation) reduces $\mathbf{L}_p$ to $\mathbf{D}_p \equiv \operatorname{sym} \mathbf{L}_p$. The elastic deformation determines the stress, for instance the Cauchy stress $\mathbf{T}$:

$$\mathbf{T} = \hat{\mathbf{T}}(\mathbf{F}_e) = \mathbf{T}(\mathbf{e}_e) \,, \tag{10}$$

and the evolution of the plastic state is determined by a *flow rule*, *e.g.*:

$$\dot{\mathbf{F}}_p = \mathbf{D}_p \mathbf{F}_p \quad \text{where} \quad \mathbf{D}_p = \hat{\mathbf{D}}_p(\mathbf{F}_p, \mathbf{T}) = \mathbf{D}_p(\mathbf{b}_p, \boldsymbol{\sigma}) \,, \tag{11}$$

where $\mathbf{F}_p$ quantifies the plastic state and $\mathbf{T}$ the driving stress. Invariance allows the reduction of the argument of $\mathbf{T}$ to, for example, the objective, elastic Almansi strain $\mathbf{e}_e = \frac{1}{2}\left(\mathbf{I} - \mathbf{b}_e^{-1}\right)$ based on the left Cauchy-Green/Finger stretch tensor $\mathbf{b}_e = \mathbf{F}_e \mathbf{F}_e^T$. Similarly, the state variable in the flow rule can be reduced by applying invariance, for example, $\mathbf{b}_p = \mathbf{F}_p \mathbf{F}_p^T$. The driving stress can be attributed to the deviatoric part of the pull-back of the Cauchy stress $\mathbf{T}$: $\boldsymbol{\sigma} = \operatorname{dev}\left[\mathbf{F}_e^{-1}\mathbf{T}\mathbf{F}_e^{-T}\right]$ which is also invariant and also coexists in the intermediate configuration with $\mathbf{D}_p$. Furthermore, a deviatoric tensor basis element, such at $\boldsymbol{\sigma}$, generates an isochoric flow which respects plastic incompressibility $\det \mathbf{F}_p \equiv 1$. Other choices of the inputs and outputs of the stress and flow functions are discussed in Results section. Typically both the stress and flow are derived potentials, despite the fact that potential usually cannot be measured directly, to ensure elastic energy conservation for the stress and associative flow for the flow rule; however, in this work we to allow for a more general flow and a non-differentiable NN model.[3]

A few basic properties are built into traditional empirical models that need to be learned in typical NN models. First, zero strain, $\mathbf{e}_e = \mathbf{0}$, implies zero stress:

$$\mathbf{T}(\mathbf{0}) = \mathbf{0} \,, \tag{12}$$

and, likewise, zero driving stress should result in zero plastic flow:

$$\mathbf{D}_p(\mathbf{F}_p, \mathbf{0}) = \mathbf{0} \,. \tag{13}$$

[3] Also worth mentioning are the complex requirements for elastic stability, see Marsden et al. [Marsden and Hughes (1983)], that we do not attempt to embed in the formulation mainly because they require a potential.

Also there is a dissipation requirement for the plastic flow. Generally speaking, the Coleman et al. [Coleman and Noll (1963)] argument, together with the first and second law of thermodynamics, applied to a free energy in terms of the elastic deformation and a plastic history variable results in: (a) The stress being conjugate to the elastic strain rate, and (b) The internal, plastic state variable, when it evolves, reduces the change in free energy to $\mathbf{M} \cdot \mathbf{L}_p \geq 0$, where $\mathbf{M}$ is the Mandel stress

$$\mathbf{M} = \det(\mathbf{F}) \, \left[\mathbf{F}_e^T \mathbf{F}_e\right] \left[\mathbf{F}^{-1}\mathbf{T}\mathbf{F}^{-T}\right] \ , \tag{14}$$

or equivalently

$$\mathbf{T} \cdot \mathbf{d}_p \geq 0 \ , \tag{15}$$

refer to Lubliner [Lubliner (2008)]. Also, given the physics of dislocation motion, it is commonly assumed that the plastic deformation is incompressible, $\det \mathbf{F}_p = 1$, which implies the flow is deviatoric

$$\operatorname{tr} \mathbf{D}_p = 0 \tag{16}$$

For more details see the texts [Lubliner (2008); Simo and Hughes (1998); Gurtin, Fried and Anand (2010)].

2.3 Application to neural network constitutive modeling

As mentioned in the Introduction, we generalize the Tensor Basis Neural Network (TBNN) formulation [Ling, Jones and Templeton (2016)] to build NN representations for the stress relation, Eq. (10), and the plastic flow rule, Eq. (11), that embed a number of symmetries and constraints. Both $\mathbf{T}$ and $\mathbf{D}_p$ are required to be isotropic functions of their arguments by invariance. As discussed in Section 2.1, classical representation theorems give the general form

$$\mathbf{f}(\mathcal{A}) = \sum_i f_i(\mathcal{I}) \, \mathbf{B}_i \ , \tag{17}$$

where $\mathcal{A} \equiv \{\mathbf{A}_1, \mathbf{A}_2, \ldots\}$ are the pre-supposed dependencies/arguments of function $\mathbf{f}$, $\mathcal{I} \equiv \{I_j\}$ is an (irreducible) set of scalar invariants of $\mathcal{A}$, and $\mathcal{B} = \{\mathbf{B}_j\}$ is the corresponding tensor basis. In Eq. (17), only the scalar coefficient functions are $\{f_i\}$ are unknown once the inputs have been selected; and, hence, they are represented with a dense NN using the selected scalar invariants $\mathcal{I}$ as inputs embedded in the overall TBNN structure. In the TBNN framework, the sum the NN functions $\{f_i(\mathcal{I})\}$ and the corresponding tensor basis elements $\{\mathbf{B}_i\}$ in Eq. (17) is accomplished by a so-called *merge layer*, and the functions $\{f_i\}$ are trained simultaneously (refer to Fig. 1 and more details will be given in Section 3.2). This formulation is in contrast to the standard, component-wise NN formulation:

$$\mathbf{f}(\mathcal{A}) = \sum_{i,j} f_{ij}([\mathbf{A}_1]_{ij}\, , [\mathbf{A}_2]_{ij}\, , \ldots) \, \mathbf{e}_i \otimes \mathbf{e}_j \ , \tag{18}$$

which is based on components of both the inputs $\{\mathbf{A}_1, \mathbf{A}_2, \ldots\}$ and the output $\mathbf{f}$.

For the stress, we assume a single symmetric tensor input selected from the Seth-Hill/Doyle-Ericksen elastic strain family, in particular $\mathbf{e}_e$, is sufficient, so that representation Eq. (7):

$$\mathbf{T} = \sigma_0(\mathcal{I})\mathbf{I} + \sigma_1(\mathcal{I})\mathbf{e}_e + \sigma_2(\mathcal{I})\mathbf{e}_e^2 \ , \tag{19}$$

is appropriate. Despite this formulation being based on strain, versus stretch, it does not embed the zero stress property, Eq. (12), and, hence, $\sigma_0(\mathcal{I})$ will need to learn that zero strain implies zero stress. We prefer to impose, rather than learn, physical constraints such as Eq. (12) since this reduces the necessary training data [Ling, Jones and Templeton (2016)], and exact satisfaction leads to conservation and other properties necessary for stability, *etc.* Exact satisfaction of Eq. (12) can be accomplished a few different ways: shifting the basis with the Cayley-Hamilton theorem (4)

$$\mathbf{T} = \sigma_1 \mathbf{e}_e + \sigma_2 \mathbf{e}_e^2 + \sigma_3 \mathbf{e}_e^3 \, , \tag{20}$$

refactoring some $\mathbf{T} = (I_2 \sigma_0') \, \mathbf{I} + \sigma_1' \mathbf{e}_e + \sigma_2' \mathbf{e}_e^2$, or all $\mathbf{T} = I_2 \left(\sigma_0'' \mathbf{I} + \sigma_1'' \mathbf{e}_e + \sigma_2'' \mathbf{e}_e^2 \right)$ of the coefficient functions $\{\sigma_i\}$ with $I_2 = \operatorname{tr} \mathbf{e}_e^2$. In general, any of these representations can be expressed on the spectral basis

$$\mathbf{T} = \sum_i \sum_{j=1}^3 \sigma_i \lambda_j^i \mathbf{a}_j \otimes \mathbf{a}_j = \sum_{j=1}^3 \left(\sum_i \sigma_i \lambda_j^i \right) \mathbf{a}_j \otimes \mathbf{a}_j \tag{21}$$

so there is a (weak) equivalence between coefficient functions of the various representations. Here, $\mathbf{e}_e = \sum_i \lambda_i \mathbf{a}_i \otimes \mathbf{a}_i$.

As mentioned in Section 2.2, we assume that the inputs to the flow rule are (a) a history variable $\mathbf{b}_p$, and (b) driving stress $\boldsymbol{\sigma}$. A general function representation from classical theory for an isotropic function of two (symmetric) tensor arguments requires ten invariants [Rivlin (1955)] (see also Boehler [Boehler (1987)]):

$$\mathcal{I} \equiv \{I_i\} = \{\operatorname{tr} \mathbf{b}_p, \operatorname{tr} \mathbf{b}_p^2, \operatorname{tr} \mathbf{b}_p^3, \operatorname{tr} \boldsymbol{\sigma}, \operatorname{tr} \boldsymbol{\sigma}^2, \operatorname{tr} \boldsymbol{\sigma}^3, \operatorname{tr} \mathbf{b}_p \boldsymbol{\sigma}, \operatorname{tr} \mathbf{b}_p^2 \boldsymbol{\sigma}, \operatorname{tr} \mathbf{b}_p \boldsymbol{\sigma}^2, \operatorname{tr} \mathbf{b}_p^2 \boldsymbol{\sigma}^2\} \tag{22}$$

and eight tensor generators/basis elements

$$\mathcal{B} \equiv \{\mathbf{B}_i\} = \{\mathbf{I}, \mathbf{b}_p, \mathbf{b}_p^2, \boldsymbol{\sigma}, \boldsymbol{\sigma}^2, \operatorname{sym} \mathbf{b}_p \boldsymbol{\sigma}, \operatorname{sym} \mathbf{b}_p^2 \boldsymbol{\sigma}, \operatorname{sym} \mathbf{b}_p \boldsymbol{\sigma}^2\} \, , \tag{23}$$

where $\operatorname{sym} \mathbf{A} \equiv \frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$. To satisfy the zero flow condition, Eq. (13), we can shift basis for the second, stress argument and eliminate all basis elements solely dependent on the first, plastic state argument:

$$\mathcal{B} = \{\boldsymbol{\sigma}, \boldsymbol{\sigma}^2, \boldsymbol{\sigma}^3, \operatorname{sym} \mathbf{b}_p \boldsymbol{\sigma}, \operatorname{sym} \mathbf{b}_p^2 \boldsymbol{\sigma}, \operatorname{sym} \mathbf{b}_p \boldsymbol{\sigma}^2\} \, . \tag{24}$$

Plastic incompressibility, in the form of deviatoric plastic flow, Eq. (16), can imposed by applying the linear operator dev, $\operatorname{dev} \mathbf{A} \equiv \mathbf{A} - \frac{1}{3} \operatorname{tr}(\mathbf{A}) \mathbf{I}$,

$$\begin{aligned} \mathbf{D}_p &= f_{01} \operatorname{dev} \boldsymbol{\sigma} + f_{11} \operatorname{sym} \operatorname{dev} \mathbf{b}_p \boldsymbol{\sigma} + f_{02} \operatorname{dev} \boldsymbol{\sigma}^2 \\ &\quad + f_{21} \operatorname{dev} \operatorname{sym} \mathbf{b}_p^2 \boldsymbol{\sigma} + f_{12} \operatorname{dev} \operatorname{sym} \mathbf{b}_p \boldsymbol{\sigma}^2 \end{aligned}$$

Dissipation of plastic flow can be strictly imposed by requiring that the flow be directly opposed to the stress in Eq. (15) which implies:

$$\mathbf{D}_p = f_1 \, \boldsymbol{\sigma} + f_3 \, \boldsymbol{\sigma}^3 \, , \tag{25}$$

and $f_1(\mathcal{I}) > 0$ and $f_3(\mathcal{I}) > 0$. In this study we will rely on the learning process to ensure the positivity of the coefficient functions f_1 and f_3 but this could be accomplished exactly with the Macauley bracket (ramp function) applied to f_1 and f_3, for example.

3 Methods

We train the NN models of plasticity with data from two traditional plasticity models. In this section we give details of (a) the traditional models, (b) the training of the NNs, and (c) numerical integration of the TBNN plasticity model.

3.1 Plasticity models

In an exploration of the fundamental properties of NNs applied to plasticity, we seek to represent responses of two models: (a) A poly-crystalline representative volume element (RVE) with grain-wise crystal plasticity (CP) response (an *unknown* closed form model since the poly-crystalline aspect of the CP model obscures its closed form), and (b) A simple visco-plasticity (VP) material point (a *known* closed form model). Both are finite deformation models so that invariance and finite rotation are important; and both are visco-plastic in the sense of lacking a well-defined yield surface and strictly dissipative character.

Briefly, crystal plasticity (CP) is a well-known meso-scale model of single crystal deformation. Here we use crystal plasticity to prescribe the response of individual crystals in a perfectly bonded polycrystalline aggregate. The theoretical development of CP is described in Taylor et al. [Taylor (1934); Kroner (1961); Bishop and Hill (1951b,a); Mandel (1965)] and the computational aspects in reviews [Dawson (2000); Roters, Eisenlohr, Hantcherli et al. (2010)].

Specifically, for the crystal elasticity, we employ a St. Venant stress rule formulated with the second Piola-Kirchhoff stress mapped to the current configuration

$$\mathbf{T} = \frac{1}{\det \mathbf{F}} \mathbf{F} \left(\mathbb{C} \mathbf{E}_e \right) \mathbf{F}^T \tag{26}$$

where the elastic modulus tensor $\mathbb{C} = C_{11}\mathbb{K} + C_{12}(\mathbb{I} - \mathbb{K}) + C_{44}(\mathbb{J} - 2\mathbb{K})$ has cubic crystal symmetries with C_{11}, C_{12}, C_{44} = 204.6, 137.7, 126.2 GPa, and $\mathbf{E}_e = \frac{1}{2}\left(\mathbf{F}_e^T \mathbf{F}_e - \mathbf{I}\right)$ is the elastic Lagrange strain. Here $[\mathbb{I}]_{ijkl} = \delta_{ij}\delta_{kl}$, $[\mathbb{J}]_{ijkl} = \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}$, $[\mathbb{K}]_{ijkl} = \delta_{ij}\delta_{kl}\delta_{ik}$ (no sum implied), δ_{ij} is the Kronecker delta, and $\mathbb{C}\mathbf{E}_e = \sum_{k,l} [\mathbb{C}]_{ijkl} [\mathbf{E}_e]_{kl} \mathbf{E}_i \otimes \mathbf{E}_j$. Plastic flow can occur on any of 12 face-centered cubic (FCC) slip planes. Each crystallographic slip system, indexed by α, is characterized by Schmid dyads $\mathbf{P}_\alpha = \mathbf{s}_\alpha \otimes \mathbf{n}_\alpha$ composed of the allowed slip direction, $\mathbf{s}_\alpha$, and the normal to the slip plane, $\mathbf{n}_\alpha$. Given the set $\{\mathbf{P}_\alpha\}$, the plastic velocity gradient is constructed via:

$$\mathbf{L}_p = \sum_\alpha \dot{\gamma}_\alpha \mathbf{P}_\alpha \ , \tag{27}$$

which is inherently volume preserving in the (incompatible) intermediate/lattice configuration. Finally, the slip rate $\dot{\gamma}_\alpha$ is related to the applied stress through the resolved shear (Mandel) stress $\tau_\alpha = \mathbf{M} \cdot \mathbf{P}_\alpha$, for that slip system. We employ a common power-law form for the slip rate relation

$$\dot{\gamma}_\alpha = \dot{\gamma}_{\alpha 0} \left| \frac{\tau_\alpha}{g_\alpha} \right|^{1/m} \tau_\alpha \ , \tag{28}$$

where $\dot{\gamma}_{\alpha 0}$ = 122.0 (MPa-s)$^{-1}$ is a reference strain rate, $m = 20$ is a rate sensitivity exponent, and g_α = 355.0 MPa is a hardness value. These parameters are representative of steel.

With this model, we simulate the polycrystalline response using a uniform mesh 20 × 20 × 20 with the texture assigned element-wise and strict compatibility enforced at the voxelated grain boundaries. Ten realizations with 15, 15, 17, 18, 18, 19, 19, 20, 20, 21, 22 grains were sampled from an average grain size ensemble and each grain was assigned a random orientation. Minimal boundary conditions to apply the various loading modes (tension, shear, etc.) were employed on the faces and edges of the cubical representative volumes. Also, we limit samples to a single, constant strain rate 1.01/s.

The simple visco-plastic (VP) model consists of a St. Venant stress rule in the current configuration with Almansi strain:

$$\mathbf{T} = \mathbb{C}\mathbf{e}_e \ , \tag{29}$$

where $\mathbb{C} = \lambda\mathbb{I} + \mu\mathbb{J}$ with isotropic parameters $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$, Young's modulus E = 200 GPa and Poisson's ratio ν = 0.3, together with a simple (associative) power law for the flow rule:

$$\mathbf{D}_p = c\|\mathbf{s}\|^p\mathbf{s} \ , \tag{30}$$

where c = 0.001 MPa$^{-(1+p)}$-s^{-1} and p = 0.1 are material constants.

3.2 Neural network representation and machine learning algorithm

A typical NN, such as the representation of each component of Eq. (18), is a two-dimensional feed-forward, directed network consisting of an input layer, output layer and L intervening hidden layers where neighboring layers are densely connected. Each layer ℓ_i consists of N nodes (ij). The vector of outputs, y_i, of the nodes $(ij), j \in (1, N)$ of layer ℓ_i is the weighted sum of the outputs of the previous layer ℓ_{i-1} offset by a threshold and passed through a ramp-like or step-like *activation* function $a(x)$:

$$\mathsf{x}_i = a(\mathsf{y}_i) \ \text{ with } \ \mathsf{y}_i = \mathsf{W}_i\mathsf{x}_{i-1} + \mathsf{b}_i \ , \tag{31}$$

where W_i is the weight matrix for the inputs of (hidden) layer ℓ_i of the state/output of nodes of the previous layer x_{i-1} and b_i is the corresponding threshold vector. In our application the input layer consists of the $N_{\mathcal{I}}$ invariants $\mathcal{I}$ and the $N_{\mathcal{B}}$ elements of the tensor basis $\mathcal{B}$. The elements of $\mathcal{I}$ form the arguments of the coefficient functions, each having a $L \times N$ neural network representation, while the elements of $\mathcal{B}$ pass through the overall network until they are combined with the coefficient functions according to Eq. (17) to form the output via a *merge* layer that does the summation of the coefficient functions and the elements of the tensor basis. After exploring the C0 step- and ramp-like *rectifying* activation functions commonly used, we employ the ramp-like (C1 continuous) Exponential Linear Unit (ELU) [Clevert, Unterthiner and Hochreiter (2015)] activation function:

$$a(x) = \begin{cases} \exp(x) - 1 & \text{if } x < 0 \\ x & \text{else} \end{cases} \tag{32}$$

to promote smoothness of the response and limit the depth of the network necessary to represent the response relative that necessary with saturating step-like functions.

Training the network weights W_i and thresholds b_i is accomplished via the standard back-propagation of errors [Werbos (1974); Rumelhart, Hinton and Williams (1986)] which, in turn, drives a (stochastic) gradient-based descent (SGD) optimization scheme to minimize the so-called *loss*/error, E. We employ the usual root mean square error (RMSE)

$$E = \frac{1}{2N_D} \sum_{(\mathsf{x}_k, \mathsf{d}_k) \in D} \|\mathsf{y}(\mathsf{x}_k) - \mathsf{d}_k\|^2 \ , \tag{33}$$

where D is the set of training data composed of inputs $\mathsf{x}_k = \{\mathcal{I}_k, \mathcal{B}_k\}$ and corresponding output d_k. The gradient algorithm relies on: (a) The change in E with respect to each weight W_i

$$\frac{\partial E}{\partial \mathsf{W}_i} = \underbrace{\frac{\partial E}{\partial \mathsf{x}_i}\frac{\partial \mathsf{x}_i}{\partial \mathsf{y}_i}}_{\Delta_i}\frac{\partial \mathsf{y}_i}{\partial \mathsf{W}_i} = \mathsf{x}_{i-1} \otimes \Delta_i \tag{34}$$

and (b) Each threshold b_i

$$\frac{\partial E}{\partial \mathsf{b}_i} = \underbrace{\frac{\partial E}{\partial \mathsf{x}_i}\frac{\partial \mathsf{x}_i}{\partial \mathsf{y}_i}}_{\Delta_i}\frac{\partial \mathsf{y}_i}{\partial \mathsf{b}_i} = \Delta_i \ , \tag{35}$$

where

$$\Delta_i = \left(\mathsf{W}_{i+1}^T \Delta_{i+1}\right) \odot a'(\mathsf{y}_i) \ \text{ for } \ i \neq L \ \text{ with } \ \Delta_L = \sum_{(\mathsf{x}_k,\mathsf{d}_k)\in D} (\mathsf{y}(\mathsf{x}_k) - \mathsf{d}_k) \odot a'(\mathsf{y}_L) \tag{36}$$

Here a' is the derivative of weight function, $[\mathsf{a} \otimes \mathsf{b}]_{ij} = a_i b_j$ is the tensor product, and $[\mathsf{a} \odot \mathsf{b}]_i = a_i b_i$ element-wise Hadamard-Schur product.[4] The gradient defined by these expressions is evaluated with random sampling of subset of training data D called *minibatches*. Also, search for a minimum along this direction is governed by a step size called the *learning rate* in the ML community. These standard constructions are trivially generalized to the TBNN structure since the inputs $\mathcal{B}$ are not directly related to W_i nor b_i, and are merely scaled by the coefficient functions to form the output y, refer to Fig. 1. For more details of the SGD algorithm see Nielsen [Nielsen (2015)].

To begin the training, the unknown weights, $\{\mathsf{W}_i\}$, and thresholds, $\{\mathsf{b}_i\}$, are initialized with normally distributed random values to break the degeneracy of the network and enable local optimization. Since multiple local minima for training are known to exist, choosing an ensemble of initial weights which are then optimized improves the chances of finding a global minimum and the distribution of the solutions indicates the robustness of the training. Also, the full set of input data is divided into a training set D, used to generate the errors for the back-propagation algorithm; a test set T, for assessing convergence of the descent algorithm; and a third set V for cross-validation, to estimate the predictive capability of the trained network. Ensuring that the errors based on T are comparable to those on V reduces the likelihood over-fitting data with a larger than necessary NN. We chose to divide the available data in a $T : D : V =$ 20:72:8 ratio. In addition, we sample individual stress-strain curves produced by the CP and VP simulators so as to maintain approximate uniform density of data along the arc-length of the stress-strain curve (*vs.* a uniform sampling over the strain range) to resolve high-gradient (elastic) and transition (yield) regimes. Also, it should be noted that we allow ourselves to train on inputs derived from the plastic deformation gradient, $\mathbf{F}_p$, despite the fact that this quantity is difficult to observe directly in experiments. A critical part of the training algorithm is normalizing the data so that the NN maps $\mathcal{O}(1)$ inputs to $\mathcal{O}(1)$ outputs since having W_i, $\mathsf{b}_i \sim \mathcal{O}(1)$ will achieve better SGD convergence. We also shift and scale the scalar invariants $\mathcal{I}$ so that they have a mean zero, variance one distribution. We normalize the other set of inputs, the tensor basis $\mathcal{B}$, using the maximum Frobenius norm of the basis generators, e.g. $\mathbf{b}_p$ and $\boldsymbol{\sigma}$,

[4] The recursion seen in Eq. (36) gives back-propagation its name.

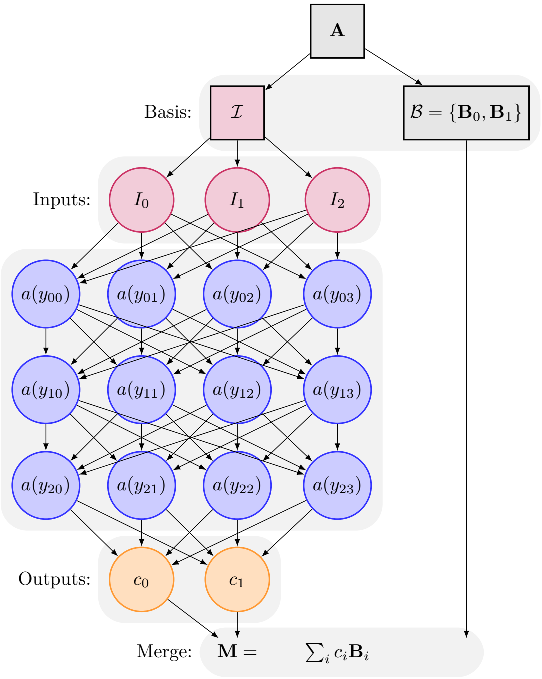


Figure 1: TBNN structure for $\mathbf{M}(\mathbf{A}) = \sum_i c_i(\mathcal{I})\mathbf{B}_i$ with 3 invariants $\mathcal{I} = \{I_0, I_1, I_2\}$, a 3×4 NN, 2 coefficient functions $\{c_0(\mathcal{I}), c_1(\mathcal{I})\}$, and 2 tensor basis elements $\mathcal{B} = \{\mathbf{B}_0, \mathbf{B}_1\}$. The linear transformation $\mathsf{y}_i = \mathsf{W}_i\mathsf{x}_{i-1} + \mathsf{b}_i$ of the outputs x_{i-1} of layer $i-1$ to the inputs y_i of layer i is denoted by the arrows connecting the nodes of layer $i-1$ to those of layer i. The nonlinearity of the activation functions $a(\mathsf{y}_i)$ is represented by $a(y_{ij})$ where y_{ij} are the components of y_i. The scaling operations described in Section 3.2 are omitted for clarity

over the training set D. During training, the output tensors are normalized similarly based on their maximum norms over D, so that

$$\mathbf{f} = \sum_i \underbrace{\frac{1}{s_\mathbf{f}} \bar{f}_i(\bar{\mathcal{I}}) s_{\mathbf{B}_i}}_{f_i(\mathcal{I})} \mathbf{B}_i \ , \tag{37}$$

where $s_\mathbf{f}$ is the scaling of output $\mathbf{f}$; $s_{\mathbf{B}_i}$ is the scaling of basis element $\mathbf{B}_i$ based on the powers of principal generator (e.g. if $\mathbf{B}_i = \mathbf{b}^a \mathbf{s}^b$ then $s_{\mathbf{B}_i} = s_\mathbf{b}^a s_\mathbf{s}^b$ where $s_\mathbf{b}$ is the scaling of $\mathbf{b}$); and $\bar{\mathcal{I}} = s_{I_i} I_i$ is the set of scaled and shifted invariants. These scales have the added benefit of coarsely encoding the range of training data so the extrapolation during prediction can be detected.

Convergence is assessed by averaging the error with respect to T over previous iterations of the SDG (in this work we average over the last 4-10 iterations) and terminating when this average converges, but not before performing a minimum number of iterations (1000 in this work). More discussion of the training approach can be found in Ling et al. [Ling, Jones and Templeton (2016)], although in that work the learning rate was held fixed rather than decaying as the training proceeds, as in this study.

3.3 Integration algorithm

We need a time-integration scheme to solve the differential-algebraic system Eq. (10) and Eq. (11). We assume it is deformation driven so that $\mathbf{F} = \mathbf{F}(t)$ is data. To form a numerical integrator, we rely on the well-known exponential map for the plastic deformation $[\mathbf{F}_p]_{n+\alpha} \equiv \mathbf{F}_p(t_{n+\alpha})$:

$$[\mathbf{F}_p]_{n+\alpha} = \exp\left(\alpha\, \Delta t\, [\mathbf{D}_p]_n\right) [\mathbf{F}_p]_n \tag{38}$$

which is an explicit/approximate solution to Eq. (11). In Tab. 1, we outline an adaptive scheme based on this update and the interpolation of the (full) deformation gradient $\mathbf{F}_{n+\alpha} = \exp(\alpha \log \Delta\mathbf{F})\, \mathbf{F}_n$ by similar means via $\log \mathbf{F}_{n+\alpha}$ = $(1-\alpha) \log \mathbf{F}_{n+1} + \alpha \log \mathbf{F}_n$ = $\log \mathbf{F}_n + \alpha \log \Delta\mathbf{F}$, with $\Delta\mathbf{F} = \mathbf{F}_{n+1}\mathbf{F}_n^{-1}$. Since we do not rely on the NN models of stress Eq. (10) and flow (11) being directly differentiable,[5] we use a simple relaxation scheme to enforce consistency of the rate $\mathbf{D}_p(\mathbf{b}_p, \boldsymbol{\sigma})$ using a residual based on the unknown $[\mathbf{F}_p]_{n+1}$ updated at each iteration given $[\mathbf{F}_p]_n$. If any step has an increase in the residual, the step size is cut; conversely, when a sub-step converges, the remainder of the interval is attempted.

4 Results

In this section we cover our investigations of: (a) Optimal network size, inputs, and representation basis; (b) Influence of training data on error and stability; and (c) The robustness and accuracy of the model predictions. As mentioned, we employ data from an unknown-form CP model (Eq. (26) and Eq. (27)) and known-form VP model (Eq. (29) and Eq. (30)). (Recall that we designate the CP model an *unknown-form* model since the response of the polycrystalline aggregates has no closed form.) Training with the data

[5] This relaxation could be improved by using the derivatives already computed by the back-propagation algorithm in a Newton solver with a trust region based on the bounds of the training data.

Table 1: Time integration algorithm with adaptive time-stepping. Note, ϵ is a convergence tolerance

```
For step $n+1$
  • **Initialize** $\mathbf{F}_p = [\mathbf{F}_p]_n$, and $\Delta\mathbf{F} = \mathbf{F}_{n+1}\mathbf{F}_n^{-1}$
  • **Sub-step**: while $\alpha < 1$, try $\alpha = 1$ with $\mathbf{F} \equiv \mathbf{F}_{n+\alpha} = \exp\left(\alpha \log\left(\Delta\mathbf{F}\right)\right)\, \mathbf{F}_n$
    Relaxation: at fixed $\alpha$, initialize $[\mathbf{F}_p]^*_{k=0} = [\mathbf{F}_p]_{n+\alpha}$ with iteration indexed by $k$:
      1. $\mathbf{b}_p^* = \left[\mathbf{F}_p\mathbf{F}_p^T\right]_k^*$ and $\mathbf{b}_e^* = \mathbf{F}\left[\left(\mathbf{F}_p^T\mathbf{F}_p\right)^{-1}\right]_k^* \mathbf{F}^T$
      2. $\mathbf{T}^* = \mathbf{T}(\mathbf{b}_e^*)$ and $\boldsymbol{\sigma}^* = \operatorname{dev}\left([\mathbf{F}_p]_k\,\mathbf{F}^{-1}\mathbf{T}^*\mathbf{F}^{-T}\,[\mathbf{F}_e]_k^T\right)$
      3. $[\mathbf{D}_p]_k^* = \mathbf{f}(\mathbf{b}_p^*, \boldsymbol{\sigma}^*)$
      4. Update $[\mathbf{F}_p]_{k+1}^* = \exp\left(\alpha\Delta t\ [\mathbf{D}_p]_k^*\right)\,\mathbf{F}_p$
      5. If $R_k \equiv \left\|[\mathbf{D}_p]_k^* - [\mathbf{D}_p]_{k-1}^*\right\| < \epsilon\left\|[\mathbf{D}_p]_0^*\right\|$ then $\alpha \mathrel{+}= \Delta\alpha$, *converged* at $t_{n+\alpha}$,
         else if $R_k > R_{k-1}$ then cut step $\alpha = 1/2\alpha$, *diverging*,
         else *continue*.
  • **Update** $\mathbf{T}_{n+1} = \mathbf{T}^*$ and $[\mathbf{F}_p]_{n+1} = [\mathbf{F}_p]^*$
```

from the CP model illustrates the NN model's ability to represent and homogenize the response of a complex system and the VP model is particularly useful for exploring NN representations since we know the true response and generating samples is computationally inexpensive.

4.1 Constructing and training the neural networks

We begin our numerical investigations with: (a) A survey of the possible representations for the models of stress and flow, and (b) Optimizing the structure and meta-parameters of the NN representations. To assess improvements in performance we used the traditional metric for evaluating NN performance, *cross-validation* error, where the training dataset D is replaced by the validation dataset V in evaluating the RMSE formula, Eq. (33).

For this study we use data from the CP model to train the stress and flow TBNNs. In particular, we collect data using 3 tension and 6 simple shear loading modes averaged over 570 random textures for each of the 10 polycrystalline realizations. As mentioned in the Methods section, we give ourselves access to the (average) plastic state variables of the CP simulations and so we train the stress and flow TBNNs independently (and not simultaneously). The trajectory for each realization is computed over a sequence of 100 uniformly spaced strains and then averaged. As stated in Section 3.2, interpolation between points in the elastic and elastic-plastic transition regimes is used to increase the density of training data in these regions. This results in approximately 2000 data points with 6 strain and 6 stress components per point, which are then split between training, test, and cross-validation datasets. This dataset is typical of experimental datasets used to calibrate traditional models but compared to typical big-data applications of neural networks, this is a relatively small dataset. It is likely that other surrogate representations, such as Gaussian Processes (GPs), would do as well at emulating the data as measured by cross validation error. Nevertheless, this work will demonstrate that appropriately sized NNs can represent the abrupt transition between the elastic and plastic regimes, which can lead to errors with other methods, e.g. global polynomials [Rizzi, Khalil, Jones et al. (2018)]. The proposed TBNN architecture is designed to enforce the underlying physical invariants and constraints in the system *exactly*, which few other surrogate models can achieve, e.g. GPs [Salloum and Templeton (2014)]. While less relevant for reducing overall representation error, respecting these invariants is crucial in forward integration of the model [Arnold (2013)] and also has benefits in understanding how the model is making its predictions in the context of classical constitutive theory. In order to optimize the dimensions of the NN to the amount of available data, the influence of the breadth and depth of the network architecture on prediction error was explored and will be discussed shortly. Since these meta-parameters can be tailored in a straightforward manner to optimize representation accuracy and avoid over-fitting, NNs appropriate to the physical problem and available data size can be constructed and evaluated for additional properties necessary for the application at hand.

Fig. 2 and Fig. 3 shows typical training data for the I3 stress and IF flow representations (refer to Tab. 2 and Tab. 3) with 3×4 and 5×8 NNs, respectively. The left column of Fig. 2 and Fig. 3 show the tension response and the right columns show the shear response. The upper panels show the (input) invariants and the (output) coefficient functions. In general, the inputs and outputs are smooth and correlated, and all coefficient functions contribute. The notable exception is the stress model in shear in which only the coefficient function of the linear basis element $\mathbf{e}_e$ appears to contribute. Note that all invariants are arguments to each coefficient function. Also, in Fig. 3 the zero invariant, $\operatorname{tr} \boldsymbol{\sigma} \equiv 0$, that becomes noise upon the input scaling described in Section 3.2. Apparently, the NN training learns to ignore this input since the outputs are smooth and regular. The lower panels show: (a) The correspondence of the model (lines) and the data (points), and (b) The error as a function of strain. The errors for each of the components are of comparable magnitude and tend to have an irregular pattern in the elastic region of the loading. Note that with a C0 activation function (e.g. the Rectifying Unit $a(x) = \max(0, x)$) we observed distinct scallops and cusps in error curves (not shown for brevity). Also it is remarkable that the errors of the flow model in shear are distinctly linear, which, perhaps, is related to the fact only the linear basis element is active.

These results are typical for a wide range of NN structures and (meta) training parameters. Fig. 4 shows the cross-validation errors of the stress and flow (scaled by $s_{\mathbf{T}}$ and $s_{\mathbf{D}_p}$, respectively) using the full representations I3 and IF (refer to Tab. 2 and Tab. 3, respectively). As Schwartz-Ziv and Tishby [Shwartz-Ziv and Tishby (2017)] remark, trying to interpret the behavior of network from a single training tends to be meaningless; hence, we evaluate parametric and structural changes with an ensemble of at least 30 replicas models $M_k \in \mathcal{M}$ in this and the following studies. (The replicas are obtained by using different random seeds to produce the initial weights and thresholds.) The insets show that the (initial) learning rate can have a strong effect on the errors, but once a small enough ($< 10^{-3}$) rate is selected the final errors are relatively insensitive to this parameter. The main panels show the typical trends in errors ranging from under-representation (too small a network) to over-fitting (too large a network).[6] For the stress TBNN, N=4 nodes appears

[6] As mentioned, we require that in the training procedure that the error on the training D and

to be an optimum even for relatively shallow networks ($N < 4$) but the optimal number of nodes is relatively insensitive for $L > 4$. The flow TBNN shows analogous behavior but with a trade-off between nodes and layers, e.g. for $L > 6$, N=4 appears to be best, while $N > 4$ is better for shallower networks. These findings are somewhat obscured by the noise in the trend lines, which persists despite using the average of 150 replica networks. Also, the convergence window (described in Section 3.2) is an important meta parameter. We obtained these results with a 4 iteration convergence window; a longer convergence window (e.g. 10 iterations) shifts the best cross-validation to smaller networks, but also induces larger variance in error between replicas. Since we want reliable error from each replica, we use a 4 iteration convergence window throughout the remainder of this work. Lastly, we do not believe cross-validation is sufficient for determining completeness of network; however, these results indicate that optimal number of nodes is less than the number of input invariants for flow but greater than this matrix rank-based criterion for the stress representation. Apparently, the NN is compressing the input for the flow network; and, hence, we conjecture that the NN is forming lower dimensional set of (alternate) invariants internally with respect to the training data.

Fig. 5 shows cross-validation error for the CP training data for various basis representation of the stress and flow functions (refer to Tab. 2 and Tab. 3 for the definition of the labels). For the stress TBNNs, all (overall) errors are comparable with the exception of the component-based representation and the one term E1 representation (with tensor basis $\mathcal{B} = \{\mathbf{e}_e\}$). Clearly, the E1 basis is not sufficient since it is akin to a one parameter Navier model of stress. From the results of the two truncated bases, I2 and ID, it appears that correlated inputs, $\mathcal{B} = \{\mathbf{I}, \mathbf{e}_e\}$ (I2), train comparably to linearly independent inputs, $\{\mathbf{I}, \operatorname{dev} \mathbf{e}_e\}$ (ID, which uses a volumetric/deviatoric split). Also, the upper panel of Fig. 5a shows that the representations without embedded satisfaction of the zero-stress at zero-strain constraint, Eq. (12), generally violate this constraint by about 1% of the maximum stress. For the flow TBNNs, all (overall) errors are comparable with the exception of the reduced scalar and tensor basis representation S1. Also the other reduced representations (R3,R1,T3,T1) have slightly higher average errors than the full representations (UF,IF,IR,SF,DS,DS,DR,DZ) albeit with reduced variance. The consistency of the representations with the zero-flow-at-zero-stress condition Eq. (12) generally follows whether powers of $\mathbf{e}_e$ are included or not. Clearly, cross-validation based on this limited dataset is not sufficient for decisive model selection but it does eliminate some representations. By comparison, the component-based representations, E_{ij} and CM, display higher errors, larger variance in the performance and poor zero-input-zero-output results. Beyond the fundamentally different functional representation, these models are likely suffering from an insufficiency of data to learn the necessary properties (the ones embedded in the generalized TBNN framework) accurately, as demonstrated in Ling et al. [Ling, Jones and Templeton (2016)]. Lastly, as discussed in the Theory section, we have embedded a number of properties in the representations, e.g. symmetry, deviatoric flow, dissipation, and, generally, the violation of the learned properties is on par with what we illustrate with the zero-stress and zero-flow conditions.

In preliminary studies we also trained networks with different inputs and outputs. In general, the cross-validation errors were comparable over a variety of choices, for example

the testing T data be comparable as failure to achieve parity in the errors is indicative of bad predictions and over-fitting.

Table 2: Stress representations: Those with $\mathbf{I}$ in their tensor basis do not satisfy the zero stress condition intrinsically, and those with less than three basis elements are not complete

	scalar	tensor
E_{ij}	component $[\mathbf{e}_e]_{ij}$	component $\mathbf{e}_i \otimes \mathbf{e}_j$
I3	full $\{\operatorname{tr}\mathbf{e}_e, \operatorname{tr}\mathbf{e}_e^2, \operatorname{tr}\mathbf{e}_e^3\}$	full $\{\mathbf{I}, \mathbf{e}_e, \mathbf{e}_e^2\}$
E3	full $\{\operatorname{tr}\mathbf{e}_e, \operatorname{tr}\mathbf{e}_e^2, \operatorname{tr}\mathbf{e}_e^3\}$	full, shifted $\{\mathbf{e}_e, \mathbf{e}_e^2, \mathbf{e}_e^3\}$
I2	full $\{\operatorname{tr}\mathbf{e}_e, \operatorname{tr}\mathbf{e}_e^2, \operatorname{tr}\mathbf{e}_e^3\}$	reduced $\{\mathbf{I}, \mathbf{e}_e\}$
ID	full $\{\operatorname{tr}\mathbf{e}_e, \operatorname{tr}\mathbf{e}_e^2, \operatorname{tr}\mathbf{e}_e^3\}$	reduced, independent $\{\mathbf{I}, \operatorname{dev}\mathbf{e}_e\}$
E1	full $\{\operatorname{tr}\mathbf{e}_e, \operatorname{tr}\mathbf{e}_e^2, \operatorname{tr}\mathbf{e}_e^3\}$	reduced $\{\mathbf{e}_e\}$

using a symmetrized Mandel stress for the driving stress input to the flow rule. We considered the rate $\dot{\overline{\mathbf{C}_p^{-1}}}$ of the inverse of the plastic right Cauchy-Green deformation tensor $\mathbf{C}_p = \mathbf{F}_p^T \mathbf{F}_p$ (as in Simo et al. [Simo and Hughes (1998)]) as the output of the flow rule and obtained similar cross-validation (and prediction) performance. Also noteworthy, we employed both the elastic and the full left Cauchy-Green stretch tensors as history inputs and the full Cauchy stress as a driving stress input. These inputs resulted in similar cross-validation except when we paired the elastic Cauchy-Green stretch with the highly correlated Cauchy stress where we observed slightly higher errors (and less variance among the errors).

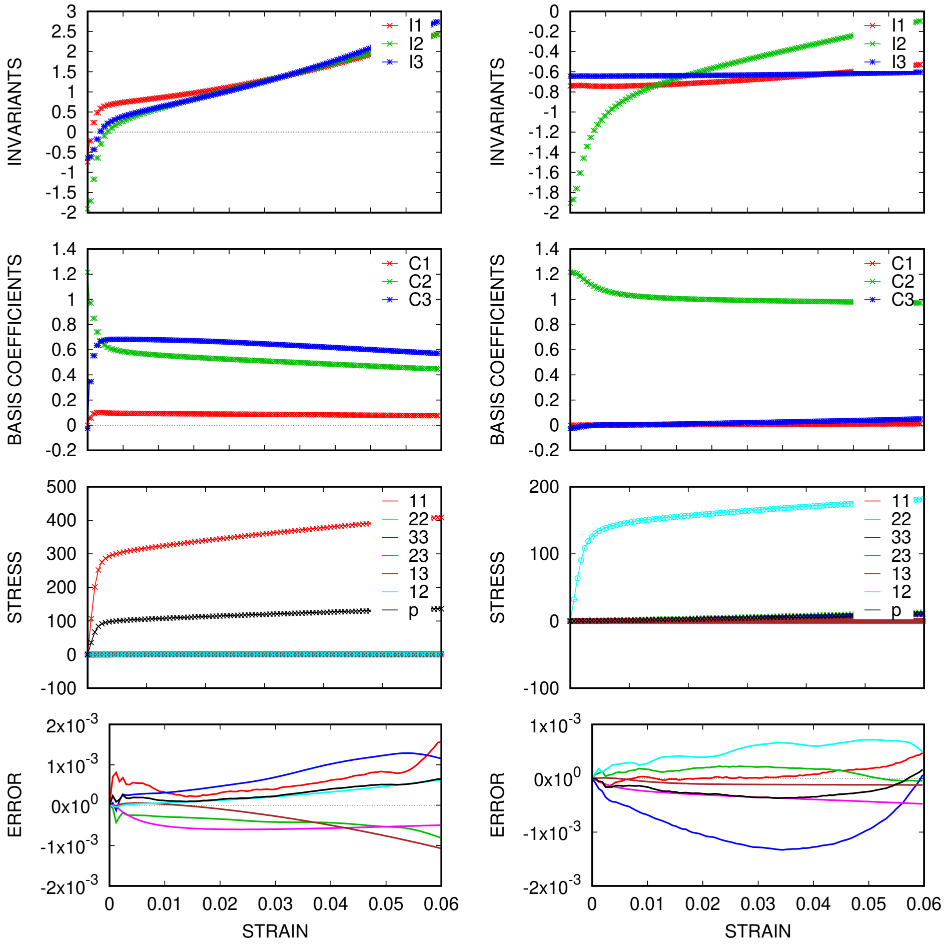
Fig. 6 and Fig. 7 show the response of the NN coefficient functions to tension and shear, for stress and flow, respectively. In these plots, each coefficient is scaled according to Eq. (37) so that coefficient functions of higher order terms can be plotted on par with those of lower order terms. First, we notice that the E3 basis achieves zero-stress at zero-strain satisfaction exactly at the expense of a more complex, larger magnitude per component response than I3, as the higher order term $\mathbf{e}_e^3$ apparently needs compensation by the component functions, refer to Eq. (21). Also, we see more evidence that the truncated representations, I2 and ID, have almost indistinguishable response despite ID having a linearly independent tensor basis. For the flow representation we only compare the DZ and T1 representations for clarity. Note that T1 is much simpler in form (one tensor basis element versus ten) than DZ, which has a complete basis, and its response is simpler while achieving comparable cross-validation error to DZ. Also evident from both tension and shear response, DZ builds a similar response to T1 by letting all/most components contribute. Also significant, the coefficient of $\boldsymbol{\sigma}^2$, C2, is essentially zero throughout the shear trajectory but not the tension, which implies that the NN may not be learning dissipation is an important property. This is in contrast with the DR representation (not shown) where the corresponding coefficient is essentially zero for both tension and shear. For both the stress and flow models, the coefficient responses generally resemble the trends in the stress and flow data, with large changes up to the elastic-plastic transition at strain > 0.002 followed by relatively constant values in the plastic regime. This is consistent with the expectation that in fully developed plastic flow (in a constant direction with negligible hardening) the elastic state and the plastic flow are constant.

4.2 Validation

Our validations studies include tests of: (a) Completeness of representation and training data, and (b) Continuity/robustness to perturbation.

Table 3: Flow representations: All but UF have a symmetric tensor basis, those with $\mathbf{I}$ or $\mathbf{b}_p$ in the tensor basis do not satisfy the zero flow constraint intrinsically, those with $\operatorname{tr}\boldsymbol{\sigma}$ include this noise invariant, those with dev applied to the tensor basis generate incompressible flows, and those with only odd powers of $\boldsymbol{\sigma}$ in the tensor basis are designed to be dissipative

CM	component basis
UF	full: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ unsymmetric,full: $\{\mathbf{I}, \mathbf{b}_p, \boldsymbol{\sigma}, \mathbf{b}_p^2, \boldsymbol{\sigma}^2, \mathbf{b}_p\boldsymbol{\sigma}, \boldsymbol{\sigma}^2\mathbf{b}_p, \mathbf{b}_p\boldsymbol{\sigma}^2\}$
IF	full: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ full: $\{\mathbf{I}, \mathbf{b}_p, \boldsymbol{\sigma}, \mathbf{b}_p^2, \boldsymbol{\sigma}^2, \operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}, \operatorname{sym}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}^2\}$
IR	no $\operatorname{tr}\boldsymbol{\sigma}$: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ full: $\{\mathbf{I}, \mathbf{b}_p, \boldsymbol{\sigma}, \mathbf{b}_p^2, \boldsymbol{\sigma}^2, \operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}, \operatorname{sym}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}^2\}$
SF	full $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ shifted, full $\{\mathbf{b}_p, \boldsymbol{\sigma}, \mathbf{b}_p^2, \boldsymbol{\sigma}^2, \operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}, \mathbf{b}_p^3, \operatorname{sym}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}^2\}$
SR	no $\operatorname{tr}\boldsymbol{\sigma}$: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ shifted,full: $\{\mathbf{b}_p, \boldsymbol{\sigma}, \mathbf{b}_p^2, \boldsymbol{\sigma}^2, \operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}, \mathbf{b}_p^3, \operatorname{sym}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}^2\}$
DS	full: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ deviatoric,shifted,full: $\{\operatorname{dev}\mathbf{b}_p, \operatorname{dev}\boldsymbol{\sigma}, \operatorname{dev}\mathbf{b}_p^2, \operatorname{dev}\boldsymbol{\sigma}^2, \operatorname{dev}\operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma},$ $\operatorname{dev}\boldsymbol{\sigma}^3, \operatorname{dev}\operatorname{sym}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{dev}\operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}^2\}$
DR	no $\operatorname{tr}\boldsymbol{\sigma}$: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ deviatoric,shifted,full: $\{\operatorname{dev}\mathbf{b}_p, \operatorname{dev}\mathbf{b}_p^2, \operatorname{dev}\boldsymbol{\sigma}, \operatorname{dev}\boldsymbol{\sigma}^2,$ $\operatorname{dev}\boldsymbol{\sigma}^3, \operatorname{dev}\operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}, \operatorname{dev}\operatorname{sym}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{dev}\operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}^2\}$
DZ	no $\operatorname{tr}\boldsymbol{\sigma}$: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ deviatoric,dissipative: $\{\operatorname{dev}\boldsymbol{\sigma}, \operatorname{dev}\boldsymbol{\sigma}^2,$ $\operatorname{dev}\boldsymbol{\sigma}^3, \operatorname{dev}\operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}, \operatorname{dev}\operatorname{sym}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{dev}\operatorname{sym}\mathbf{b}_p\boldsymbol{\sigma}^2\}$
R3	full: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ reduced, dissipative: $\{\boldsymbol{\sigma}, \operatorname{dev}\boldsymbol{\sigma}^3\}$
R1	full: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ reduced, dissipative: $\{\boldsymbol{\sigma}\}$
T3	no $\operatorname{tr}\boldsymbol{\sigma}$: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ reduced, dissipative: $\{\boldsymbol{\sigma}, \boldsymbol{\sigma}^3\}$
T1	no $\operatorname{tr}\boldsymbol{\sigma}$: $\{\operatorname{tr}\mathbf{b}_p, \operatorname{tr}\mathbf{b}_p^2, \operatorname{tr}\mathbf{b}_p^3, \operatorname{tr}\boldsymbol{\sigma}^2, \operatorname{tr}\boldsymbol{\sigma}^3, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p, \operatorname{tr}\boldsymbol{\sigma}\mathbf{b}_p^2, \operatorname{tr}\boldsymbol{\sigma}^2\mathbf{b}_p^2\}$ reduced, dissipative: $\{\boldsymbol{\sigma}\}$
S1	reduced: $\{\operatorname{tr}\mathbf{b}_p\}$ reduced, dissipative: $\{\boldsymbol{\sigma}\}$

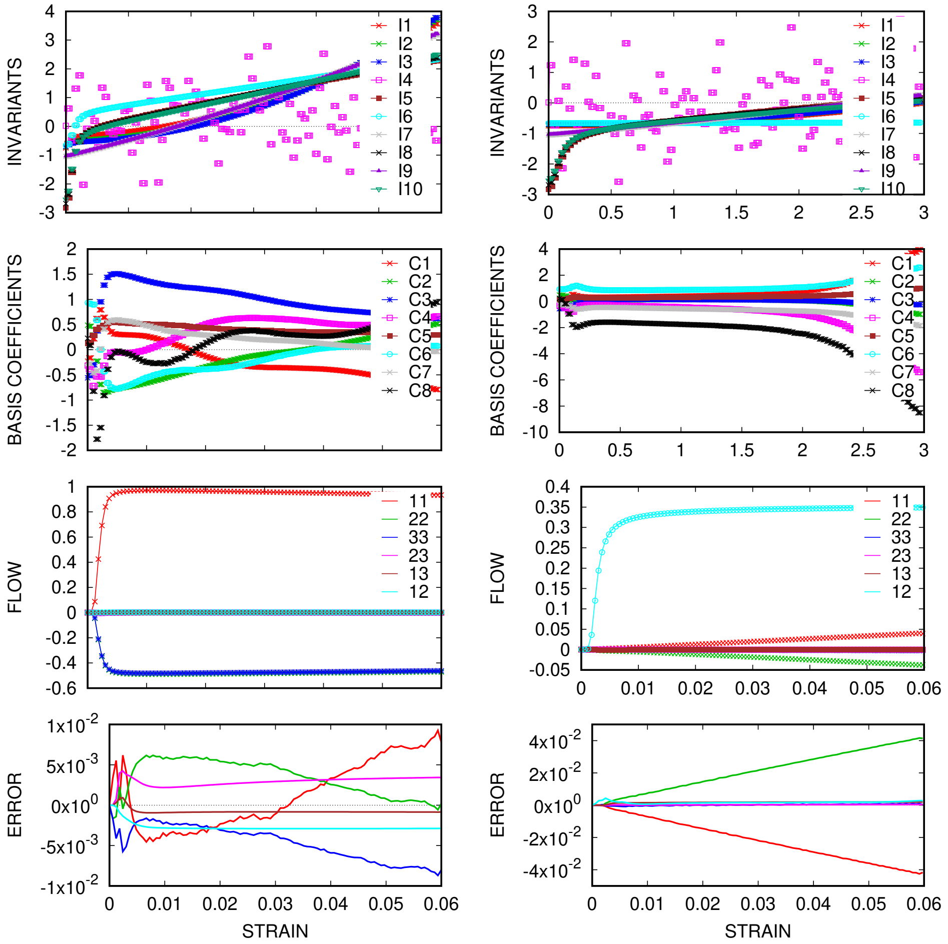


(a) tension

(b) shear

Figure 2: Stress training data: (a) Tension and (b) Shear stress evolution with strain for CP model. Top panels: Scaled input invariants. Second panels: Scaled trained tensor basis coefficient functions. Third panels: Stress $\mathbf{T}$ response (lines: model, points: data). Bottom panel: Error as function of strain scaled by $s_{\mathbf{T}}$

As we already have indications that a training set composed of only tension and shear may be insufficient, we computed the (E3) TBNN and (E_{ij}) component-based NN stress models' response to bimodal stretch $\mathbf{e}_e(\epsilon_1, \epsilon_2) = \epsilon_1 \mathbf{M}_1 + \epsilon_2 \mathbf{M}_2$, where the modes are the CP (tension) training modes $\mathbf{M}_i(\epsilon) = \epsilon\left(\mathbf{e}_i \otimes \mathbf{e}_i - \nu\left(\mathbf{I} - \mathbf{e}_i \otimes \mathbf{e}_i\right)\right)$. Hence, the (tension) training data aligns with axes and here the models represent the data well. Fig. 8 shows that the model responses are significantly different away from the training data, the limits of which are denoted by the white box outline on the contour plots. In particular, the component model does not give a symmetric response, compare Fig. 8a and Fig. 8b, which is to be expected since each component is independently (and imperfectly) trained. Both models have regions of negative stress and large stresses outside the training limits. Fig. 8c shows expectation for the stress from the crystal elasticity underlying the response of the polycrystalline aggregate. The TBNN response has more complex trends likely due to its



(a) tension

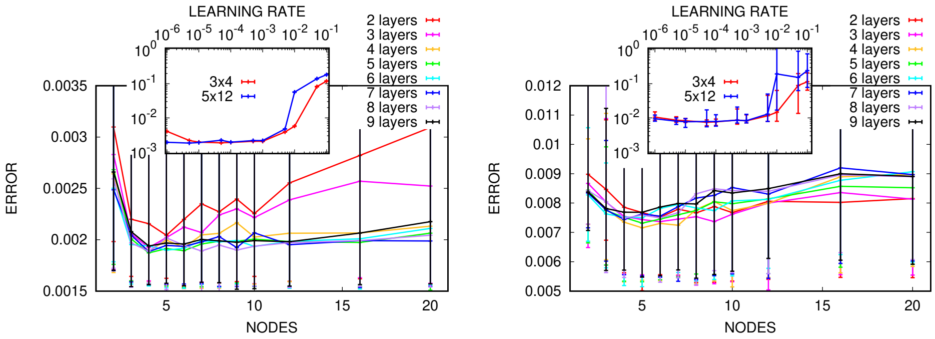
(b) shear

Figure 3: Flow training data: (a) Tension and (b) Shear flow evolution with strain for CP model. Top panels: Scaled input invariants. Second panels: Scaled trained tensor basis coefficient functions. Third panels: Flow $\mathbf{D}_p$ response (lines: model, points: data). Bottom panel: Error as function of strain scaled by $s_{\mathbf{D}_p}$

formulation on invariants and the 11-stress response of the component model arguably represents the expected stress best, albeit with a distinct error in the gradient. This result illustrates that acceptable cross-validation errors along limited training data does not necessarily lead to comparably acceptable interpolation, nor extrapolation, with NNs.

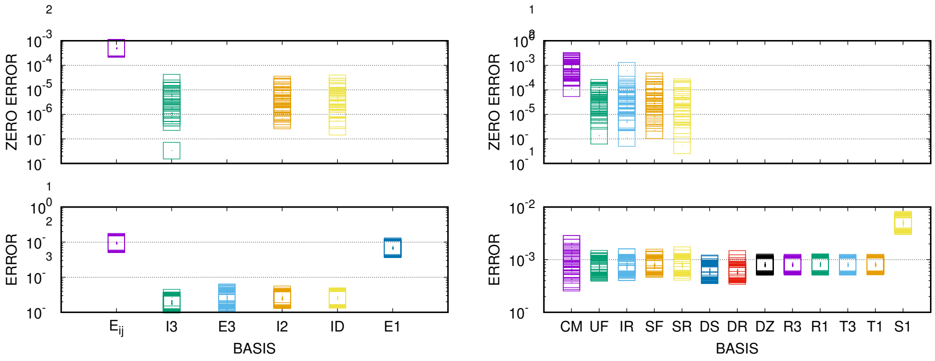
To further investigate how much data and what variety of data is needed to sufficiently train the constitutive models, we employed the simple VP underlying model to generate data for loading modes that are symmetric and monotonic: $\mathbf{F}(t) = t\sum_{i=1}^{3} \lambda_i \mathbf{e}_i \otimes \mathbf{e}_i$. In particular, the training datasets D_n are comprised of n trajectories with 100 state points each. The λ_i for each trajectory were uniformly sampled on the 2-sphere (using minimum energy quadrature points [Sloan and Womersley (2004)]).[7] The testing dataset T consisted

[7] Note the uniform sampling points nest, in the sense that a larger set D_m contains all the points of



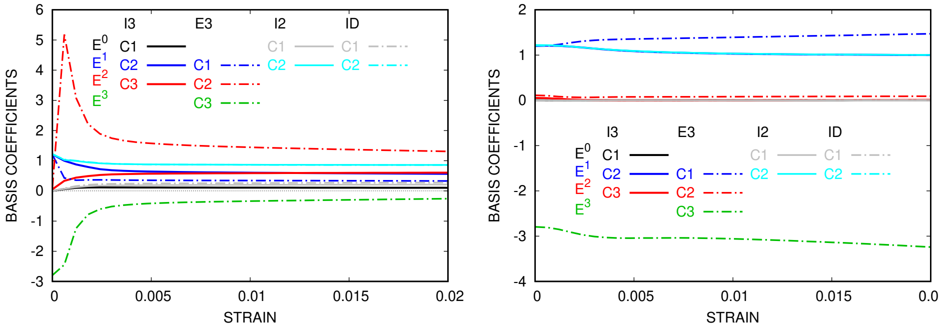
(a) stress model (b) flow model

Figure 4: Network optimization for 150 realizations of full representations for both stress and flow on CP data. Training meta parameter for full representations for (I3) stress and (SF) flow on CP data. Error bars denote min and max of error, and errors are scaled by $s_{\mathbf{T}}$ and $s_{\mathbf{D}_p}$, respectively



(a) stress (b) flow

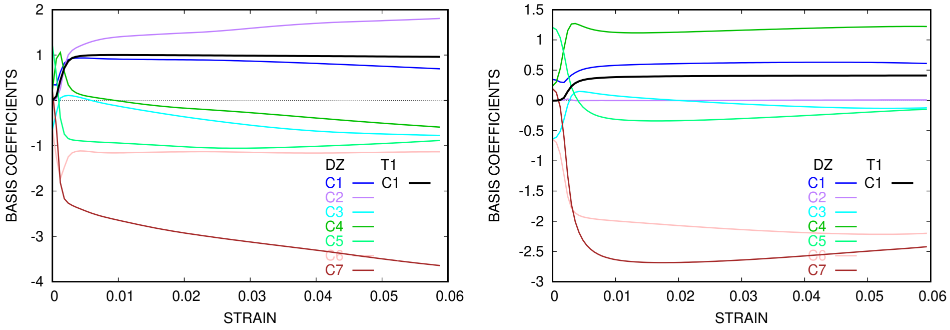
Figure 5: Cross validation error as a function of representation for (a) stress, (b) flow for CP data. Errors are scaled by $s_{\mathbf{T}}$ and $s_{\mathbf{D}_p}$, respectively Refer to Tab. 2 for stress representations and Tab. 3 for the flow representations



(a) tension (b) shear

Figure 6: Stress tensor basis coefficients in shear and tension using different bases

a smaller set D_n, $m > n$.



(a) tension

(b) shear

Figure 7: Flow Tensor basis coefficients in shear and tension using DZ and T1 bases

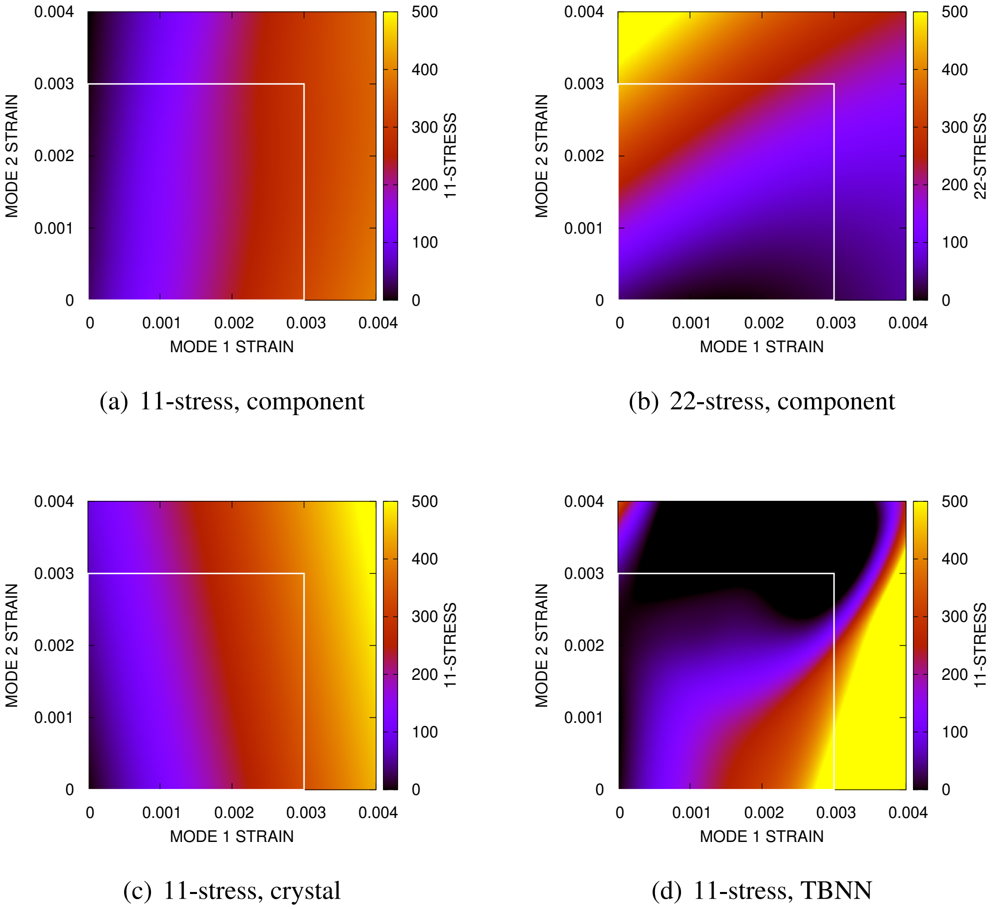
of 10 trajectories given by random samples on the 2-sphere. Fig. 9 shows the change in accuracy of the model relative to the random sample test T set and the information gain with increasing the size of the training D_n dataset. The decreasing errors in Figs. 9a and 9b with more training suggests completeness of the representation, and the slightly higher rate of convergence for the larger network indicates the complexity of the underlying function. Also the variability of the models is decreasing with more data, which gives context for the variability of the models trained only with the CP data. The decrease rate is modest, $\sim n^{-a}$, where $a \in [0.2, 0.5]$ and n is the number of trajectories that each contain 100 points, but the variability in response is also decreasing with more data.

To measure of how much information has been gained by training the NN relative to its untrained state, we use the Kullback-Leibler (KL) divergence:

$$g_j(D_n) = \int p(T|D_n) \log \frac{p(T|D_n)}{p(T)} \mathrm{d}\mathsf{y}_j \tag{39}$$

evaluated with the assistance of standard kernel density estimators. Here D_n is a training set, T is the independent test data, and $p(T|D_n)$ is the probability density function (PDF) of the predictions y_j using an ensemble of models $M_k \in \mathcal{M}$ and the (fixed) data inputs x_j, j indexes the state and prescribed strain, and $p(T) \equiv p(T|D_0)$. In Fig. 9c,d we see that: (a) Both the stress and flow models are steadily differentiating themselves from their untrained state with increased training data, (b) The largest changes appear to occur in the initial increases in training data and yet convergence of the KL divergence is not reached even with D_{64}, and (c) The stress is gaining more information from the low strain data and the flow model is gaining the most information from the post-yield data, which is physically intuitive.

As a prelude to studying the dynamic stability of our plasticity TBNN model, we test the TBNN formulations' sensitivity to noise by randomly perturbing the inputs by 1% using the CP training data. The response to the perturbations is fairly uniform over the range strains (not shown). As Fig. 10 shows, the output variance for most of the models is on-par with the input variance, the exceptions being tied to the presence of the noisy invariant $\operatorname{tr} \boldsymbol{\sigma}$. Clearly, pruning ill-conditioned invariants is crucial for stability.



(a) 11-stress, component

(b) 22-stress, component

(c) 11-stress, crystal

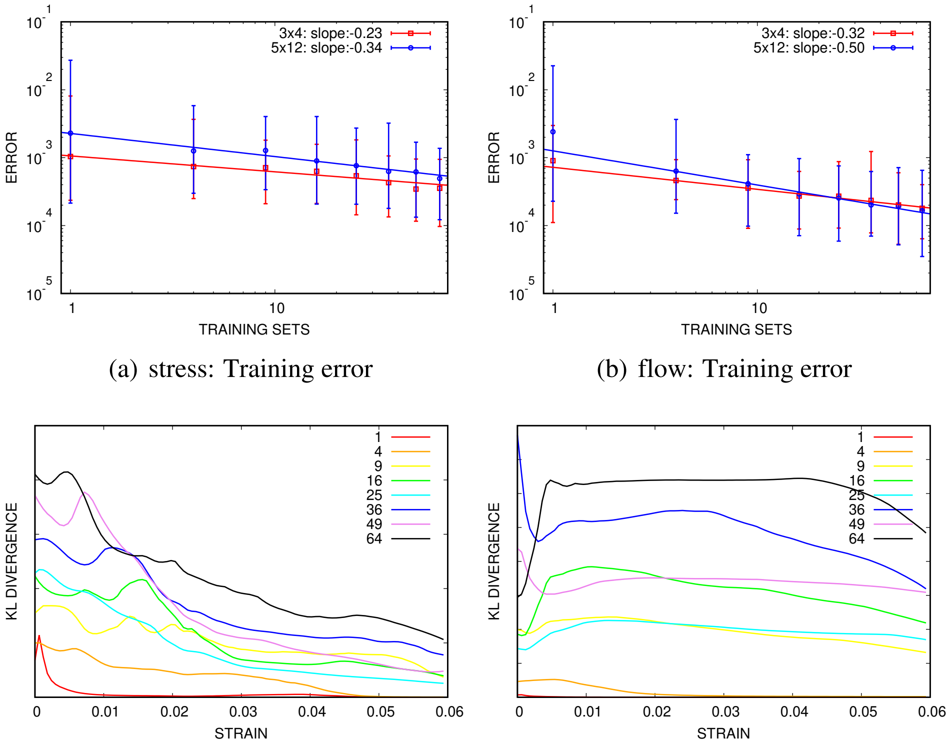
(d) 11-stress, TBNN

Figure 8: Stress response of (a),(b) component E_{ij}, (c) underlying crystal, and (d) TBNN E3 models to bimodal stretch. The TBNN 11-stress (c) and 22-stress (not shown) responses are symmetric across the diagonal. Note white box outlines limit of training data which lies along the axes

4.3 Prediction

Generally speaking, errors in the predictions of the proposed TBNN plasticity models come from: errors in the elastic model, those in the flow rule, and those engendered by the integration scheme. In preliminary work, we integrated the rate given by training data to tune the tolerances of the integration scheme and ensure the integrator error is negligible.

In Fig. 11 we show Lyapunov-like stability tests using a 3×4 E3/5×8 T1 TBNN model trained on a D_{64} dataset from the known closed-form VP model. First, we perturb the initial conditions of state $\mathbf{F}_p(0)$ for a random (monotomic) loading mode $\mathbf{F}(t)$ and compare the response of the underlying model (gray lines) to that of an analytic stress (Eq. (29))/TBNN flow model hybrid (colored) for this ensemble of initial conditions. The TBNN response is on par with that of the true model, albeit with a distinct bias toward higher stress. Second, we compare the same models with a $\mathbf{F}_p(0) = \mathbf{I}$ initial condition but with an imperfect



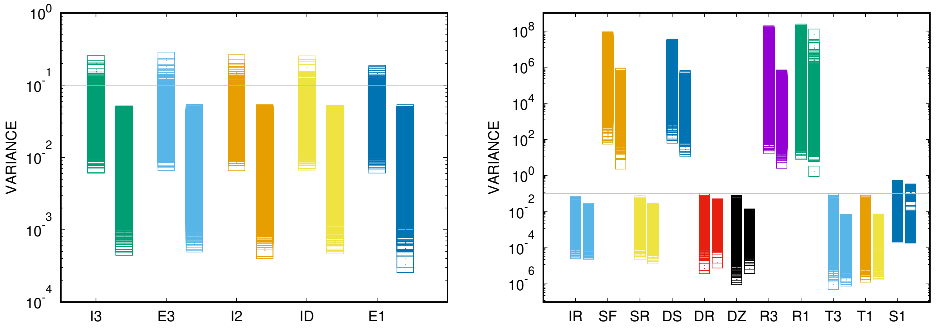
(a) stress: Training error

(b) flow: Training error

(c) stress: K-L divergence

(d) flow: K-L divergence

Figure 9: Error as a function of training data span and network size (a,b), and model information content relative to an uninformed/untrained model (c,d). Note each training set/trajectory has 100 state samples from the VP (known underlying) model and the convergence rates are reported in terms of number training sets (not number of state samples). Also, the error bars reflect the variance in the training errors across the ensemble of NN



(a) variance: stress

(b) variance: flow

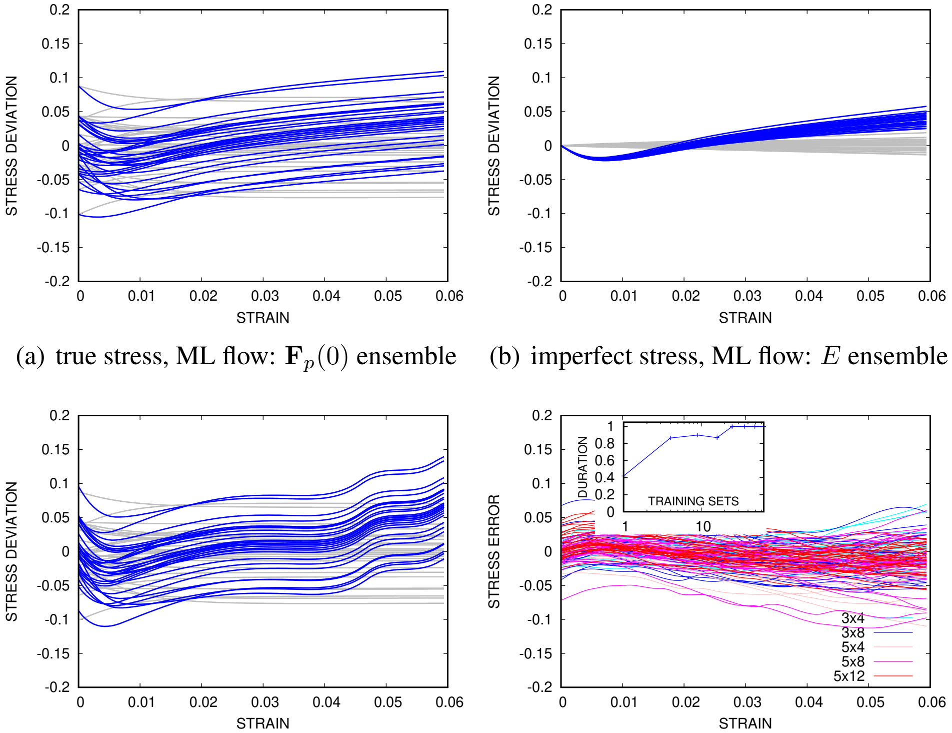
Figure 10: Variance of models in response to 1% Gaussian input noise: (a) Stress, (b) Flow, on-diagonal components on left, off-diagonal components on right

stress model created by randomly perturbing the Young's modulus E of the analytic stress model. Here again, the TBNN response is on par with the true model and yet artifacts in the trajectories are clearly present. Third, we repeated the first investigation with at TBNN with both a ML flow and a ML stress model. The results are largely similar to the response of the TBNN with the true stress model albeit with additional artifacts in trajectories. Lastly, we explore the sensitivity of the trajectory errors to flow model network size. Fig. 11d shows that the trajectory errors for the flow model trained on VP data are relatively insensitive to the NN dimensions. Also, since the variance of the results does not increase with strain, the errors are apparently primarily due to the stress representation in this loading mode. The inset of Fig. 11d demonstrates the necessity of sufficient variety of training data. Here we plot the fraction of the models that stably reach double the training strain. Apparently, in this application, training on at least 25 trajectories is necessary to achieve robust predictions outside the training data.

Lastly, we return to models trained on the tension and shear CP data. Fig. 12 shows the predictions of a TBNN 3×4 E3 stress model with NN flow models of various sizes. Fig. 12a and 12b demonstrate that the predictions are essentially self-consistent with the training data. Also the fanning out of the trajectories is generally consistent with accumulation of errors from integrating an imperfect model. It appears that, as the plastic flow develops, non-smooth transitions occur which make some trajectories jump to paths neighboring the true/training path. Fig. 12c and 12d show the results for bona fide predictions: (c) Illustrates a combined simple shear and tension loading mode, where $[\mathbf{F}(t)]_{11} = (1+t)$, $[\mathbf{F}(t)]_{21} = 1/2t$, and the other directions have traction-free boundary conditions; and (d) Illustrates a non-monotonic tension-then-compression mode at a different rate, $[\mathbf{F}(t)]_{11} = (1+3t)$ for $t \in [0, 0.02]$ and $[\mathbf{F}(t)]_{11} = (1.06 - 3t)$ for $t \in [0.02, 0.06]$. For these modes the results are considerably less stable, especially in the mixed tension-shear mode which points to the stress model being the main issue (as discussed in the previous section). Even the tension phase of the non-monotonic loading leads to decreased stability and accuracy compared to the tension only case, apparently due to the change in strain rate. Lastly, in these modes none of the larger 5×12 network flow models tested were stable, which gives more evidence that the main issue is a lack of sufficient variety in the training data.

5 Discussion

In this work we generalized the TBNN framework to fully take advantage of classical representation theory. By embedding constraints and properties directly in the structure and formulation of the NNs for stress and plastic flow, we were able to reduce the amount of training required for valid models compared to current component-based NN models. The constraints of plasticity phenomenology and the trade-offs between learning and embedding the properties lead to a variety of models and hence to a model selection process. We showed that traditional cross-validation errors are not sufficient for the down-selection process and, for example, stability with respect to perturbation needs to be considered for a viable model. We also illustrated the facts that: Given limited data, the formulations are insensitive to a number of meta-parameters and variables, in particular the selected functional inputs; and, there are trade-offs between model complexity and property preservation, such as preserving zero-stress-at-zero-strain. Using a known underlying data model, we demonstrated that the enhanced TBNN framework can provide robust and accurate predictions given sufficient data. This also supports the speculation that richer training sets may enhance the ability to discriminate between model representations.



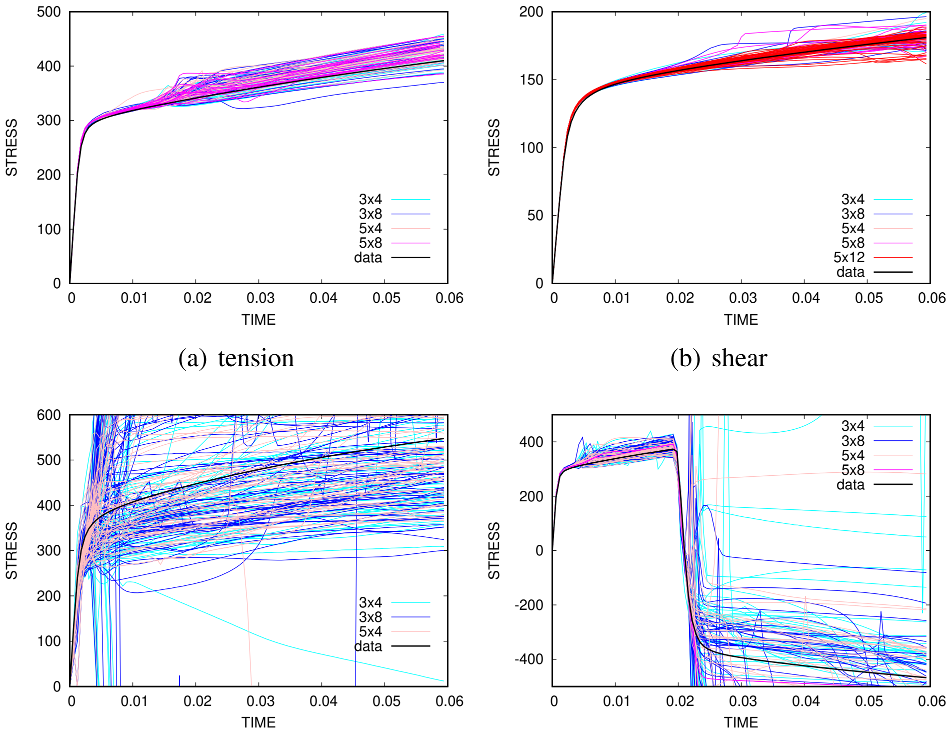
(a) true stress, ML flow: $\mathbf{F}_p(0)$ ensemble (b) imperfect stress, ML flow: E ensemble

(c) ML stress, ML flow: $\mathbf{F}_p(0)$ ensemble (d) ML stress, ML flow: model ensemble

Figure 11: Lyapunov bundle of trajectories for models on VP (known model) data: (a) A perfect stress model and a NN flow model with perturbed initial conditions, (b) Imperfect stress models (modulus E random) and a NN flow model, (c) A NN stress model and a NN flow model with perturbed initial conditions. (d) Ensemble of (unperturbed) NN stress and NN flow models. Deviation is with respect to an unperturbed trajectory, gray lines: Exact model, colored lines: TBNN. Inset of (d) shows the fraction of the models that reach the double duration of the training data as a function of the amount of training data to illustrate the models' stability in extrapolation

In addition, we demonstrated that the tension (and shear) experiments traditionally used in model calibration are likely insufficient to fully train a NN model, as formulated in the TBNN framework or via a component formulation that generally displayed worse performance. Some other practical findings of this work:

- formulation of NNs based on invariants, such as the TBNN, reduce the required amount of training data but also impose non-linear relations between the components of the inputs and output,
- an ensemble of trained models give a truer indication of the performance of a given NN formulation and architecture than a single calibrated model,
- cross-validation errors generally do not discriminate between functionally equivalent constitutive formulations given limited data,



(a) tension

(b) shear

(c) shear+tension

(d) tension,compression

Figure 12: Prediction of TBNN with ML stress and ML flow models trained on CP tension and shear data: (a) Tension, (b) Shear, (c) Combined tension and shear, (d) Tension and then compression. Note loading in (d) is at a different rate from training data

- Good cross-validation error based on a given dataset does not ensure stability since predictions typically visit neighboring trajectories,

- Cross-validation indications of over-fitting by overly large NNs are corroborated by instabilities in predictions,

- Pruning badly conditioned invariants is crucial for stability when neighboring inputs are sampled but inclusion of these inputs does not always affect overall cross-validation errors,

- Enforcing zero-output at zero-input with a higher order basis results in more complex coefficient functions,

- Apparently, training a general invariant formulation on limited data does not result in simplicity in the component functions,

- Rank considerations, such as using the same number of network nodes as input invariants, should only be a starting point since limited data can lead to NN to compress the required number of invariants internally,

- A reduced basis can perform well if the data complexity and coverage is low and is similar to the prediction regime,
- Metrics, such as the KL divergence of the NN weights, can be used to indicate what data is helping inform the particular constitutive functions, and what is not, thus leading to adaptive sampling schemes and design of experiments,
- Common experimental data, *e.g.* tension tests, are likely insufficient to train robust NN models, even accounting for over-fitting, since these tests explore a small fraction of the input space.

Given these results and the lack of a distinctly optimal formulation and network architecture, our best recommendation, given tension and shear data, is to employ a 3×4 E3 stress model that has the same number of inputs as invariants and satisfies the zero-output condition intrinsically, together with a 3×8 T1 flow model which is apparently compressing the space of input invariants but requires depth to fully represent the phenomenology.

We speculate a profitable next step would be to use dropout layers and other pruning methods, such as compressive sensing/L1 regularization, to reduce the model complexity and hopefully increase stability for a given network's dimensions. Such an approach may also help identify better performing representations by eliminating weak connections and their associated basis coefficients. Along the same lines, the present work indicates that assessing which coefficient functions are informed by the data, and how the addition of more complex trajectories that fill the input space affects training, may lead to more optimal representations. The source of this additional data could be multi-axial experiments or full-field data, such as from digital image correlation. Lastly, developing an implicit time integrator based on derivatives of the neural network is another logical step in the development of viable NN based constitutive models.

Acknowledgement: We relied on Albany (https://github.com/gahansen/Albany), Dream3d (http://dream3d.bluequartz.net), Lasagne (https://lasagne.readthedocs.io/en/latest/), and Theano (http://deeplearning.net/software/theano/), to accomplish this work. We also wish to thank J. Ling for helping to upgrade the TBNN package to support the modeling approach described in this paper. Clay Sanders acknowledges support from the United States Department of Energy through the Computational Sciences Graduate Fellowship (DOE CSGF) under grant number: DE-FG02-97ER25308. This work was supported by the LDRD program at Sandia National Laboratories, and its support is gratefully acknowledged. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

Adeli, H. (2001): Neural networks in civil engineering: 1989-2000. *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 2, pp. 126-142.

Adeli, H.; Yeh, C. (1989): Perceptron learning in engineering design. *Computer-Aided Civil and Infrastructure Engineering*, vol. 4, no. 4, pp. 247-256.

Alharbi, H. F.; Kalidindi, S. R. (2015): Crystal plasticity finite element simulations using a database of discrete fourier transforms. *International Journal of Plasticity*, vol. 66, pp. 71-84.

Arnold, V. I. (2013): *Mathematical Methods of Classical Mechanics*, volume 60. Springer, Berlin.

Asgharzadeh, A.; Aval, H. J.; Serajzadeh, S. (2016): A study on flow behavior of AA5086 over a wide range of temperatures. *Journal of Materials Engineering and Performance*, vol. 25, no. 3, pp. 1076-1084.

Bartók, A. P.; Csányi, G. (2015): Gaussian approximation potentials: A brief tutorial introduction. *International Journal of Quantum Chemistry*, vol. 115, no. 16, pp. 1051-1057.

Bessa, M.; Bostanabad, R.; Liu, Z.; Hu, A.; Apley, D. W. et al. (2017): A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality. *Computer Methods in Applied Mechanics and Engineering*, vol. 320, pp. 633-667.

Bishop, J.; Hill, R. (1951): A theoretical derivation of the plastic properties of a polycrystalline face-centred metal. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 42, no. 334, pp. 1298-1307.

Bishop, J.; Hill, R. (1951): XLVI. A theory of the plastic distortion of a polycrystalline aggregate under combined stresses. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 42, no. 327, pp. 414-427.

Bobbili, R.; Ramakrishna, B.; Madhu, V.; Gogia, A. (2015): Prediction of flow stress of 7017 aluminium alloy under high strain rate compression at elevated temperatures. *Defence Technology*, vol. 11, no. 1, pp. 93-98.

Boehler, J.-P. (1987): Representations for isotropic and anisotropic non-polynomial tensor functions. *Applications of Tensor Functions in Solid Mechanics*, pp. 31-53. Springer, Berlin.

Cheu, R. L.; Ritchie, S. G. (1995): Automated detection of lane-blocking freeway incidents using artificial neural networks. *Transportation Research Part C: Emerging Technologies*, vol. 3, no. 6, pp. 371-388.

Clevert, D.-A.; Unterthiner, T.; Hochreiter, S. (2015): Fast and accurate deep network learning by exponential linear units (ELUs). *arXiv preprint: 1511.07289.*

Coleman, B. D.; Noll, W. (1963): The thermodynamics of elastic materials with heat conduction and viscosity. *Archive for Rational Mechanics and Analysis*, vol. 13, no. 1, pp. 167-178.

Dawson, P. R. (2000): Computational crystal plasticity. *International Journal of Solids and Structures*, vol. 37, no. 1-2, pp. 115-130.

Desu, R. K.; Guntuku, S. C.; Aditya, B.; Gupta, A. K. (2014): Support vector regression based flow stress prediction in austenitic stainless steel 304. *Procedia Materials Science*, vol. 6, pp. 368-375.

Doyle, T.; Ericksen, J. L. (1956): Nonlinear elasticity. *Advances in Applied Mechanics*, vol. 4, pp. 53-115. Elsevier, Amsterdam.

Duraisamy, K.; Zhang, Z. J.; Singh, A. P. (2015): New approaches in turbulence and transition modeling using data-driven techniques. *53rd AIAA Aerospace Sciences Meeting*, pp. 1284.

Furukawa, T.; Yagawa, G. (1998): Implicit constitutive modelling for viscoplasticity using neural networks. *International Journal for Numerical Methods in Engineering*, vol. 43, no. 2, pp. 195-219.

Ghaboussi, J.; Pecknold, D. A.; Zhang, M.; Haj-Ali, R. M. (1998): Autoprogressive training of neural network constitutive models. *International Journal for Numerical Methods in Engineering*, vol. 42, no. 1, pp. 105-126.

Goodman, R.; Wallach, N. R. (1998): *Representations and Invariants of the Classical Groups*, vol. 68. Cambridge University Press, Cambridge.

Goodman, R.; Wallach, N. R. (2009): *Symmetry, Representations, and Invariants*, vol. 255. Springer, Berlin.

Gurtin, M. E. (1982): *An Introduction to Continuum Mechanics*, vol. 158. Academic Press, Cambridge, MA.

Gurtin, M. E.; Fried, E.; Anand, L. (2010): *The Mechanics and Thermodynamics of Continua*. Cambridge University Press, Cambridge.

Hajela, P.; Berke, L. (1991): Neurobiological computational models in structural analysis and design. *Computers and Structures*, vol. 41, no. 4, pp. 657-667.

Haupt, P.; Tsakmakis, C. (1989): On the application of dual variables in continuum mechanics. *Continuum Mechanics and Thermodynamics*, vol. 1, no. 3, pp. 165-196.

Haupt, P.; Tsakmakis, C. (1996): Stress tensors associated with deformation tensors via duality. *Archives of Mechanics*, vol. 48, no. 2, pp. 347-384.

Hill, R. (1968): On constitutive inequalities for simple materials-I. *Journal of the Mechanics and Physics of Solids*, vol. 16, no. 4, pp. 229-242.

Itskov, M. (2007): *Tensor Algebra and Tensor Analysis for Engineers*. Springer, Berlin.

Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D. et al. (2013): Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, vol. 1, no. 1.

Johnson, G. C.; Bammann, D. J. (1984): A discussion of stress rates in finite deformation problems. *International Journal of Solids and Structures*, vol. 20, no. 8, pp. 725-737.

Khotanzad, A.; Hong, Y. H. (1990): Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489-497.

Kirchdoerfer, T.; Ortiz, M. (2016): Data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, vol. 304, pp. 81-101.

Koh, P. W.; Liang, P. (2017): Understanding black-box predictions via influence functions. *International Conference on Machine Learning*.

Kroner, E. (1961): On the plastic deformation of polycrystals. *Acta Metallurgica*, vol. 9, no. 2, pp. 155-161.

Lee, E. H. (1969): Elastic-plastic deformation at finite strains. *Journal of Applied Mechanics*, vol. 36, no. 1, pp. 1-6.

Li, H.-Y.; Wang, X.-F.; Wei, D.-D.; Hu, J.-D.; Li, Y.-H. (2012): A comparative study on modified Zerilli–Armstrong, Arrhenius-type and artificial neural network models to predict high-temperature deformation behavior in T24 steel. *Materials Science and Engineering: A*, vol. 536, pp. 216-222.

Lin, Y.; Zhang, J.; Zhong, J. (2008): Application of neural networks to predict the elevated temperature flow behavior of a low alloy steel. *Computational Materials Science*, vol. 43, no. 4, pp. 752-758.

Ling, J.; Jones, R.; Templeton, J. (2016): Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, vol. 318, pp. 22-35.

Ling, J.; Kurzawski, A.; Templeton, J. (2016): Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, vol. 807, pp. 155-166.

Ling, J.; Templeton, J. (2015): Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty. *Physics of Fluids*, vol. 27, no. 8.

Lowe, D. G. (1999): Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pp. 1150-1157.

Lubarda, V. A. (2004): Constitutive theories based on the multiplicative decomposition of deformation gradient: Thermoelasticity, elastoplasticity, and biomechanics. *Applied Mechanics Reviews*, vol. 57, no. 2, pp. 95-108.

Lubliner, J. (2008): *Plasticity Theory*. Dover, Mineola, NY.

Mandel, J. (1965): Généralisation de la théorie de plasticité de WT Koiter. *International Journal of Solids and Structures*, vol. 1, no. 3, pp. 273-295.

Marsden, J. E.; Hughes, T. J. (1983): *Mathematical Foundations of Elasticity*. Prentice-Hall, Upper Saddle River, NJ.

Milano, M.; Koumoutsakos, P. (2002): Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, vol. 182, no. 1, pp. 1-26.

Nielsen, M. A. (2015): *Neural Networks and Deep Learning*. Determination Press, San Francisco.

Olver, P. J. (2000): *Applications of Lie Groups to Differential Equations*, volume 107. Springer, Berlin.

Pipkin, A.; Wineman, A. (1963): Material symmetry restrictions on non-polynomial constitutive equations. *Archive for Rational Mechanics and Analysis*, vol. 12, no. 1, pp. 420-426.

Raccuglia, P.; Elbert, K. C.; Adler, P. D.; Falk, C.; Wenny, M. B. et al. (2016): Machine-learning-assisted materials discovery using failed experiments. *Nature*, vol. 533, no. 7601, pp. 73.

Rivlin, R.; Smith, G. (1969): Orthogonal integrity basis for N symmetric matrices. *Contributions to Mechanics: Markus Reiner Eightieth Anniversary Volume*, pp. 121.

Rivlin, R. S. (1955): Further remarks on the stress-deformation relations for isotropic materials. *Journal of Rational Mechanics and Analysis*, vol. 4, pp. 681-702.

Rivlin, R. S.; Ericksen, J. L. (1955): Stress-deformation relations for isotropic materials. *Journal of Rational Mechanics and Analysis*, vol. 4, pp. 323-425.

Rivlin, R. S.; Smith, G. F. (1997): On identities for 3×3 matrices. *Collected Papers of RS Rivlin*, pp. 1550-1558. Springer, Berlin.

Rizzi, F.; Khalil, M.; Jones, R. E.; Templeton, J. A.; Ostien, J. T. et al. (2018): Bayesian modeling of inconsistent plastic response due to material variability. *Data Analysis, Statistics and Probability*.

Roters, F.; Eisenlohr, P.; Hantcherli, L.; Tjahjanto, D. D.; Bieler, T. R. et al. (2010): Overview of constitutive laws, kinematics, homogenization and multiscale methods in crystal plasticity finite-element modeling: Theory, experiments, applications. *Acta Materialia*, vol. 58, no. 4, pp. 1152-1211.

Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. (1986): Learning representations by back-propagating errors. *Nature*, vol. 323, no. 6088, pp. 533-538.

Saal, J. E.; Kirklin, S.; Aykol, M.; Meredig, B.; Wolverton, C. (2013): Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD). *JOM*, vol. 65, no. 11, pp. 1501-1509.

Salloum, M.; Templeton, J. A. (2014): Inference and uncertainty propagation of atomistically informed continuum constitutive laws, part 2: Generalized continuum models based on gaussian processes. *International Journal for Uncertainty Quantification*, vol. 4, no. 2.

Sattinger, D. H.; Weaver, O. L. (2013): *Lie Groups and Algebras with Applications to Physics, Geometry, and Mechanics*, vol. 61. Springer, Berlin.

Seth, B. (1961): Generalized strain measure with applications to physical problems. Technical report, Wisconsin University-Madison, Mathematics Research Center, 1961.

Shaughnessy, M.; Jones, R. (2016): Efficient use of an adapting database of ab initio calculations to generate accurate Newtonian dynamics. *Journal of Chemical Theory and Computation*, vol. 12, no. 2, pp. 664-675.

Shwartz-Ziv, R.; Tishby, N. (2017): Opening the black box of deep neural networks via information. *Machine Learning*.

Simo, J.; Hughes, T. (1998): *Computational Inelasticity*. Springer, New York.

Sloan, I. H.; Womersley, R. S. (2004): Extremal systems of points and numerical integration on the sphere. *Advances in Computational Mathematics*, vol. 21, no. 1-2, pp. 107-125.

Smith, G. (1965): On isotropic integrity bases. *Archive for Rational Mechanics and Analysis*, vol. 18, no. 4, pp. 282-292.

Smith, G.; Rivlin, R. (1964): Integrity bases for vectors-the crystal classes. *Archive for Rational Mechanics and Analysis*, vol. 15, no. 3, pp. 169-221.

Smith, G.; Rivlin, R. S. (1957): The anisotropic tensors. *Quarterly of Applied Mathematics*, vol. 15, no. 3, pp. 308-314.

Smith, G.; Rivlin, R. S. (1957): Stress-deformation relations for anisotropic solids. *Archive for Rational Mechanics and Analysis*, vol. 1, no. 1, pp. 107-112.

Smith, J.; Xiong, W.; Yan, W.; Lin, S.; Cheng, P. et al. (2016): Linking process, structure, property, and performance for metal-based additive manufacturing: Computational approaches with experimental support. *Computational Mechanics*, vol. 57, no. 4, pp. 583-610.

Spencer, A. (1971): Part III. Theory of invariants. *Continuum Physics*, vol. 1, pp. 239-353.

Spencer, A. (1982): The formulation of constitutive equation for anisotropic solids. *Mechanical Behavior of Anisotropic Solids/Comportment Méchanique des Solides Anisotropes*, pp. 3-26. Springer, Berlin.

Spencer, A. (1987): Isotropic polynomial invariants and tensor functions. *Applications of Tensor Functions in Solid Mechanics*, pp. 141-169. Springer, Berlin.

Spencer, A.; Rivlin, R. (1962): Isotropic integrity bases for vectors and second-order tensors. *Archive for Rational Mechanics and Analysis*, vol. 9, no. 1, pp. 45-63.

Spencer, A. J. M.; Rivlin, R. (1958): Finite integrity bases for five or fewer symmetric 3×3 matrices. *Archive for Rational Mechanics and Analysis*, vol. 2, no. 1, pp. 435-446.

Spencer, A. J. M.; Rivlin, R. S. (1958): The theory of matrix polynomials and its application to the mechanics of isotropic continua. *Archive for Rational Mechanics and Analysis*, vol. 2, no. 1, pp. 309-336.

Svendsen, B. (1994): On the representation of constitutive relations using structure tensors. *International Journal of Engineering Science*, vol. 32, no. 12, pp. 1889-1892.

Szabo, L.; Balla, M. (1989): Comparison of some stress rates. *International Journal of Solids and Structures*, vol. 25, no. 3, pp. 279-297.

Taylor, G. I. (1934): The mechanism of plastic deformation of crystals. part i. theoretical. *Proceedings of the Royal Society of London. Series A*, vol. 145, no. 855, pp. 362-387.

Theocaris, P.; Panagiotopoulos, P. (1993): Neural networks for computing in fracture mechanics. methods and prospects of applications. *Computer Methods in Applied Mechanics and Engineering*, vol. 106, no. 1-2, pp. 213-228.

Tracey, B. D.; Duraisamy, K.; Alonso, J. J. (2015): A machine learning strategy to assist turbulence model development. *53rd AIAA Aerospace Sciences Meeting*, pp. 1287.

Truesdell, C.; Noll, W. (2004): *The Non-linear Field Theories of Mechanics*, pp. 1-579. Springer, Berlin.

Versino, D.; Tonda, A.; Bronkhorst, C. A. (2017): Data driven modeling of plastic deformation. *Computer Methods in Applied Mechanics and Engineering*, vol. 318, pp. 981-1004.

Wang, C.-C. (1969): On a general representation theorem for constitutive relations. *Archive for Rational Mechanics and Analysis*, vol. 33, no. 1, pp. 1-25.

Wang, C.-C. (1970): A new representation theorem for isotropic functions: An answer to professor GF Smith's criticism of my papers on representations for isotropic functions. *Archive for Rational Mechanics and Analysis*, vol. 36, no. 3, pp. 166-197.

Wang, J.-X.; Wu, J.; Ling, J.; Iaccarino, G.; Xiao, H. (2017): A comprehensive physics-informed machine learning framework for predictive turbulence modeling. *Fluid Dynamics*.

Werbos, P. (1974): *Beyond Regression: New Tools for Prediction and Analysis in the Behavior Science (Ph. D. Thesis).* Harvard University. Cambridge, MA.

Wineman, A. S.; Pipkin, A. (1964): Material symmetry restrictions on constitutive equations. *Archive for Rational Mechanics and Analysis*, vol. 17, no. 3, pp. 184-214.

Zhang, J.; Rychlewski, J. (1990): Structural tensors for anisotropic solids. *Archives of Mechanics*, vol. 42, no. 3, pp. 267-277.

Zheng, Q.-S. (1994): Theory of representations for tensor functions–a unified invariant approach to constitutive equations. *Applied Mechanics Reviews*, vol. 47, no. 11, pp. 545-587.