Granular Matter (2022) 24:1
https://doi.org/10.1007/s10035-021-01137-y

ORIGINAL PAPER

Data-driven discovery of interpretable causal relations for deep learning material laws with uncertainty propagation

Xiao Sun[1] · Bahador Bahmani[2] · Nikolaos N. Vlassis[2] · WaiChing Sun[2] · Yanxun Xu[1]

Received: 13 November 2020 / Accepted: 17 June 2021

Abstract
This paper presents a computational framework that generates ensemble predictive mechanics models with uncertainty quantification (UQ). We first develop a causal discovery algorithm to infer causal relations among time-history data measured during each representative volume element (RVE) simulation through a directed acyclic graph. With multiple plausible sets of causal relationships estimated from multiple RVE simulations, the predictions are propagated in the derived causal graph while using a deep neural network equipped with dropout layers as a Bayesian approximation for UQ. We select two representative numerical examples (traction-separation laws for frictional interfaces, elastoplasticity models for granular assembles) to examine the accuracy and robustness of the proposed causal discovery method for the common material law predictions in civil engineering applications.

Keywords Causal discovery · Granular materials · Knowledge graph · Path-dependent responses · Soil mechanics

1 Introduction

Computer simulations for mechanics problems often require material (constitutive) laws that replicate the local constitutive responses of the materials. These material laws can be used to replicate the responses of an interface (e.g., traction-separation laws or cohesive zone models) or bulk materials (e.g., elastoplasticity models for solids, porosity-permeability relationship and water retention curve). A computer model is then completed by incorporating these local constitutive laws into a discretized form of balance principles (balance of mass, linear momentum and energy) where discretized numerical solutions can be sought by a proper solver.

Constitutive laws, such as stress-strain relationship for bulk materials, traction-separation laws for interface, porosity-permeability for porous media, are often derived following a set of axioms and rules [82]. In these handcrafted models, phenomenological observations are incorporated into constitutive laws (e.g., critical state theory for soil mechanics [8, 55, 68, 75], void-growth theory for ductile damage [26]). While the earlier simpler models are often amended by newer and more comprehensive models [13] in order to improve the performance (e.g. accuracy, more realistic interpretation of mechanisms), these improvements are often a trade-off that may unavoidably increase the number of parameters, leading to increasing difficulty for the calibration, verification and validation processes, as well as the uncertainty quantification (UQ) [6, 7, 10, 13, 89, 91].

The rise of big data and the great promises of machine learning have led to a new generation of approaches that either bypass the usages of constitutive laws via model-free data-driven methods (e.g., [2, 28, 38, 39]) or replace parts of the modeling efforts/components with models generated from supervised learning (e.g., [19, 43, 48, 51, 83, 84, 88, 90, 99]). However, one critical issue of these machine learning and data-driven approaches is the lack of sufficient interpretability of predictions. While there is no universally accepted definition of interpretability, we will herein employ the definition used in [53] which refers interpretability as the degree to which a human can understand the cause of the prediction.

This article is part of the topical collection: Physics-informed artificial intelligence for granular matter.

✉ WaiChing Sun

✉ Yanxun Xu
yanxun.xu@jhu.edu

[1] Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, USA

[2] Department of Civil Engineering and Engineering Mechanics, Columbia University, New York, NY, USA

Published online: 02 November 2021

One possible way to boost the interpretability is to introduce a proper medium to represent knowledge that can be understood by human [80]. Causal graph, also known as causal Bayesian network, is one such medium in which the causal relations among different entities are mathematically represented by a directed graph. In the application of computational mechanics [30, 88, 90] have derived a mathematical framework to decompose a complex prediction task into multiple easier predictions represented by subgraphs within graphs. More recent work such as [89, 92] introduce a deep reinforcement learning approach that employs the Monte Carlo Tree Search (MCTS) to assemble a directed graph that generates a sequence of interconnected predictions of physical quantities to emulate a hand-crafted constitutive law. However, these directed graphs are generated to optimize a given performance metric (e.g. accuracy, calculation speed and forward prediction robustness), but not necessarily reveal the underlying causal relations among physical quantities.

Discovering causal relations from observational data is an important problem for many fields of science, such as social science [56], finance [23], and biomedicine [70]. The standard way to discover causality is through randomized controlled experiments. However, conducting such experiments can be either impractical, unethical, and/or very expensive in many disciplines [96, 97]. For mechanics problems, the major issues include the time and labor cost for physical experiments, the lack of facilities or equipment to complete the required tests and the difficulties to obtain specimens [54, 63, 95]. As a result, an alternative approach, which is adopted in this study, is to use sub-scale simulations as the digital representation that generates auxiliary data sets to build material laws or forecast engine for the macroscopic material responses [15, 47, 49, 90]. Classic methods for causal discovery are based on probabilistic graphical modeling [61], the structure of which is a directed acyclic graph (DAG) with nodes representing random variables and edges representing conditional dependencies between variables. Learning a DAG from observational data is highly challenging since the number of possible DAGs is super-exponential to the number of nodes. There are two main approaches for causal discovery: the constraint-based approach and the score-based approach. The constraint-based approach aims to recover a Markov equivalence class through inferring conditional indep3endence relationships among the variables, and the resulting Markov equivalence class may contain multiple DAGs that indicate the same conditional independence relationships [11, 42]. On the other hand, the score-based approach uses a scoring function, such as the Bayesian Information Criterion (BIC), to search for the DAG that best fits the data [29, 33].

In this paper, we aim to discover causal relations that can explain the underlying mechanism of a history-dependent macroscopic constitutive law upscaled from direct numerical simulations at the meso-scale. The most common method for constructing the causal relations from time-series data is Granger causality [24], which assumes a number of lagged effects and analyzes the data in a unit no more than that number of lags. See [66] for a review of causal discovery methods on time-series data. However, most of these causal discovery methods assume that the data are generated from a stationary process, meaning that the data are generated by a distribution that does not change with time. Such an assumption does not hold in many physical processes, in which the mechanisms or parameters in the causal model may change over time. Several methods have been proposed recently to tackle time-varying causal relations in non-stationary processes [22, 34, 36]. However, they either assume linear causal models, or does not offer the flexibility of incorporating known physical knowledge, limiting their applicability to the nonlinear path-dependent relations in learning material constitutive laws.

In this work, we offer two major innovations. First, we introduce a new decoupled discovery/training approach where the discovery of causal relations represented by a DAG is enabled by a causal discovery algorithm that deduces plausible causal relations from non-stationary time series data and incorporates known physical knowledge. Second, we leverage the obtained causal graph as the representation of mechanics knowledge and adopt a Bayesian approximation using the dropout layer technique [20] to propagate epistemic uncertainty in the causal graph and generate quantitative predictions with uncertainty quantification.

The rest of the paper is organized as follows. Section 2 first introduces the two data sets (learning traction-separation law and hypo-plasticity of granular materials) used for our numerical experiments. This is followed by the description of theory and implementation of the proposed causal discovery algorithm used to deduce the causal relations from non-stationary time series data (Sect. 3). The setup of the deep neural network model for the prediction tasks and the uncertainty propagation are included in Sects. 4 and 5 respectively. The proposed framework is then tested against two numerical experiments (Sect. 6), which is then followed by the conclusion.

2 Causal relations and constitutive laws

As demonstrated in previous studies such as [30, 83, 88–90, 92], the relationships in a constitutive model can be represented by a network of unidirectional information flow, i.e., a DAG $G = (\boldsymbol{V}, \boldsymbol{E})$ where $\boldsymbol{V}$ represents a vertex set and $\boldsymbol{E}$ denotes an edge set. With appropriate assumptions that will be discussed later, the DAG can be identified as a causal graph [61]. The causal relations are not only useful

to explain the underlying mechanism of a process but also provide us a basis to formulate multi-step transfer learning to predict constitutive responses. This strategy can be beneficial because one can leverage more data gathered from physical numerical experiments to train the prediction model. For instance, while a black-box prediction of stress-strain curves only leverages the stress-strain pair for supervised learning, the introduction of knowledge graphs may introduce multiple supervised learning tasks where measurements of porosity, fabric tensors or any other physical and geometrical attributes measured during the experiments can be leveraged to improve the training. For completeness, we briefly describe the procedure to consider the data set as vertex sets in graphs and the causal discovery process used to create the directed edge set in a knowledge graph through two examples.

Note that many of the physical quantities that become the vertices in the knowledge graphs are graph metrics obtained from analyzing the connectivity topology of the granular system. For brevity, we will not provide a review on the applications of graph theory for granular matter here. Interested readers may refer to Appendix 2 for the definitions of the graph metrics and [1, 41, 58, 67, 81, 86] and [83] for reviews on the graph theory applied to particulate and granular systems.

2.1 Dataset for traction-separation law

In the first example, our goal is to conduct a numerical experiment to verify whether the causal discovery algorithm is able to re-discover the well-known causal relation that links the plastic dilatancy and contraction to the frictional behaviors [62, 69] with a small data set.

Following [88–90, 92], we consider the vertex set consists of five elements, the displacement jump/separation $\boldsymbol{U}$, the traction $\boldsymbol{T}$, and three geometric measures, i.e.,

1. Displacement jump $\boldsymbol{U}$, the relative displacement of an interface of two in-contact bodies.
2. Porosity ϕ, the ratio between the volume of the void and the total volume of RVE.
3. Coordination number (averaged) $CN = N_{\text{contact}}/N_{\text{particle}}$ where N_{contact} is the number of particle contacts and $N_{particle}$ is the number of particles in the RVE.
4. Fabric tensor $\mathbf{A}_f = \frac{1}{N_{\text{contact}}} \sum_{c=1}^{N_{\text{contact}}} \mathbf{n}^c \otimes \mathbf{n}^c$ where $\mathbf{n}^c$ is the normal vector of a particle contact c in the RVE. The symbol '$\otimes$' denotes a juxtaposition of two vectors (e.g., $\boldsymbol{a} \otimes \boldsymbol{b} = a_i b_j$) or two symmetric second order tensors [e.g., $(\boldsymbol{\alpha} \otimes \boldsymbol{\beta})_{ijkl} = \alpha_{ij}\beta_{kl}$].
5. Traction $\boldsymbol{T}$, the traction vector acts on the interface.

To generate a machine learning based traction-separation law, we identify the displacement jump as the root and the traction as the leaf of the causal graph. The causal graph is a DAG $G = (\boldsymbol{V}, \boldsymbol{E})$ where $\boldsymbol{V}$ is the set consisting of the physical quantities $\boldsymbol{U}, \phi, CN, \boldsymbol{A}_f$ and $\boldsymbol{T}$. Meanwhile, $\boldsymbol{E} \subseteq \boldsymbol{V} \times \boldsymbol{V}$ is a set of directed edges that connect any two elements from $\boldsymbol{V}$, and $\boldsymbol{E}$ is determined from the causal discovery algorithm outlined in Sect. 3.

The dataset is generated using an open-source code YADE. In total, there are 100 traction-separation law simulations run with different loading paths performed on the same RVE. This RVE consists of spherical particles with radii between 1 ± 0.3 mm with uniform distribution. The RVE has a height of 20 mm in the normal direction of the frictional surface and is initially consolidated to an isotropic pressure of 10 MPa. The inter-particle interaction is controlled by Cundall's elastic-frictional contact model [12] with an inter-particle elastic modulus of $E_{eq} = 1\,\text{GPa}$, a ratio between shear and normal stiffness of $k_s/k_n = 0.3$, a frictional angle of $\phi = 30°$, a density $\rho = 2600\,\text{kg/m}^3$, and a Cundall damping coefficient $\alpha_{\text{damp}} = 0.2$. For brevity, the generation and setup of the simulations are not included in this paper. Interested readers please refer to [89] for more information. The data required to replicate the results of this paper and for 3rd-party validation can be found in the Mendeley Data repository [76].

2.2 Dataset for hypo-plasticity of granular materials

While the first data set is used to determine whether the causal graph algorithm may re-discover known physical relations in the literature, the second problem is designed to test whether the causal graph algorithm may successfully investigate new plausible causal relations not known *a prior* in the literature.

For this purpose, we run 60 discrete element simulations and use 30 of them for calibrations and 30 for blind forward predictions. In addition to the conventional microstructural attributes (e.g., porosity and fabric tensor) typically used for hand-crafted constitutive laws [8, 14, 50, 55, 75], we have also recorded the evolution of the particle contact pairs in each incremental time step of the discrete element simulations. The particle contact connectivity is itself an undirected graph $G_{\text{contact}} = (\boldsymbol{V}_{\text{particle}}, \boldsymbol{E}_{\text{contact}})$ where $\boldsymbol{V}_{\text{particle}}$ is the set of particles and $\boldsymbol{E}_{\text{contact}}$ is the set of particle contacts, one for each contact between two contacting convex particle represented. They are undirected edges. To facilitate new discovery, we compute 15 different graph metrics of G_{contact} (see Appendix 2 for definition) that have not been used for composing constitutive laws and see if (1) whether the causal discovery algorithm may discover causal relations among these new physical quantities and (2) whether the

new discovery helps improve the accuracy, robustness and consistency of the forward predictions enabled by neural networks trained according to the discovered causal relations.

In total, there are 11 types of time-history data in which 3 of them are second-order tensors (strain, stress and the strong fabric tensor), and the rest are scalar (porosity, coordination number graph density, graph local efficiency, graph average clustering, graph degree assortativity coefficient, graph transitivity and graph clique number). As such there are 11 elements in the vertex set and the goal of the causal discovery is to establish the edge set to complete the causal graph. A sequence of supervised learning is then used to generate predictions via deep learning.

3 Causal discovery and knowledge graph constructions

3.1 Notations and assumptions

Let $G = (\boldsymbol{V}, \boldsymbol{E})$ be a DAG containing only directed edges and has no directed cycles. For each $V_i \in \boldsymbol{V}$, let PA^i denotes the set of parents of V_i in G. Since our data are time-history dependent, we assume that the joint probability distribution of $\boldsymbol{V}$ at each time point according to G can factorize as $p(\boldsymbol{V}) = \prod_{i=1}^{m} p(V_i \mid \mathrm{PA}^i)$, where m is the number of vertices in G. *Here $p(V_i \mid \mathrm{PA}^i)$ can be regarded as "causal mechanism." For non-stationary time series data, the causal mechanism $p(V_i \mid \mathrm{PA}^i)$ can change over time, and the changes may be due to the involved functional models or the causal strengths.*

Throughout this section, we use the example of traction-separation law to illustrate the proposed causal discovery method without loss of generality. Therefore, $\boldsymbol{V}$ is the set consisting of displacement jump $\boldsymbol{U}$, porosity ϕ, coordination number CN, fabric tensor $\boldsymbol{A}_f$, and traction $\boldsymbol{T}$. [35] developed a constraint-based causal discovery algorithm for non-stationary time series data to identify changing causal modules and recover the causal structure. In this paper, we extend the algorithm in [35] such that the proposed causal discovery algorithm not only handles non-stationary time-history data but also incorporates certain physical constraints. For example, in constructing the traction-separation law, we have the prior knowledge that the dynamic changes in $\boldsymbol{U}$ can cause changes in other variables, not vice versa. Therefore, if there exists a directed edge between the displacement jump $\boldsymbol{U}$ and any other variable V_i, then $\boldsymbol{U} \to V_i$.

Denote $\boldsymbol{V}_{-\boldsymbol{U}}$ to include all other variables in $\boldsymbol{V}$ excluding $\boldsymbol{U}$ (e.g., porosity, fabric tensor). *Since the causal mechanism can change over time, we assume that the changes can be explained by certain time-varying confounders, which can be written as functions of time. As we have the prior knowledge that $\boldsymbol{U}$ itself is a time-dependent variable and could affect all other variables, we regard $\boldsymbol{U}$ as such a confounder and* assume that the causal relation for each $V_i \in \boldsymbol{V}_{-\boldsymbol{U}}$ can be represented by the following structural equation model:

$$V_i = g_i(\mathrm{PA}^i, \theta_i(\boldsymbol{U}), \epsilon_i), \tag{1}$$

where PA^i includes $\boldsymbol{U}$ if the changes in $\boldsymbol{U}$ can affect the changes in V_i, $\theta_i(\boldsymbol{U})$ denotes a function of $\boldsymbol{U}$ that influences V_i as effective parameters, ϵ_i is a noise term that is independent of $\boldsymbol{U}$ and PA^i. The ϵ_i's are assumed to be independent. As we treat $\boldsymbol{U}$ as a random variable, there is a joint distribution over $\boldsymbol{V} \cup \{\theta_i(\boldsymbol{U})\}_{i:V_i \in \boldsymbol{V}_{-\boldsymbol{U}}}$. Denote G^{aug} to be the graph by adding $\{\theta_i(\boldsymbol{U})\}_{i:V_i \in \boldsymbol{V}_{-\boldsymbol{U}}}$ to G, and for each i, adding an arrow from $\theta_i(\boldsymbol{U})$ to V_i. Note that G is the induced subgraph of G^{aug} over $\boldsymbol{V}$. Denote the joint distribution of G^{aug} to be p^{aug}.

In order to apply any conditional independence test on the variable set $\boldsymbol{V}$ for recovering causal structure, we set the following assumptions [73].

Assumption 1 (Causal Markov condition) G^{aug} and the joint distribution p^{aug} on $\boldsymbol{V} \cup \{\theta_i(\boldsymbol{U})\}_{i:V_i \in \boldsymbol{V}_{-\boldsymbol{U}}}$ satisfy the causal Markov condition if and only if a vertex of G^{aug} is probabilistically independent of all its non-descendants in G^{aug} given the set of all its parents.

Assumption 2 (Faithfulness) G^{aug} and the joint distribution p^{aug} satisfy the faithfulness condition if and only if no conditional independence holds unless entailed by the causal Markov condition.

Assumption 3 (Causal sufficiency) The common causes of all variables in $\boldsymbol{V} \cup \{\theta_i(\boldsymbol{U})\}_{i:V_i \in \boldsymbol{V}_{-\boldsymbol{U}}}$ are measured.

3.2 Recovery of the causal skeleton

In this section, we propose a constraint-based method building upon the PC algorithm [73] to first identify the skeleton of G, defined as the obtained undirected graph if we ignore the directions of edges in a DAG G. We prove that given Assumptions 1–3, we can apply conditional independence tests to $\boldsymbol{V}$ to recover the skeleton of G. Algorithm 1 describes the proposed method, which is supported by Theorem 1. The proof is provided in Appendix 1 following [35].

Algorithm 1 Obtain the undirected skeleton of G

1: Object: To obtain the undirected skeleton of G
2: Build a complete undirected graph U_G with variables $\boldsymbol{V}$
3: **for** each node $V_i \in \boldsymbol{V}_{-\boldsymbol{U}}$ **do**
4: **if** V_i and $\boldsymbol{U}$ are independent given a subset of $\{V_k | V_k \in \boldsymbol{V}_{-\boldsymbol{U}}, k \neq i\}$ **then**
5: Remove the edge between V_i and $\boldsymbol{U}$
6: **end if**
7: **end for**
8: **for** every $V_i, V_j \in \boldsymbol{V}_{-\boldsymbol{U}}$ **do**
9: **if** V_i and V_j are independent given a subset of $\{V_k \mid V_k \in \boldsymbol{V}_{-\boldsymbol{U}}, k \neq i, k \neq j\} \cup \boldsymbol{U}$ **then**
10: Remove the edge between V_i and V_j
11: **end if**
12: **end for**
13: **return** U_G

Theorem 1 *Given Assumptions 1–3, for every $V_i, V_j \in \boldsymbol{V}_{-\boldsymbol{U}}$, V_i and V_j are not adjacent in G if and only if they are independent conditional on some subset of $\{V_k \mid V_k \in \boldsymbol{V}_{-\boldsymbol{U}}, k \neq i, k \neq j\} \cup \{\boldsymbol{U}\}$.*

In lines 3–7 of Algorithm 1, we determine whether the changes in $\boldsymbol{U}$ cause changes in V_i. If not, $\boldsymbol{U}$ is not in the parent set of V_i and there is no edge between $\boldsymbol{U}$ and V_i in G. The lines 8–12 of Algorithm 1 aims to identify the causal skeleton between variables in $\boldsymbol{V}$ except $\boldsymbol{U}$. Since how other variables change with $\boldsymbol{U}$ and the relations between these variables are usually unknown and potentially very complex, we use a nonparametric conditional independence test, kernel-based condition independence (KCI) test developed by [98], to determine the dependence between variables throughout this paper. This nonparametric approach can not only capture the linear/nonlinear correlations between variables by testing for zero Hilbert-Schmidt norm of the partial cross-covariance operator, but also handle multidimensional data that are common in mechanics problems.

3.3 Determination of causal directions

After obtaining the skeleton U_G, we need to determine the causal directions of edges. [52] provided a set of orientation rules to determine the directions of undirected edges in a graph based on conditional independence tests. However, the Meek rule [52] is only applicable to edges that satisfy its conditions. In this section, we first introduce the Meek rule [52], then propose an algorithm to orient the edges that are not covered by the Meek rule after incorporating known physical knowledge.

Denote $\leftrightarrow$ to be an undirected edge. The Meek rule has the following principles:

1. For all triples $V_i \leftrightarrow V_j \leftrightarrow V_k$, if V_i and V_k are marginally independent but conditionally dependent given V_j, then $V_i \rightarrow V_j \leftarrow V_k$;
2. If $V_i \rightarrow V_j \leftrightarrow V_k$ and there is no edge between V_i and V_k, then orient $V_j \rightarrow V_k$;
3. If $V_i \rightarrow V_j \leftrightarrow V_k$ and there is an edge between V_i and V_k, then orient $V_i \rightarrow V_k$;
4. If $V_i \rightarrow V_j \leftarrow V_k$, $V_i \leftrightarrow V_k \leftrightarrow V_k$, and $V_k \leftrightarrow V_j$, then $V_k \rightarrow V_j$.

Now we describe our algorithm on how to determine the edge directions in the obtained skeleton U_G. Firstly, for any node V_i adjacent to $\boldsymbol{U}$, we orient $\boldsymbol{U} \rightarrow V_i$ due to the prior physical knowledge that only $\boldsymbol{U}$ affects other variables, not vice versa. Then we apply the Meek rule to the obtained graph after orienting the edges from $\boldsymbol{U}$ to its neighbours. For instance, suppose $\boldsymbol{U} \rightarrow V_i \leftrightarrow V_j$, if V_j and $\boldsymbol{U}$ are independent given a set of variables including V_i, then we orient $V_i \rightarrow V_j$; if V_j and $\boldsymbol{U}$ are independent given a set of variables excluding V_i, then we have $V_j \rightarrow V_i$.

Next, we discuss how to determine the edge direction between two adjacent variables if they are both adjacent to $\boldsymbol{U}$, i.e., $V_i \leftrightarrow V_j$, $\boldsymbol{U} \rightarrow V_i$, and $\boldsymbol{U} \rightarrow V_j$, since such a scenario is not covered by the Meek rule. The modularity property of causal systems [61] demonstrated that if there are no confounders for *cause* and *effect*, then $p(cause)$ and $p(effect \mid cause)$ are either fixed or change independently. Based on this principle, since both V_i and V_j change with $\boldsymbol{U}$, we can test the conditional independence between $p(V_i \mid \theta_i(\boldsymbol{U}))$ and $p(V_j \mid V_i, \theta_j(\boldsymbol{U}))$, as well as between $p(V_j \mid \theta_j(\boldsymbol{U}))$ and $p(V_i \mid V_j, \theta_i(\boldsymbol{U}))$ to determine the direction between V_i and V_j. That says, if $p(V_i \mid \theta_i(\boldsymbol{U}))$ and $p(V_j \mid V_i, \theta_j(\boldsymbol{U}))$ are conditionally independent but $p(V_j \mid \theta_j(\boldsymbol{U}))$ and $p(V_i \mid V_j, \theta_i(\boldsymbol{U}))$ are not, then $V_i \rightarrow V_j$. [35] developed a kernel embedding of non-stationary conditional distributions and extended the Hilbert Schmidt Independence Criterion (HSIC, [25]) to measure the dependence between distributions, based on which the causal directions can be determined. For example, if we have two random variables V_1 and V_2, we can compute the dependence between

$p(V_1)$ and $p(V_2 \mid V_1)$ using the normalized HSIC, denoted by $\hat{\Delta}_{V_1 \to V_2}$. By the same token, we can compute the dependence between $p(V_2)$ and $p(V_1 \mid V_2)$ using the normalized HSIC, denoted by $\hat{\Delta}_{V_2 \to V_1}$. If $\hat{\Delta}_{V_1 \to V_2} < \hat{\Delta}_{V_2 \to V_1}$, we orient $V_1 \to V_2$; otherwise $V_2 \to V_1$. After orienting all possible edges, we can get the Markov equivalent class of the DAG G. Algorithm 2 summarizes the proposed method on how to determine causal directions.

Remark 1 In the numerical examples, we setup a threshold inclusion probability (20%) below which the causality relation is not included in the hierarchical neural network models. This treatment allows us to ensure that the causalities with sufficient likelihoods are included but the less prominent relationship is omitted to improve the efficiency and simplicity of the resultant model. This threshold can be viewed as a hyperparameter. A highly threshold may yield a DAG with less vertices and therefore reduce the total number of required supervised training at the expense of being less precise on the causality relations among the data.

Algorithm 2 Obtain the Markov equivalence class of the DAG G

```
Object: To orient the directions in the causal skeleton U_G
Input: The undirected skeleton output from Algorithm 1
for any node V_i adjacent to U do
    Orient U → V_i
end for
for all other undirected edges do
    Apply the Meek rule
end for
Find all nodes that are adjacent to U and have undirected edges with other nodes, denoted by S
if S is empty then
    return G
else
    repeat
        for each node V ∈ S do
            Consider the set Z of nodes that either are directed parents of V or have undirected edges to V
            Calculate the normalized HSIC using the node V as the effect and the set of nodes Z as the cause
        end for
        Pick the node V with the smallest normalized HSIC
        Orient all edge directions from the nodes in Z to node V
        Remove the node V from S
    until S is empty
end if
return G
```

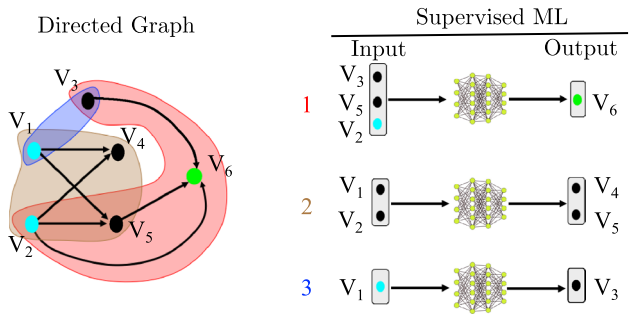


Fig. 1 **a** shows a direct graph partitioned into three sub-graphs. **b** indicates how each sub-graph is used in a separate supervised machine learning task to predict the downstream node(s) from the upstream node(s)

4 Supervised learning for path-dependent material laws

Once causal relations are identified, a directed graph $G = (\boldsymbol{V}, \boldsymbol{E})$ can be established where there is an edge $e_{ij} \in \boldsymbol{E}$ from the node $V_i \in \boldsymbol{V}$ to $V_j \in \boldsymbol{V}$ if V_i is a direct cause of V_j. Denote the leaf node to be the vertex that is not the cause to any other vertices, the root node to be the vertex that is not the target of any other vertices. Figure 1 demonstrates a directed graph indicating an information flow how leaf

node(s) is related to root node(s) via some intermediate nodes, e.g., in Fig. 1 $\{V_1, V_2\}, \{V_6\}$, and $\{V_3, V_4, V_5\}$ are sets of root, leaf, and intermediate nodes.

Along with the similar idea introduced in [89], we aim to discover all sub-graphs that sequentially pass the information from the root to leaf, see Algorithm 3. Each of these subgraphs will contain leaf and root but without any intermediate nodes. As such, supervised learning can help us train neural network that predict the root of each subgraph with the corresponding leaf (or leaves) as input(s). To identify all sub-graphs, we traverse the graph backward from leaf to root nodes. The leaf with its immediate predecessors formed a directed tree which will be added into the list of potential sub-graphs. Then, we remove edges of the founded tree in the graph G. In the next step, we select a new leaf node from the updated G and do the process just described until there is not any edges in G. For the case shown in Fig. 1 we have the following potential sub-graphs:

$$G_a = (\{V_2, V_3, V_5, V_6\}, \{e_{26}, e_{36}, e_{56}\}), \tag{2}$$

$$G_b = (\{V_1, V_4, V_5\}, \{e_{14}, e_{15}\}), \tag{3}$$

$$G_c = (\{V_2, V_4, V_5\}, \{e_{24}, e_{25}\}), \tag{4}$$

$$G_d = (\{V_1, V_3\}, \{e_{13}\}). \tag{5}$$

Those sub-graphs (directed trees) that share common upstream nodes will be merged into a bigger sub-graph. For the graph in Fig. 1 final sub-graphs are:

$$G_1 = G_a = (\{V_2, V_3, V_5, V_6\}, \{e_{26}, e_{36}, e_{56}\}), \tag{6}$$

$$G_2 = G_b \cup G_c = (\{V_1, V_2, V_4, V_5\}, \{e_{14}, e_{15}, e_{24}, e_{25}\}), \tag{7}$$

$$G_3 = G_d = (\{V_1, V_3\}, \{e_{13}\}). \tag{8}$$

For each sub-graph, we have a separate supervised machine learning (ML) task. In each ML task, inputs and outputs features are upstream and downstream nodes in each directed sub-graph, as shown in Fig. 1(b).

Algorithm 3 Obtain all supervised learning input-output pairs

Input: Directed graph $G = (\boldsymbol{V}, \boldsymbol{E})$ ▷ result of causal graph
$\mathcal{S} \leftarrow \emptyset$ ▷ a set of 2-tuples (input nodes, output nodes) for ML tasks
while $\boldsymbol{E} \neq \emptyset$ **do**
  $V_l \leftarrow$ get a leaf node of G ▷ if G has multiple leaf nodes returns a random leaf
  $\bar{\boldsymbol{V}} \leftarrow \{V_l\}$
  $\hat{\boldsymbol{V}} \leftarrow \{V_i | e_{il} \in \boldsymbol{E}\}$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\hat{\boldsymbol{V}}, \bar{\boldsymbol{V}})\}$ ▷ $\hat{\boldsymbol{V}}$ is input node(s), and $\bar{\boldsymbol{V}}$ is the output node for a potential ML task
  $\boldsymbol{E} \leftarrow \boldsymbol{E} \setminus \{e_{il} | V_i \in \hat{\boldsymbol{V}}\}$
end while
Modification: Elements i and j of $\mathcal{S}$ known as $(\hat{\boldsymbol{V}}_i, \bar{\boldsymbol{V}}_i)$ and $(\hat{\boldsymbol{V}}_j, \bar{\boldsymbol{V}}_j)$, respectively, are merged into one element as $(\hat{\boldsymbol{V}}_i, \bar{\boldsymbol{V}}_j \cup \bar{\boldsymbol{V}}_j)$ if $\hat{\boldsymbol{V}}_i \equiv \hat{\boldsymbol{V}}_j$.
return $\mathcal{S}$

In the training step, we use the same architecture for all ML tasks consisting of five layers GRU-Dropout-GRU-Dropout-Dense. Each GRU layer has 32 neurons, and the linear activation function is used for the Dense layer. Gated Recurrent Unit (GRU) is one type of Recurrent Neural Networks (RNN) to model history dependence [9]. We use the GRU to strengthen the robustness of the ML black-box in dealing with any possible path-dependence [92]. The dropout rate in the GRU is controlled to quantify uncertainty in the model prediction, which will be detailed in Sect. 5. All the input and output feature columns for each supervised learning task are normalized to zero mean and unity standard deviation. The loss function is defined as the mean squared root error between output prediction and ground truth. This loss is minimized by the Adam optimizer inside Keras library with Tensorflow 2.0 as its backend. Optimization process is done by the mini-batch stochastic gradient descent algorithm with a batch size 256 during 1000 epochs. The described neural network architecture and hyperparameters are chosen to be as close as possible to the ones used in [89]. *Note that, the tuning of the hyperparameters (e.g. number of neurons, number of layers, type of activation) may have a significant effect on the performance of the neural network models. The best combination of hyperparameters can be estimated via a variety of approaches such as the greedy search, random search* [3], *random forest* [64], *Bayesian optimization* [40], *meta-gradient iteration or deep reinforcement learning* [18, 93]. *In this work, we adopt the random search approach in* [3] *to fine-tune the hyperparameters (cf. Sec. 6.1) A rigorous hyperparameter study that compares different hyperparameter tuning for neural networks that generates constitutive laws may provide further insights on the optimal setup of the hyperparameters but is out of the scope of this study.*

For the blind prediction, after training, we start from the root (e.g., $\boldsymbol{U}$ in the traction-separation law) and sequentially predict intermediate nodes via their corresponding sub-graph trained neural networks (NN) until reaching the leaf node. For example in the case shown in Fig. 1: NN 3 predicts V_3 from input V_1; V_4 and V_5 are predicted by NN 2 from inputs V_1 and V_2; and finally NN 1 is used to predict the target variable V_6 from input V_2 and the obtained intermediate nodes V_3 and V_5.

5 Uncertainty propagation in causal graph with dropout layers

As described in Sect. 4, we use the deep learning method, GRU in the training step and prediction to handle path-dependent predictions. However, the GRU itself is not designed to capture prediction uncertainty, which is of crucial importance in learning material law. In machine learning and statistics, Bayesian methods as probabilistic models provide us a natural way to quantify the model uncertainty through computing the posterior distribution of unknown parameters [96]. However, these methods often suffer from a prohibitively high computational cost. In this paper, we show that the dropout technique [74] used in the GRU can quantify uncertainty in prediction as a Bayesian approximation.

Dropout, a regularization method that randomly masks or ignores neurons during training, has been widely used in many deep learning models to avoid over-fitting and improve prediction [5, 31, 46]. [20] firstly prove the link between dropout and a well known probabilistic model, the Gaussian process [65], and show that the use of dropout in the feed forward neural networks can be interpreted as a Bayesian approximation of Gaussian processes. In the context of RNNs, [21] treated RNNs as probabilistic models by assuming network weights as random variables with a Gaussian mixture prior (with one component fixed at zero with a small variance). Such a technique is similar to the spike-and-slab prior in Bayesian statistics for variable selection [37]. Then [21] show that optimizing the objective in the variational inference [4] for approximating the posterior distribution over the weights is equivalent to conducting dropout in the respective RNNs, and demonstrate the implementation in one commonly-used RNN model, the long short-term memory [32]. In this section, we propose to extend the technique developed in [21] in the context of GRUs for uncertainty quantification (UQ) in the prediction.

Given training inputs $\boldsymbol{X}$ and the corresponding output $\boldsymbol{Y}$, suppose that we aim to predict an output $\boldsymbol{y}^*$ for a new input $\boldsymbol{x}^*$. From the Bayesian point of view, the prediction uncertainty can be characterized by the posterior predictive distribution of $\boldsymbol{y}^*$ as follows:

$$p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y}) = \int p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \boldsymbol{\omega}) p(\boldsymbol{\omega} \mid \boldsymbol{X}, \boldsymbol{Y}) \mathrm{d}\boldsymbol{\omega}, \tag{9}$$

where $\boldsymbol{\omega}$ includes all unknown model parameters, $p(\boldsymbol{\omega} \mid \boldsymbol{X}, \boldsymbol{Y})$ is the posterior distribution of $\boldsymbol{\omega}$. In the GRU, all unknown weights can be viewed as $\boldsymbol{\omega}$. As the posterior distribution $p(\boldsymbol{\omega} \mid \boldsymbol{X}, \boldsymbol{Y})$ is generally intractable, the variational inference method approximates it by proposing a variational distribution $q(\boldsymbol{\omega})$ and then finding the optimal parameters in the variational distribution through minimizing the Kullback-Leibler (KL) divergence between the approximating distribution and the full posterior distribution:

$$\begin{aligned} &\mathrm{KL}(q(\boldsymbol{\omega}) \| p(\boldsymbol{\omega} \mid \boldsymbol{X}, \boldsymbol{Y})) \propto \\ &- \int q(\boldsymbol{\omega}) \log p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} + \mathrm{KL}(q(\boldsymbol{\omega}) \| p(\boldsymbol{\omega})), \end{aligned} \tag{10}$$

where $p(\boldsymbol{\omega})$ is the prior distribution of $\boldsymbol{\omega}$.

Given an input sequence $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T]$ of length T, the hidden state $\boldsymbol{h}_t$ at time step t in the GRU neural network can be generated as follows:

$$\begin{aligned} \boldsymbol{z}_t &= \sigma\left(\boldsymbol{W}_z \boldsymbol{x}_t + \boldsymbol{U}_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z\right), \\ \boldsymbol{r}_t &= \sigma\left(\boldsymbol{W}_r \boldsymbol{x}_t + \boldsymbol{U}_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right), \\ \boldsymbol{i}_t &= \tanh\left(\boldsymbol{W}_h \boldsymbol{x}_t + \boldsymbol{U}_h\left(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}\right) + \boldsymbol{b}_h\right), \\ \boldsymbol{h}_t &= \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + \left(1 - \boldsymbol{z}_t\right) \odot \boldsymbol{i}_t, \end{aligned} \tag{11}$$

where σ denotes the sigmoid function and $\odot$ denotes the element-wise product. Also, we assume that the model output at time step t can be written as $f_{\boldsymbol{Y}}(\boldsymbol{h}_t) = \boldsymbol{h}_t \boldsymbol{W}_{\boldsymbol{Y}} + \boldsymbol{b}_{\boldsymbol{Y}}$. Then the unknown parameters in the GRU are $\boldsymbol{\omega} = \left\{\boldsymbol{W}_Y, \boldsymbol{W}_z, \boldsymbol{U}_z, \boldsymbol{W}_r, \boldsymbol{U}_r, \boldsymbol{W}_h, \boldsymbol{U}_h, \boldsymbol{b}_Y, \boldsymbol{b}_z, \boldsymbol{b}_r, \boldsymbol{b}_h\right\}$. We write $\boldsymbol{h}_t = f_{\boldsymbol{h}}^{\boldsymbol{\omega}}\left(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}\right)$ and $f_{\boldsymbol{Y}}^{\boldsymbol{\omega}}$ for the output in order to make the dependence on $\boldsymbol{\omega}$ clear.

Then the right hand of (10) can be written as follows:

$$-\int q(\boldsymbol{\omega}) \log p\left(\boldsymbol{Y} \mid f_{\boldsymbol{Y}}^{\boldsymbol{\omega}}\left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T, f_{\boldsymbol{h}}^{\boldsymbol{\omega}}\left(\boldsymbol{x}_T, f_{\boldsymbol{h}}^{\boldsymbol{\omega}}(\ldots f_{\boldsymbol{h}}^{\boldsymbol{\omega}}(\boldsymbol{x}_1, \boldsymbol{h}_0) \ldots)\right)\right)\right) \mathrm{d}\boldsymbol{\omega} + \mathrm{KL}(q(\boldsymbol{\omega}) \| p(\boldsymbol{\omega})), \tag{12}$$

which can be approximated by Monte Carlo integration with the generated samples $\hat{\boldsymbol{\omega}}^b \sim q(\boldsymbol{\omega})$ and plug in the sampled $\hat{\boldsymbol{\omega}}^b$'s to (12).

Following [21], we use a mixture of Gaussian distributions as the variational distribution for every weight matrix row $\boldsymbol{\omega}_k$:

$$q(\boldsymbol{\omega}) = \prod_{k=1}^{K} q(\boldsymbol{\omega}_k), \quad q\left(\boldsymbol{\omega}_k\right) = \pi \mathcal{N}\left(\boldsymbol{\omega}_k; \boldsymbol{0}, \tau^2 I\right) + (1 - \pi) \mathcal{N}\left(\boldsymbol{\omega}_k; \boldsymbol{m}_k, \tau^2 I\right), \tag{13}$$

where π is the dropout probability, $\boldsymbol{m}_k$ is the variational parameter (row vector), and τ^2 is a small variance. We optimize over $\boldsymbol{m}_k$ by minimizing the KL divergence in (12). Sampling each row of $\hat{\boldsymbol{\omega}}^b$ is equivalent to randomly mask rows in each weight matrix, i.e., conducting dropout. Then the predictive posterior distribution can be approximated by

$$\begin{aligned} &p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y}) \\ &= \int p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \boldsymbol{\omega}) p(\boldsymbol{\omega} \mid \boldsymbol{X}, \boldsymbol{Y}) \mathrm{d}\boldsymbol{\omega} \approx \frac{1}{B} \sum_{b=1}^{B} p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \hat{\boldsymbol{\omega}}^b), \end{aligned} \tag{14}$$

where $\hat{\boldsymbol{\omega}}^b \sim q(\boldsymbol{\omega})$ and B is the total number of generated samples.

To implement the dropout in the GRU, we re-parametrize (11) as follows:

$$\begin{pmatrix} \boldsymbol{z}_t \\ \boldsymbol{r}_t \\ \boldsymbol{i}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \tanh \end{pmatrix} \left(\begin{pmatrix} \boldsymbol{x}_t \circ \boldsymbol{m}_{\boldsymbol{x}} \\ \boldsymbol{h}_{t-1} \circ \boldsymbol{m}_{\boldsymbol{h}} \end{pmatrix} \cdot \boldsymbol{\omega} \right), \tag{15}$$

where $\boldsymbol{m}_{\boldsymbol{x}}$ and $\boldsymbol{m}_{\boldsymbol{h}}$ denote randomly masks repeated at all time steps.

We take the prediction tasks in Fig. 1 for example. The posterior predictive distribution of V_3 given V_1 is straightforward by (14) since only root and leaf nodes are involved. When involving the intermediate nodes, e.g., the subgroup G_1, the posterior predictive distribution of V_6 given the root nodes (V_1 and V_2) can be computed by

$$\begin{aligned} p\left(V_6 \mid V_1, V_2\right) = \int & p\left(V_6 \mid V_2, V_3, V_5, \boldsymbol{\omega}\right) p(V_3 \mid V_1, \boldsymbol{\omega}) \\ & p(V_5 \mid V_1, V_2, \boldsymbol{\omega}) p(\boldsymbol{\omega} \mid V_1, V_2) \mathrm{d}V_3 \mathrm{d}V_5 \mathrm{d}\boldsymbol{\omega}. \end{aligned} \tag{16}$$

Specifically, when we generate B samples from the posterior distribution of $\boldsymbol{\omega}$, we also generate B samples for the intermediate nodes. Those B samples of $\boldsymbol{\omega}$ and the corresponding

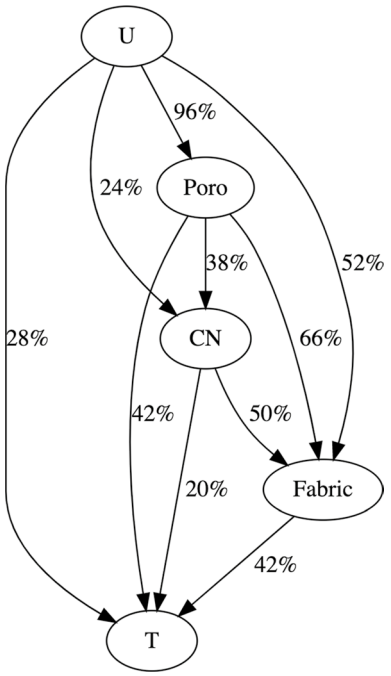


Fig. 2 Final causal graph for the traction-separation law deduced from time-history of displacement, traction, porosity, coordination number, and fabric tensor. The number on each edge represents the edge inclusion probabilities among all possible causal relations from the training data sets

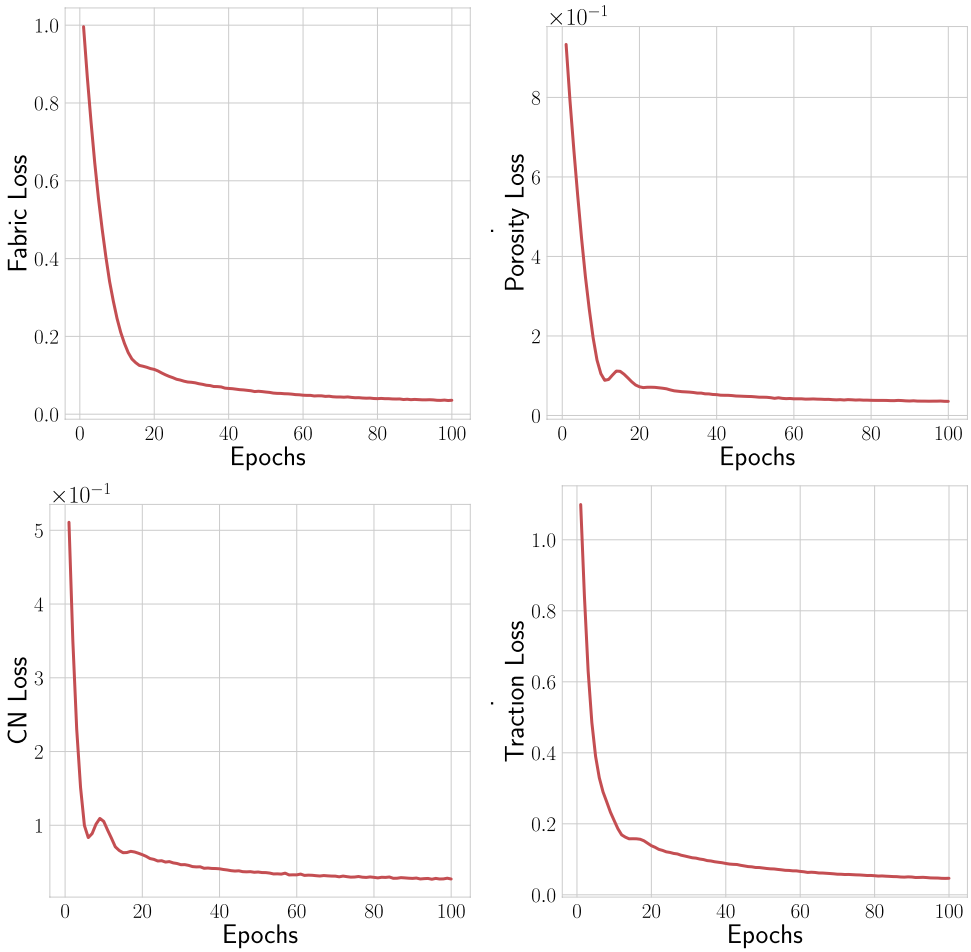


Fig. 3 Training loss convergence behavior for four supervised learning tasks deduced from the causal graph. Top left: fabric is predicted based on displacement, porosity and coordination number; Top right: porosity is predicted based on displacement; Bottom left: coordination number is predicted based on displacement and porosity; Bottom right: traction is predicted based on displacement, coordination number, fabric, and porosity

intermediate nodes can be then used in Monte Carlo integration to calculate (16). Throughout this paper, we set B=200.

6 Numerical examples

In this section, we conduct two numerical experiments to test our proposed framework that combines causal discovery with deep learning to build constitutive laws for granular materials. In Sect. 6.1, the causal discovery is conducted to determine the constitutive relationships for an RVE interface composed of spherical grains. The subsequent supervised machine learning then leverages the causal relations learned from the causal discovery algorithm to establish a serial of supervised learning that constitutes a forecast engine for traction. The propagation of uncertainty is enabled by the dropout technique that approximates Monte Carlo simulations to determine the confidence intervals for a given dropout rate. In Sect. 6.2, the same exercise is repeated for another data set to generate hypoplasticity surrogate model for a discrete element assembly where new topological measures are computed and incorporated into the proposed framework to (1) discover new physical mechanisms and (2) determine the benefit of the new discovery on the accuracy, robustness, and consistency of the forward predictions on unseen events.

Table 1 This table reports the mean and standard deviation of the validation loss among different configurations which are different based on their number of units utilized for each GRU layer when the optimal learning rate 0.001 and batch size 32 is chosen. Notice that we randomly conduct 100 trials for each subgraph with different hyperparameters. The last column shows the validation loss when the neural network has the same architecture suggested in Sec. 4. Based on the standard deviation values, we observe that the number of units in GRU layers has a marginal effect on the performance. The suggested fixed neural network architecture in Sec. 4 for all sub-graphs has almost the same performance as the best optimal configurations

Subgraph	Mean	Standard deviation	Number of configurations	Suggested NN
Porosity	1.6e − 6	7.36e − 7	10	3.7e − 6
CN	1.04e − 3	2.3e − 5	5	1.07e − 3
Fabric	2.2e − 3	3.98e − 4	7	2.4e − 3
Traction	5.1e − 5	1.01e − 5	8	9.1e − 5

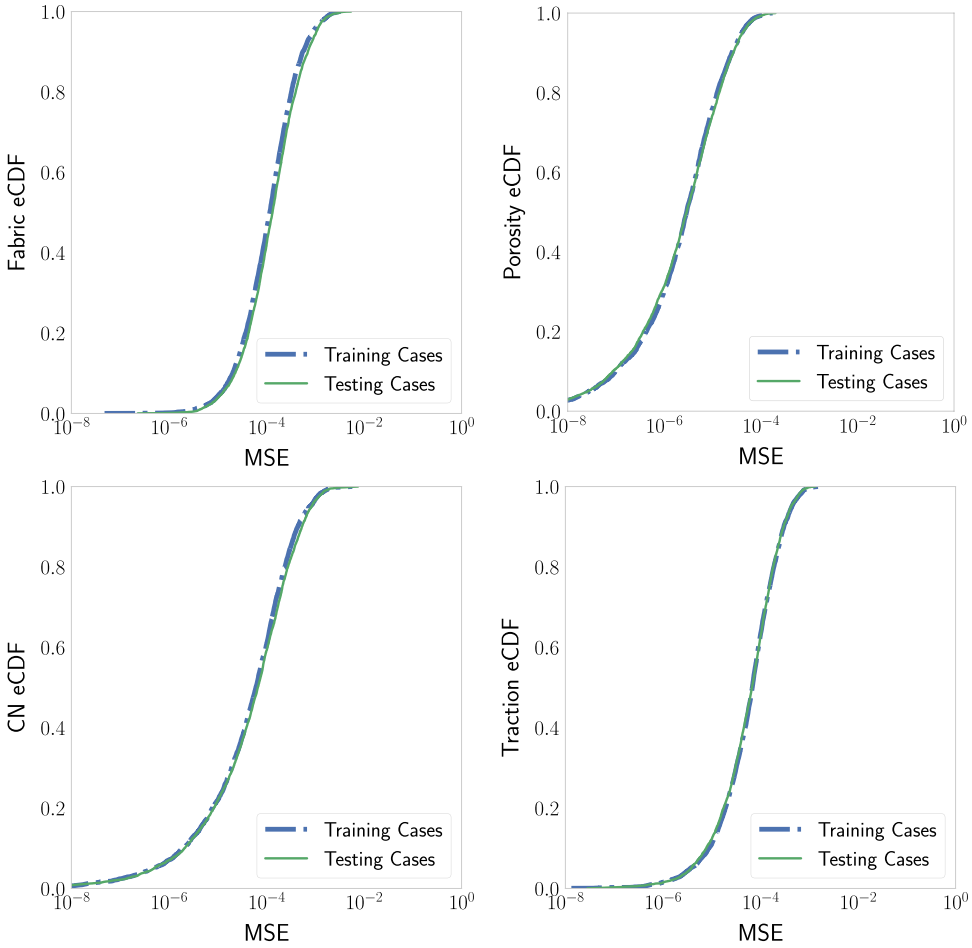


Fig. 4 Empirical Cumulative Distribution Function (eCDF) for prediction on training data sets and mean value of predictions on test data sets. We use 50 data sets for training and 50 other data sets for testing purposes. For each test case, we perform 200 feed-forward predictions with the drop-out rate 0.2

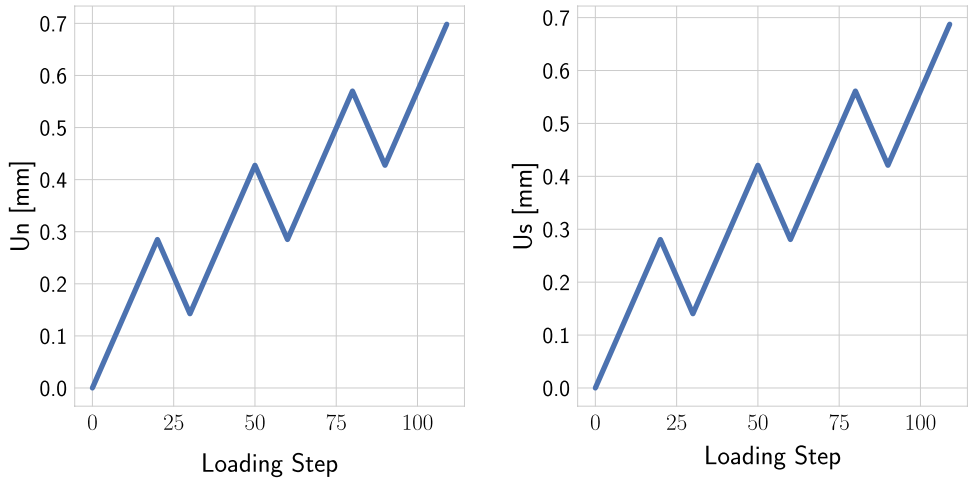


Fig. 5 Applied normal and shear displacements in one of the experimental cases. Increment in normal jump indicates more compression

6.1 Numerical example 1: machine learning traction-separation law

Traction-separation laws are key ingredients of cohesive fracture models [59, 60]. Generally speaking, a traction-separation law constitutes the relation between the traction and displacement jump across a surface. There exist many hand-crafted models developed by experts for different applications [60], while no interpretable machine learning framework had been developed until recently by [89] in which reinforcement learning is used to deduce an interpretable knowledge graph that doubles as the forecast engine. Also, in some applications such as granular materials, more descriptors, e.g., porosity or fabric tensor, should

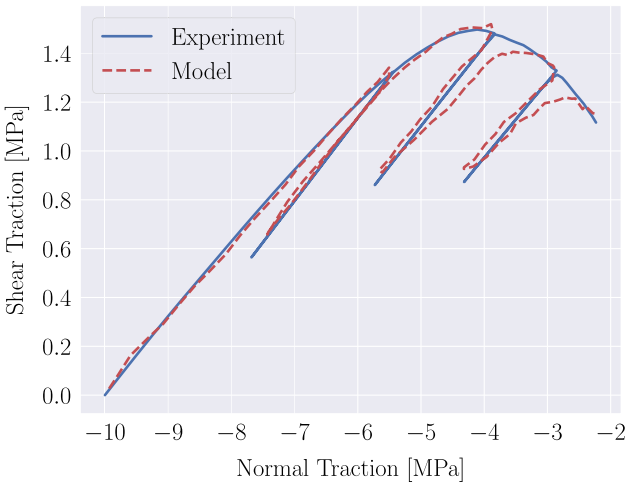


Fig. 6 Comparison of normal-shear traction between model and experiment in one case

be considered in these constitutive laws [77] to derive more predictive models.

Our first test for the data-driven causal discovery model with dropout UQ is on the traction-separation law data publicly available in the repository Mendeley data (cf. [76]). This dataset has also been used in [89] where the traction-separation law is determined from reinforcement learning. Our major point of departure is three-fold. Firstly, we develop a causal discovery algorithm to identify causal relations among history-dependent physical quantities in RVE simulations. Secondly, we decouple the causal discovery from the training of the neural network such that we now first discover causal relations, then utilize the discovered relationships to generate quantitative predictions using the method detailed in Sect. 4. Thirdly, we introduce the Bayesian approximation using the dropout technique to propagate

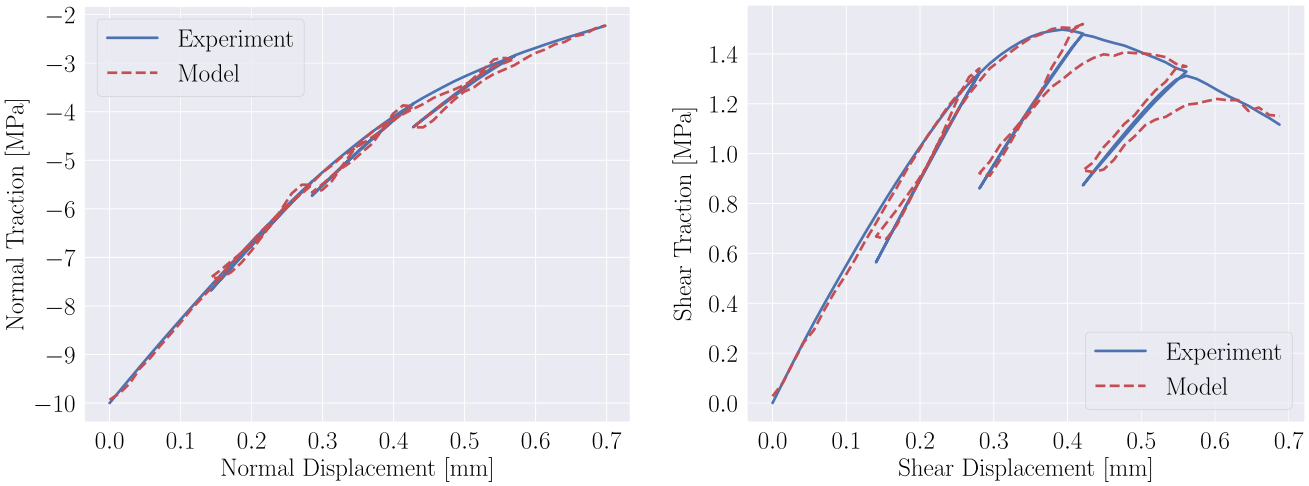
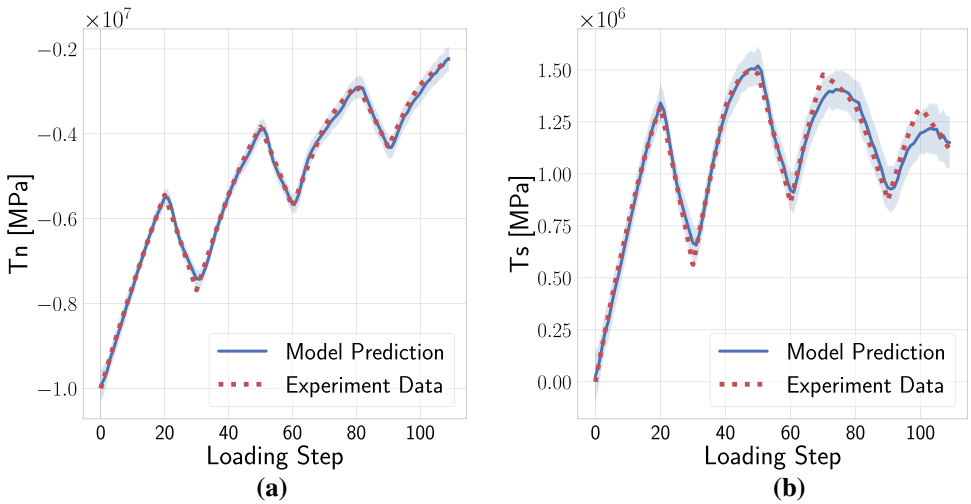


Fig. 7 Comparison of traction-displacement between model and experiment in one case

Fig. 8 Model predictions for normal **a** and shear **b** traction values. Shaded area includes predictions within 95% confidence interval



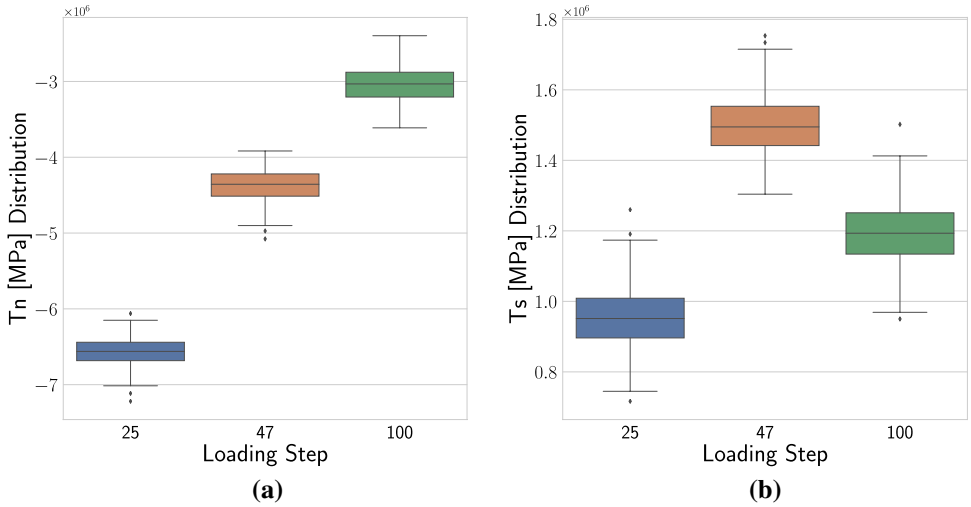


Fig. 9 Box plots of density distributions of normal **a** and shear **b** traction distributions at three load steps.The top line is maximum value and the bottom line is the minimum value. The box composed of three thick lines are separately first quantile, median, third quantile

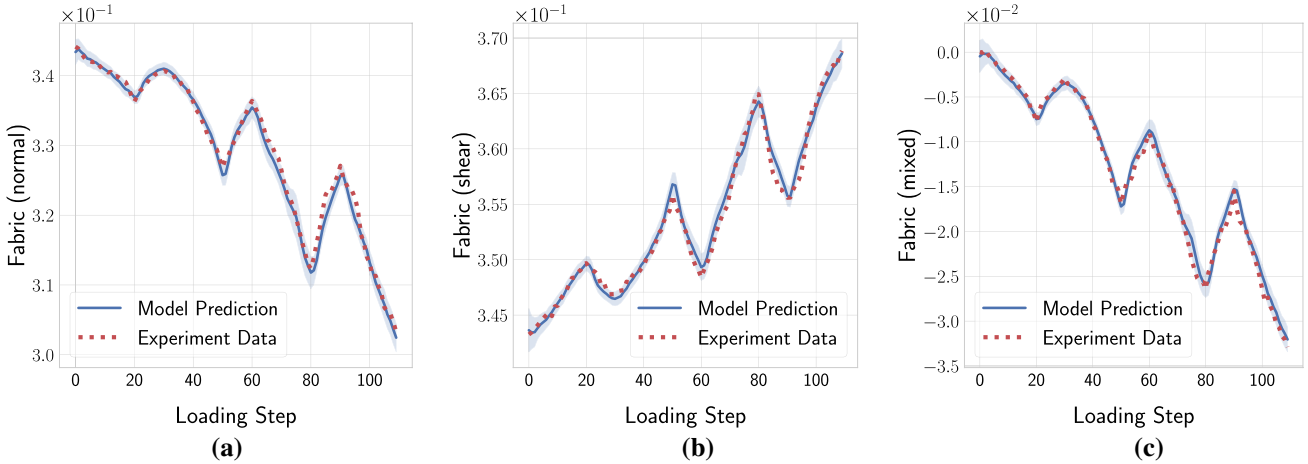


Fig. 10 Model prediction for components of symmetric fabric tensor $\boldsymbol{A}$. Normal **(a)**, shear **(b)**, and mixed **(c)** components of fabric tensors are A_{nn}, A_{ss}, and A_{ns}, respectively. Shaded area includes predictions within 95% confidence interval

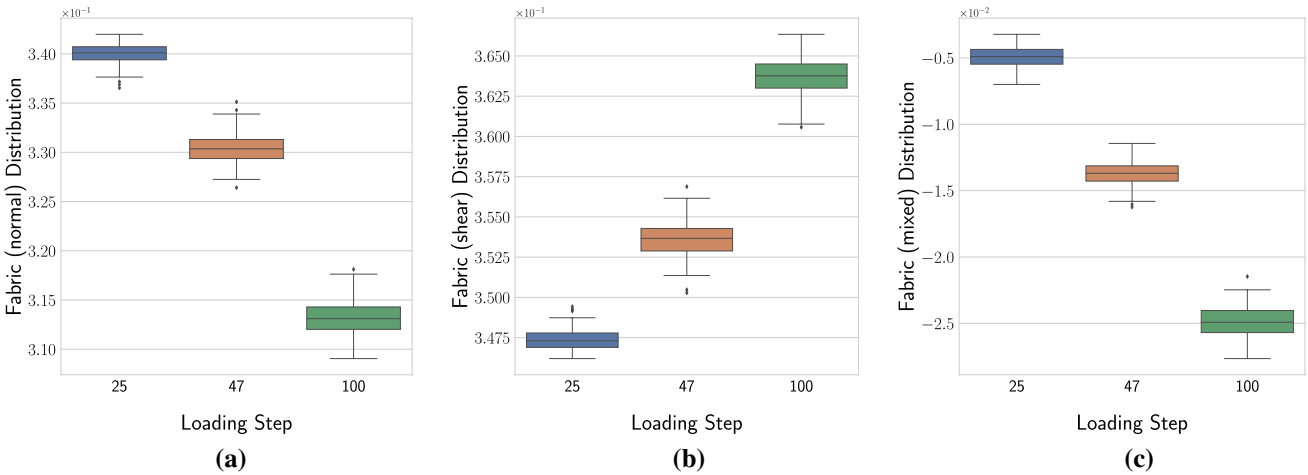


Fig. 11 Box plots of density distributions of fabric's components

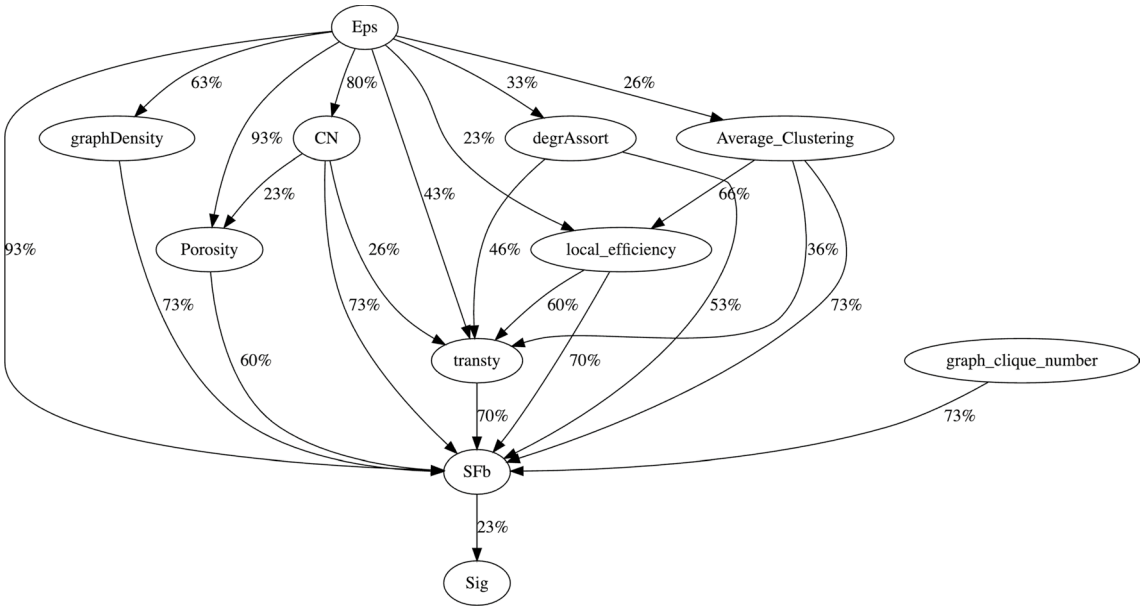


Fig. 12 Final Causal graph for the hypoplasticity relations deduced from time-history of strain, stress, and 9 other measures of microstructural and topological properties. The number on each edge represents the edge inclusion probabilities among all possible causal relations from the training data sets

the uncertainty in the causal graph, building upon the theoretical framework established in [21].

The database includes 100 DEM experiments. In each DEM experiment, the time history of all the variables included in the DAG are recorded. Each experiment is conducted by a different ratio of normal to tangential loading rate and loading-unloading cycles on the same representative volume element of granular materials. As such, the total number of time-history data points in these experiments may vary from 51 to 111. The interested reader is referred to the appendix in [89] for more information. In our study, feature space consists of displacement jump vector, traction vector, coordination number, symmetric part of fabric tensor, and porosity. We use half of the experiments for causal discovery and training artificial neural networks, and the rest is used for test and validation. In the causal discovery step, as different experimental setups may lead to different causal relations among variables, we apply the proposed causal discovery algorithm (Algorithms 1 and 2 in Sect. 3) to each of the training experiment, and then report the final causal graph by calculating the inclusion probabilities of directed edges appearing in all training experiments. The inclusion probability of one edge is defined as the proportion of causal graphs containing this edge. The directed edges with inclusion probabilities being larger than a pre-defined threshold (20% in our paper) are kept in the final causal graph. When

Table 2 Hyperparameters used to train the neural network

NN setting description	Abbreviation	Values
Neuron type subset	*NeuronType*	GRU
Hidden layers subset	*numHiddenLayers*	3
Number of neurons per layer	*numNeuronsPerLayer*	32
Dropout rate subset	*DropOutRate*	0.0
Optimizer type subset	*Optimizer*	Adam
Activation functions subset	*Activation*	*relu*
Batch sizes subset	*BatchSize*	128
Minimum Learning rate	*ReduceLROnPlateau*	0.95

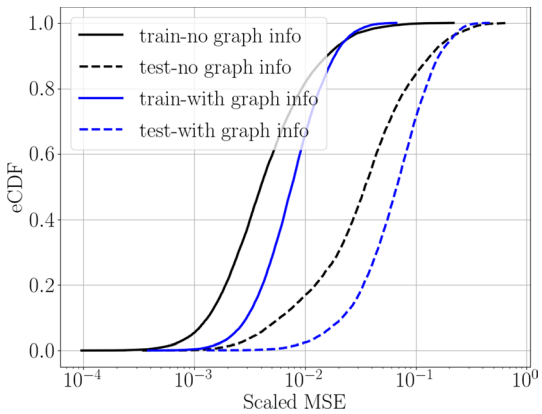


Fig. 13 Empirical Cumulative Distribution Function (eCDF) for prediction on training data sets and mean value of predictions on test data sets. There are 60 simulations, 30 used for calibration and 30 for blind forward testing. Blue curves are predictions made from neural networks generated according to the causal graph, black curve is the control experiment counterpart generated from predictions that takes on strain fabric tensor and porosity as inputs to predict Cauchy stress

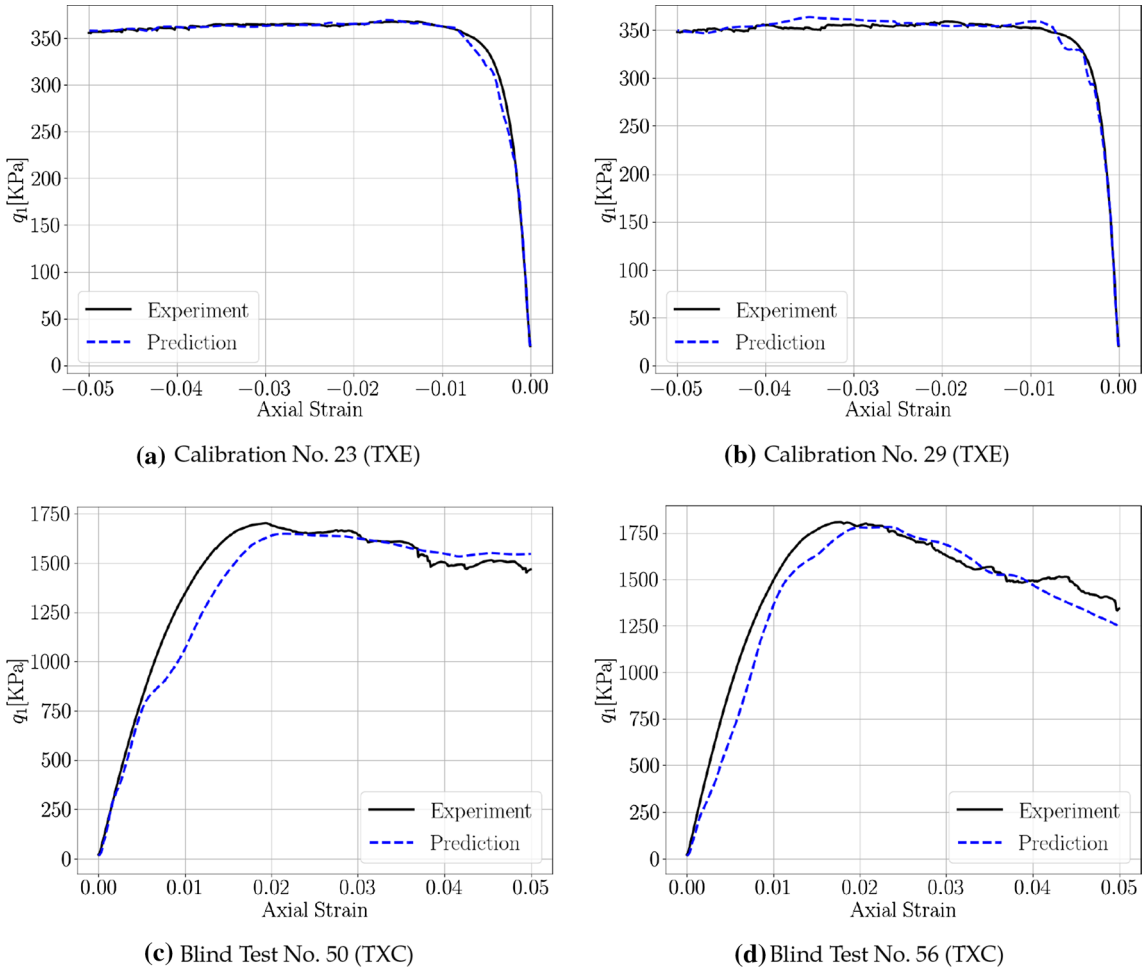


Fig. 14 Difference between the major and minor principal stress vs. axial strain. Compressive strain has positive sign convention

both edge directions between two variables appear with positive inclusion probabilities (e.g., $V_i \to V_j$ and $V_j \to V_i$ both exist), we keep the edge direction that has a higher inclusion probability. The goal of the resultant model is to predict the same granular assembly responds to a different cyclic loading path unseen in the training. As such, the focus of this model is to generate a surrogate for one representative element volume.

Figure 2 plots the final causal graph on the training data sets with edge inclusion probabilities. The strong confidence (96%) in the edge starting from the displacement jump vector to porosity is consistent with the common field knowledge, i.e., the immediate consequence of displacement jump is the volume change. The displacement jump vector, as the only control variable, affects all the intermediate physical quantities and traction vector. This observation may seem to be trivial, but it is not always the case which will be shown in the next example. The causal effects of fabric and coordination number on traction is aligned with the modern Critical State Theory [45] which is obtained without expert interpretation by the causal algorithm. Note that fabric encodes microstructural information in more detail such as directional dependence due to its tensorial nature, rather than porosity which smears out information into one scalar quantity. Therefore, it is reasonable to see that fabric has a considerable contribution in describing material behavior with a complex arrangement of force chains at the microstructural level.

Remark 2 To make sure the suggested neural network architecture in Sect. 4 works satisfactorily in this problem, we trained several neural networks with different hyper-parameters for each sub-graph learning task in Fig. 2. We used the random search approach [3] implemented in Keras Tunner package [57] for this study. The number of GRU layers is kept fixed and equal to two, and in the training stage, the dropout rate is set to zero. The number of epochs is also set to 200. The parameters under this study are as follows: the number of units in GRU layers are sampled from the set $\{8, 16, 32, 64\}$, the Adam learning rate is sampled from the

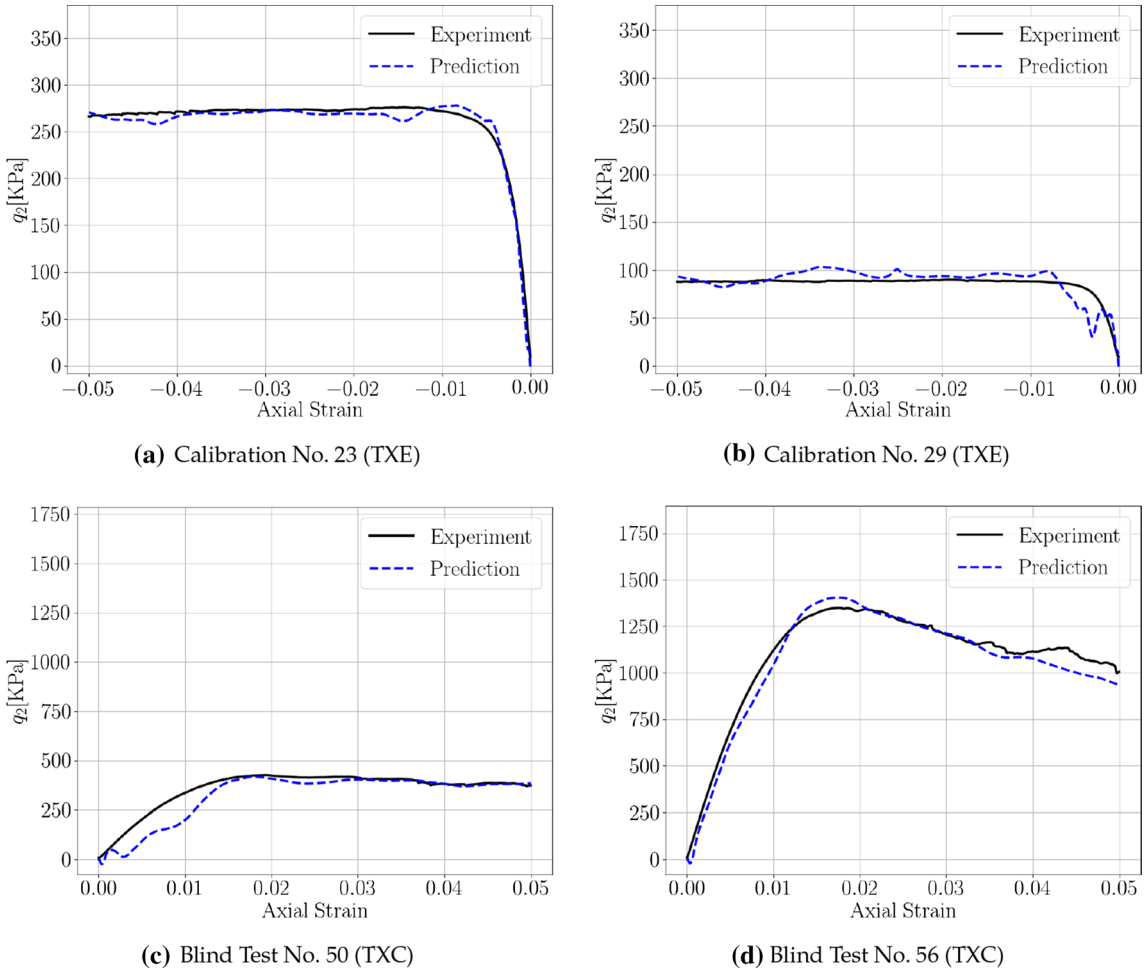


Fig. 15 Difference between the major and immediate principal stress vs. axial strain. Compressive strain has positive sign convention

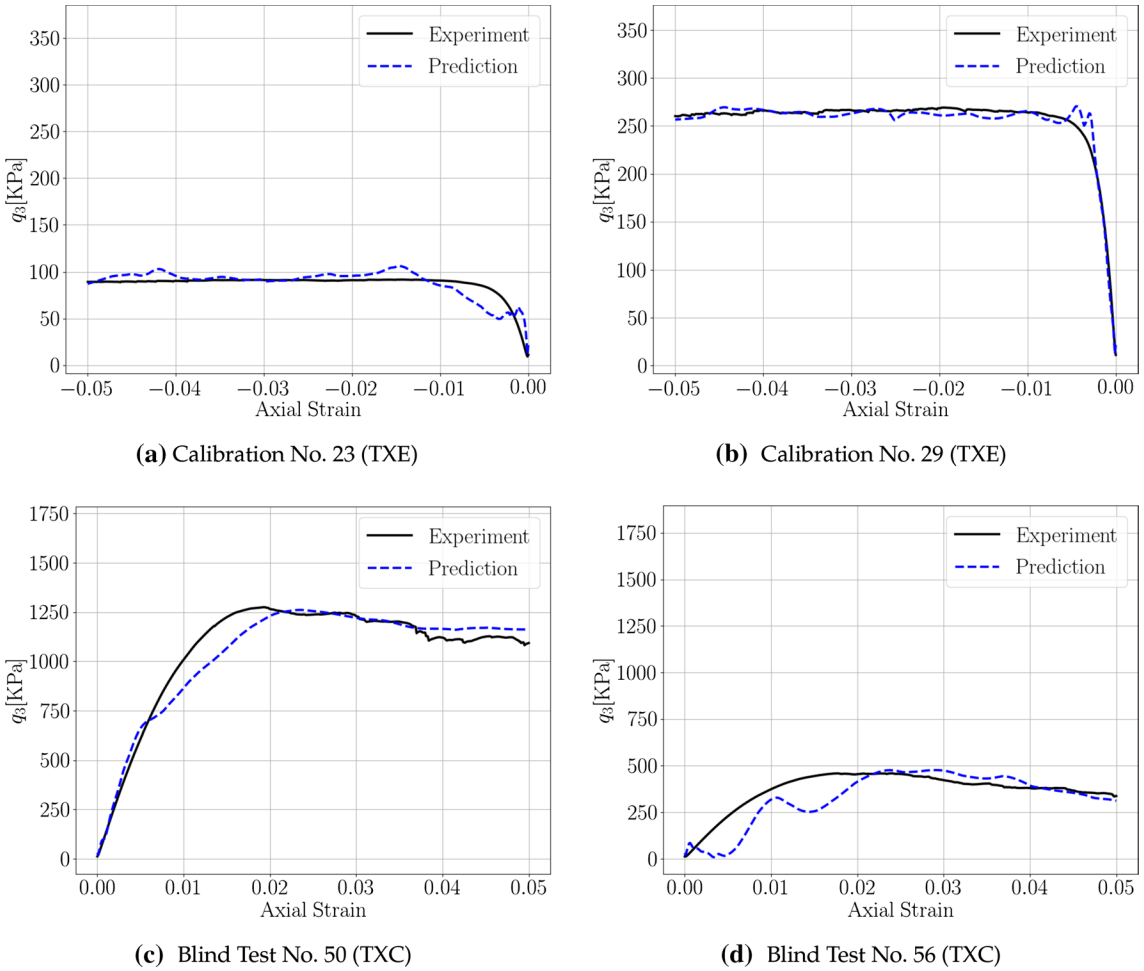
set $\{0.01, 0.001, 0.0001\}$, the batch size for the SGD algorithm is sampled from the set $\{32, 64, 128, 256, 512\}$. Based on these hyperparameter ranges, each subgraph learning task has 240 different configurations; however, in the random search algorithm, we set the number of trials to 100 for each subgraph training task to reduce overall computational time. For this hyperparameter tuning task, we choose 50 data sets as the training set and another 50 data sets as the validation set. Our metric for selecting the best configuration is the minimum validation loss. We found that the learning rate 0.001 and batch size 32 are common among all the best configurations of subgraphs. In Table 1 we study the effect of number of units in GRU layers when learning rate and batch size have their optimal values.

Applying Algorithm 3 to Fig. 2, we need to perform four supervised learning tasks: (1) predict Poro from the input $\boldsymbol{U}$; (2) predict CN from the input $\boldsymbol{U}$ and the intermediate node Poro; (3) predict fabric from the input $\boldsymbol{U}$ and the intermediate nodes Poro and CN; and (4) predict the target variable $\boldsymbol{T}$ from the input $\boldsymbol{U}$ and the obtained intermediate nodes Poro, CN, and fabric. We then use the GRU to train each sub-graph with the dropout rate being 0.2 for both training and feed-forward predictions. Figure 3 confirms that the neural network architecture proposed in Sect. 4 yields satisfactory performance for all supervised tasks. To examine the generalization performance of the trained neural networks, we study the empirical cumulative distribution functions (eCDFs) for training and test data sets following [89].

We define the point-wise scaled mean squared error (MSE) between a set of ground-truth values with size N and its corresponding approximation set as:

$$e_i = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{S}(y_i^{\text{true}}) - \mathcal{S}(y_i^{\text{appx}})), \tag{17}$$

where $\mathcal{S}$ is a scaling function. In this paper, the scaling function linearly transforms a set of values into a new set where all values are in the range [0, 1]. We perform 200



(a) Calibration No. 23 (TXE)

(b) Calibration No. 29 (TXE)

(c) Blind Test No. 50 (TXC)

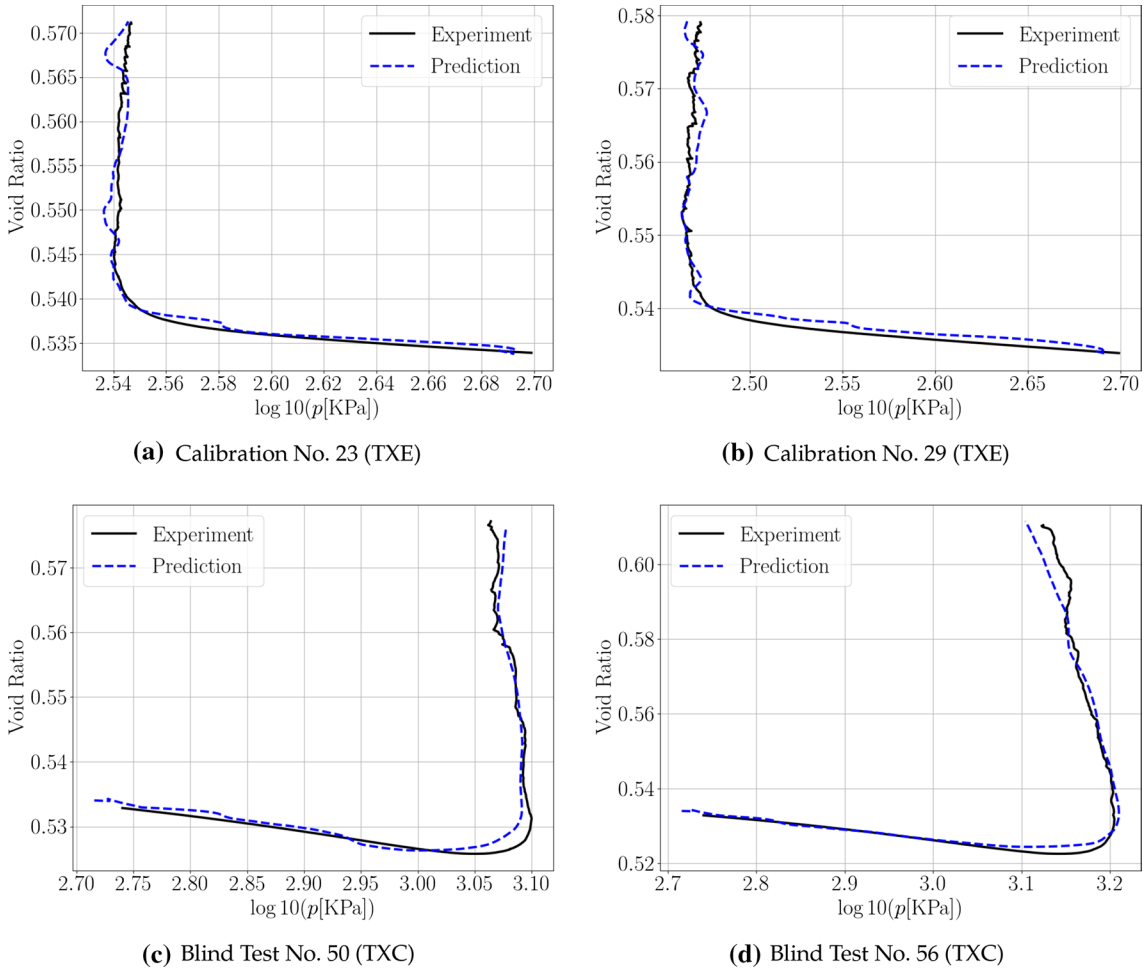
(d) Blind Test No. 56 (TXC)

Fig. 16 Difference between the immediate and minor principal stress vs. axial strain. Compressive strain has positive sign convention

feed-forward predictions to obtain the distribution of each feature output at a specified load-step. For eCDF calculation only, we use the average of these 200 predictions to approximate the feature output. In this way, the discrete eCDF of a target output feature, such as porosity, at data point i is defined as $F_N(e_i) = \frac{1}{M}\sum_{j=1}^{M} \mathbb{1}(e_i \geq e_j)$ where e_i is the point-wise scaled MSE between the feature ground-truth value and its predictions' average, M is the total number of instances (i.e., the total number of data points across 50 training data sets) used for eCDF calculations, and $\mathbb{1}(\cdot)$ is the indicator function. Figure 4 plots eCDFs for all feature outputs in training and testing modes. In these plots, the eCDFs for test and training cases are almost the same, indicating no under-fitting or over-fitting issue exists. Note that the use of dropout in the GRU is not only for uncertainty quantification in prediction, but also to improve model generalization performance.

In the following, we present prediction results for one of the test cases where its applied normal and shear displacements are plotted in Fig. 5. Normal and shear displacement jumps experience cyclic loading-unloading path and are kept equal in magnitude.

We focus on the average of model predictions in Figs. 6 and 7. In Fig. 6, we see that the initial friction angle is close to 16.7 degree which is almost half of the inter-particle friction angle. This reduction in the overall friction angle might be due to the induced dilation in the normal displacement. Another reason could be related to initial confining pressure: the higher the confining pressure is, the lower the friction angle is. In each loading-unloading branch, the behavior is almost linear without any energy dissipation, but further loading after a level makes the behavior nonlinear. If we only follow the loading path we observe the strain-softening which is the dominant mechanism of a dense granular assemblage; see Figs. 6 and 7b. In other words, the material shows an unstable peak shear strength which is followed by a softening behavior until it reaches the critical state. The sign of changes in normal traction (Fig. 9a) and shear traction (Fig. 9b) are in agreement with the fabric normal (Fig. 10a) and shear (Fig. 10b) components, respectively.



(a) Calibration No. 23 (TXE)

(b) Calibration No. 29 (TXE)

(c) Blind Test No. 50 (TXC)

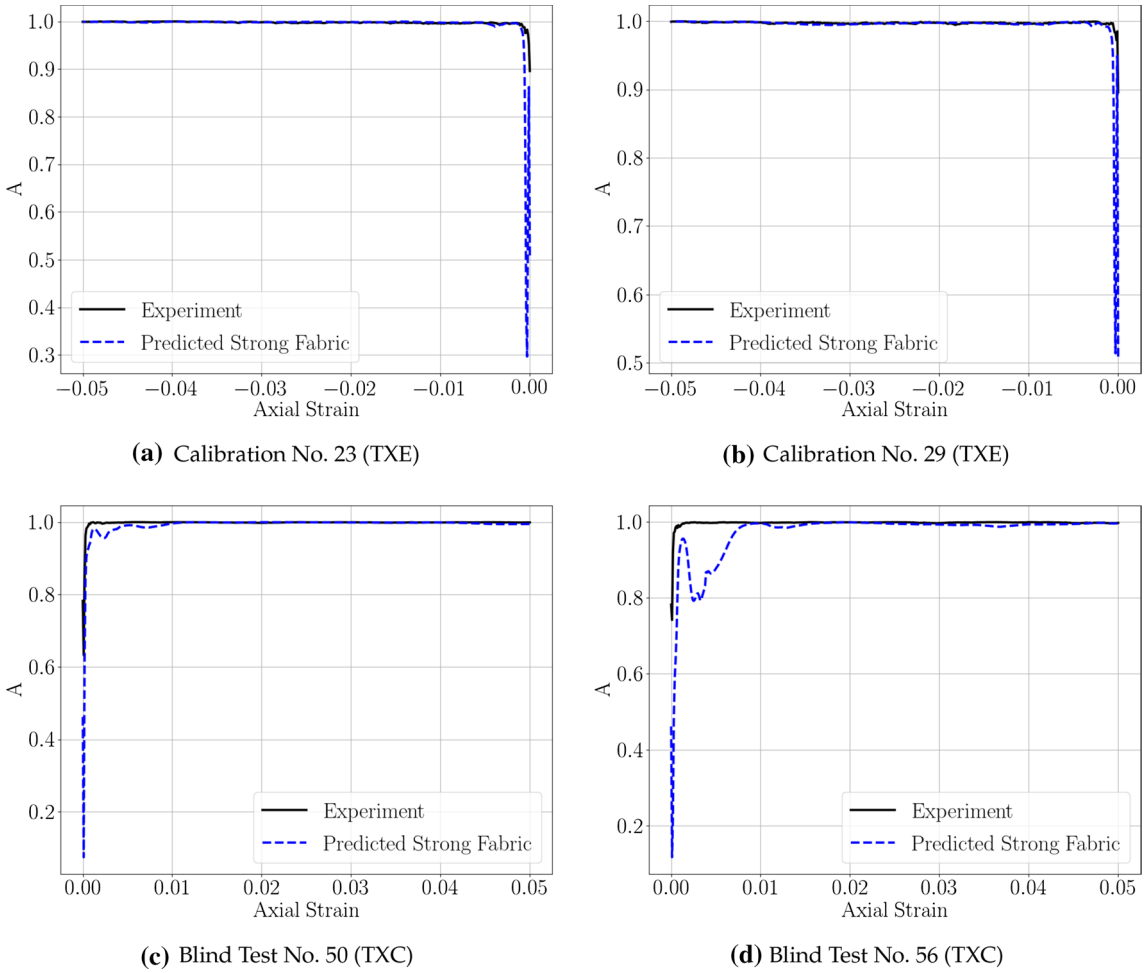
(d) Blind Test No. 56 (TXC)

Fig. 17 State path (void ratio vs. logarithm of mean pressure). Compressive strain has positive sign convention

This confirms the tendency of fabric tensor to trace the load direction [44, 45, 87]. Overall the proposed data-driven scheme can replicate main features of a realistic experiment, and there exists a good agreement between the model and experiment. However, there is an issue corresponding to second loading-unloading cycle where hysteresis is predicted by the model while experiment shows almost zero energy dissipation. This is mainly due to the neural network capacity and design and can be resolved by enriching the neural network architecture with wider neurons or deeper layers or hyper-parameter tuning. Note that one needs to be aware of the over-fitting issue when the model complexity increases by increasing the number of neurons. Generally, a more complex neural network should be trained with more data.

The uncertainty in traction vector prediction is shown in Fig 8. Density distributions of traction vector at three loading steps are plotted in Fig 9. In this figure, steps 25, 47, and 100 belong to the first unloading, second peak, the last peak conditions, respectively (see Fig. 8). Figure 8 suggests that the model is able to track the path-dependent behavior of experiments with narrow variation bands in most of the loading steps. This figure also suggests that the uncertainty for shear traction is higher than normal traction and increases at peak loads. We know that, from mechanics, the shear mode of deformation is more complex and nonlinear than the normal mode and consequently deserves higher uncertainty, which agrees with these results (also see step 47 and 100 in Fig. 9 for a quantitative comparison). At peak values, the complexity is more profound due to the cyclic loading or softening, so more uncertainty is expected.

Model prediction for fabric tensor is plotted in Fig. 10. Density distributions of fabric tensor at three loading steps are plotted in Fig. 11. The uncertainty in fabric tensor has narrow variation bands in most of the loading steps. Similar to the traction prediction, the uncertainties in normal, shear, and mixed modes are higher at peak loads due to the cyclic loading or softening. However, comparing the normal, shear, and mixed modes, we do not observe significant differences in uncertainty at the three loading steps (Fig. 11). Interestingly, we observe that traction predictions have less



(a) Calibration No. 23 (TXE)

(b) Calibration No. 29 (TXE)

(c) Blind Test No. 50 (TXC)

(d) Blind Test No. 56 (TXC)

Fig. 18 Normalized fabric anisotropy variable vs. axial strain. Compressive strain has positive sign convention

uncertainty at initial load steps, step 0 to 20, comparing to fabric while fabric is an intermediate node for traction prediction. This means that traction prediction is potentially less dependent on the fabric at initial loading steps and neural network weights are automatically adjusted to make predictions with high confidence as much as possible by an appropriate combination of porosity, displacement, and fabric. We observe an almost linear correlation between normal displacement jump and porosity, so we have not presented porosity prediction results due to its simplicity. Such a correlation is expected since the normal displacement is the boundary condition in this problem, and dilation is explicitly controlled during the experiment.

6.2 Numerical example 2: machine learning hypoplasticity

In the second numerical experiment, we attempt to generate a predictive surrogate model for one numerical granular assembly undergoing monotonic true triaxial compression loading. For convenient purpose, discrete element simulations are used as replacement of physical tests. These discrete element simulations are run via the open-source software YADE [72].

In total, we conduct 60 true triaxial compression tests with loading path that varying the principle stress σ_1, σ_2 and σ_3 are performed on the same numerical specimen. Before the shearing phase, the material is subjected to hydrostatic loading to compress the assembly hydrostatically to reach the initial confining pressure. Following this step, a vertical compression or extension or a change of the applied tractions on the side walls are prescribed to generate different stress paths. To facilitate third-party validation and re-production of the simulation results, the data used for the causal discovery are given access to the public via Mendeley Data [85].

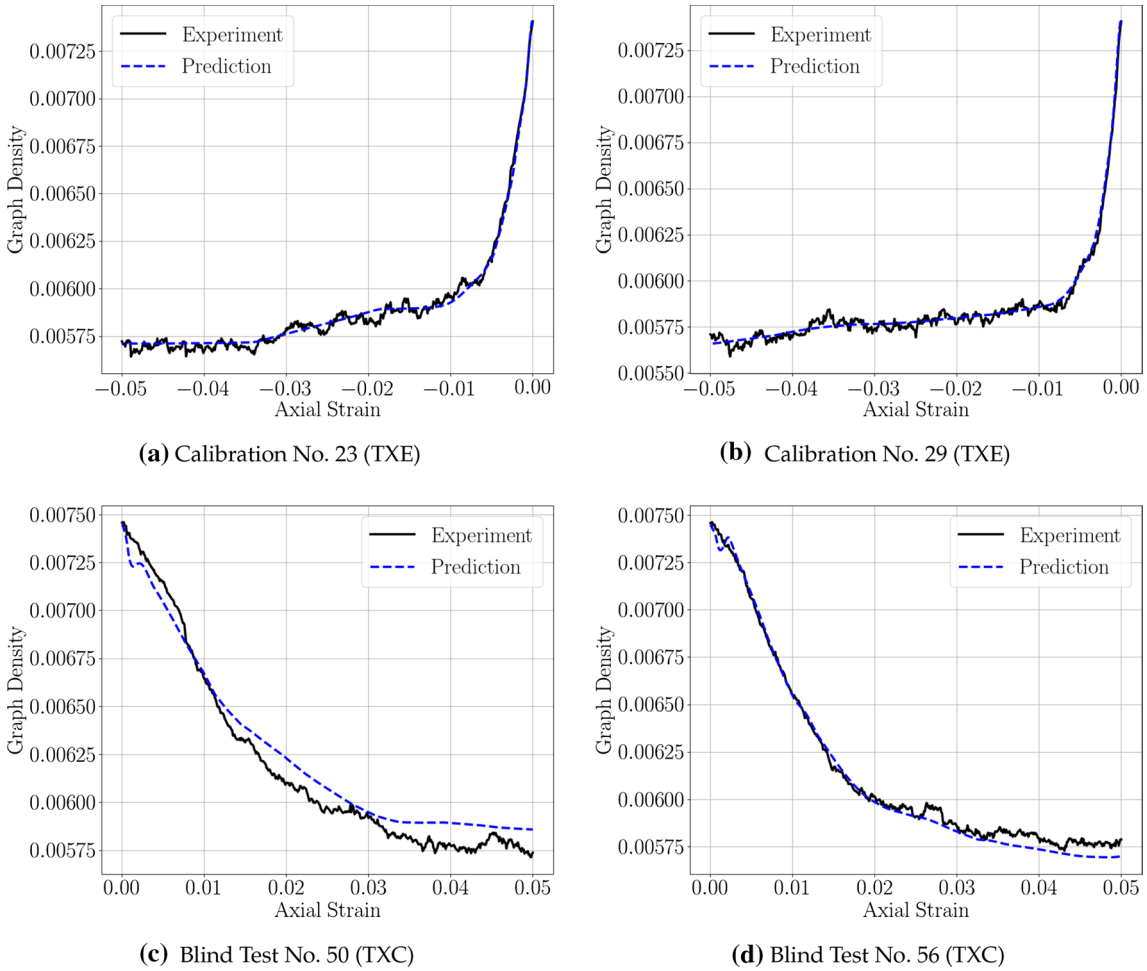


Fig. 19 Graph density vs. axial strain. Compressive strain has positive sign convention

6.2.1 Data-driven causal relations of granular matter

Figure 12 shows the final causal graph of the causal discovery algorithm applied to the true triaxial test data generated from discrete element simulations. The number on each edge represents the edge inclusion probability in the calibration experimental data sets.

The causal discovery driven by the small set of calibration data reveals a number of key observations that are worth-noticing. First, the causal discovery algorithm does re-discover the conventional wisdoms, such as the fact that (1) the changes of coordination number is due to the expansion of the void space; (2) both the coordination number and the porosity changes may cause changes on the fabric tensors; and (3) the dominate role of the strong fabric tensors on the resultant stress. These observations are consistent with previous findings in a number of discrete element simulation literature [41, 71, 77, 94] and the anisotropic critical state theory [17, 45, 100].

In addition to the rediscoveries of known knowledge, the causal discovery algorithm also finds a few causal relationships not known in the existing literature (to the best knowledge of the authors). For instance, the causal discovery algorithm is able to establish a casual relationship that changes in average clustering coefficient may affect the local efficiency of the particle connectivity, whereas the degree of assortativity coefficient, a measure of the similarity of the connections of the graphs, may affect the graph transitivity. Interestingly, the causal discovery algorithm also finds that changes of the strong fabric tensor may be caused by changes of the strain (93%), porosity (60%), coordination number (73%), graph density (73%), local efficiency (70%) and graph transitivity (70%), degree of assortativity (70%) as well as graph clique number (73%). This discovery indicates that the changes of the strong fabric tensors are driven by the changes of the underlying connectivity topology and the volume changes of the void space.

Furthermore, another interesting discovery is that the changes of the stress tensor is only conditionally

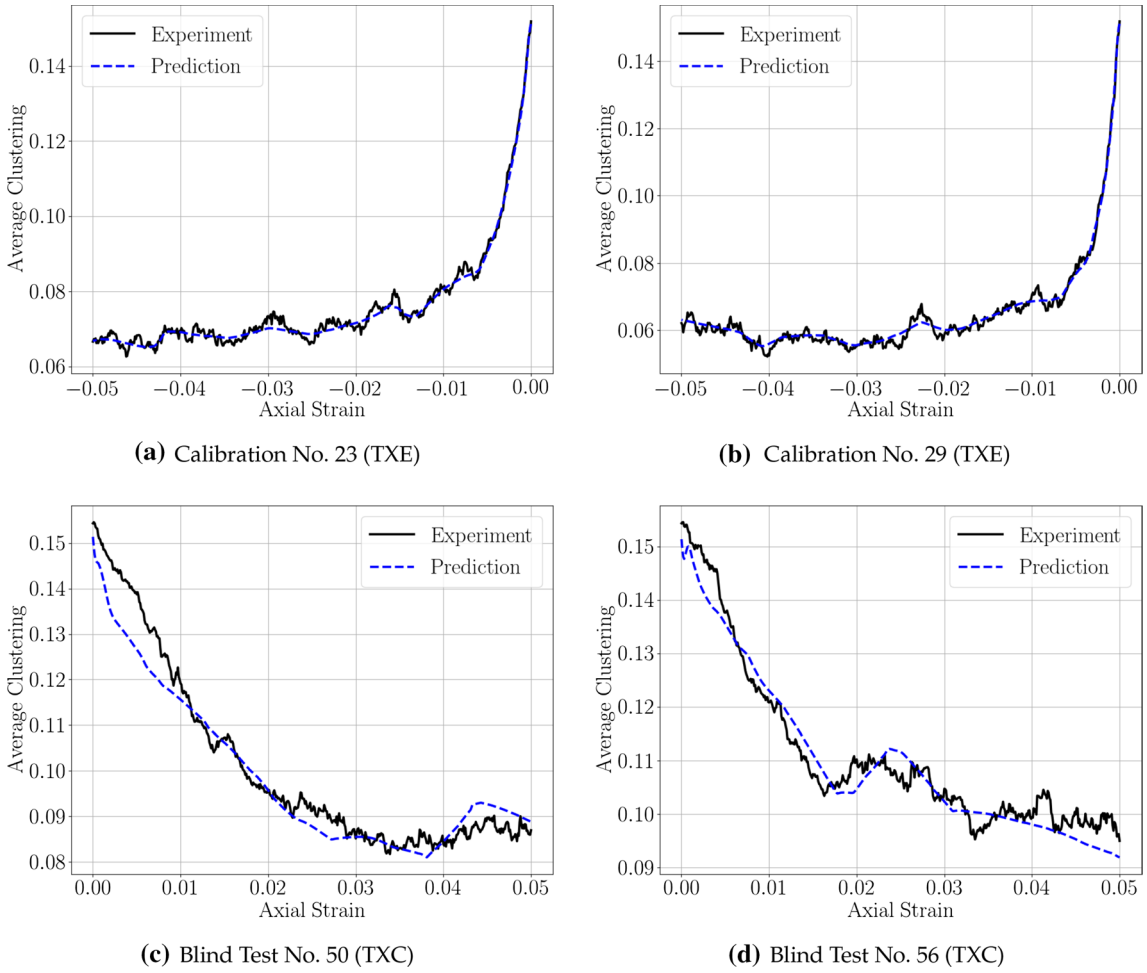


Fig. 20 Average Clustering vs. axial strain. Compressive strain has positive sign convention

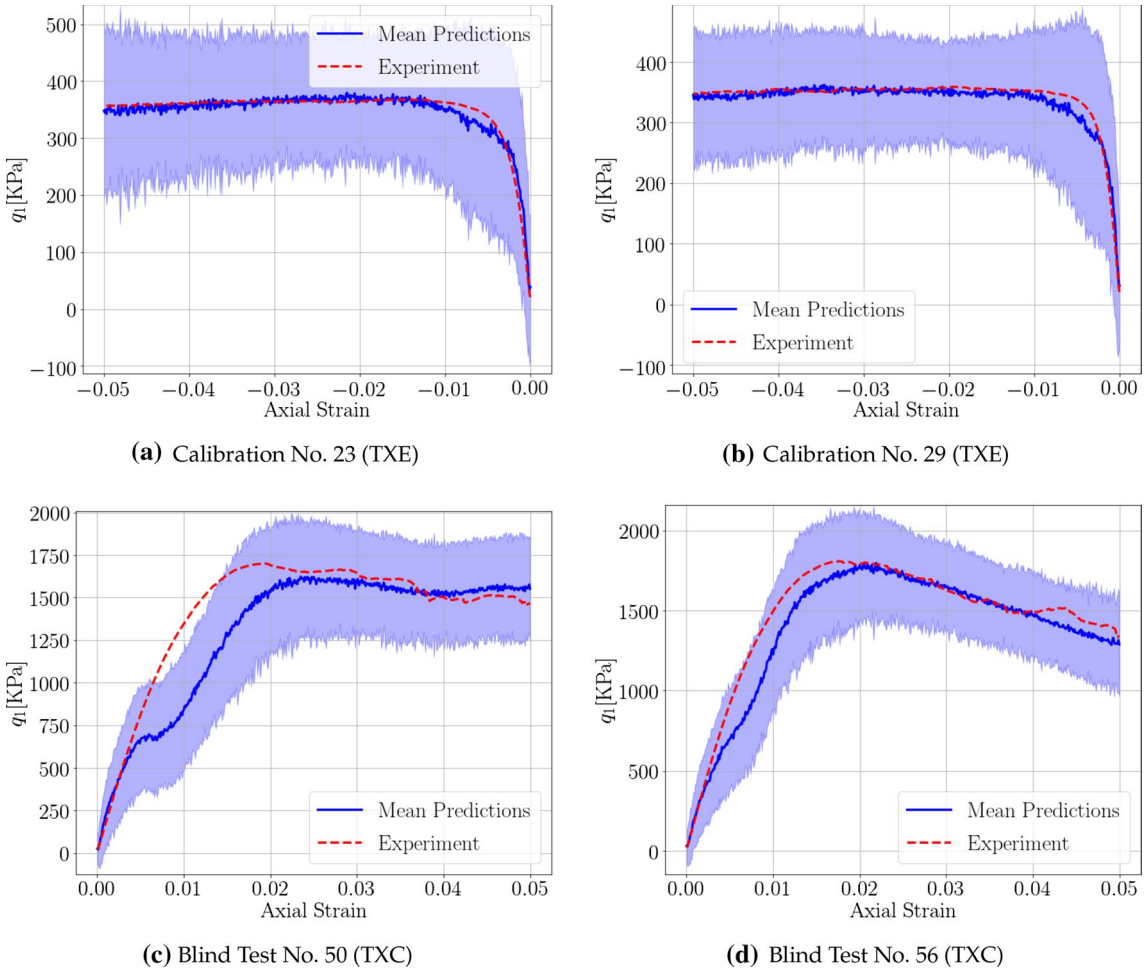
independently caused by the changes of the strong fabric tensor. This result is consistent with the previous finding of 2D granular materials reported in [71] where it is shown that (1) the principal direction of the strong fabric tensor (but not necessarily other fabric tensors) is coaxial with the homogenized Cauchy stress, and (2) the fabric tensor and stress tensor are related by a scalar coefficient that may vary according to the mean pressure.

6.2.2 Predictions based on discovered causal relationships

Here we investigate the accuracy, robustness and the limitations of the machine learning predictions generated based on the deduced causal relations. For comparison purposes, we complete the training of two sets of neural networks—one employs the newly discovered causal relationships into the predictions, another one employs only the strain, fabric tensor and porosity to predict the stress. The latter neural network is then used as a controlled experiment for the former one. The supervised learning procedures used to train the two models are identical.

We first do not introduce the usage of dropout layer in the GRU and hence the dropout rate is zero. The hyperparameters are obtained from repeated trial-and-errors and they are summarized in Table 2. All the sub-graph predictions, regardless of the number of input variables, are trained by the neural network with the identical architecture listed in Table 2. After the predictions, we conduct a cross-validation in which the trained neural networks are tasked to predict both the homogenized Cauchy stress obtained from the calibration and testing simulation data. The results are shown in Fig. 13. Unlike the traction-separation law examples, the predicted stress-strain curves for the true triaxial test exhibit profound over-fitting regardless of whether the additional graph metrics are used for the predictions.

The roughly 2-order of difference in stress predictions suggests that either regularization strategy or more data is needed to circumvent the mismatch of accuracy on the calibration and blind prediction data. Notice that expanding the



(a) Calibration No. 23 (TXE)

(b) Calibration No. 29 (TXE)

(c) Blind Test No. 50 (TXC)

(d) Blind Test No. 56 (TXC)

Fig. 21 Difference between the major and minor principal stress vs. axial strain. Results are obtained for active dropout layers. Shaded area includes predictions within 95% confidence interval. Compressive strain has positive sign convention

data set is not difficult for discrete element simulations, it is certainly very difficult to conduct 60 true triaxial tests physically in a typical laboratory. As such, the results indicate the difficulty to create forecast engine to predict stress responses for unconventional stress paths even when the simulations are free of the issues, noises and errors exhibited in physical experiments.

Interestingly, the predictions from the neural network with the new graph measures do not help significantly on the mean errors of predictions. However, a closer examination of the tail of the eCDF on the two testing curves in Fig. 13 does indicate that the neural network armed with the new knowledge produces a less catastrophic worst-case scenario. Figures 14, 15, and 16 show the predicted principal stress difference against the benchmark data, in which $q_1 = \sigma_1 - \sigma_2$, $q_2 = \sigma_1 - \sigma_3$, and $q_3 = \sigma_2 - \sigma_2$ where $\sigma_1 \geq \sigma_2 \geq \sigma_3$ are principal stresses. In both the calibration and the testing cases, the discrepancy of the principal stress difference are minor.

For brevity, we do not intend to present all the 30 forward prediction results. Here, we pick three samples, two calibrations (Test No. 23 and 29) and two blind predictions (Test No. 50 and 56) for close examination. Figures 14, 15, and 16 compare the difference of the three principal stress inferred from the recurrent neural network and obtained from discrete element simulations. In these figures, TXC and TXE denote triaxial compression and extension tests. For simplicity, this test does not contain cyclic loading, as a result, the prediction task is much simpler. Nevertheless, despite of the relatively small data set, the trained neural networks in the causal graph is capable of predicting important characteristics, such as hardening/softening properly. The predictions also exhibit more fluctuation, which is undesirable. However, this can be presumably suppressed with a different set of activation functions and other regularization strategies.

The second important characteristics that warrants attention is the state path in the void ratio vs. logarithm of mean pressure. Here, we consider compressive pressure as positive

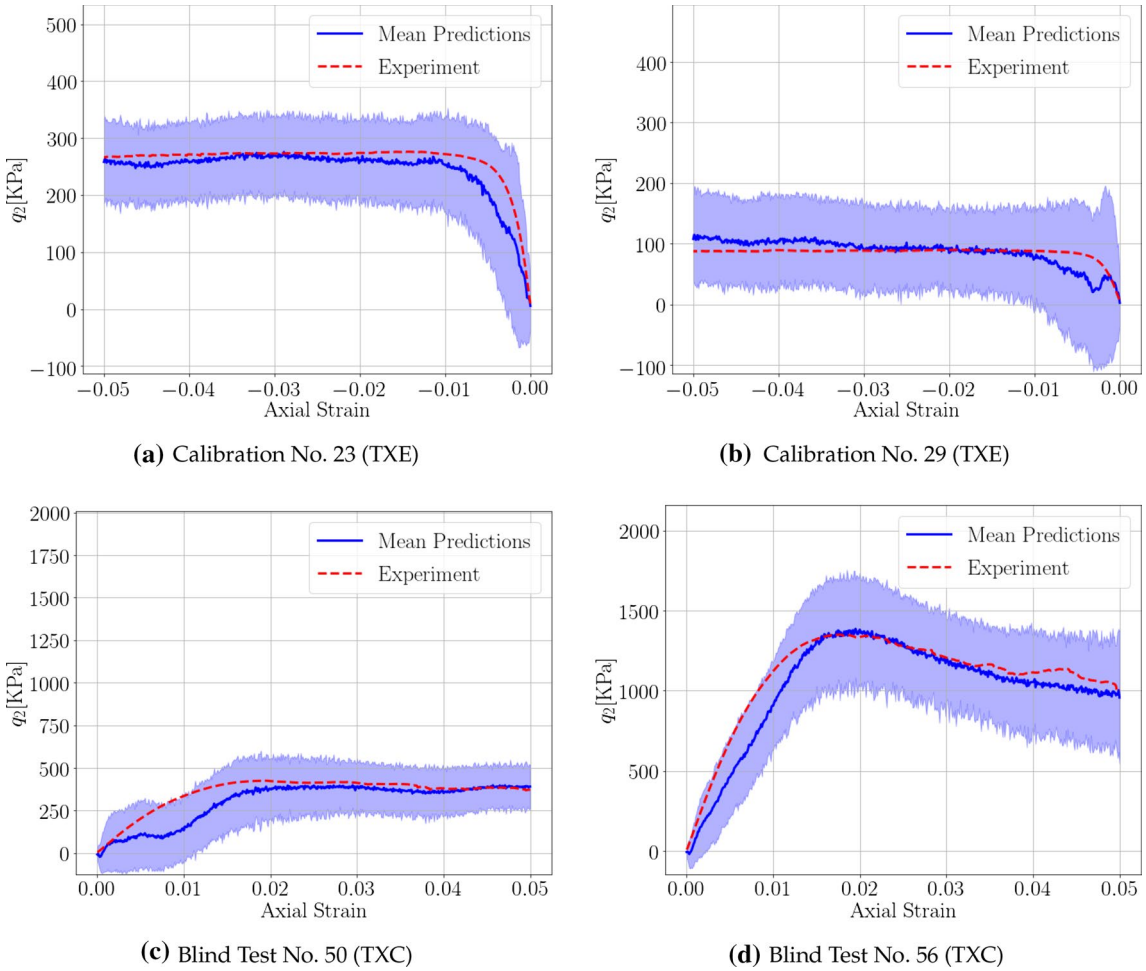


Fig. 22 Difference between the major and immediate principal stress vs. axial strain. Results are obtained for active dropout layers. Shaded area includes predictions within 95% confidence interval. Compressive strain has positive sign convention

and the results are shown in Fig. 17. Again, the predictions indicate that the trained neural network is able to predict the elastic compression followed by the plastic dilatancy in the triaxial compression (TXC) cases and the elastoplastic expansion in the triaxial extension (TXE) cases.

Next, we examine the strong fabric tensor and its relationships with graph measures. Here, the fabric tensor $\boldsymbol{F}$ is computed by the summation of the dyadic product of the branch vectors $\boldsymbol{n}$ divided by the number of grain contacts n_c, i.e.,

$$\boldsymbol{F} = \frac{1}{n_c} \sum_{i=1}^{n_c} \boldsymbol{n} \otimes \boldsymbol{n}. \tag{18}$$

The strong fabric tensor is obtained by considering only a subset of the contact of which the contact normal force is larger than a threshold value. In this work, this threshold value is set to be the averaged contact force. For brevity, we only show the normalized fabric anisotropy variable, which measures the alignment between the fabric tensor and the normalized deviatoric component of the stress $\boldsymbol{n}^{\text{dev}}$,

$$A = \frac{1}{\sqrt{\boldsymbol{F} : \boldsymbol{F}}} \boldsymbol{F} : \boldsymbol{n}^{\text{dev}}, \tag{19}$$

in Fig. 18. Recall that the normalized fabric anisotropy variable $A = 1$ is a necessary condition for a material to reach the critical state [16, 45, 100], hence the predictions of A may indicate how accurate the neural network in the causal graph predicts the critical state. Comparing the predictions of A in the calibration cases and blind tests indicates that the neural network prediction tends to delay the predicted onset of the critical state. This may explain the over-fitting exhibited in Fig. 13.

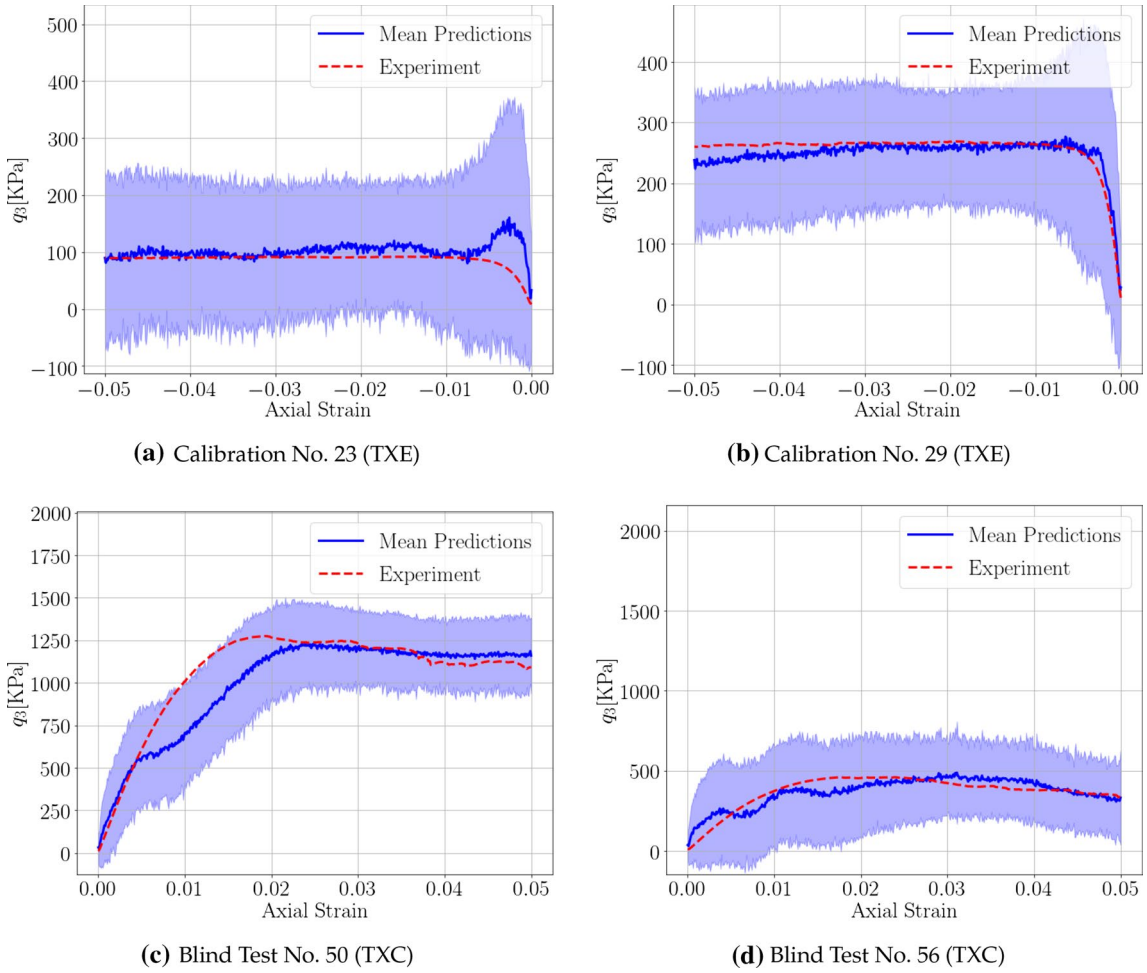


Fig. 23 Difference between the immediate and minor principal stress vs. axial strain. Results are obtained for active dropout layers. Shaded area includes predictions within 95% confidence interval. Compressive strain has positive sign convention

Finally, we examine the predictions of the graph measures most likely to be influential to the predictions of the fabric tensors. Figure 19 shows that the graph density reduces during the shear phase in both triaxial compression and extension tests. According to the causal graph, the deformation is causing the graph density changing which in turn affects the fabric tensors. These causal relationships seem reasonable as the deformation during the shear phase is likely to cause plastic dilatancy and therefore reduces the number of contacts, which explains the drop in the graph density and the resultant changes in fabric tensors.

A similar reasoning can also be used to explain the drops in the average clustering shown in Fig. 20 where the shear deformation tends to reduce the tendency of the particles to cluster together and that in return leads to the evolution of the fabric tensor.

These results indicate that, while the causal discovery may reveal potential causal relations not apparent to domain experts, the knowledge from causal relationships does not necessarily lead to more accurate predictions. Factors such as the choices of the supervised machine learning methods and the availability of data are also key factors that affect the usefulness of the new knowledge for predictions.

6.2.3 Uncertainty propagation with dropout layer

As the final numerical experiment, we activate dropout layers to collect results of stochastic forward passes through the model. This gives us a Monte Carlo estimate of the predictions. Note that the activation of the dropout layers will lead to a different set of neuron weights even the data used for the training of the neural network are identical.

Figures 21, 22, and 23 show the confidence interval for 200 Monte Carlo predictions of principal stress differences, q_1, q_2 and q_3 vs. axial strain for 4 selected triaxial extension and compression loading paths.

In most of the cases shown in Figs. 21, 22, and 23, the mean paths of the stochastic predictions generated by the

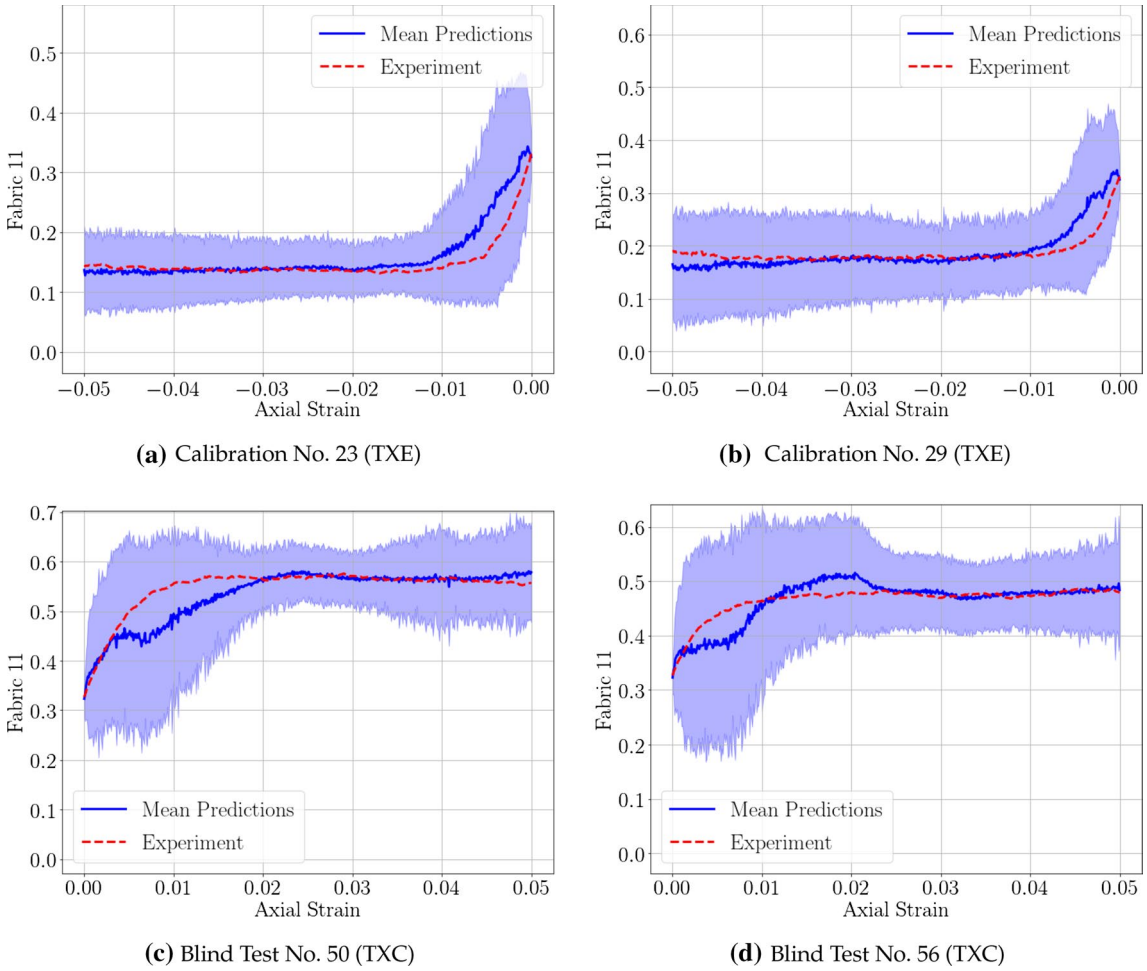


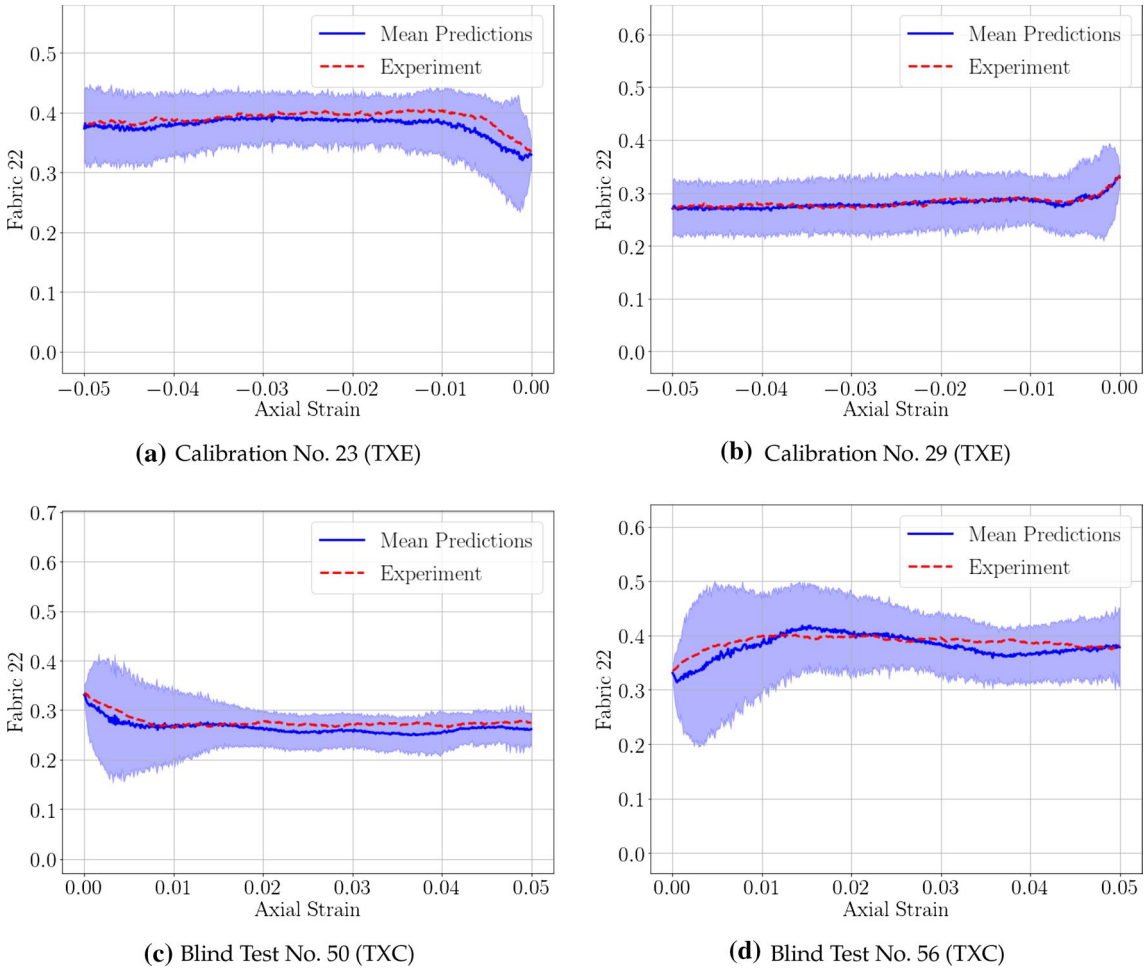
Fig. 24 Component 11 of fabric tensor vs. axial strain. Results are obtained for active dropout layers. Shaded area includes predictions within 95% confidence interval. Compressive strain has positive sign convention

dropout layer is able to match qualitatively with the experimental benchmarks. Furthermore, in most cases, the principal stress differences observed from experiments are within the 95% confidence interval. It should nevertheless be noted that the blind test is less accurate than the calibration cases, indicating that the neural networks may have been over-fitted.

To examine how uncertainty is propagated in the causal graph, we plot the diagonal components of the fabric tensor and the results are shown in Figs. 24, 25, and 26. For brevity, the off-diagonal components of the fabric tensor, which are much smaller than the diagonal components, are not provided here. Comparing the 95% confidence interval of the fabric tensor and that of the principal stress difference, one can easily see that the predictions of stress tend to be more accurate when the fabric tensor can be more precisely determined with a narrower confidence interval.

7 Conclusions

In this paper, we introduce, for the first time, a data-driven framework that combines (1) the causal discovery algorithm that detects unknown causal relations, (2) the Bayesian approximation for uncertainty quantification enabled by the dropout technique, and (3) the recurrent neural network technique to analyze, interpret, and forecast the path-dependent responses of granular materials. Numerical experiments conducted on idealized granular system have indicated that the data-driven framework is able to investigate and discover new hidden causal relationships and propagate uncertainty generated from a sequence of structured neural network predictions within a casual graph. This approach has potentials to help modelers and experimentalists to spot hidden mechanisms not apparent to human eyes as well as deduce complex casual relationships in a high-dimensional parametric space where intuition and domain knowledge are not sufficient due to the dimensionality of the data. Further work may include



(a) Calibration No. 23 (TXE)

(b) Calibration No. 29 (TXE)

(c) Blind Test No. 50 (TXC)

(d) Blind Test No. 56 (TXC)

Fig. 25 Component 22 of fabric tensor vs. axial strain. Results are obtained for active dropout layers. Shaded area includes predictions within 95% confidence interval. Compressive strain has positive sign convention

improvement and comparisons of different causal inferences, *extension to recover causal relations when both instantaneous and lagged causal relations exist,* as well as the applications to more complex granular systems where particles are of different shapes and properties.

Appendix: Proof of Theorem 1

Theorem 2 *Given Assumptions 1–3, for every* $V_i, V_j \in \boldsymbol{V}_{-\boldsymbol{U}}$, V_i *and* V_j *are not adjacent in* G *if and only if they are independent conditional on some subset of* $\{V_k \mid V_k \in \boldsymbol{V}_{-\boldsymbol{U}}, k \neq i, k \neq j\} \cup \{\boldsymbol{U}\}$.

Proof From equation (1), any variable V_i in $\boldsymbol{V}_{-\boldsymbol{U}}$ can be written as a function of $\{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1}$ and $\{\epsilon_i\}_{i=1}^{m-1}$, where $m-1$ is the number of vertices included in $\boldsymbol{V}_{-\boldsymbol{U}}$ since $\boldsymbol{V}$ includes m vertices. Therefore, the distribution of $\boldsymbol{V}_{-\boldsymbol{U}}$ at each value of $\boldsymbol{U}$ is determined by the distribution of $\epsilon_1, \ldots, \epsilon_{m-1}$ and the values of $\{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1}$. For any $V_i, V_j \in \boldsymbol{V}_{-\boldsymbol{U}}$ and $S \subseteq \{V_k \mid V_k \in \boldsymbol{V}_{-\boldsymbol{U}}, k \neq i, k \neq j\}$, $p(V_i, V_j \mid S \cup \{\boldsymbol{U}\})$ is determined by $\prod_{i=1}^{m-1} p(\epsilon_i)$ and $\{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1}$. Since $\prod_{i=1}^{m-1} p(\epsilon_i)$ does not change with $\boldsymbol{U}$, we have

$$p(V_i, V_j \mid S \cup \{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1} \cup \{\boldsymbol{U}\}) = p(V_i, V_j \mid S \cup \{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1}). \tag{20}$$

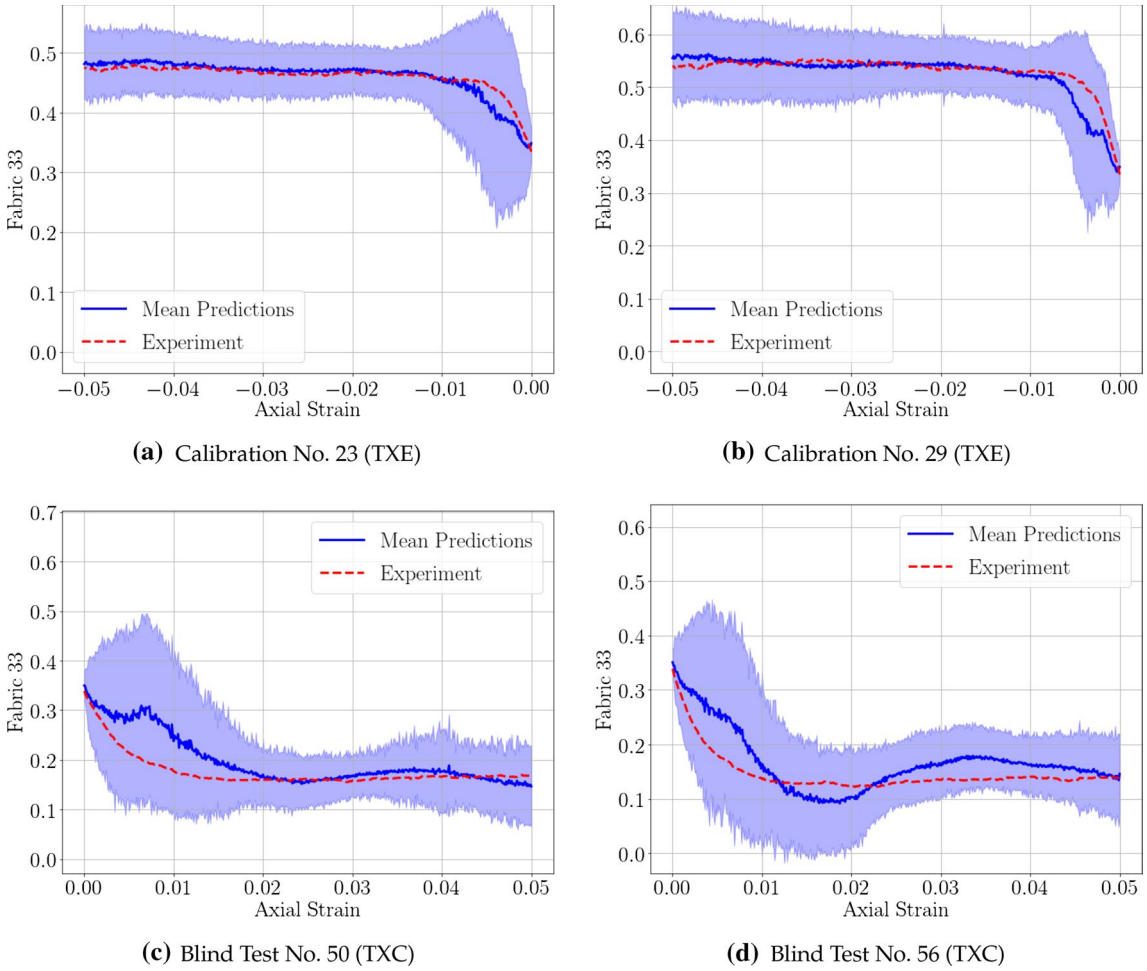
Denote $\perp\!\!\!\perp$ to indicate independence, it follows that

$$\boldsymbol{U} \perp\!\!\!\perp (V_i, V_j) \mid S \cup \{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1}. \tag{21}$$

Applying the weak union property of conditional independence, we have

$$\boldsymbol{U} \perp\!\!\!\perp V_i \mid \{V_j\} \cup S \cup \{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1}.$$

Suppose that V_i and V_j are not adjacent in G, then they are not adjacent in G^{aug}. There exists a set $S \subseteq \{V_k \mid V_k \in \boldsymbol{V}_{-\boldsymbol{U}}, k \neq i, k \neq j\}$ such that $S \cup \{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1}$ d-separates V_i and V_j. Because of Assumption 1, we have



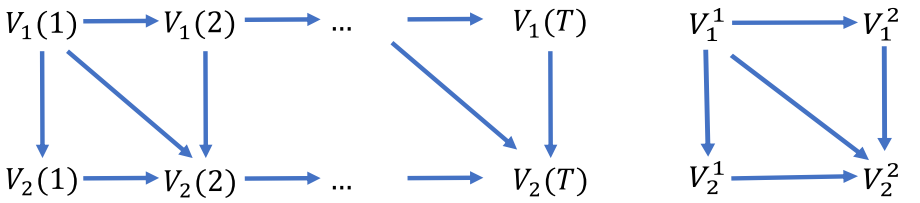
(a) Calibration No. 23 (TXE)

(b) Calibration No. 29 (TXE)

(c) Blind Test No. 50 (TXC)

(d) Blind Test No. 56 (TXC)

Fig. 26 Component 33 of fabric tensor vs. axial strain. Results are obtained for active dropout layers. Shaded area includes predictions within 95% confidence interval. Compressive strain has positive sign convention



(a) Repetitive causal graph.

(b) Unit causal graph.

Fig. 27 An illustration of lagged causal relationships with two vertices in a DAG and the lag time being 1 ($L = 1$)

$$V_i \perp\!\!\!\perp V_j \mid S \cup \{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1}. \tag{22}$$

Since all $\theta_i(\boldsymbol{U})$ are deterministic functions of $\boldsymbol{U}$, we have $p(V_i, V_j \mid S \cup \boldsymbol{U}) = p(V_i, V_j \mid S \cup \{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1} \cup \boldsymbol{U})$.

Equations (21) and (22) imply that

$$V_i \perp\!\!\!\perp (\boldsymbol{U}, V_j) \mid S \cup \{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1}.$$

By the weak union property of conditional independence, we have

$$V_i \perp\!\!\!\perp V_j \mid S \cup \{\theta_i(\boldsymbol{U})\}_{i=1}^{m-1} \cup \{\boldsymbol{U}\}.$$

Since all $\theta_i(\boldsymbol{U})$ are deterministic functions of $\boldsymbol{U}$, it follows that

$$V_i \perp\!\!\!\perp V_j \mid S \cup \{\boldsymbol{U}\}.$$

Now we prove that if V_i and V_j are conditionally independent given a subset S of $\{V_k \mid V_k \in \boldsymbol{V}_{-\boldsymbol{U}}, k \neq i, k \neq j\} \cup \{\boldsymbol{U}\}$, V_i and V_j are not adjacent in G. Because of Assumption 2 (faithfullness), V_i and V_j are not adjacent in G^{aug}. Therefore, they are not adjacent in G. □

Appendix: Graph metric definitions

In this section, we provide brief review of the the terminology of the graph measures obtained from the grain connectivity graph generated in each time step of a discrete element simulation. These graph measures are used to create the knowledge graph for the machine learning constitutive law in Section 6.2.

The following graph metrics were calculated using the open-source software networkX [27] for exploration and analysis of graph networks.

Definition 1 The **degree assortativity coefficient** measures the similarity of the connections in a graph with respect to the node degree.

Definition 2 The **graph transitivity** is the fraction of all possible triangles present in the graph over the number of triads. Possible triangles are identified by the number of triads—two edges with a shared vertex.

Definition 3 The **density** for undirected graphs is defined as:

$$d = \frac{2m}{n(n-1)}, \tag{23}$$

where n in the number of nodes and m is the number of edges of the graph.

Definition 4 The **average clustering coefficient** of the graph is defined as:

$$C = \frac{1}{n}\sum_{v \in G} c_n, \tag{24}$$

where n in the number of nodes and is c_n is the clustering coefficient of node n defined as:

$$c_n = \frac{2T(n)}{\deg(n)(\deg(n)-1)}, \tag{25}$$

where $T(n)$ is the number of triangles passing through node n and $\deg(n)$ is the degree of node n.

Definition 5 A **clique** is a subset of nodes of an undirected graph such that every two distinct nodes in the clique are adjacent. The **graph clique number** is the size of largest clique in the graph.

Definition 6 The **efficiency** of a pair of nodes is defined as the reciprocal of the shortest path distance between the nodes. The **local efficiency** of a node in the graph is the average global efficiency of the subgraph induced by the neighbours of the node. The **average local efficiency**, used in this work, is the average of the local efficiency calculated for every node in the graph.

Appendix: Loading conditions in bulk plasticity experiments

The database used for causal discovery and training of the neural network forecast engines for numerical example in Section 6.2 includes 60 true triaxial numerical experiments conducted via the YADE' DEM simulator. These experiments differ according to the applied axial strain rate $\dot{\epsilon}_{11}$, initial confining pressure p_0, initial void ratio e_0, and a parameter $b = \frac{\sigma_{22}-\sigma_{33}}{\sigma_{11}-\sigma_{33}}$ that controls applied stress conditions. In all 60 cases we set $\dot{\sigma}_{33} = \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$. The setup of them are listed below. The tests with the bold font are the one discussed in Section 6.2. The first 30 test (labelled T0-T29) are used to train the neural network, while T30-T59 are used for forward predictions.

T0 $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -300kPa, e_0 = 0.539.$
T1 $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -400kPa, e_0 = 0.536.$
T2 $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -500kPa, e_0 = 0.534.$
T3 $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -300kPa, e_0 = 0.539.$
T4 $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -400kPa, e_0 = 0.536.$
T5 $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -500kPa, e_0 = 0.534.$
T6 $\dot{\epsilon}_{11} < 0, b = 0.5, p_0 = -300kPa, e_0 = 0.539.$
T7 $\dot{\epsilon}_{11} < 0, b = 0.5, p_0 = -400kPa, e_0 = 0.536.$
T8 $\dot{\epsilon}_{11} < 0, b = 0.5, p_0 = -500kPa, e_0 = 0.534.$
T9 $\dot{\epsilon}_{11} > 0, b = 0.5, p_0 = -300kPa, e_0 = 0.539.$
T10 $\dot{\epsilon}_{11} > 0, b = 0.5, p_0 = -400kPa, e_0 = 0.536.$
T11 $\dot{\epsilon}_{11} > 0, b = 0.5, p_0 = -500kPa, e_0 = 0.534.$
T12 $\dot{\epsilon}_{11} < 0, b = 0.1, p_0 = -300kPa, e_0 = 0.539.$
T13 $\dot{\epsilon}_{11} < 0, b = 0.1, p_0 = -400kPa, e_0 = 0.536.$
T14 $\dot{\epsilon}_{11} < 0, b = 0.1, p_0 = -500kPa, e_0 = 0.534.$
T15 $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -300kPa, e_0 = 0.539.$
T16 $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -400kPa, e_0 = 0.536.$

T17 $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -500kPa, e_0 = 0.534.$
T18 $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -300kPa, e_0 = 0.539.$
T19 $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -400kPa, e_0 = 0.536.$
T20 $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -500kPa, e_0 = 0.534.$
T21 $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -300kPa, e_0 = 0.539.$
T22 $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -400kPa, e_0 = 0.536.$
T23 $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -500kPa, e_0 = 0.534.$
T24 $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -300kPa, e_0 = 0.539.$
T25 $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -400kPa, e_0 = 0.536.$
T26 $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -500kPa, e_0 = 0.534.$
T27 $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -300kPa, e_0 = 0.539.$
T28 $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -400kPa, e_0 = 0.536.$
T29 $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -500kPa, e_0 = 0.534.$
T30 $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -350kPa, e_0 = 0.539.$
T31 $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -450kPa, e_0 = 0.536.$
T32 $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -550kPa, e_0 = 0.534.$
T33 $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -350kPa, e_0 = 0.539.$
T34 $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -450kPa, e_0 = 0.536.$
T35 $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -550kPa, e_0 = 0.534.$
T36 $\dot{\epsilon}_{11} < 0, b = 0.5, p_0 = -350kPa, e_0 = 0.539.$
T37 $\dot{\epsilon}_{11} < 0, b = 0.5, p_0 = -450kPa, e_0 = 0.536.$
T38 $\dot{\epsilon}_{11} < 0, b = 0.5, p_0 = -550kPa, e_0 = 0.534.$
T39 $\dot{\epsilon}_{11} > 0, b = 0.5, p_0 = -350kPa, e_0 = 0.539.$
T40 $\dot{\epsilon}_{11} > 0, b = 0.5, p_0 = -450kPa, e_0 = 0.536.$
T41 $\dot{\epsilon}_{11} > 0, b = 0.5, p_0 = -550kPa, e_0 = 0.534.$
T42 $\dot{\epsilon}_{11} < 0, b = 0.1, p_0 = -350kPa, e_0 = 0.539.$
T43 $\dot{\epsilon}_{11} < 0, b = 0.1, p_0 = -450kPa, e_0 = 0.536.$
T44 $\dot{\epsilon}_{11} < 0, b = 0.1, p_0 = -550kPa, e_0 = 0.534.$
T45 $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -350kPa, e_0 = 0.539.$
T46 $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -450kPa, e_0 = 0.536.$
T47 $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -550kPa, e_0 = 0.534.$
T48 $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -350kPa, e_0 = 0.539.$
T49 $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -450kPa, e_0 = 0.536.$
T50 $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -550kPa, e_0 = 0.534.$
T51 $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -350kPa, e_0 = 0.539.$
T52 $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -450kPa, e_0 = 0.536.$
T53 $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -550kPa, e_0 = 0.534.$
T54 $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -350kPa, e_0 = 0.539.$
T55 $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -450kPa, e_0 = 0.536.$
T56 $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -550kPa, e_0 = 0.534.$
T57 $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -350kPa, e_0 = 0.539.$
T58 $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -450kPa, e_0 = 0.536.$
T59 $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -550kPa, e_0 = 0.534.$

Appendix: Extension to lagged causal relationships

In this section, we briefly discuss an extension of the proposal causal discovery approach to allow both instantaneous and lagged causal relations. Without loss of generality, we use the example of traction-separation law to illustrate the method. Recall that $\boldsymbol{V}_{-\boldsymbol{U}}$ includes all other variables in $\boldsymbol{V}$ excluding $\boldsymbol{U}$ (e.g., porosity, fabric tensor). Assume that there are m vertices in $\boldsymbol{V}_{-\boldsymbol{U}}$, and we denote the values of vertices at the tth time point to be $\boldsymbol{V}_{-\boldsymbol{U}}(t) = (V_1(t), \dots, V_m(t))$, where $t = 1, \dots, T$. Since there exist both instantaneous and lagged causal relations over these variables, we assume that the largest lag time to be L. Furthermore, we denote the ith variable from the lth time point to the $(T - L + l - 1)$th time point to be $V_i^l = (V_i(l), V_i(l+1), \dots, V_i(T - L + l - 1))$, $i = 1, \dots, m$ and $l = 1, \dots, L + 1$. Then we introduce a new set of variables $\tilde{\boldsymbol{V}}_{-\boldsymbol{U}} = \{\tilde{\boldsymbol{V}}^1\}_{l=1}^{L+1}$, where $\tilde{\boldsymbol{V}}^l = \{V_1^l, V_2^l, \dots, V_m^l\}$. Fig. 27 illustrates the lagged causal relationships using a DAG with only two vertices. Fig. 27(a) shows the repetitive causal graph over two time series $\boldsymbol{V}(t) = (V_1(t), V_2(t))$, $t = 1, \dots, T$ when the lag time $L = 1$. Fig. 27(b) shows the unit causal graph over the newly introduced variables $\tilde{\boldsymbol{V}} = \tilde{\boldsymbol{V}}^1 \cup \tilde{\boldsymbol{V}}^2$. In this case, our goal is to not only recover the instantaneous causal relations between V_1^2 and V_2^2 as what did in the main manuscript, but also the lagged causal relations from V_1^1 to V_2^2, and from V_1^1 to V_1^2.

Recall that $\boldsymbol{U}$ is the known input (i.e., displacement jump), and we have the prior knowledge that the dynamic changes in $\boldsymbol{U}$ can cause changes in other variables, not vice versa. Therefore, $\boldsymbol{U}$ can be used as a surrogate variable to help identify the causal relations. To recover the causal skeleton for both instantaneous and lagged causal relations, we modify Algorithm 1 as follows. Firstly, a complete undirected graph U_G is built with the variable $\boldsymbol{U}$ and the newly introduced variables $\tilde{\boldsymbol{V}}_{-\boldsymbol{U}}$. Then for each i, we test for the marginal and conditional independence between V_i^{L+1} and $\boldsymbol{U}$, $i = 1, \dots, m$. If they are independent, the edge between V_i^{L+1} and $\boldsymbol{U}$ is removed; meanwhile the edge between V_i^l and $\boldsymbol{U}$ is also removed, $l = 1, \dots, L$. Next, we recover lagged causal relations by testing the marginal and conditional independence between the variables in $\tilde{\boldsymbol{V}}^{L+1}$ and the variables in $\tilde{\boldsymbol{V}}^{L-l+1}$ for each lth lagged relation, $l = 1, \dots, L$. In particular, if V_i^{L+1} and V_j^{L-l+1} are independent, their edge is removed from U_G; meanwhile the edge between V_i^{L-k+1} and $V_j^{L-l+1-k}$ is removed, $k = 0, \dots, L - l$. Lastly, we recover the instantaneous causal relations by testing for the marginal and conditional independence between V_i^{L+1} and V_j^{L+1}, $i \neq j$. If they are independent, their edge is removed; meanwhile the edge between V_i^k and V_j^k is removed, $k = 1, \dots, L$. After the causal skeleton is determined, we can apply Algorithm 2 in the paper to recover the causal directions for instantaneous causal relations. For lagged ones, they follow the rule that past causes future.

Acknowledgements The members of the Columbia research group involved in this research are supported by National Science Foundation under Grant contracts CMMI-1846875 and OAC-1940203, the Earth Materials and Processes program from the US Army Research Office under Grant contract W911NF-18-2-0306, and the Dynamic Materials and Interactions Program from the Air Force Office of Scientific

Research under Grant contracts FA9550-17-1-0169. The members of the Johns Hopkins University are supported by National Science Foundation under Grant contract 1940107. These supports are gratefully acknowledged. The authors would also like to thank Dr. Kun Wang from Los Alamos National Laboratory for providing the data for the traction-separation law.

The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the sponsors, including the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Author Contributions Xiao Sun and Bahador Bahmani contribute equally as first authors. All authors have contributed to the planning/writing/reviewing/editing of this manuscript.

Data Availability Statement The causal discovery algorithm can be found at [79]. The recurrent neural network is built via Tensorflow and the code to complete the training and the generation of the forecast engine can be found at [78]. The discrete element simulations data can be found in the Mendeley data repositories [76, 85].

Declaration

Conflict of interest The authors confirm that there are no relevant financial or non-financial competing interests to report.

References

1. Bagi, K.: Stress and strain in granular assemblies. Mech. Mater. **22**(3), 165–177 (1996)
2. Bahmani, B.: Sun, WaiChing: A kd-tree-accelerated hybrid data-driven/model-based approach for poroelasticity problems with multi-fidelity multi-physics data. Comput. Methods Appl. Mech. Eng. **382**, 113868 (2021)
3. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**, 281–305 (2012)
4. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)
5. Boluki, S., Ardywibowo, R., Dadaneh, Siamak Z., Zhou, M., Qian X.: Learnable bernoulli dropout for bayesian deep learning. arXiv preprint arXiv:2002.05155 (2020)
6. Borja, R.I., Sun, W.C.: Estimating inelastic sediment deformation from local site response simulations. Acta Geotech. **2**(3), 183 (2007)
7. Borja, R.I., Sun, W.C.: Coseismic sediment deformation during the, loma prieta earthquake. J. Geophys. Res.: Solid Earth **113**(B8), 2008 (1989)
8. Bryant, E.C., Sun, W.C.: A micromorphically regularized cam-clay model for capturing size-dependent anisotropy of geomaterials. Comput. Methods Appl. Mech. Eng. **354**, 56–95 (2019)
9. Chung, J., Gulcehre, C., Cho, K., Kyung H., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
10. Clément, A., Soize, C., Yvonnet, J.: Uncertainty quantification in computational stochastic multiscale analysis of nonlinear elastic materials. Comput. Methods Appl. Mech. Eng. **254**, 61–82 (2013)
11. Cui, R., Groot, P., Heskes, T.: Copula PC algorithm for causal discovery from mixed data. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 377–392. Springer (2016)
12. Cundall, P.A., Strack, O.D.L.: A discrete numerical model for granular assemblies. Geotechnique **29**(1), 47–65 (1979)
13. Dafalias, Y.F.: Modelling cyclic plasticity: simplicity versus sophistication. Mech. Eng. Mater. 153178 (1984)
14. Dafalias, Y.F., Manzari, M.T.: Simple plasticity sand model accounting for fabric change effects. J. Eng. Mech. **130**(6), 622–634 (2004)
15. Frankel, A.L., Jones, R.E., Alleman, C., Templeton, J.A.: Predicting the mechanical response of oligocrystals with deep learning. Comput. Mater. Sci. **169**, 109099 (2019)
16. Pengcheng, F., Dafalias, Y.F.: Fabric evolution within shear bands of granular materials and its relation to critical state theory. Int. J. Numer. Anal. Methods Geomech. **35**(18), 1918–1948 (2011)
17. Pengcheng, F., Dafalias, Y.F.: Relationship between void-and contact normal-based fabric tensors for 2d idealized granular materials. Int. J. Solids Struct. **63**, 68–81 (2015)
18. Fuchs, A., Heider, Y., Wang, K., Sun, W.C., Kaliske, M.: Dnn2: A hyper-parameter reinforcement learning game for self-design of neural network based elasto-plastic constitutive descriptions. Comput. Struct. **249**, 106505 (2021). https://doi.org/10.1016/j.compstruc.2021.106505.
19. Furukawa, T., Yagawa, G.: Implicit constitutive modelling for viscoplasticity using neural networks. Int. J. Numer. Meth. Eng. **43**(2), 195–219 (1998)
20. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning, pp. 1050–1059, (2016a)
21. Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In: Advances in Neural Information Processing Systems, pp. 1019–1027, (2016b)
22. Ghassami, A., Kiyavash, N., Huang, B., Zhang, K.: Multi-domain causal structure learning in linear systems. In: Advances in Neural Information Processing Systems, pp. 6266–6276, (2018)
23. Gong, M., Zhang, K., Schölkopf, B., Glymour, C., Tao, D.: Causal discovery from temporally aggregated time series. In: Uncertainty in Artificial Intelligence: Proceedings of the Conference. Conference on Uncertainty in Artificial Intelligence, volume 2017. NIH Public Access, (2017)
24. Granger, C.W.J.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica J. Econom. Soc. **73**(3), 424–438 (1969)
25. Gretton, A., Fukumizu, K., Teo, L.S., Choon H., Schölkopf, B., Smola, A.J.: A kernel statistical test of independence. In: Advances in Neural Information Processing Systems, pp. 585–592, (2008)
26. Gurson, A.L.: Continuum theory of ductile rupture by void nucleation and growth: Part I—Yield criteria and flow rules for porous ductile media. J. Eng. Mater. Technol. **99**(1), 2–15 (1977)
27. Hagberg, A., Swart, P., Chult, D.S.: Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008)
28. He, Q., Chen, J.-S.: A physics-constrained data-driven approach based on locally convex reconstruction for noisy database. Comput. Methods Appl. Mech. Eng. **363**, 112791 (2020)
29. Heckerman, D., Geiger, D., Chickering, D.M.: Learning bayesian networks: The combination of knowledge and statistical data. Mach. Learn. **20**(3), 197–243 (1995)
30. Heider, Y., Wang, K., Sun, W.C.: So (3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials. Comput. Methods Appl. Mech. Eng. **363**, 112875 (2020)

31. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, (2012)
32. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
33. Huang, B., Zhang, K., Lin, Y., Schölkopf, B., Glymour, C.: Generalized score functions for causal discovery. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1551–1560, (2018)
34. Huang, B., Zhang, K., Gong, M., Glymour, C.: Causal discovery and forecasting in nonstationary environments with state-space models. Proc. Mach. Learn. Res. **97**, 2901 (2019a)
35. Huang, B., Zhang, K., Zhang, J., Ramsey, J., Sanchez-Romero, R., Glymour, C., Schölkopf, B.: Causal discovery from heterogeneous/nonstationary data.CoRR, abs/1903.01672, (2019b)
36. Huang, B., Zhang, K., Zhang, J., Ramsey, J., Sanchez-Romero, R., Glymour, C., Schölkopf, B.: Causal discovery from heterogeneous/nonstationary data. J. Mach. Learn. Res. **21**(89), 1–53 (2020)
37. Ishwaran, H., Rao, J.S.: Spike and slab variable selection: frequentist and bayesian strategies. Ann. Stat. **33**(2), 730–773 (2005)
38. Karapiperis, K., Stainier, L., Ortiz, M., Andrade, J.E.: Data-driven multiscale modeling in mechanics. J. Mech. Phys. Solids **147**, 104239 (2021)
39. Kirchdoerfer, T., Ortiz, M.: Data-driven computational mechanics. Comput. Methods Appl. Mech. Eng. **304**, 81–101 (2016)
40. Klein, A., Falkner, S., Bartels, S., Hennig, P., Hutter, F.: Fast bayesian optimization of machine learning hyperparameters on large datasets. In: Artificial Intelligence and Statistics, pp. 528–536 (2017)
41. Kuhn, M.R., Sun, W.C., Wang, Q.: Stress-induced anisotropy in granular materials: fabric, stiffness, and permeability. Acta Geotechnica **10**(4), 399–419 (2015)
42. Le, T., Hoang, T., Li, J., Liu, L., Liu, H., Hu, S.: A fast pc algorithm for high dimensional causal discovery with multi-core pcs. In: IEEE/ACM Transactions on Computational Biology and Bioinformatics, (2016)
43. Lefik, M., Schrefler, B.A.: Artificial neural network as an incremental non-linear constitutive model for a finite element code. Comput. Methods Appl. Mech. Eng. **192**(28–30), 3265–3283 (2003)
44. Li, X., Li, X.-S.: Micro-macro quantification of the internal structure of granular materials. J. Eng. Mech. **135**(7), 641–656 (2009)
45. Li, X.S., Dafalias, Y.F.: Anisotropic critical state theory: role of fabric. J. Eng. Mech. **138**(3), 263–275 (2012)
46. Li, Z., Gong, B., Yang, T.: Improved dropout for shallow and deep learning. In: Advances in Neural Information Processing Systems, pp. 2523–2531, (2016)
47. Liu, Y., Sun, W.C., Fish, J.: Determining material parameters for critical state plasticity models based on multilevel extended digital database. J. Appl. Mech. **83**, 011003–2 (2016)
48. Logarzo, H.J., Capuano, G., Rimoli, J.J.: Smart constitutive laws: Inelastic homogenization through machine learning. Comput. Methods Appl. Mech. Eng. **373**, 113482 (2021)
49. Ma, R., Sun, W.C.: Computational thermomechanics for crystalline rock. Part ii: Chemo-damage-plasticity and healing in strongly anisotropic polycrystals. Comput. Methods Appl. Mech. Eng. **369**, 113184 (2020)
50. Manzari, M.T., Dafalias, Y.F.: A critical state two-surface plasticity model for sands. Geotechnique **47**(2), 255–272 (1997)
51. Masi, F., Stefanou, I., Vannucci, P., Maffi-Berthier, V.: Thermodynamics-based artificial neural networks for constitutive modeling. J. Mech. Phys. Solids **147**, 104277 (2021)
52. Meek C.: Causal inference and causal explanation with background knowledge. In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, pp. 403–410 (1995)
53. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artif. Intell. **267**, 1–38 (2019)
54. Mitchell, J.K., Soga, K., et al.: Fundamentals of Soil Behavior, vol. 3. Wiley, New York (2005)
55. Na, S.H., Bryant, E.C., Sun, W.C.: A configurational force for adaptive re-meshing of gradient-enhanced poromechanics problems with history-dependent variables. Comput. Methods Appl. Mech. Eng. **357**, 112572 (2019)
56. Oktay, H., Taylor, B.J., Jensen, D.D.: Causal discovery in social media using quasi-experimental designs. In: Proceedings of the First Workshop on Social Media Analytics, pp. 1–9, (2010)
57. O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al.: Keras tuner. Retrieved May, **21**:2020, (2019)
58. O'Sullivan, C.: Particulate Discrete Element Modelling: A Geomechanics Perspective. CRC Press, Florida (2011)
59. Pandolfi, A., Guduru, P.R., Ortiz, M., Rosakis, A.J.: Three dimensional cohesive-element analysis and experiments of dynamic fracture in c300 steel. Int. J. Solids Struct. **37**(27), 3733–3760 (2000)
60. Park, K., Paulino, G.H.: Cohesive zone models: a critical review of traction-separation relationships across fracture surfaces. Appl. Mech. Rev. **64**(6), 060802 (2011)
61. Pearl, J.: Causality: Models, Reasoning and Inference, vol. 9, pp. 10–11. Cambridge University Press, Cambridge (2000)
62. Popov, V.L.: Contact Mechanics and Friction. Springer, Berlin (2010)
63. Powrie, W.: Soil Mechanics: Concepts and Applications. CRC Press, Florida (2018)
64. Probst, P., Wright, M.N., Boulesteix, A.-L.: Hyperparameters and tuning strategies for random forest. Wiley Interdisciplinary Reviews: Data Mining Knowl. Discovery **9**(3), e1301 (2019)
65. Rasmussen, C.E.: Gaussian processes in machine learning. In: Carbonell, J.G., Siekmann, J. (eds) Summer School on Machine Learning, pp. 63–71. Springer (2003)
66. Runge, J.: Causal network reconstruction from time series: From theoretical assumptions to practical estimation. Chaos Interdiscip. J. Nonlinear Sci. **28**(7), 075310 (2018)
67. Satake, M.: Constitution of mechanics of granular materials through the graph theory. In Proc. US-Japan Seminar on Continuum Mech. Stat. Appr. Mech. Granul. Mater., Sendai, pp. 203–215, (1978)
68. Schofield, A., Wroth, P.: Critical State Soil Mechanics. McGraw-hill, New York (1968)
69. Scholz, C.H.: Earthquakes and friction laws. Nature **391**(6662), 37–42 (1998)
70. Shen, X., Ma, S., Vemuri, P., Simon, G.: Challenges and opportunities with causal discovery algorithms: Application to alzheimer's pathophysiology. Sci. Rep. **10**(1), 1–12 (2020)
71. Shi, J., Guo, P., Stolle, D.: Noncoaxiality between fabric and stress in two-dimensional granular materials. J. Eng. Mech. **144**(9), 04018092 (2018)
72. Šmilauer, V., Catalano, E., Chareyre, B., Dorofeenko, S., Duriez, J., Gladky, A., Kozicki, J., Modenese, C., Scholtès, L., Sibille, L., et al.: Yade reference documentation. Yade Document. **474**(1), (2010)
73. Spirtes, P., Glymour, C.N., Scheines, R., Heckerman, D.: Causation, Prediction, and Search. MIT Press, Cambridge (2000)
74. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

75. Sun, W.C.: A unified method to predict diffuse and localized instabilities in sands. Geomech. Geoeng. **8**(2), 65–75 (2013)
76. Sun, W.C., Wang, K.: Discrete element traction-separation data for meta-modeling game, (2019). URL https://data.mendeley.com/datasets/n5v7hyny8n/1
77. Sun, W.C., Kuhn, M.R., Rudnicki, J.W.: A multiscale dem-lbm analysis on permeability evolutions inside a dilatant shear band. Acta Geotech. **8**(5), 465–480 (2013)
78. Sun, X., Bahmani, B., Vlassis, N.N., Sun, W.C., Xu, Y.: Forecast engine for data-driven discovery of interpretable causal relations. https://github.com/bbhm-90/MultiGraphRNN, (2020a)
79. Sun, X., Bahmani, B., Vlassis, N.N., Sun, W.C., Xu, Y.: Data-driven discovery of interpretable causal relations for deep learning material laws with uncertainty quantification. https://github.com/YanxunXu/MaterialLawCausal, 2020b)
80. Sussillo, D., Barak, O.: Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. Neural Comput. **25**(3), 626–649 (2013)
81. Tordesillas, A., Walker, D.M., Lin, Q.: Force cycles and force chains. Phys. Rev. E **81**(1), 011302 (2010)
82. Truesdell, C., Noll, W.: The non-linear field theories of mechanics. In: Antman, S.S. (ed) The Non-linear Field Theories of Mechanics, pp. 1–579. Springer (2004)
83. Vlassis, N., Ma, R., Sun, W.C.: Geometric deep learning for computational mechanics part i: anisotropic hyperelasticity. Comput. Methods Appl. Mech. Eng. **371**, 113299 (2020a)
84. Vlassis, N.N., Sun, W.C.: Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. Comput. Methods Appl. Mech. Eng. **377**, 113695 (2021)
85. Vlassis, N.N., Bahmani, B., Sun, W.C.: Discrete element method hypoplasticity data for data-driven causal relation discovery (2020b). https://data.mendeley.com/datasets/755bk3tvz9/1
86. Walker, D.M., Tordesillas, A.: Topological evolution in dense granular materials: a complex networks perspective. Int. J. Solids Struct. **47**(5), 624–639 (2010)
87. Wang, K., Sun, W.C.: A semi-implicit micropolar discrete-to-continuum method for granular materials. In: Papadrakakis, M., Papadopoulos, V., Stefanou, G., Plevris, V., (Eds.), Proceedings of European Congress on Computational Methods in Applied Science and Engineering, number June, pp. 5–10, Crete Island, (2016)
88. Wang, K., Sun, W.C.: A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. Comput. Methods Appl. Mech. Eng. **334**, 337–380 (2018)
89. Wang, K., Sun, W.C.: Meta-modeling game for deriving theory-consistent, microstructure-based traction-separation laws via deep reinforcement learning. Comput. Methods Appl. Mech. Eng. **346**, 216–241 (2019a)
90. Wang, K., Sun, W.C.: An updated lagrangian lbm-dem-fem coupling model for dual-permeability fissured porous media with embedded discontinuities. Comput. Methods Appl. Mech. Eng. **344**, 276–305 (2019b)
91. Wang, K., Sun, W.C., Salager, S., Na, S.H., Khaddour, G.: Identifying material parameters for a micro-polar plasticity model via x-ray micro-computed tomographic (ct) images: lessons learned from the curve-fitting exercises. Int. J. Multiscale Comput. Eng. **14**(4), 389–413 (2016)
92. Wang, K., Sun, W.C., Qiang, D.: A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with ai-guided experimentation. Comput. Mech. **64**(2), 467–499 (2019)
93. Wang, K., Sun, W.C., Du, Q.: A non-cooperative meta-modeling game for automated third-party calibrating, validating, and falsifying constitutive laws with parallelized adversarial attacks. arXiv preprint arXiv:2004.09392, (2020)
94. Wang, R., Pengcheng, F., Zhang, J.-M., Dafalias, Y.F.: Evolution of various fabric tensors for granular media toward the critical state. J. Eng. Mech. **143**(10), 04017117 (2017)
95. Wood, D.M.: Soil Behaviour and Critical State Soil Mechanics. Cambridge University Press, Cambridge (1990)
96. Xie, F., Xu, Y.: Bayesian projected calibration of computer models. J. Am. Stat. Assoc. (2020). https://doi.org/10.1080/01621459.2020.1753519
97. Xu, Y., Zhang, J., Yuan, Y., Mitra, R., Müller, P., Ji, Y.: A bayesian graphical model for integrative analysis of tcga data. In: Proceedings 2012 IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS), pp. 135–138. IEEE, (2012)
98. Kun Z., Jonas P., Dominik J., and Bernhard S.: Kernel-based conditional independence test and application in causal discovery. arXiv preprint arXiv:1202.3775, (2012)
99. Zhang, P., Yin, Z.-Y., Jin, Y.-F., Ye, G.-L.: An ai-based model for describing cyclic characteristics of granular materials. Int. J. Numer. Anal. Methods Geomech. **44**(9), 1315–1335 (2020)
100. Zhao, J., Guo, N.: Unique critical state characteristics in granular media considering fabric anisotropy. Géotechnique **63**(8), 695–704 (2013)