Computational Mechanics
https://doi.org/10.1007/s00466-019-01723-1

ORIGINAL PAPER

A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with AI-guided experimentation

Kun Wang[1] · WaiChing Sun[1] · Qiang Du[2]

Received: 29 January 2019 / Accepted: 13 May 2019

Abstract
We introduce a multi-agent meta-modeling game to generate data, knowledge, and models that make predictions on constitutive responses of elasto-plastic materials. We introduce a new concept from graph theory where a modeler agent is tasked with evaluating all the modeling options recast as a directed multigraph and find the optimal path that links the source of the directed graph (e.g. strain history) to the target (e.g. stress) measured by an objective function. Meanwhile, the data agent, which is tasked with generating data from real or virtual experiments (e.g. molecular dynamics, discrete element simulations), interacts with the modeling agent sequentially and uses reinforcement learning to design new experiments to optimize the prediction capacity. Consequently, this treatment enables us to emulate an idealized scientific collaboration as selections of the optimal choices in a decision tree search done automatically via deep reinforcement learning.

Keywords Directed multigraph · Data-driven constitutive modeling · Multi-agent deep reinforcement learning · Combinatorial optimization · Computational combinatorics

1 Introduction

In single-physics solid mechanics problems, the balance of linear momentum is often used to provide constraints for the motion of a body in the space-time continuum, while a constitutive law is often supplied to replicate constitutive responses at a selected material point of the body. Many successful commercial and open-source codes now introduce mechanisms or gateways that simplify the incorporation of material point constitutive models into predefined solid mechanics solvers (e.g. UMAT in ABAQUS) [14,15,28,57,66,67,81,83,99]. Once a constitutive law is formulated, algorithms are then designed to approximate the mathematical model such that a computer can be used to run simulations. The algorithms that approximate or enforce the constitutive laws are then verified, validated and eventually used in engineering practice [31,34,94].

Conventionally, a constitutive model that replicates the relation between the kinetic and kinematics quantities is derived from a finite set of fundamental principles, assumptions and phenomenological equations [87,88]. For instance, the laws of thermodynamics, material frame indifference, and balance laws are universal principles that are widely believed to be true for all materials under common circumstances. After enforcing those universal principles, there often remains a finite set of choices a modeler can make to construct a constitutive model. In particular, different types of experiments are designed such that a proper set of additional constraints can be generated. These constraints may not be fully explained by universal principles but are added to ensure the compatibility between observed and simulated mechanical responses. In reality, the universal principles alone are insufficient to complete most of the constitutive laws, regardless of the spatial scales these constitutive laws are designed for. As a result, phenomenological relations are introduced such that all constraints imposed by principles and observations can both be enforced.

✉ WaiChing Sun
wsun@columbia.edu

Kun Wang
kw2534@columbia.edu

Qiang Du
qd2125@columbia.edu

[1] Department of Civil Engineering and Engineering Mechanics, Columbia University, 614 SW Mudd, Mail Code 4709, New York, NY 10027, USA

[2] Department of Applied Physics and Applied Mathematics, and Data Science Institute, Columbia University, New York, NY 10027, USA

Published online: 25 May 2019

1.1 Rationales of phenomenological relations

Even though phenomenological relations cannot be fully justified via universal-principle arguments, it is understandable that proposers of these phenomenological relations often seek justifications by introducing new theories or incorporating microstructural information. For instance, the most commonly used family of soil models, the critical-state plasticity, relies on the existence of a critical state line in the state path (i.e. specific volume against the natural logarithmic of effective mean pressure) such that soil in the numerical simulations may develop plastic shear strain without volumetric deformation when reaching the critical state and exhibit the plastic dilatancy or contraction at different void ratio and overconsolidation ratio [7,12,44,56,69,80]. Experimental evidences are then sought to either justify the claim (cf. [100]), or redefine the applicability of the theory in light of new evidences (c.f. [42,54,102]). The incorporation of fabric tensors in critical state plasticity is another example where sub-scale information is incorporated to enhance forward prediction quality [42,92]. Other types of information incorporated into the constitutive law may come from microstructural attributes or the kinematics of microstructures. A classical example is crystal plasticity where the kinematics of the plastic flow is restricted by the orientations of the slip systems [2,9,52,57].

Finally, for practical reasons or due to lack of sufficient experimental evidences to prove otherwise, assumptions are sometimes made to interpret a phenomenological relation. A classical example for this type of phenomenological approach is the effective displacement theory commonly used in traction-separation models where one assumes that a scalar kinematic measure, often a weighted norm of normal and tangential displacements, can be used to determine a scalar traction measure that leads to the traction vector [59,62]. Nevertheless, the distinction between phenomenology that only enhances curve-fitting in calibration and the counterpart that leads to more accurate, robust and reliable predictions is often a blurred line and might be subject to debate [88,95–97]. Furthermore, the popularity of a model in the short term is also not necessarily purely based on the prediction quality, but also ties to the difficulty in calibration and interpretation of the model [39], the demand of experimental data [58,97], as well as the social, cultural and personal influences (cf. [50]), among other factors . In the case where a limited subset of data might be chosen to make a constitutive law or theory sound plausible or consistent with a physical phenomenon, the true forward prediction quality of the model might take a toll while the apparent capacity of the model could be exaggerated [55]. The underlying problem is that this issue is very difficult to detect unless all the models are compared objectively in the same benchmark study and subjected to a universally agreed validation metric [11,60]. Hence, a validation procedure that employs blind predictions is critical, *regardless of the type of models used for predictions*.

1.2 Data-driven approaches as alternatives

An alternative to the conventional modeling approach is the data-driven modeling in which constitutive responses are predicted primarily based on the available data either by black-box neural networks [22,25,41,93,94] or via minimization problems in the phase space [34]. While the latter approach, as outlined in [34,35], has shown great promises to handle hyperelasticity problems, the extension to plasticity problems likely requires either imposing further constraints (e.g. perfect plasticity [31]) or creating a sufficiently large database to capture the phase space of the history-dependent responses. On the other hand, Lefik and Schrefler [41] has demonstrated that a neural network can generate cyclic elasto-plastic responses with some level of success. Nevertheless, despite the fact that a multi-layer neural network can be considered as a universal approximator, as pointed out in [29], this does not imply that the training of the neural network is always successful. In fact, failure to complete the training is quite common and it might be caused by, for example, (1) higher demand of data for the neural network training compared to the material parameter identification in conventional modeling, (2) the curse of high dimensionality that leads to inconsistency between calibration and forward prediction performance, (3) issues related to under-fitting and over-fitting, and (4) the vanishing gradient issues that make the algorithm unable to locate the global minimizer of the loss function [94]. Furthermore, without special treatment to extend the database for training the neural network, the resultant models often exhibit dependence on coordinate systems. Even though this issue has been addressed recently using the spectral decomposition of tensorial inputs and outputs in recurrent neural networks [94], this lack of consistency with theory indicates that the domain expertise remains critical for evaluating the quality of the machine learning model and finding remedies for issues not immediately apparent for nonspecialists.

In the aforementioned data-driven approach, the demand for big data remains an ongoing challenge [47,77,86]. In particular, machine learning models, especially those in most generic forms (i.e. model-free approaches), may suffer a lack of constraints imposed by material theory, thus increasing the demand for data to generate the constraints. Hence, it is important for modelers to be able to estimate the least amount of data required to complete the training of a specific model. The introduction of the two-player cooperative game in this paper can provide a practical solution to find the required amount and type of data for path-dependent materials.

1.3 The hybridized theoretical/data-driven approach

In this paper, our goal is to (1) introduce a meta-modeling method to generate algorithms that **hybridize** theory, phenomenological relations, and universal principles to **automatically** generate constitutive laws that fulfill a specific objective defined by the loss (objective function) in a quantitatively optimal manner and (2) incorporate the reinforcement learning technique to select experiments that lead to improvement in prediction capacity. We do not limit ourselves to the approach in which the neural network model is either used to replace the entire constitutive law or not being used at all (cf. [24,34,94]). Instead, our goal is to find the optimal way out of all the possible choices to construct a constitutive law for a given material data.

To reach our goal, we employ two techniques of discrete mathematics that are less commonly used in computational mechanics, the directed multigraph and decision tree learning. First, the directed multigraph is used to recast the available choices of constitutive laws as a family of possible ways to configure a graph of information flow from the upstream (the source or input, such as the relative displacement or strain) to the downstream (the target or output, such as the traction or stress). A model is a path (in the terminology of graph theory) of this directed multigraph that optimizes an objective function. As such, a model is associated with a collection of physical quantities (vertices in the directed graph) linked by either mathematical expressions or machine learning models that connect the upstream to the downstream (edges in the directed graph) (cf. [94]).

Within our framework, a black-box neural network model, for instance, is simply a model in which there are no human-interpretable quantities connecting the input and output. Many classical neural network models such as Ghaboussi et al. [24], Lefik and Schrefler [41] and Wang and Sun [94] all belong to this category, as neurons are the only media that propagate the information flow. Meanwhile, a classical theory-based constitutive law can be viewed as a directed graph (or a particular path of the directed multigraph) in which all the edges are mappings that can be written as mathematical expressions formulated by human. On the other hand, a hybridized model could have a subset of neural network edges while having the rest edges theoretically based.

Since the optimal configuration of the directed graph for a given problem and the corresponding objective function is not known a priori, we introduce mechanisms to hierarchically explore the possible modeling choices using a decision tree. A decision tree is simply an explicit representation of all possible scenarios such that the sequence of decisions (in our case the modeling choices and data explorations) is evaluated by an agent who then takes account of the possible observations (e.g. experimental observations), and state changes (e.g. the changes of validation metrics or loss function values) to estimate the best choices.

In this work, our major contributions are threefold. First, we introduce the concept of labeled directed multigraph to represent relational knowledge. Such a mechanism provides a convenient mean to hybridize theory-based and data-driven models to yield optimal forward predictions. Second, we recast the reinforcement learning as the process of formulating constitutive laws as a combinatorial optimization problem for making a large number of modeling choices. Through an automated trial-and-error process, the AI agent continues to improve its decision-making ability automatically without human intervention. The resultant meta-modeling approach therefore enables the AI to discovery model building knowledge via the Edisonian approach, while overcoming the low efficiency of the Edisonian approach through automation. The application of concepts from graph theory, such as directed graph and directed multigraph gives us hierarchical information that helps understand the causal connections among events and mechanisms. As point out in [38], this model-building approach has an advantage over the model-free machine learning approaches in interpretability. As considerable evidence has indicated that the model-based planning, such as the one introduced in this paper, is not only an essential ingredient of human intelligence, but also the key step to enable flexible adaptation for new tasks and goals. The importance of the usage of multigraph is that it enables us to form complex idea, knowledge, prediction, inference and response with a rather small set of simple elements. This kind of application of the principle of combinatorial generalization has long been regarded as the key signature of intelligence [6,13,30,38]. Third, we also introduce a cooperative mechanism to integrate the data exploration into the modeling process. In this way, the framework can not only generate constitutive models to make the best predictions among the limited data, but also estimate the most efficient way to select experiments such that the most needed information is included to generate the knowledge closure.

1.4 Content organization

The rest of the paper is organized as follows. We first introduce the meta-modeling cooperative game, including the method to recast all possible modeling options as directed multigraph, and the generation of decision tree (Sect. 2). Following this, we will introduce the detailed design of the data collection/meta-modeling game for modeling the collaboration of the AI data agent and the AI modeler agent (Sect. 3). In Sect. 4, we then review the multi-agent reinforcement learning algorithms that enable us to find the optimal decision for constitutive models, as well as the corresponding optimal actions the data agent takes to max-

imize the prediction quality of the AI-generated model. We then present numerical experiments to assess the accuracy and robustness of the blind predictions of the model generated via our meta-modeling algorithm operated on the directed multigraph. To check whether our approach is able to deal with a wide spectrum of situations and can be generalized for different materials, the multigraph meta-modeling algorithm is tested with distinctive types of data (e.g. synthetic data from elasto-plastic models and discrete element simulations). To aid the reproducibility of our numerical experiments by the third party, these data will be open source upon the publication of this article.

1.5 Notations and terminologies

For convenience, we provide a minimal review of the essential terminologies and concepts from graph theory that are used throughout this paper. Their definitions can be found in, for instance, [4,27,98].

Definition 1 A ***n*-tuple** is a sequence or ordered list that consists of n element where n is a non-negative integer and that (unlike a set) may contain multiple instances of the same element.

Definition 2 A **directed graph** (digraph) is an ordered pair (2-tuple) $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ where $\mathbb{V}$ is a nonempty set of vertices and $\mathbb{E}$ is a *set* of *ordered* pairs of vertices (directed edges) where each edge in $\mathbb{E}$ connects a pair of source (beginning) and target (end) vertices in a specific direction. Both vertices connected by an edge in $\mathbb{E}$ must be elements of $\mathbb{V}$ and the edge connecting them must be unique.

Definition 3 A **directed acyclic graph** is a directed graph where edges do not form any directed cycle. In a directed acyclic graph, there is no path that can start from a vertex and eventually loop back to the same vertex.

Definition 4 A **directed multigraph** with a distinctive edge identity (also called multi digraph) is an ordered 4-tuple $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \boldsymbol{s}, \boldsymbol{t})$ where $\mathbb{V}$ is a set of vertices, $\mathbb{E}$ is a set of edges that connect source and target vertices, $\boldsymbol{s} : \mathbb{E} \to \mathbb{V}$ is a mapping that maps each edge to its source node, and $\boldsymbol{t} : \mathbb{E} \to \mathbb{V}$ is a mapping that maps each edge to its target node.

Definition 5 An **underlying graph** $\mathbb{U}$ of a directed multigraph $\mathbb{G}$ is a multigraph whose edges are without directions.

Definition 6 A **subgraph** $\mathbb{G}'$ of a directed multigraph $\mathbb{G}$ is a directed multigraph whose vertex set $\mathbb{V}'$ is a subset of $\mathbb{V}$ ($\mathbb{V}' \subseteq \mathbb{V}$), and whose edge set $\mathbb{E}'$ is a subset of $\mathbb{E}$ ($\mathbb{E}' \subseteq \mathbb{E}$).

Definition 7 A **labeled directed multigraph** is a directed multigraph with labeled vertices and edges which can be mathematically expressed as an 8-tuple $\mathbb{G} = (\mathbb{L}_\mathbb{V}, \mathbb{L}_\mathbb{E}, \mathbb{V}, \mathbb{E}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{n_V}, \boldsymbol{n_E})$ where $\mathbb{V}$ and $\mathbb{E}$ are the sets of vertices and edges, $\mathbb{L}_\mathbb{V}$ and $\mathbb{L}_\mathbb{E}$ are the sets of labels for the vertices and edges, $s : \mathbb{E} \to \mathbb{V}$ and $t : \mathbb{E} \to \mathbb{V}$ are the mappings that map the edges to the source and target vetrices, $n_V : \mathbb{V} \to \mathbb{L}_\mathbb{V}$ and $n_E : \mathbb{E} \to \mathbb{L}_\mathbb{E}$ are the mappings that give the vertices and edges the corresponding labels in $\mathbb{L}_\mathbb{V}$ and $\mathbb{L}_\mathbb{E}$ accordingly.

As for notations and symbols, bold-faced letters denote tensors (including vectors which are rank-one tensors); the symbol '$\cdot$' denotes a single contraction of adjacent indices of two tensors (e.g. $\boldsymbol{a} \cdot \boldsymbol{b} = a_i b_i$ or $\boldsymbol{c} \cdot \boldsymbol{d} = c_{ij} d_{jk}$); the symbol ':' denotes a double contraction of adjacent indices of tensor of rank two or higher (e.g. $\boldsymbol{C} : \boldsymbol{\epsilon^e} = C_{ijkl}\epsilon^e_{kl}$); 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. $(\boldsymbol{\alpha} \otimes \boldsymbol{\beta})_{ijkl} = \alpha_{ij}\beta_{kl}$). Moreover, $(\boldsymbol{\alpha} \oplus \boldsymbol{\beta})_{ijkl} = \alpha_{jl}\beta_{ik}$ and $(\boldsymbol{\alpha} \ominus \boldsymbol{\beta})_{ijkl} = \alpha_{il}\beta_{jk}$. We also define identity tensors $(\boldsymbol{I})_{ij} = \delta_{ij}$, $(\boldsymbol{I}^4)_{ijkl} = \delta_{ik}\delta_{jl}$, and $(\boldsymbol{I}^4_{\text{sym}})_{ijkl} = \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{kj})$, where δ_{ij} is the Kronecker delta. As for sign conventions, unless specified otherwise, we consider the direction of the tensile stress and dilative pressure as positive.

2 Meta-modeling: deriving material laws from a directed multigraph

In this section, we describe the concepts behind the proposed automated meta-modeling procedure and the mechanism of the learning process. The key departures of our newly proposed method via the neural network approaches for constitutive laws (e.g. [24,25,40,41,94]) is the introduction of labeled directed multigraph that represents all possible theories under consideration for modeling a physical process, the acyclic directed graph that represents the most plausible knowledge on the relationships among physical quantities, and the data agent which enables users to estimate the amount of data required to reach the point where additional information no longer enhances prediction capacity for a given action space. In this paper, our focus is limited to the class of materials that exhibits elasto-plastic responses while damage can be neglected. We assume that the deformation is infinitesimal and the material is under isothermal condition. The proposed methodology, however, can be extended to other more complex materials.

2.1 Material modeling algorithm as a directed multigraph

The architecture of an algorithm is often considered as a directed multigraph [18]. In essence, a material model can

be thought as a procedure that employs organized knowledge to make predictions such that relationships of components and the universally accepted principles govern the outcomes of predictions. For instance, we may consider a traction-separation model as an information flow in a directed graph where physical attributes, such as porosity, plastic flow, permeability, are considered as vertices and their relationships are considered as edges [94]. The input and output of the models (e.g. relative displacement history and traction) are then considered as the sources and targets of the directed graph.

However, in some circumstances, a physical relation can be modeled by more than one methods, theories or constitutive relations. To reflect the availability of options, a generalized representation of the thought process is needed when we try to use artificial intelligence algorithm to replace human to write constitute models. This generalized thought process, which we refer as meta-modeling (i.e. modeling the process of writing a model), can be recast as a labeled directed multigraph. The latter can be used where a pair of connected vertices are not necessarily connected by one edge but by multiple edges, each represents a specific model that connects two physical quantities (e.g. porosity-permeability relationship). A formal statement can be written as follows:

Possible configurations of constitutive laws as a labeled directed multigraph Given a data set which measures a set of physical quantities defined as $\mathbb{V}$ with a corresponding set of labels $\mathbb{L}_{\mathbb{V}}$ where $n_{\mathbb{V}} : \mathbb{V} \to \mathbb{L}_{\mathbb{E}}$ is a bijective mapping that maps the vertices to the labels. Let $\mathbb{V}_R \subset \mathbb{V}$ and $\mathbb{V}_L \subset \mathbb{V}$ be the source(s) and target(s) of the directed multigraph. All possible ways to write constitutive laws that map the input V_R (e.g. strain history) to output V_L (e.g. stress) as information flow can be defined by the sets of directed edges where each edge that links two physical quantities $\mathbb{E}$, the mappings $\boldsymbol{s} : \mathbb{E} \to \mathbb{V}$ and $\boldsymbol{t} : \mathbb{E} \to \mathbb{V}$ that provide the direction of the information flow, and the surjective mapping $\boldsymbol{n}_{\mathbb{E}} : \mathbb{E} \to \mathbb{L}_{\mathbb{E}}$ that assigns the edge labels (names) to the edges.

A simple example for traction-separation law can be found in the Appendix A.

2.2 Recasting the process of writing constitutive laws as selecting subgraphs in a directed multigraph

In the first meta-modeling game introduced in this work, we consider a scenario where a set of experimental data is given. This experimental data include measurement of different physical quantities, but the inherent relationships are unknown to the modeler. Furthermore, in the process of writing the constitutive law, the modeler must follow a set of rules coined as universal principles (e.g. thermodynamic principles, material frame indifference) [34,94]. Here, we first assume that an objective of writing the constitutive model is well defined and hence a score system is available for the deep Q-learning. We then idealize the process of writing a constitutive law *with a fixed set of data* as a two-step process. First, we consider all the possible ways to write a constitutive law and represent all these possibilities in a labeled directed multigraph. This labeled directed multigraph define the action space of the meta-modeling game. Second, among all the possible ways to write a constitutive law, i.e., on the labeled directed multigraph, we seek the optimal configuration that will lead to the best outcome measured by an objective function. If the total number of possible configurations is sufficiently small, then the optimal configuration can be sought by building all the possible configurations and comparing their performance afterward. However, this brute force approach becomes infeasible when the total number of configurations is very large as in the case of the game of chess and Go [70,72]. As a result, the proposed procedure of finding the optimal configuration of a constitutive law is given as follow.

Instants of constitutive laws are considered as directed graphs. Given a dataset that contains the time history of measurable physical quantities of n types of data stored in the vertices labeled by the vertex label $l_i \in \mathbb{L}_{\mathbb{V}}$ and the labeled direct graph defined by the 8-tuple $\mathbb{G} = (\mathbb{L}_{\mathbb{V}}, \mathbb{L}_{\mathbb{E}}, \mathbb{V}, \mathbb{E}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{n_V}, \boldsymbol{n_E})$, and objective function SCORE and constraints to enforce universal principles. Find an subgraph $\mathbb{G}'$ of $\mathbb{G}$ consists of vertices $\boldsymbol{V} \in \mathbb{V}^s \subseteq \mathbb{V}$ and edges $\boldsymbol{E} \in \mathbb{E}^s \subseteq \mathbb{E}$ such that 1) $\mathbb{G}'$ is a directed acyclic graph, 2) a score metric is maximized under a set of m constraints $f_i(l_1, l_2, \ldots, l_n) = 0, i = 1, \ldots, m$ where , i.e.,

$$\begin{aligned} &\underset{l_i}{\text{maximize}} \quad \text{SCORE}(l_1, l_2, \ldots, l_n) \\ &\text{subject to} \quad f_i(i_i) = 0, \ i = 1, \ldots, m. \end{aligned} \tag{1}$$

Example 2 Game Action for traction-separation Laws. Given an 8-tuple $\mathbb{G} = (\mathbb{L}_{\mathbb{V}}, \mathbb{L}_{\mathbb{E}}, \mathbb{V}, \mathbb{E}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{n_V}, \boldsymbol{n_E})$ with elements defined in (66), (67), (72), (71). Find the subgraph $\mathbb{G}'$ of $\mathbb{G}$ such that this subgraph becomes the directed acyclic graph that maximizes the blind prediction accuracy defined by an objection function. □

3 Two-player meta-modeling game for the discovery of elasto-plastic models through modeling and automated experiments

In this work, we conceptualize the process of writing, calibrating and validating constitutive laws as a cooperative two-player game played by one modeler and one experimentalist (data) agent. These two agents, in theory, can be played by either a human or an artificial intelligence (AI) machine. To simplify the problems, we consider only virtual experiments such as discrete element simulations [46,82,91,94,101,105] and that the agents are not constrained by the number of virtual experiment tests they might conduct. The control of the experimental cost and the ability to automate the execution of experiments are important topics but are both out of the scope of this work.

As such two AI agents must be able to cooperate such that they can find the hierarchical relationships among available data and (2) come up with the experiment plan that helps improve the performance of the blind predictions made by the directed graph model, as shown in Fig. 1. This lead to a multi-agent multi-objective problem that can be solved by a deep reinforcement learning framework [65,85]. In this work, the deep reinforcement learning algorithm is based on a model-free policy gradient algorithm that employs a neural network to estimate the Q values of the policies (cf. [72,95]). In principle, it is possible to use other Bayesian reinforcement learning approaches, such as Thompson sampling, Bayesian upper confidence bounds, Bayesian sparse sampling and other different decision making algorithms to optimize the learning process. Finding the optimal strategy for the deep reinforcement learning in specific applications is an active research area, but is out of the scope of this study. Interested readers are referred to [26] for a comprehensive review on these reinforcement learning algorithms.

3.1 Data collection game for experimentalist agent

This section presents a design of the data collection game involving the common decision-making activities of experimentalists in testing the mechanical properties of a material. The goal of this game is for the experimentalist agent to find the optimal subset of tests for model generation and parameter calibration within a set of candidate tests on the material. The key ingredients of the game are detailed as follows.

3.1.1 Game board for experimentalist

Consider a set of possible mechanical experiments on a material $\mathbf{T} = \{T_1, T_2, T_3, \ldots, T_n\}$. The experiments can be divided into two types: (1) a subset $\mathbf{T}_c$ of calibration experiments for material parameter identification in a constitutive model, (2) a subset $\mathbf{T}_v$ of validation experiments for testing the forward prediction accuracy of the constitutive model. $\mathbf{T} = \mathbf{T}_c \cup \mathbf{T}_v$, $\mathbf{T}_c \cap \mathbf{T}_v = \emptyset$, $\mathbf{T}_c \neq \emptyset$ and $\mathbf{T}_v \neq \emptyset$. Suppose the experimentalist has a priori preselected the elements in both categories: $\mathbf{T}_c = \mathbf{T}_c^0 = \{T_{c1}, T_{c2}, T_{c3}, \ldots, T_{cn}\}$ and $\mathbf{T}_v = \mathbf{T}_v^0 = \{T_{v1}, T_{v2}, T_{v3}, \ldots, T_{vn}\}$. This selection could be based on the availability of laboratory equipment, i.e., $\mathbf{T}_c^0$ includes all tests that the experimentalist can perform in the laboratory, while $\mathbf{T}_v^0$ includes other tests that can only be acquired from literature or third-party laboratories. The experimentalist then chooses the final set of experiments $\mathbf{T}_c \subset \mathbf{T}_c^0$ which could generate necessary and sufficient data for the modeler agent to develop and calibrate a constitutive model with the highest model score. The final validation set $\mathbf{T}_v$ contains both experiments in $\mathbf{T}_v^0$ and those not selected in $\mathbf{T}_c$, i.e., $\mathbf{T}_v = \mathbf{T}_v^0 \cup (\mathbf{T}_c^0 \backslash \mathbf{T}_c)$. Hence the set $\mathbf{T}_c^0$ constitutes the "game board" for the experimentalist agent to play the data collection game.

3.1.2 Game state for experimentalist

The mathematical description of the current state of the game board is a list of binary indicators $s = [i_{c1}, i_{c2}, i_{c3}, \ldots, i_{cn}, i_{terminate}]$ representing whether a test $T_{ci} \in \mathbf{T}_c^0$ is selected to be one of the calibration tests, and also whether the game is terminated. If $T_{ci} \in \mathbf{T}_c$, the corresponding indicator $i_{ci} = 1$, if $T_{ci} \notin \mathbf{T}_c$ $i_{ci} = 0$. If $i_{terminate} = 1$, the game reaches the end, otherwise the experimentalist can continue. The initial state of the game is $i_{ci} = 0$, $\forall T_{ci} \in \mathbf{T}_c^0$ and $i_{terminate} = 0$. A special final state in which $i_{ci} = 0$, $\forall T_{ci} \in \mathbf{T}_c^0$ and $i_{terminate} = 1$ indicates that no data is available for model generation and calibration, hence the reward for this state is set to 0.

3.1.3 Game action for experimentalist

At each state s, the experimentalist can select the next calibration test $T_{ci} \in \mathbf{T}_c$, by changing the indicator i_{ci} from 0 to 1, or decide to stop the selection immediately, by changing $i_{terminate}$ from 0 to 1.

3.1.4 Game rule for experimentalist

Generally, there are no specific rules constraining the selection of experiments for model parameter calibration. But the game designer could always customize certain rules that prohibit the coexistence of certain experiments in $\mathbf{T}_c$. The game rule can be reflected by a list of binaries $LegalActions(s) = [ii_{c1}, ii_{c2}, ii_{c3}, \ldots, ii_{cn}, ii_{terminate}]$, indicating whether an element i_{ci} of the state s can be changed in the next action step.

- If $i_{ci} = 0$ in the current state s, then $ii_{ci} = 1$ in $LegalActions(s)$.

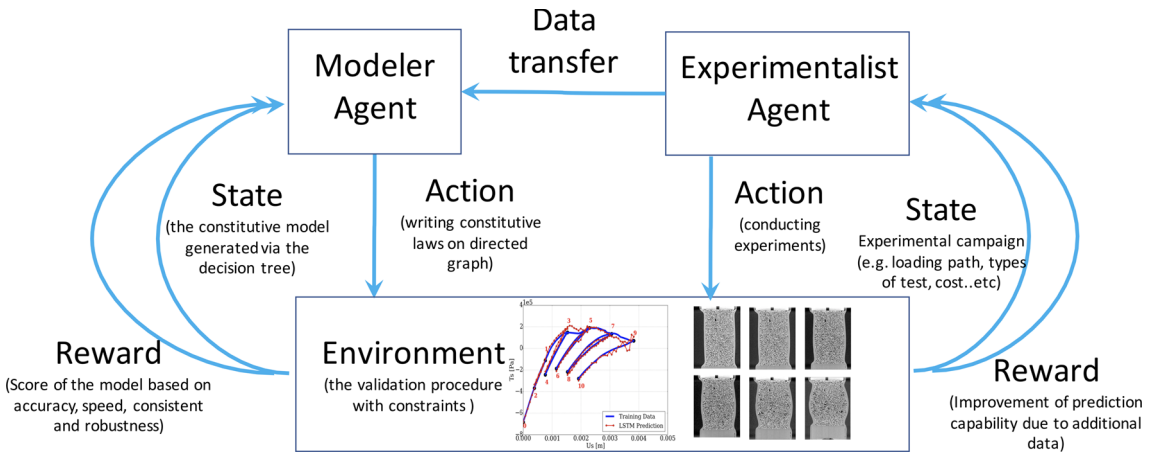


Fig. 1 Scheme of the reinforcement learning algorithm in which two agents interact with environment and receives rewards for their corresponding actions (writing models and conducting experiments)

- If $i_{ci} = 1$, then $ii_{ci} = 0$.
- if $i_{terminate} = 0$, then $ii_{terminate} = 1$.
  We enforce a game rule that require the two tests T_{ci} and T_{cj} are mutually exclusive in $\mathbf{T}_c$.
- If $i_{ci} = 1$, then $ii_{cj} = 0$, and vice versa.
  The initial legal actions are $ii_{ci} = 1,\ \forall T_{ci} \in \mathbf{T}_c^0$ and $ii_{terminate} = 1$.

3.1.5 Game reward for experimentalist

The reward from the game environment to the experimentalist agent should consider the scores of the constitutive models generated by the modeler, given the calibration data and validation data prepared by the experimentalist. For each result of the data collection game $\mathbf{T}_c$ (hence its pair $\mathbf{T}_v = \mathbf{T}\backslash\mathbf{T}_c$), the modeler could generate a number of different constitutive models with scores $[\text{SCORE}_{i,\ i=1,2,3,\ldots}]_{\mathbf{T}_c}$. The reward should also consider the total cost of the calibration tests $\mathbf{T}_c$. This reward for multiple objectives can be measured by a weighted sum $\text{COST}(\mathbf{T}_c) = \sum^{\mathbf{T}_c^0} w_{ci}^{cost} * i_{ci}$, where w_{ci}^{cost} is the normalized cost for test $T_{ci} \in \mathbf{T}_c^0$, $\sum^{\mathbf{T}_c^0} w_{ci}^{cost} = 1$, $w_{ci}^{cost} \in [0, 1]$.

If the experimentalist and the modeler are fully cooperative on generating the constitutive model with the highest score, the reward r is a function of the maximum model scores for all possible $\mathbf{T}_c \subset \mathbf{T}_c^0$ and the total experimental cost of $\mathbf{T}_c$. Suppose that, since the beginning of the two-payer cooperative game (Fig. 1), the experimentalist have experienced a number of calibration test sets $\mathbf{T}_c$ (they constitute a set $\mathbb{T}_c^{\text{history}}$), and the modeler have generated constitutive models and evaluated their scores for these calibration test sets ($[\text{SCORE}_{i,\ i=1,2,3,\ldots}]_{\mathbf{T}_c},\ \forall \mathbf{T}_c \in \mathbb{T}_c^{\text{history}}$). Then, both agents have the knowledge of the highest model score for each $\mathbf{T}_c$: $\text{SCORE}_{\mathbf{T}_c}^{\max} = \max([\text{SCORE}_{i,\ i=1,2,3,\ldots}]_{\mathbf{T}_c})$. Thus they know the highest model score in the history of self-played games: $\text{SCORE}^{\max} = \max(\text{SCORE}_{\mathbf{T}_c}^{\max}),\ \forall \mathbf{T}_c \in \mathbb{T}_c^{\text{history}}$. Then the agents can identify a set $\mathbb{T}_c^{\max} \subset \mathbb{T}_c^{\text{history}}$ in which the elements are all calibration test sets that can lead to maximum scores comparable to the highest score, i.e., $\mathbf{T}_c \in \mathbb{T}_c^{\max}$, if $|\text{SCORE}_{\mathbf{T}_c}^{\max} - \text{SCORE}^{\max}| \leq \text{TOL}$, where TOL is a small tolerance criteria.

From the perspective of the experimentalist agent, for a fully cooperative game, $\mathbf{T}_c$ (represented by the state s) is winning the data collection game if it is an element of the set $\mathbb{T}_c^{\max}$, and its total cost is the lowest among all elements in $\mathbb{T}_c^{\max}$. Hence the reward is designed as

$$r(s) = \begin{cases} 1, & \text{if } \mathbf{T}_c \in \mathbb{T}_c^{\max} \text{and } \text{COST}(\mathbf{T}_c) \leq \text{COST}(\forall \mathbf{T}_c^i \in \mathbb{T}_c^{\max}) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

3.1.6 Game choices for experimentalist

The elements in the set $\mathbf{T} = \{T_1, T_2, T_3, \ldots, T_n\}$ could be all possible mechanical experiments performed on a material. For example, for granular materials, the candidates can include the following common types of tests in soil laboratories:

1. Drained conventional triaxial test ($\dot{\epsilon}_{11} \neq 0, \dot{\sigma}_{22} = \dot{\sigma}_{33} = \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$).
2. Drained true triaxial test ($\dot{\epsilon}_{11} \neq 0, b = \frac{\sigma_{22}-\sigma_{33}}{\sigma_{11}-\sigma_{33}}, \dot{\sigma}_{33} = \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$).
3. Undrained triaxial test ($\dot{\epsilon}_{11} \neq 0, \dot{\epsilon}_{11} + \dot{\epsilon}_{22} + \dot{\epsilon}_{33} = 0, \dot{\sigma}_{22} = \dot{\sigma}_{33}, \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$).
4. One-dimensional test ($\dot{\epsilon}_{11} \neq 0, \dot{\epsilon}_{22} = \dot{\epsilon}_{33} = \dot{\epsilon}_{12} = \dot{\epsilon}_{23} = \dot{\epsilon}_{13} = 0$).
5. Simple shear test ($\dot{\epsilon}_{12} > 0, \dot{\sigma}_{11} = \dot{\sigma}_{22} = \dot{\epsilon}_{33} = \dot{\epsilon}_{23} = \dot{\epsilon}_{13} = 0$).

The loading conditions are represented by constraints on the components of the stress rate and strain rate tensors

$$\dot{\boldsymbol{\epsilon}} = \begin{bmatrix} \dot{\epsilon}_{11} & \dot{\epsilon}_{12} & \dot{\epsilon}_{13} \\ & \dot{\epsilon}_{22} & \dot{\epsilon}_{23} \\ \text{sym} & & \dot{\epsilon}_{33} \end{bmatrix}, \ \dot{\boldsymbol{\sigma}} = \begin{bmatrix} \dot{\sigma}_{11} & \dot{\sigma}_{12} & \dot{\sigma}_{13} \\ & \dot{\sigma}_{22} & \dot{\sigma}_{23} \\ \text{sym} & & \dot{\sigma}_{33} \end{bmatrix}. \tag{3}$$

Remarks on implementation In the numerical testing of the constitutive models, the above material test conditions are applied via a linearized integration technique for loading constraints of laboratory experiments $\boldsymbol{S}d\boldsymbol{\sigma} + \boldsymbol{E}d\boldsymbol{\epsilon} = d\boldsymbol{Y}$, combined with incremental constitutive equations, as proposed in [5]. $d\boldsymbol{\sigma}$ and $d\boldsymbol{\epsilon}$ are Voigt forms of incremental stress and strain, respectively. $\boldsymbol{S}$ and $\boldsymbol{E}$ are matrices of constraints on incremental stress and strain, respectively. $d\boldsymbol{Y}$ is a vector of constraint values. See [5] for their formulations for different loading constraints in geomechanics tests (e.g., drained and undrained triaxial tests).

3.2 Meta-modeling game for modeler agent

This section presents a design of the constitutive modeling game involving the common decision-making activities of modelers in developing models to approximate the mechanical properties of a material. The goal of this game is for the modeler agent to find the optimal configuration of the directed graph from a predefined directed multigraph (Sect. 2) with its structure inherited from the graphs of the classical infinitesimal strain elasto-plasticity models. The key ingredients of the meta-modeling game consist of *game agents, game board, game state, game actions, game Rules, game reward and game choices* such that it constitutes an agent-environment interactive system [8,95] which are detailed as follows.

3.2.1 Game board for modeler

A constitutive model in the generalized elasto-plasticity framework [63,104] requires four essential components of "phenomenological relations" : (1) elasticity law (2) loading direction (3) plastic flow direction (4) hardening modulus. The process of obtaining a directed graph (the final state of the game) from the game board, i.e., the direct multigraph of the proposed framework is presented in Fig. 2. The quantities are presented in the incremental form at discrete time steps. A quantity a at the current time step t_n is denoted as $a_n = a(t_n)$. The next time step is t_{n+1} with the time increment $\Delta t = t_{n+1} - t_n$. Then the increment of the quantity a within Δt is denoted as $\Delta a_{n+1} = a_{n+1} - a_n$. The essential "definition" edges in the direct multigraph are written as

$$\begin{aligned}
&(1) \quad \Delta\boldsymbol{\sigma}_{n+1} = \boldsymbol{C}^e_n : \Delta\boldsymbol{\epsilon}^e_{n+1} \\
&(2) \quad \Delta\boldsymbol{\epsilon}^e_{n+1} = \Delta\boldsymbol{\epsilon}_{n+1} - \Delta\boldsymbol{\epsilon}^p_{n+1} \\
&(3) \quad \Delta\boldsymbol{\epsilon}^p_{n+1} = \Delta\lambda_{n+1}\boldsymbol{m}^{flow}_n \\
&(4) \quad \Delta\lambda_{n+1} = \begin{cases} \frac{\boldsymbol{n}^{load}_n : \boldsymbol{C}^e_n : \Delta\boldsymbol{\epsilon}_{n+1}}{H_n + \boldsymbol{n}^{load}_n : \boldsymbol{C}^e_n : \boldsymbol{m}^{flow}_n} & \text{if plastic loading} \\ 0 & \text{if elastic loading} \end{cases},
\end{aligned} \tag{4}$$

where $\Delta\lambda_{n+1}$ is the plastic multiplier and H_n is the generalized plastic modulus.

The "elastic loading" and "plastic loading" states are determined via the projection of the trial elastic stress increment $\Delta\boldsymbol{\sigma}^e_{n+1} = \boldsymbol{C}^e_n : \Delta\boldsymbol{\epsilon}_{n+1}$ on the loading direction $\boldsymbol{n}^{load}_n$. If there is no assumed yield surface, then

$$\begin{cases} \Delta\boldsymbol{\sigma}^e_{n+1} : \boldsymbol{n}^{load}_n \neq 0 \rightarrow \text{plastic loading} \\ \Delta\boldsymbol{\sigma}^e_{n+1} : \boldsymbol{n}^{load}_n = 0 \rightarrow \text{elastic loading} \end{cases}, \tag{5}$$

or if there exists a yield surface $f(\boldsymbol{\sigma}, \boldsymbol{q}^{piv}_n(\boldsymbol{\xi}^{piv}_n))$, then

$$\begin{cases} f(\boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}^e_{n+1}, \boldsymbol{q}^{piv}_n(\boldsymbol{\xi}^{piv}_n)) > 0 \rightarrow \text{plastic loading} \\ f(\boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}^e_{n+1}, \boldsymbol{q}^{piv}_n(\boldsymbol{\xi}^{piv}_n)) \leq 0 \rightarrow \text{elastic loading} \end{cases}, \tag{6}$$

where $\boldsymbol{\xi}^{piv}_n$ is a vector of strain-like plastic internal variables and $\boldsymbol{q}^{piv}_n$ is a vector of stress-like plastic internal variables conjugate to $\boldsymbol{\xi}^{piv}_n$. $\boldsymbol{\xi}^{piv}_n$ may include the following internal state variables accumulated during the deformations from the initial time t_0 to the current time t_n,

$$(5) \begin{cases} \lambda_n = \int_0^{t_n} \dot{\lambda} dt \\ \bar{\epsilon}^p_n = \int_0^{t_n} ||\dot{\boldsymbol{\epsilon}}^p|| dt \\ \bar{\epsilon}^p_{v_n} = \int_0^{t_n} \text{tr}(\dot{\boldsymbol{\epsilon}}^p) dt \\ \bar{\epsilon}^p_{s_n} = \int_0^{t_n} ||\dot{\boldsymbol{\epsilon}}^p - \frac{1}{3}\text{tr}(\dot{\boldsymbol{\epsilon}}^p)|| dt \\ e_n = e_0 + \int_0^{t_n} \dot{e} dt = e_0 + \int_0^{t_n} (1+e)\dot{\epsilon}_v dt \end{cases}, \tag{7}$$

where $\bar{\epsilon}^p$, $\bar{\epsilon}^p_v$ and $\bar{\epsilon}^p_s$ are accumulated total, volumetric and deviatoric plastic strains, respectively. e is the void ratio for granular materials, defined as the ratio between volume of the void and the solid constituent. We assume that the yield function is isotropic and therefore can be expressed in terms of stress invariants [9]. As a result, the phenomenological relations can be represented as functions of a stress invariants $\boldsymbol{\sigma}^{ivr}_n$, which may include

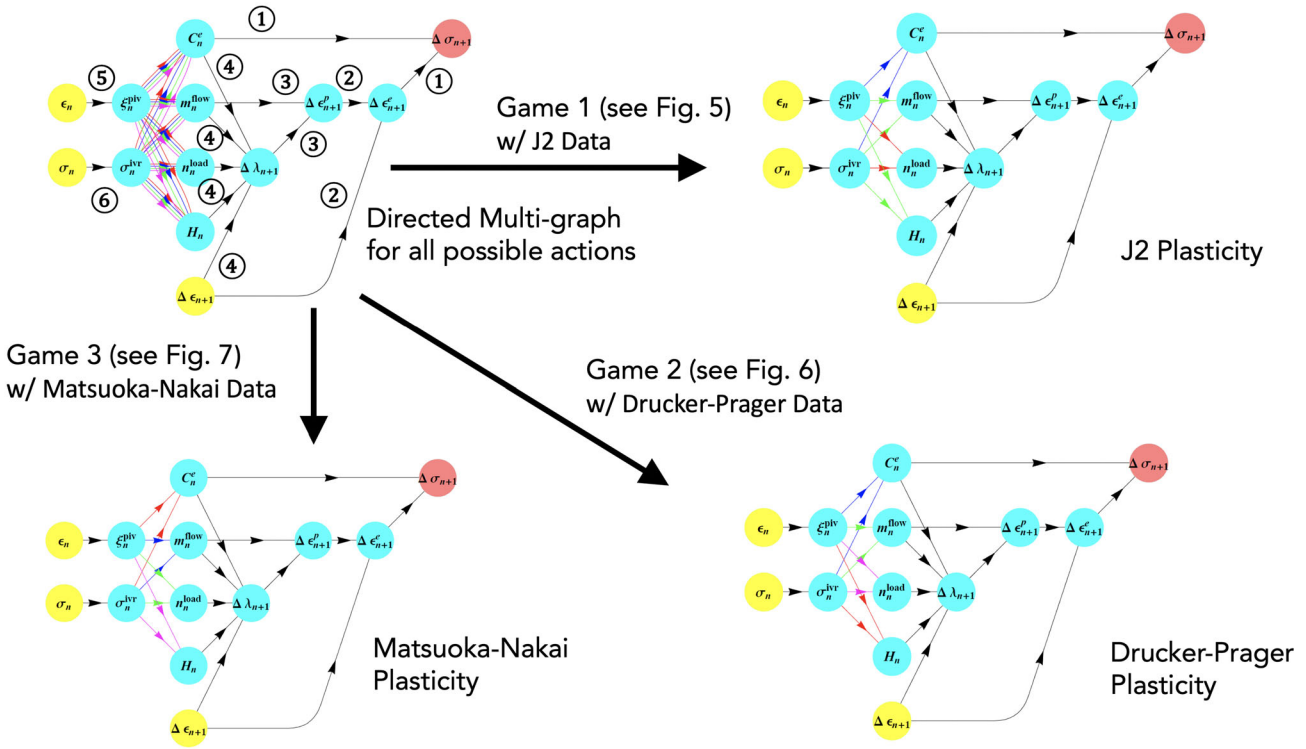


Fig. 2 Directed multigraph of an elasto-plasticity model.The yellow nodes of the strain $\boldsymbol{\epsilon}_n$, stress $\boldsymbol{\sigma}_n$ and strain increment $\Delta\boldsymbol{\epsilon}_{n+1}$ refer to the root nodes, the pink node of the stress increment $\Delta\boldsymbol{\sigma}_{n+1}$ refers to the leaf node, and the cyan nodes refer to intermediate nodes. The black arrows refer to "definition" edges. The color arrows refer to "phenomenological relations" edges. In the Meta-modeling game, the modeler AI agent generates the optimal configuration of the model from the labeled directed multi-graph for a given set of data. In the case of reverse engineering, the modeler AI agent should be able to recover the original constitutive laws when given the corresponding types of data. (Color figure online)

$$\textcircled{6}\begin{cases} p_n = \dfrac{\mathrm{tr}(\boldsymbol{\sigma}_n)}{3} \\ q_n = \sqrt{3J_2} = \sqrt{\dfrac{3}{2}}||\boldsymbol{s}_n|| \\ \theta_n = \dfrac{1}{3}\sin^{-1}(-\dfrac{3\sqrt{3}}{2}\dfrac{J_3}{J_2^{3/2}}),\ -\dfrac{\pi}{6} \le \theta \le \dfrac{\pi}{6} \end{cases} \tag{8}$$

where $J_2 = \frac{1}{2}\mathrm{trace}(\boldsymbol{s}_n^2)$, $J_3 = \frac{1}{3}\mathrm{trace}(\boldsymbol{s}_n^3)$, $\boldsymbol{s}_n = \boldsymbol{\sigma}_n - p_n\boldsymbol{I}$ and θ_n is the Lode's angle, the smallest angle between the line of pure shear and the projection of stress tensor in the deviatoric plane [49]. The constitutive relation between the loading direction $\boldsymbol{n}^{load}$ and the state variables $\boldsymbol{\xi}_n^{piv}$, $\boldsymbol{\sigma}_n^{ivr}$ can be defined either by formulating a yield surface f such that,

$$\boldsymbol{n}^{load} = \frac{\partial f}{\partial \boldsymbol{\sigma}}||\frac{\partial f}{\partial \boldsymbol{\sigma}}||^{-1}, \tag{9}$$

or, in the case yield surface is absence, directly inferred from observations as those in the generalized plasticity framework (cf. [44,48,63]),

$$\boldsymbol{n}^{load} = n_v^{load}\boldsymbol{n}_v + n_s^{load}\boldsymbol{n}_s. \tag{10}$$

where

$$\begin{cases} \boldsymbol{n}_v = \dfrac{\partial p}{\partial \boldsymbol{\sigma}} = \dfrac{1}{3}\boldsymbol{I} \\ \boldsymbol{n}_s = \dfrac{\partial q}{\partial \boldsymbol{\sigma}} = \dfrac{\sqrt{3}}{2\sqrt{J_2}}\boldsymbol{S}. \end{cases} \tag{11}$$

Similarly, the constitutive relation between the plastic flow direction $\boldsymbol{m}^{flow}$ and the state variables $\boldsymbol{\xi}_n^{piv}$, $\boldsymbol{\sigma}_n^{ivr}$ can be defined either by formulating a plastic potential surface g such that,

$$\boldsymbol{m}^{flow} = \frac{\partial g}{\partial \boldsymbol{\sigma}}||\frac{\partial g}{\partial \boldsymbol{\sigma}}||^{-1}. \tag{12}$$

or directly inferred from observations as those in the generalized plasticity framework (cf. [44,48,63])

$$\boldsymbol{m}^{flow} = m_v^{flow}\boldsymbol{n}_v + m_s^{flow}\boldsymbol{n}_s. \tag{13}$$

3.2.2 Game states for modeler

The mathematical description of the current state of the game board is a list of binary indicators $s = [i_{e1}, i_{e2}, i_{e3}, \ldots, i_{en}]$

representing whether a labeled edge E_{ei} in the labeled edge set $\mathbb{L}_{\mathbb{E}}$ of the directed multigraph $\mathbb{G}$ is selected in the final directed graph $\mathbb{G}'$. If E_{ei} is included in $\mathbb{G}'$, the corresponding indicator $i_{ei} = 1$, otherwise $i_{ei} = 0$. The initial state of the game is $i_{ei} = 0,\ \forall E_{ei} \in \mathbb{L}_{\mathbb{E}}$.

3.2.3 Game actions for modeler

At each state s, the modeler can select the next labeled edge $E_{ei} \in \mathbb{L}_{\mathbb{E}}$, by changing the indicator i_{ei} from 0 to 1.

3.2.4 Game rules for modeler

The modeling choices for the four essential components in an elasto-plasticity model are not fully compatible with each other. For example, a J2 yield surface only has the yield stress as the stress-like plastic internal variable, while a strain hardening law for a Drucker–Prager yield surface has both frictional and cohesion hardening laws. These restrictions on compatible edge choices are specified by a list of binaries $LegalActions(s) = [ii_1, ii_2, ii_3, \ldots, ii_n]$ of legal choices for each state. Another set of game rules consist of universal principles on the constitutive models. For example, thermodynamic consistency would require that rate of mechanical dissipation remains non-negative for isothermal process [9,73], i.e.

$$\mathcal{D} = \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}} - \frac{d\psi}{dt} \geq 0, \tag{14}$$

where $\boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}}$ represents the stress power per unit volume and $\frac{d\psi}{dt}$ represents the rate of change of Helmholtz free energy, which may take the following form,

$$\psi(\boldsymbol{\epsilon}^e, \boldsymbol{\xi}) = \psi^e(\boldsymbol{\epsilon}^e) + \psi^p(\boldsymbol{\xi}). \tag{15}$$

where $\psi^e(\boldsymbol{\epsilon}^e)$ and $\psi^p(\boldsymbol{\xi})$ are the elastic and plastic contributions of the Helmoltz free energy and $\boldsymbol{xi}$ is a collection of history-dependent internal variables. In our implementation, we assume that the deformation is infinitesimal. As a result, the additive decomposition of the total strain rate into the elastic and plastic components is valid and Eq. (14) could also be rewritten as [9],

$$\mathcal{D} = (\boldsymbol{\sigma} - \frac{\partial \psi^r}{\partial \boldsymbol{\epsilon}^e}) : \dot{\boldsymbol{\epsilon}} + \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}} - \frac{\partial \psi^p}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} \geq 0. \tag{16}$$

Readers interested to obtain further information on the constraints due to the thermodynamic laws are referred to [9,73,88]. In the cooperative game, the thermodynamic laws are converted into game rules then enforced implicitly by introduce a penalty t the model score. If the final model in an episode violates this rule, the final model score is set to be 0. This low score is then used as training material for the mastermind modeler agent such that it reduces the policy probabilities of the choices that violate universal principles as shown in Fig. 3. As the deep reinforcement learning progresses, the modeler agent will gradually learn to avoid generating models that violate the thermodynamic rules through the low policy values. Since the training of the constitutive law can only be completed if the score of the best candidate model is sufficiently high, this prevents the meta-modeling algorithm from generating any model that violates the first principles.

Note that the thermodynamic laws are not the only game rules in the cooperative game. Another physical law we enforced in this game is the frame indifference, first discussed in [40]. In this work, the frame indifference is enforced by representing tensors in spectral forms, then using Lie-algebra to establishing the mapping from one orthogonal basis to the other. The detailed operations have been documented in [94,95] and will not repeat in this paper for brevity.

3.2.5 Game reward for modeler

A score system must be introduced to evaluate the generated directed graphs for constitutive models such that the accuracy and credibility in replicating the mechanical behavior of real-world materials can be assessed. This score system may also serve as the objective function that defines the rewards for the deep reinforcement learning agent to improve the generated digraphs and resultant constitutive laws. In this work, we define the score as a positive real-valued function of the range [0, 1] which depends on the measures A_i $(i = 1, 2, 3, \ldots, n)$ of n important features of a constitutive model,

$$\text{SCORE} = F(A_1, A_2, A_3, \ldots, A_n), \tag{17}$$

where $0 \leq A_i \leq 1$. Some features are introduced to measure the performance of a model such as the accuracy and computation speed. Other features are introduced to enforce constraints to ensure the admissibility of a constitutive model, such as the frame indifference and the thermodynamic consistency. Suppose there are n_{pfm} measures of performance features A_i^{pfm} and n_{crit} measures of critical features A_i^{crit} in the measure system of constitutive models, the score takes the form,

$$\text{SCORE} = \left(\prod_{j=1}^{n_{\text{crit}}} A_j^{\text{crit}} \right) \cdot \left(\sum_{i=1}^{n_{\text{pfm}}} w_i A_i^{\text{pfm}} \right), \tag{18}$$

where $w_i \in [0, 1]$ is the weight associated with the measure A_i^{pfm}, and $\sum_{i=1}^{n_{\text{pfm}}} w_i = 1$.

For example, for measures of accuracy A_{accuracy} of calibrations and forward predictions, we introduce a cross-

validation procedure in which the dataset used for training the models (e.g. identifying material parameters (e.g. [45,97]) or adjusting weights of neurons in recurrent neural networks (e.g. [40,94]) is mutually exclusive to the testing dataset used to evaluate the quality of blind predictions. Both calibration and blind prediction results are compared against the target data. The mean squared error (MSE) commonly used in statistics and also as objective function in machine learning is chosen as the error measure for each data sample i in this study, i.e.,

$$\mathrm{MSE}_i = \frac{1}{N_{\text{feature}}} \sum_{j=1}^{N_{\text{feature}}} [\mathcal{S}_j(Y_{i_j}^{\text{data}}) - \mathcal{S}_j(Y_{i_j}^{\text{model}})]^2, \tag{19}$$

where $Y_{i_j}^{\text{data}}$ and $Y_{i_j}^{\text{model}}$ are the values of the jth feature of the ith data sample, from target data value and predictions from constitutive models, respectively. N_{feature} is the number of output features. $\mathcal{S}_j$ is a scaling operator (standardization, min-max scaling, ...) for the output feature $\{Y_{i_j}\}$, $i \in [1, N_{\text{data}}]$.

The empirical cumulative distribution functions (eCDFs) are computed for MSE of the entire dataset $\{\mathrm{MSE}_i\}$, $i \in [1, N_{\text{data}}]$, for MSE of the training dataset $\{\mathrm{MSE}_i\}$, $i \in [1, N_{\text{traindata}}]$ and for MSE of the test dataset $\{\mathrm{MSE}_i\}$, $i \in [1, N_{\text{testdata}}]$, with the eCDF defined as [33],

$$F_N(\mathrm{MSE}) = \begin{cases} 0, & \mathrm{MSE} < \mathrm{MSE}_1, \\ \dfrac{r}{N}, & \mathrm{MSE}_r \le \mathrm{MSE} < \mathrm{MSE}_{r+1},\ r = 1, \ldots, N-1, \\ 1, & \mathrm{MSE}_N \le \mathrm{MSE}, \end{cases} \tag{20}$$

where $N = N_{\text{data}}$, or $N_{\text{traindata}}$, or N_{testdata}, and all $\{\mathrm{MSE}_i\}$ are arranged in increasing order. A measure of accuracy is proposed based on the above statistics,

$$A_{\text{accuracy}} = \max\left(\frac{\log[\max(\varepsilon_{P\%}, \varepsilon_{\text{crit}})]}{\log \varepsilon_{\text{crit}}}, 0\right), \tag{21}$$

where $\varepsilon_{P\%}$ is the Pth percentile (the MSE value corresponding to $P\%$ in the eCDF plot) of the eCDF on the entire, training or test dataset. $\varepsilon_{\text{crit}} \ll 1$ is the critical MSE chosen by users such that a model can be considered as "satisfactorily accurate" when $\varepsilon_{P\%} \le \varepsilon_{\text{crit}}$.

Once a complete constitutive model is generated, the model score is evaluated. The final reward is defined as: if the current score is higher than the average score of models from a group of already played games by the agent, then the current game wins and $r_T = 1$, otherwise, the current game loses and $r_T = -1$. The average score can be initialized to 0 for the first game.

3.2.6 Game choices for modeler

This section specifies the candidate edges in the directed multi-graph of elasto-plasticity models (Fig. 2) for the modeler agent to choose during deep reinforcement learning. The edges are categorized into four groups representing the four essential constitutive relations in the model. The edges $\boldsymbol{\sigma}_n^{ivr} \to \boldsymbol{C}_n^e$ and $\boldsymbol{\xi}_n^{piv} \to \boldsymbol{C}_n^e$ represent the elasticity law. The edges $\boldsymbol{\sigma}_n^{ivr} \to \boldsymbol{n}_n^{load}$ and $\boldsymbol{\xi}_n^{piv} \to \boldsymbol{n}_n^{load}$ represent the definition of the loading direction. The edges $\boldsymbol{\sigma}_n^{ivr} \to \boldsymbol{m}_n^{flow}$ and $\boldsymbol{\xi}_n^{piv} \to \boldsymbol{m}_n^{flow}$ represent the definition of the plastic flow direction. The edges $\boldsymbol{\sigma}_n^{ivr} \to H_n$ and $\boldsymbol{\xi}_n^{piv} \to H_n$ represent the hardening law. Each edge allows multiple choices extracted from the phenomenological relations developed in the computational plasticity literature. In this paper, for simplicity of illustration of the meta-modeling game framework, the edge choices are not exhaustive. The following lists only contain common representative choices for geomaterials. But the designer of the meta-modeling game is always free to add more edge choices to expand the action space.

The edges for elasticity law ($\boldsymbol{\sigma}_n^{ivr} \to \boldsymbol{C}_n^e$ and $\boldsymbol{\xi}_n^{piv} \to \boldsymbol{C}_n^e$) represent the definition and evolution of the elastic stiffness tensor

$$\boldsymbol{C}_n^e = K\boldsymbol{I} \otimes \boldsymbol{I} + 2G\left(\boldsymbol{I}_{sym}^4 - \frac{\boldsymbol{I} \otimes \boldsymbol{I}}{3}\right), \tag{22}$$

where K is the tangential elastic bulk modulus and G is the tangential shear modulus.

Three common formulations of the elastic stiffness tensor for granular materials are available for model choice:

(E1) Linear elasticity

$$\begin{cases} K = K_0 \\ G = G_0 \end{cases}, \tag{23}$$

where K_0 and G_0 are constants.

(E2) Nonlinear elasticity with dependence on the mean pressure p [51]

$$\begin{cases} K = K_0 \left(\dfrac{p}{p_{at}}\right)^a \\ G = G_0 \left(\dfrac{p}{p_{at}}\right)^a \end{cases}, \tag{24}$$

where p_{at} is the atmospheric pressure ($\approx -100\,\mathrm{kPa}$) and a is a material constant.

(E3) Nonlinear elasticity with dependence on the mean pressure p and the void ratio e [19]

$$\begin{cases} K = \dfrac{2(1+\nu)}{3(1-2\nu)} G \\ G = G_0 p_{at} \dfrac{(2.97-e)^2}{1+e} \left(\dfrac{p}{p_{at}}\right)^{1/2} \end{cases}, \tag{25}$$

where ν is the constant Poisson's ratio.

The edges ($\boldsymbol{\sigma}_n^{ivr} \rightarrow \boldsymbol{n}_n^{load}$ and $\boldsymbol{\xi}_n^{piv} \rightarrow \boldsymbol{n}_n^{load}$) represent the definition and evolution of the loading direction. $\boldsymbol{n}_n^{load}$ can be either derived from an assumed yield surface $f \leq 0$ or defined explicitly in the space of stress invariants $\boldsymbol{\sigma}_n^{ivr}$.

The following common formulations of loading direction for granular materials are considered for model choices:

(L1) Yield surface of J2 plasticity $f = q - \sigma_y$ and linear hardening law

$$\sigma_y = \sigma_{y0} + H_0 \bar{\epsilon}^p, \tag{26}$$

where σ_{y0}, H_0 are material parameters.

(L2) Yield surface of J2 plasticity $f = q - \sigma_y$ and σ_y is the solution of the power law equation

$$\frac{\sigma_y}{\sigma_{y0}} = \left(\frac{\sigma_y}{\sigma_{y0}} + \frac{3G}{\sigma_{y0}} \bar{\epsilon}^p\right)^n, \tag{27}$$

where σ_{y0}, n are material parameters, G is the elastic shear modulus.

(L3) Yield surface of J2 plasticity $f = q - \sigma_y$ and Voce hardening law

$$\sigma_y = \sigma_{y0} + H_0 \bar{\epsilon}^p + H_\infty (1 - \exp(-b\bar{\epsilon}^p)), \tag{28}$$

where σ_{y0}, H_0, H_∞, b are material parameters.

(L4) Yield surface of Drucker–Prager plasticity $f = q + \alpha p$ and α evolves according to

$$\alpha = a_0 + a_1 \bar{\epsilon}^p \exp(a_2 p - a_3 \bar{\epsilon}^p), \tag{29}$$

where a_0, a_1, a_2, a_3 are material parameters [89].

(L5) Yield surface of Drucker–Prager plasticity $f = q + \alpha p$ and α evolves according to

$$\alpha = a_0 + 2a_1 \frac{\sqrt{k\bar{\epsilon}^p}}{k + \bar{\epsilon}^p}, \tag{30}$$

where a_0, a_1, k are material parameters [9].

(L6) Yield surface of three-invariant Matsuoka–Nakai model [10]

$$\begin{cases} f = (k_1 I_3)^{1/3} - (I_1 I_2)^{1/3} \\ k_1 = c_0 + \kappa_1 \left(\dfrac{p_{at}}{I_1}\right)^m \\ \kappa_1 = a_1 \bar{\epsilon}^p \exp(a_2 I_1) \exp(-a_3 \bar{\epsilon}^p) \end{cases}, \tag{31}$$

where c_0, a_1, a_2, a_3, m are material parameters.

(L7) Yield surface of Nor-Sand [1,32]

$$\begin{cases} f = \zeta q + \eta p \\ \zeta = \dfrac{(1+\rho) + (1-\rho)\cos 3(\theta + \pi/6)}{2\rho} \\ \eta = \begin{cases} M[1 + \log(p_i/p)] & \text{if } N = 0 \\ (M/N)[1 - (1-N)(p/p_i)^{N/(1-N)}] & \text{if } N > 0 \end{cases} \\ \dot{p}_i = -\sqrt{\dfrac{2}{3}} h (p_i - p_i^*) ||\dot{\boldsymbol{e}}^p||, \ \dot{\boldsymbol{e}}^p = \dot{\boldsymbol{\epsilon}}^p - \dfrac{1}{3} tr(\dot{\boldsymbol{\epsilon}}^p) \boldsymbol{I} \\ \dfrac{p_i^*}{p} = \begin{cases} \exp(\bar{\alpha}\psi_i / M) & \text{if } \bar{N} = N = 0 \\ (1 - \bar{\alpha}\psi_i N / M)^{(N-1)/N}] & \text{if } 0 \leq \bar{N} \leq N \neq 0 \end{cases} \\ \bar{\alpha} = -3.5 \dfrac{1 - \bar{N}}{1 - N} \\ \psi_i = e - e_{c0} + \tilde{\lambda}(p_i / p_{at})^a \end{cases} \tag{32}$$

where $\rho, N, \bar{N}, M, h, e_{c0}, \tilde{\lambda}, a$ are material parameters.

(L8) Yield surface in the shape of a small cone [19]

$$\begin{cases} f = ||\boldsymbol{S} - p\boldsymbol{\alpha}|| - \sqrt{2/3} pm \\ \dot{\boldsymbol{\alpha}} = \dot{\lambda}(2/3) h (\boldsymbol{\alpha}_\theta^b - \boldsymbol{\alpha}) \\ \boldsymbol{\alpha}_\theta^b = \sqrt{2/3} [\dfrac{1}{\zeta} M \exp(-n^b \psi) - m] \boldsymbol{n} \\ \zeta = \dfrac{(1+\rho) + (1-\rho)\cos 3(\theta + \pi/6)}{2\rho} \\ \boldsymbol{n} = \dfrac{\frac{\boldsymbol{S}}{p} - \boldsymbol{\alpha}}{\sqrt{2/3} m} \\ \psi = e - e_{c0} + \tilde{\lambda}(p / p_{at})^a \end{cases}, \tag{33}$$

where $\rho, m, M, n^b, h, e_{c0}, \tilde{\lambda}, a$ are material parameters.

(L9) Loading direction defined as [63,104]

$$\begin{cases} n_v^{load} = \dfrac{d_f}{\sqrt{1 + d_f^2}} \\ n_s^{load} = \dfrac{1}{\sqrt{1 + d_f^2}} \\ d_f = (1+\alpha)(M_f + q/p) \end{cases}, \tag{34}$$

where α, M_f are material parameters.

(L10) Loading direction defined as [44]

$$\begin{cases} n_v^{load} = \dfrac{d_f}{\sqrt{1+d_f^2}} \\ n_s^{load} = \dfrac{1}{\sqrt{1+d_f^2}} \\ d_f = (1+\alpha)(M_f \exp(m_f(1-e)) + q/p) \end{cases}, \quad (35)$$

where α, M_f, m_f are material parameters.

(L11) Loading direction given by a neural network trained with data inversely computed from experimental data (described later in the definition of plastic modulus edges).

The edges ($\boldsymbol{\sigma}_n^{ivr} \rightarrow \boldsymbol{m}_n^{flow}$ and $\boldsymbol{\xi}_n^{piv} \rightarrow \boldsymbol{m}_n^{flow}$) represent the definition and evolution of the plastic flow direction. $\boldsymbol{m}_n^{flow}$ can be either derived from an assumed plastic potential surface $g = 0$ or defined explicitly in the space of stress invariants $\boldsymbol{\sigma}_n^{ivr}$.

The following common formulations of the plastic flow direction for granular materials are considered for model choices:

(P1) Plastic potential surface of J2 plasticity $g = q - c_g$ and c_g is a parameter to ensure that the stress point is on the potential surface when the plastic deformation occurs.

(P2) Plastic potential surface of Drucker–Prager plasticity $g = q + \beta p - c_g$ and $\beta = \alpha - \beta_0$, where α can be defined through Eq. (29) or (30), and β_0 is an additional material parameter.

(P3) Plastic potential surface of three-invariant Matsuoka–Nakai model [10]

$$\begin{cases} g = (k_2 I_3)^{1/3} - (I_1 I_2)^{1/3} \\ k_2 = c_0 + \kappa_2 \left(\dfrac{p_{at}}{I_1}\right)^m \\ \kappa_2 = \alpha \kappa_1 \end{cases}, \quad (36)$$

where κ_1 can be defined through Eq. 31 and β_0 is an additional material parameter.

(P4) Plastic potential surface of Nor-Sand [1,32]

$$\begin{cases} g = \bar{\zeta} q + \bar{\eta} p \\ \bar{\zeta} = \dfrac{(1+\bar{\rho}) + (1-\bar{\rho})\cos 3(\theta + \pi/6)}{2\bar{\rho}} \\ \bar{\eta} = \begin{cases} M[1 + \log(\bar{p}_i/p)] & \text{if } \bar{N} = 0 \\ (M/\bar{N})[1 - (1-\bar{N})(p/\bar{p}_i)^{\bar{N}/(1-\bar{N})}] & \text{if } \bar{N} > 0 \end{cases} \end{cases}, \quad (37)$$

where $\bar{\rho}, \bar{N}, M$ are material parameters and $\bar{p}_i$ is a free parameter to ensure $g = 0$ when the material is undergoing plastic deformation.

(P5) Plastic flow direction defined as [19]

$$\begin{cases} \boldsymbol{m}^{flow} = B\boldsymbol{n} - C\left(\boldsymbol{n}^2 - \dfrac{1}{3}\boldsymbol{I}\right) + \dfrac{1}{3}D\boldsymbol{I} \\ B = 1 + \dfrac{3}{2}\dfrac{1-c}{c\xi}\cos 3(\theta + \pi/6) \\ C = 3\sqrt{3/2}\dfrac{1-c}{c\xi} \\ D = A_d(\boldsymbol{\alpha}_\theta^d - \boldsymbol{\alpha}) : \boldsymbol{n} \\ \boldsymbol{\alpha}_\theta^d = \sqrt{2/3}\left[\dfrac{1}{\zeta}M\exp(n^d\psi) - m\right]\boldsymbol{n} \end{cases}, \quad (38)$$

where $\rho, m, M, n^d, A_d, e_{c0}, \tilde{\lambda}, a$ are material parameters.

(P6) Plastic flow direction defined as [63,104]

$$\begin{cases} m_v^{flow} = \dfrac{d_g}{\sqrt{1+d_g^2}} \\ m_s^{flow} = \dfrac{1}{\sqrt{1+d_g^2}} \\ d_g = (1+\alpha)(M_g + q/p) \end{cases}, \quad (39)$$

where α, M_g are material parameters.

(P7) Plastic flow direction defined as [44]

$$\begin{cases} m_v^{flow} = \dfrac{d_g}{\sqrt{1+d_g^2}} \\ m_s^{flow} = \dfrac{1}{\sqrt{1+d_g^2}} \\ d_g = (1+\alpha)(M_g \exp m_g\psi + q/p) \\ \psi = e - e_{c0} + \tilde{\lambda}(p/p_{at})^a \end{cases}, \quad (40)$$

where $\alpha, M_g, m_g, e_{c0}, \tilde{\lambda}, a$ are material parameters.

(P8) Plastic flow direction given by a neural network trained with data inversely computed from experimental data (described later in the definition of plastic modulus edges).

The edges ($\boldsymbol{\sigma}_n^{ivr} \rightarrow H_n$ and $\boldsymbol{\xi}_n^{piv} \rightarrow H_n$) represent the definition of the generalized hardening modulus. H_n can be either derived from an assumed yield surface $f \leq 0$ or defined explicitly.

The following common formulations of hardening modulus for granular materials are considered for model choices:

(H1) Hardening modulus derived from classical yield surface $f(\boldsymbol{\sigma}, \boldsymbol{\epsilon}^p)$ and a chosen $\boldsymbol{m}^{flow}$.

$$H = -\frac{\partial f/\partial \boldsymbol{\epsilon}^p : \boldsymbol{m}^{flow}}{||\partial f/\partial \boldsymbol{\sigma}||}. \tag{41}$$

(H2) Hardening modulus defined as [63,104]

$$\begin{cases} H = H_0(-p)H_f(H_v + H_s) \\ H_f = \left(1 + \dfrac{q}{pM_f}\dfrac{\alpha_f}{1+\alpha_f}\right)^4 \\ H_v = 1 + \dfrac{q}{pM_g} \\ H_s = \beta_0\beta_1 \exp(-\beta_0\bar{\epsilon}_s^p) \end{cases}, \tag{42}$$

where $\alpha_f, M_f, H_0, e_{c0}, M_g, \beta_0, beta_1$ are material parameters.

(H3) Hardening modulus defined as [44]

$$\begin{cases} H = H_0\sqrt{p/p_{at}}H_f\left(1 + \dfrac{q}{pM_b}\right) \\ M_b = M_g \exp(-m_b\psi) \\ H_0 = H_{L0}\exp(m_0(1-e)) \\ H_f = \left(1 + \dfrac{q}{pM_f}\dfrac{\alpha_f}{1+\alpha_f}\right)^4 \end{cases}, \tag{43}$$

where $\alpha_f, M_f, H_{L0}, m_0, M_g, m_b, e_{c0}, \tilde{\lambda}, a$ are material parameters.

(H4) Hardening modulus given by a neural network trained with data inversely computed from experimental data.

The stress increment at each time step is known from the experimental data $\Delta\boldsymbol{\sigma}_{n+1}^{data} = \boldsymbol{\sigma}_{n+1}^{data} - \boldsymbol{\sigma}_n^{data}$. For a chosen elasticity law $\boldsymbol{C}_n^e(\boldsymbol{\sigma}_n^{ivr}, \boldsymbol{\xi}_n^{piv})$, the data of incremental plastic strain at each time step is given by [using Eq. (4)]

$$\Delta\boldsymbol{\epsilon}_{n+1}^p = \Delta\boldsymbol{\epsilon}_{n+1} - (\boldsymbol{C}_n^e)^{-1} : \Delta\sigma_{n+1}^{data}. \tag{44}$$

Then the incremental plastic multiplier is $\Delta\lambda_{n+1} = ||\Delta\boldsymbol{\epsilon}_{n+1}^p||$ and the plastic flow direction is obtained by $\boldsymbol{m}_n^{flow} = \Delta\boldsymbol{\epsilon}_{n+1}^p/\Delta\lambda_{n+1}$. Assuming associative flow rule, then $\boldsymbol{n}_n^{load} = \boldsymbol{m}_n^{flow}$. In this way, the plastic modulus can be uniquely inversely computed as

$$H_n = \frac{\boldsymbol{n}_n^{load} : \boldsymbol{C}_n^e : \Delta\boldsymbol{\epsilon}_{n+1}}{\Delta\lambda_{n+1}} - \boldsymbol{n}_n^{load} : \boldsymbol{C}_n^e : \boldsymbol{m}_n^{flow}. \tag{45}$$

3.2.7 Game choice alternatives: training neural network edges

In addition to the mathematical edges described above, we also consider the possibility of replacing any part of the elasto-plastic model with machine learning edges. In this framework, the machine learning models are not used to directly map strain history to stress, but are used for each individual edge in the directed graph to map the input vertices to the output vertices. For instance, the mapping of variables in the generalized plasticity framework can be obtained by training a recurrent neural network that represents the path-dependent constitutive relation between the history of input vertices of $\boldsymbol{\sigma}_n^{ivr}$ (p, q, θ) and $\boldsymbol{\xi}_n^{piv}$ $(\bar{\epsilon}^p, \bar{\epsilon}_v^p, \bar{\epsilon}_s^p, e)$ and the output vertices of $\boldsymbol{n}_n^{load}$, $\boldsymbol{m}_n^{flow}$ and H_n. The details of training data preparation, network design, training and testing are specified in the previous work on the meta-modeling framework for traction-separation models with data of microstructural features [95]. In this framework, all neural network edges are generated using the same neural network architecture, i.e., two hidden layers of 64 GRU (Gated recurrent unit) neurons in each layer, and the output layer as a dense layer with linear activation function. All input and output data are pre-processed by standard scaling using mean values and standard deviations. Each input feature considers its current value and 19 history values prior to the current loading step. Each neural network is trained for 1000 epochs using the Adam optimization algorithm, with a batch size of 256. Finally, it should be noticed that one can further generalize the meta-modeling game by considering multiple neural network architectures as possible edges in the meta-modeling game. This generalization will be considered in the future but is out of the scope of the current study.

3.2.8 Directed labeled multi-graph and sub-graph training via supervised machine learning

To create the directed labelled multi-graph that represents the action space of the modeling agent, we first consider the directed graph that represents the hierarchical relationships among all the vertices, i.e. $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ where

$$\mathbb{V} = \{\boldsymbol{\epsilon}_n, \boldsymbol{\sigma}_n, \boldsymbol{\xi}_n^{piv}, \boldsymbol{\sigma}_n^{ivr}, \boldsymbol{C}_n^e, \boldsymbol{m}_n^{\text{flow}}, \boldsymbol{n}_n^{\text{load}}, H_n, \boldsymbol{\Delta\epsilon}_{n+1}, \Delta\lambda_{n+1}, \boldsymbol{\Delta\epsilon}_{\boldsymbol{n+1}}^{\boldsymbol{p}}, \boldsymbol{\Delta\epsilon}_{n+1}^e, \boldsymbol{\Delta\sigma}_{n+1}\} \tag{46}$$

$$\mathbb{E} = \mathbb{E}_1 \cup \mathbb{E}_2 \cup \mathbb{E}_3 \cup \mathbb{E}_3 \cup \mathbb{E}_4 \cup \mathbb{E}_5 \cup \mathbb{E}_6 \cup \mathbb{E}_{\text{game}} \tag{47}$$

$$\mathbb{E}_1 = \{\boldsymbol{C}_n^e \to \boldsymbol{\Delta\sigma}_{n+1}, \boldsymbol{\Delta\epsilon}_{n+1}^e \to \boldsymbol{\Delta\sigma}_{n+1}\} \tag{48}$$

$$\mathbb{E}_2 = \{\boldsymbol{\Delta\epsilon}_{\boldsymbol{n+1}}^{\boldsymbol{p}} \to \boldsymbol{\Delta\epsilon}_{n+1}^e, \boldsymbol{\Delta\epsilon}_{n+1} \to \boldsymbol{\Delta\epsilon}_{n+1}^e\} \tag{49}$$

$$\mathbb{E}_3 = \{\boldsymbol{m}_n^{\text{flow}} \to \boldsymbol{\Delta\epsilon}_{n+1}^e, \Delta\lambda_{n+1} \to \boldsymbol{\Delta\epsilon}_{n+1}^e\} \tag{50}$$

$$\mathbb{E}_4 = \{\boldsymbol{\Delta\epsilon}_{n+1} \to \Delta\lambda_{n+1}, \boldsymbol{C}_n^e \to \Delta\lambda_{n+1}, \boldsymbol{m}_n^{\text{flow}} \to \Delta\lambda_{n+1}, \boldsymbol{n}_n^{\text{load}} \to \Delta\lambda_{n+1}, H_n \to \Delta\lambda_{n+1}\} \tag{51}$$

$$\mathbb{E}_5 = \{\boldsymbol{\epsilon}_n \to \boldsymbol{\xi}_n^{piv}\} \tag{52}$$

$$\mathbb{E}_6 = \{\boldsymbol{\sigma}_n \to \boldsymbol{\sigma}_n^{ivr}\} \tag{53}$$

$$\mathbb{E}_{\text{game}} = \{\boldsymbol{\xi}_n^{piv} \to \boldsymbol{C}_n^e, \boldsymbol{\sigma}_n^{ivr} \to \boldsymbol{C}_n^e, \boldsymbol{\xi}_n^{piv}$$

$$\rightarrow \boldsymbol{m}_n^{\text{flow}}, \boldsymbol{\sigma}_n^{ivr} \rightarrow \boldsymbol{m}_n^{\text{flow}}, \boldsymbol{\xi}_n^{piv} \rightarrow \boldsymbol{n}_n^{\text{load}}, \boldsymbol{\sigma}_n^{ivr} \rightarrow \boldsymbol{n}_n^{\text{load}}, \boldsymbol{\xi}_n^{piv} \rightarrow H_n, \boldsymbol{\sigma}_n^{ivr} \rightarrow H_n\}. \tag{54}$$

In this directed graph, the labels corresponding to the edge elements in the edge sets $\{\mathbb{E}_1, \mathbb{E}_2, \mathbb{E}_3, \mathbb{E}_4, \mathbb{E}_5, \mathbb{E}_6$ area already pre-determined as they are simply mathematical definitions or rules that are of sufficient certainty (cf. [94]). Therefore, they can be excluded from the meta-modeling games. The rest of the edges are elements of the edge subset $\mathbb{E}_{\text{game}}$. The goal of the modeling agent is therefore to find the subset of the edge set $\mathbb{E}_{\text{game}}$ and identify the optimal "edge labels" for each element in this subset such that the performance of the constitutive laws measured by the objective function can be optimized. Note that these edge labels can be mathematical expressions or machine-learning-generated operators, each of them provides a mapping that links the first and second elements of the ordered pair of vertices.

As a result, the directed labeled multi-graph that represents all the possible choices of modeling choices considered in the meta-modeling game are generated the directed graph $\mathbb{G}_{game} = \{\mathbb{V}_{\text{game}}, \mathbb{E}_{\text{game}}\}$, a sub-graph of $\mathbb{G}$, where $\mathbb{V}_{\text{game}} = \{\boldsymbol{\xi}_n^{piv}, \boldsymbol{\sigma}_n^{ivr}, \boldsymbol{C}_n^e, \boldsymbol{m}_n^{\text{flow}}, \boldsymbol{n}_n^{\text{load}}, H_n\}$. Meanwhile, the edge labels $\mathbb{L}_{\mathbb{E}}$ we considered in this game are discussed in Sects. 3.2.6 and 3.2.7, i.e.,

$$\begin{aligned}\mathbb{L}_{\mathbb{E}} = \{&\text{E1, E2, E3, L1, L2, L3, L4, L5, L6, L7, L8, L9, L10,}\\ &\text{L11, P1, P2, P3, P4, P5, P6, P7, P8, H1, H2, H3, H4}\}\end{aligned} \tag{55}$$

After the experimental agent make the decision on the data, the modeler agent first generate a new directed graph based on the Q values estimation for each label in $\mathbb{L}_{\mathbb{E}}$, then a local inverse problem is solved to obtain either the material parameters for each selected edges or to complete the training of the neural network (if a neural net edge is chosen). Note that the edges are not necessarily trained individually. For each child vertex in the directed graph, all the edges that connects the parent vertices to it must be trained together. These subgraphs that connect one generation of relationship are generated via Algorithm 1.

Remarks on implementation An elasto-plasticity model, once generated from AI, needs to be numerically integrated to compute the predicted stresses under different types of tests. Since the loading directions, plastic flow directions and hardening modulus can have a large number of options and may be exceedingly complex, we adopt a general-purpose explicit integration algorithm for all AI generated models, instead of using different implicit integration techniques necessary for different models. This algorithm is a combination of (1) the explicit integration with sub-stepping and automatic error control [74,75] (2) explicit integration of (potentially non-smooth) hardening laws [89] (3) integration of generalized plasticity models [20,53] (4) linearized integration for loading constraints [5]. The algorithm is detailed in Algorithm 2. This explicit scheme is versatile and stable, but not as accurate as fully implicit return mapping algorithms, hence for the evaluation of model accuracy scores, small time steps are required for the numerical integration.

Algorithm 1 Backtracking generation of sub-graphs for training of sub-models

Require: AI-generated directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ of the elasto-plasticity model.
$\mathbb{V} = \{\Delta\boldsymbol{\sigma}_{n+1}, \boldsymbol{\sigma}_n, \Delta\boldsymbol{\epsilon}_{n+1}, \Delta\boldsymbol{\epsilon}_{n+1}^e, \Delta\boldsymbol{\epsilon}_{n+1}^p, \Delta\lambda_{n+1}, \boldsymbol{n}_n^{load}, \boldsymbol{m}_n^{flow}, H_n, p_n, q_n, \theta_n, e_n\}$ (see Sect. 3.1.1 for detailed definitions)

1: Identify all paths $Path = \{path_1, path_2, \ldots\}$ from the source nodes $\Delta\boldsymbol{\epsilon}_{n+1}$ and $\boldsymbol{\sigma}_{n+1}$ to the target node $\Delta\boldsymbol{\sigma}_{n+1}$ in the directed graph $\mathbb{G}$.
2: Identify the "node ranks" $NodeRank = [V_1, V_2, V_3, \ldots]$ in descendant rank order (each node is assigned a higher rank than the highest ranked node that point to it) for every nodes in all paths $Path$.
3: **for** V_i in $NodeRank$ **do**
4: Regard V_i as the successor node $V_i^s = V_i$, find its all predecessor nodes V_i^p.
5: Identify the sub-graph $\mathbb{G}_{V_i^s}^{V_i^p}$ which has V_i^p as the source nodes (Input) and the target node V_i^s (Output).
6: Train the sub-neural-network or calibrate the sub-model $SubModel_{V_i^s}^{V_i^p}$ represented by the sub-graph $\mathbb{G}_{V_i^s}^{V_i^p}$ with input data V_i^p and output data V_i^s.
7: Collect all sub-neural-networks or sub-models $SubModel_{V_i^s}^{V_i^p}, \forall V_i \in NodeRank$
8: Exit

4 Deep reinforcement learning for the two-player meta-modeling game

With the two-player game completely defined in the previous section, a deep reinforcement learning (DRL) algorithm is employed as a guidance of taking actions of both experimentalist and modeler in the game to maximize the final model score (Fig. 3). The learning is completely free of human interventions after the game settings. This tactic is considered one of the key ideas leading to the major breakthrough in AI playing the game of Go (AlphaGo Zero) [70], Chess and shogi (Alpha Zero) [71] and many other games. In [95], the key ingredients (Policy/Value network, confidence bound for Q-value, Monte Carlo Tree Search) of the DRL technique are applied to a meta-modeling game for modeler agent only, focusing on finding the optimal topology of physical relations from fixed training/testing datasets. In this work, the game design is further extended that (1) the modeling game also involves the "component selection" from a set of candidate edge choices having the same source and

Algorithm 2 Explicit Integration Scheme

Require: AI-generated directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ of the elasto-plasticity model.
Require: Initial values of stress $\boldsymbol{\sigma}_n$, strain $\boldsymbol{\epsilon}_n$, plastic internal variables $\boldsymbol{\xi}_n^{piv}$, and stress invariants $\boldsymbol{\sigma}_n^{ivr}$.
Identify elasticity model $\boldsymbol{C^e}(\boldsymbol{\sigma}^{ivr}, \boldsymbol{\xi}^{piv})$, loading direction $\boldsymbol{n}^{load}(\boldsymbol{\sigma}^{ivr}, \boldsymbol{\xi}^{piv})$ (or yield surface $f(\boldsymbol{\sigma}^{ivr}, \boldsymbol{\xi}^{piv})$ if it exists), plastic flow direction $\boldsymbol{m}^{flow}(\boldsymbol{\sigma}^{ivr}, \boldsymbol{\xi}^{piv})$ (or plastic potential $g(\boldsymbol{\sigma}^{ivr}, \boldsymbol{\xi}^{piv})$ if it exists), generalized hardening modulus $H(\boldsymbol{\sigma}^{ivr}, \boldsymbol{\xi}^{piv})$ from the directed graph $\mathbb{G}$.
Define matrices of loading constraints $\boldsymbol{E}$, $\boldsymbol{S}$, and $d\boldsymbol{Y}$ [5].
Update the elastic stiffness tensor $\boldsymbol{C}_n^e(\boldsymbol{\sigma}_n^{ivr}, \boldsymbol{\xi}_n^{piv})$, its Voigt form $\boldsymbol{D^e}$, and the hardening modulus $H_n(\boldsymbol{\sigma}_n^{ivr}, \boldsymbol{\xi}_n^{piv})$.
Solve for the strain increment tensor $\Delta\boldsymbol{\epsilon}_{n+1}$ from Voigt form equation $(\boldsymbol{SD^e} + \boldsymbol{E}) \cdot d\boldsymbol{\epsilon} = d\boldsymbol{Y}$.
Compute the trial elastic stress increment $\Delta\boldsymbol{\sigma}_{n+1}^e = \boldsymbol{C}_n^e : \Delta\boldsymbol{\epsilon}_{n+1}$ and the trial stress state $\boldsymbol{\sigma}_{n+1}^e = \boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}_{n+1}^e$.
Determine the 'Elastic Loading' or 'Plastic Loading' condition from Eqs. (5) and (6).
if Elastic Loading **then**
    $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^e$, $\boldsymbol{\epsilon}_{n+1} = \boldsymbol{\epsilon}_n + \Delta\boldsymbol{\epsilon}_{n+1}$, $\boldsymbol{\xi}_{n+1}^{piv} = \boldsymbol{\xi}_n^{piv}$.
    Exit
if Plastic Loading **then**
    if yield surface f exists **then**
        Find α such that $f(\boldsymbol{\sigma}_n + \alpha\Delta\boldsymbol{\sigma}_{n+1}^e, \boldsymbol{\xi}_n^{piv}) = 0$ (Pegasus intersection scheme [75]).
    else
        Set $\alpha = 0$.
    Set $T = 0$, $\Delta T = 1$, $\Delta\lambda = 0$, $\boldsymbol{\sigma}_T = \boldsymbol{\sigma}_n + \alpha\Delta\boldsymbol{\sigma}_{n+1}^e$, $\boldsymbol{\epsilon}_T = \boldsymbol{\epsilon}_n + \alpha\Delta\boldsymbol{\epsilon}_{n+1}$.
    while $T < 1$ **do**
        Compute $\boldsymbol{n}^{load}(\boldsymbol{\sigma}_T, \boldsymbol{\xi}_n^{piv})$, $\boldsymbol{m}^{flow}(\boldsymbol{\sigma}_T, \boldsymbol{\xi}_n^{piv})$.
        Update the elasto-plastic stiffness tensor $\boldsymbol{C^{ep}} = \boldsymbol{C^e} - \frac{1}{\chi}\boldsymbol{C^e} : \boldsymbol{m}^{flow} \otimes \boldsymbol{n}^{load} : \boldsymbol{C^e}$, where $\chi = \boldsymbol{n}^{load} : \boldsymbol{C^e} : \boldsymbol{m}^{flow} + H_n$ and its Voigt form $\boldsymbol{D^{ep}}$.
        Solve for the strain increment tensor $\Delta\epsilon_T$ from Voigt form equation $(\boldsymbol{SD^{ep}} + \boldsymbol{E}) \cdot d\boldsymbol{\epsilon}_T = (1-\alpha) \cdot \Delta T \cdot d\boldsymbol{Y}$.
        Update $\Delta\boldsymbol{\sigma}^e \leftarrow \boldsymbol{C^e} : \Delta\epsilon_T$.
        Compute $\Delta\lambda_1 = \frac{1}{\chi}\boldsymbol{n}^{load} : \Delta\boldsymbol{\sigma}^e$, $\Delta\boldsymbol{\sigma}_1 = \Delta\boldsymbol{\sigma}^e - \boldsymbol{C^e} : \Delta\lambda_1\boldsymbol{m}^{flow}$
        Compute $\boldsymbol{n}^{load}(\boldsymbol{\sigma}_T + \Delta\boldsymbol{\sigma}_1, \boldsymbol{\xi}_n^{piv})$, $\boldsymbol{m}^{flow}(\boldsymbol{\sigma}_T + \Delta\boldsymbol{\sigma}_1, \boldsymbol{\xi}_n^{piv})$.
        Compute $\Delta\lambda_2 = \frac{1}{\chi}\boldsymbol{n}^{load} : \Delta\boldsymbol{\sigma}^e$, $\Delta\boldsymbol{\sigma}_2 = \Delta\boldsymbol{\sigma}^e - \boldsymbol{C^e} : \Delta\lambda_2\boldsymbol{m}^{flow}$
        Compute $\tilde{\boldsymbol{\sigma}}_{T+\Delta T} = \boldsymbol{\sigma}_T + 0.5 \cdot (\Delta\boldsymbol{\sigma}_1 + \Delta\boldsymbol{\sigma}_2)$, and determine the relative error $R_{T+\Delta T} = \max(\frac{||\Delta\boldsymbol{\sigma}_2 - \Delta\boldsymbol{\sigma}_1||}{2||\tilde{\boldsymbol{\sigma}}_{T+\Delta T}||}, EPS)$
        if $R_{T+\Delta T} \leq STOL$ **then**
            $\boldsymbol{\sigma}_{T+\Delta T} = \tilde{\boldsymbol{\sigma}}_{T+\Delta T}$, $\boldsymbol{\epsilon}_{T+\Delta T} = \boldsymbol{\epsilon}_T + \Delta\epsilon_T$, $\Delta\lambda \leftarrow \Delta\lambda + 0.5 \cdot (\Delta\lambda_1 + \Delta\lambda_2)$.
            $T \leftarrow T + \Delta T$
            Determine new ΔT for the next substep [75].
        else
            Determine new ΔT for this failed substep [75].
    $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_T$, $\boldsymbol{\epsilon}_{n+1} = \boldsymbol{\epsilon}_T$, update $\boldsymbol{\xi}_{n+1}^{piv}$ from Eq. (7).
    Exit

target nodes (derive a directed graph from a directed multi-graph) and (2) the choice of training dataset is carried out by an additional experimentalist agent. Since DRL needs to figure out the optimal strategies for two agents, the algorithm is extended to multi-agent multi-objective DRL [21,84,85]. The AI for the experimentalist and modeler are separate agents, each has its own Policy/Value network and decision tree search. But their intelligence are improved simultaneously during the self-plays of the entire Data collection/Meta-modeling game, according to the individual rewards they receive from the game environment and the communications between themselves (Fig. 3). The strategies of both agents can be cooperative or competitive of different degrees, depending on the design of the game reward system (for example, the video game of Pong in [84]). In this work, we consider only the learning of fully cooperative strategies, as shown in the game reward system designed in Sects. 3.1 and 3.2.

The pseudocode of the reinforcement learning algorithm to play the two-player meta-modeling game is presented in Algorithm 3. This is an extension of the algorithm in [95]. As demonstrated in Algorithm 3, each complete DRL procedure involves $numIters$ number of training iterations and one final iteration for generating the final converged digraph model. Each iteration involves $numEpisodes$ number of game episodes that construct the training example set $trainExamples$ for the training of the policy/value network f_θ. For decision makings in each game episode, the action probabilities are estimated from $numMCTSSims$ times of MCTS simulations.

Remark 1 *Non-cooperative meta-modeling game and Nash equilibrium.* In the case of the cooperative game where both agents share the same goal or score system, there is no need to determine the Nash equilibrium as the joint actions of the experimentalist/modeler group takes a collective of payoffs. However, in many realistic situations in modern-day research, it is possible that the data and modeler agents may have different or even conflicting goals and hence finding the best strategies the two agents take is equivalent to finding the Nash equilibrium. The meta-modeling model, in this case, is not only helpful for generating models but also helps us understand the relationships among objectives between the data and modeler agents, the resultant actions taken by both players, and the outcomes, assuming each player is acting in a rational manner.

5 Numerical experiments

In this section, we present two cooperative modeling games with different data to demonstrate the intelligence, robustness and efficiency of the deep reinforcement learning algorithm on improving the accuracy and consistency of the generated elasto-plasticity models through self-plays. In the first example, synthetic data computed from selected J2, Drucker–Prager and Matsuoka–Nakai plasticity are used to train the data and model agents, to validate the meta-modeling

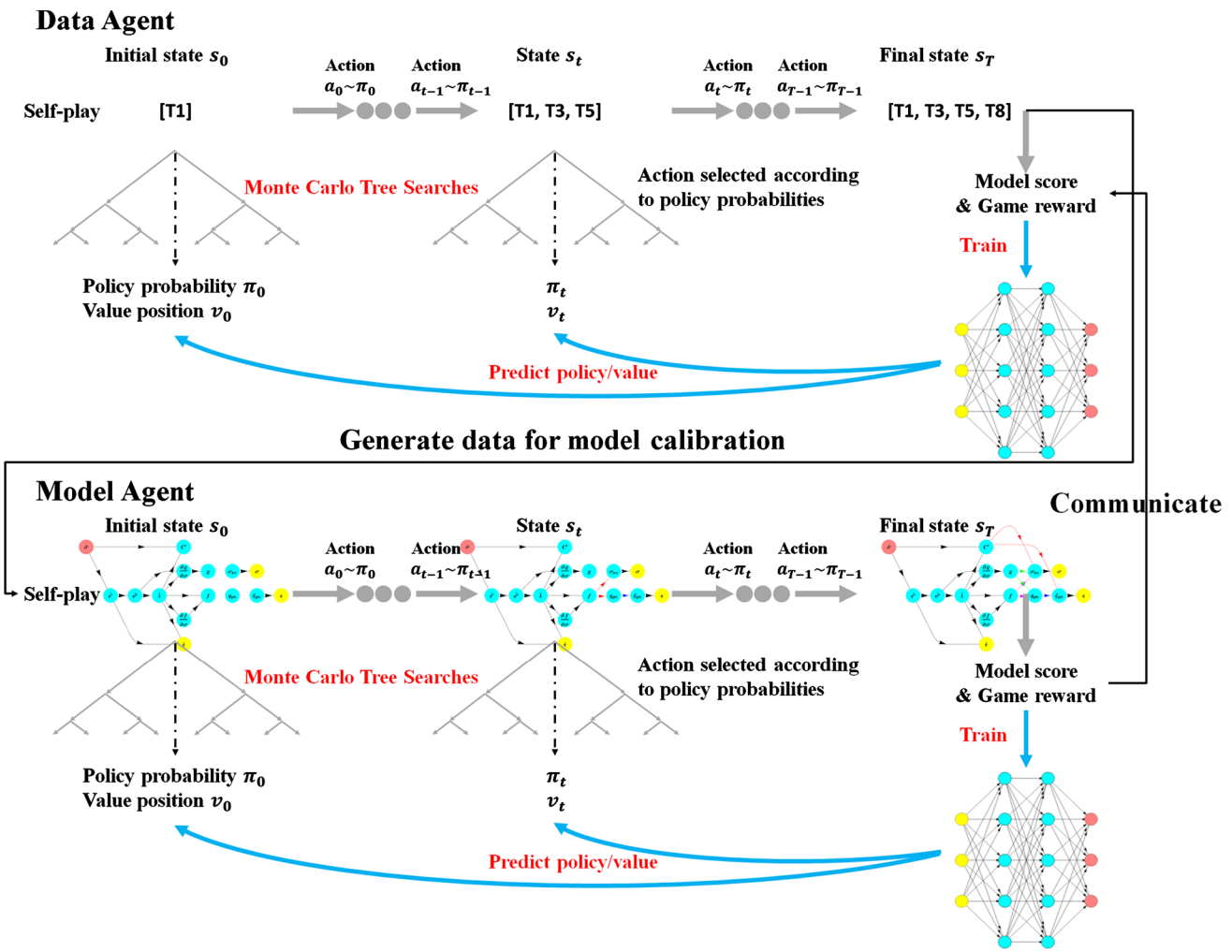


Fig. 3 Multi-player interactive deep reinforcement learning for generating optimal strategy to automate the modeling, calibration and validation of an elasto-plastic model. In this framework, two deep neural networks are used to make decisions as the mastermind modeler and data agents, while the modeler agents may also employ different strategies, including neural networks, mathematical expressions or other forms of mapping operators to complete a constitutive law

framework and show that the AI has the ability to react appropriately such that the correct interpretation (i.e. the model itself if the data is generated from that model) can be recovered from the data. In the second example, sub-scale discrete element simulations (DEM) are used to generate synthetic benchmark data for model calibrations and blind prediction evaluations to mimic data from real-world granular materials.

5.1 Numerical experiment 1: verification and testing the ability of AI to reverse engineering constitutive laws

The correctness of the implementation proposed meta-modeling framework is first verified through a series of tests on "virtual materials" having exact elasto-plastic constitutive behaviors.

This numerical experiment, as illustrated in Fig. 3.2, is introduced to (1) validate the implementation and (2) check whether the Monte Carlo tree search works when the modeling agent is given a sufficient amount of "perfect data". Note that fulfilling both requirements (1) and (2) is a necessary but not sufficient condition for the modeler agent to generate predictive models. On the other hand, if the modeling agent can recover the right model each time it fed with the corresponding model, then this ability can be leveraged to a new finite element library architecture in which material model library can be replaced by the model component library and a material model can be automatically generated and calibrated simultaneously by running the meta-modeling game. The test models for the AI to recover are, respectively,

Algorithm 3 Self-play reinforcement learning of the cooperative data collection/meta-modeling game

Require: The definitions of the cooperative data collection/meta-modeling game: game boards (Sects. 3.1.1 and 3.2.1), states (Sects. 3.1.2 and 3.2.2), actions (Sects. 3.1.3 and 3.2.3), game rewards (Sects. 3.1.5 and 3.2.5), game rules (Sects. 3.1.4 and 3.2.4).

Initialize the policy/value networks f_θ^{Data} and f_θ^{Model} for both data agent and model agent. For fresh learning, the networks are randomly initialized. For transfer learning, load pre-trained networks instead.
Initialize empty sets of the training examples for both data agent and model agent $trainExamples^{\text{Data}} \leftarrow []$, $trainExamples^{\text{Model}} \leftarrow []$.
for i in [0,…, $numIters$ (number of training iterations)-1] **do**
  for j in [1,…, $numEpisodes$ (number of game episodes)] **do**
    Initialize the starting game state s.
    Initialize empty tree of the Monte Carlo Tree search (MCTS), set the temperature parameter $\tau = 1$ for "exploration and exploitation".
    while True **do**
      Check for all legal actions at current state s according to the game rules.
      Get the action probabilities $\pi(s, \cdot)$ for all legal actions by performing $numMCTSSims$ times of MCTS simulations.
      Sample action a from the probabilities $\pi(s, \cdot)$
      Modify the current game state to a new state s by taking the action a.
      if s is the end state of a game **then**
        Evaluate the score of the constructed digraph.
        Evaluate the reward r of this game episode according to the model score.
        Break.
    Append the history in this game episode $[s, a, \pi(s, \cdot), r]$ to $trainExamples^{\text{Data}}$ and $trainExamples^{\text{Model}}$.
  Train the policy/value networks f_θ^{Data} and f_θ^{Model} with $trainExamples^{\text{Data}}$ and $trainExamples^{\text{Model}}$.
Use the final trained networks f_θ^{Data} and f_θ^{Model} in MCTS for one more iteration of "competitive gameplays" ($numEpisodes$ games) to generate the final converged digraph model.
Exit

1. J2 model with Von Mises yield function and an isotropic hardening with power law.
2. Drucker–Pager model with frictional hardening.
3. Three-invariant Matsuoka–Nakai model.

In the reverse engineering numerical experiments, we first implemented three implicit return mapping algorithms and calibrated them to generate a fixed set of material parameters. The experimentalist agent is then given the executable files of the return mapping algorithms of these models and run these executable files to generate artificial data. We intentionally withhold the information about the data, such as the material parameters and the corresponding graph representation of the constitutive laws, from the modeling agent to text whether it can recover such information. In each experimentalist's turn, an experimentalist is given the option of either generating an additional set of data listed below or terminating its involvement of the game (i.e. stop generating new data). If (1) the modeling agent is capable of learning the graph-based knowledge and (2) the artificial data are generated from sub-graphs of the labeled directed multi-graph, then the modeling agent must be able to recover all these subgraphs without any prior knowledge and obtain perfect scores, whenever given sufficient data. Failure to do so indicate the failure of the reinforcement learning process. This set of experiments is conducted to check whether the modeling agent can pass this necessary test. The choices of experimental tests available for the experimentalist agent are listed below:

T1: One-dimensional extension test ($\dot{\epsilon}_{11} > 0, \dot{\epsilon}_{22} = \dot{\epsilon}_{33} = \dot{\epsilon}_{12} = \dot{\epsilon}_{23} = \dot{\epsilon}_{13} = 0$, $p0 = -200kPa$)
T2: One-dimensional compression test ($\dot{\epsilon}_{11} < 0, \dot{\epsilon}_{22} = \dot{\epsilon}_{33} = \dot{\epsilon}_{12} = \dot{\epsilon}_{23} = \dot{\epsilon}_{13} = 0$, $p0 = -200kPa$).
T3: Drained triaxial extension test ($\dot{\epsilon}_{11} > 0, \dot{\sigma}_{22} = \dot{\sigma}_{33} = \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$, $p0 = -200kPa$).
T4: Drained triaxial compression test ($\dot{\epsilon}_{11} < 0, \dot{\sigma}_{22} = \dot{\sigma}_{33} = \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$, $p0 = -200kPa$).
T5: Undrained triaxial extension test ($\dot{\epsilon}_{11} > 0, \dot{\epsilon}_{11} + \dot{\epsilon}_{22} + \dot{\epsilon}_{33} = 0, \dot{\sigma}_{22} = \dot{\sigma}_{33}, \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$, $p0 = -200kPa$).
T6: Undrained triaxial compression test ($\dot{\epsilon}_{11} < 0, \dot{\epsilon}_{11} + \dot{\epsilon}_{22} + \dot{\epsilon}_{33} = 0, \dot{\sigma}_{22} = \dot{\sigma}_{33}, \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$, $p0 = -200kPa$).
T7: Simple shear test ($\dot{\epsilon}_{12} > 0, \dot{\sigma}_{11} = \dot{\sigma}_{22} = \dot{\epsilon}_{33} = \dot{\epsilon}_{23} = \dot{\epsilon}_{13} = 0$, $p0 = -200kPa$).

The modeling choices available for the modeler agent are specified in the *Game Choices* of the Sect. 3.2. The model score is defined as:

$$\text{SCORE} = 0.5 * A_{\text{accuracy}}^{\text{calibration}} + 0.5 * A_{\text{accuracy}}^{\text{prediction}}, \tag{56}$$

where $P\% = 80\%$ and $\varepsilon_{\text{crit}} = 1e^{-5}$ for Eq. (21) of accuracy evaluations. The DRL meta-modeling procedure (Algorithm 3) contains $numIters = 10$ training iterations of "exploration and exploitation" of game strategies, by setting the temperature parameter τ to 1. Then an iteration of "competitive gameplay" ($\tau = 0.01$) is conducted to showcase the performance of the final trained AI agent. Each iteration consists of $numEpisodes = 30$ self-play episodes of the game. Hence one execution of the entire DRL procedure contains $numIters * numEpisodes = 10 * 30 = 300$ game episodes for training the policy/value neural network. Each game starts with a randomly initialized neural network for the policy/value predictions, and each play step requires $numMCTSSims = 30$ MCTS simulations. Then the play steps and corresponding final game rewards are appended to the set of training examples for the training of the policy/value network.

Figures 4, 5 and 6 present the example model predictions and calibration tests during the DRL improvement of the

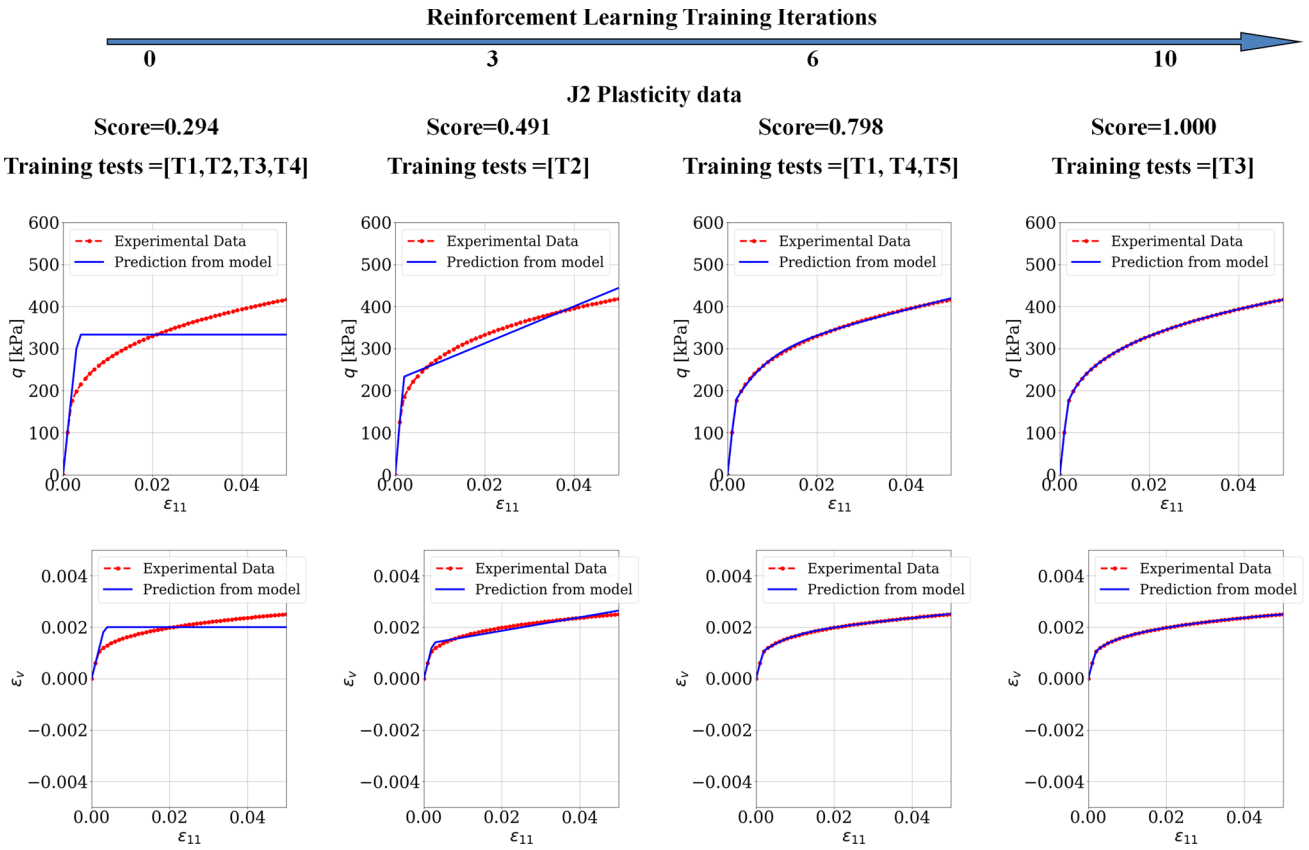


Fig. 4 Knowledge of elasto-plastic models learned by deep reinforcement learning in Numerical Experiment 1 using synthetic data from J2 plasticity. Four representative games played during the DRL iterations and their prediction accuracy against synthetic data are presented

experimentalist and modeler agents. Both agents try out different combinations of calibration data and model choices, and evaluate their model scores and individual game rewards. The agents learn from all the gameplay results that they have experienced and converge their individual strategies to the optimal ones that eventually generate the optimal set of experiment tests for model calibration and exactly recover the plasticity model used to generate the synthetic data. The "cooperative" convergence of the strategies of both agents is of crucial importance, since the calibration dataset and the selected model must be simultaneously optimal for the final model score to be maximum. Although the gameplays could be different in each separate run of the two-player DRL algorithm due to the randomness in initial Policy/Value networks and the action possibilities involved in Monte Carlo Tree Search, the optimal strategies are always recovered if the exploration is sufficient. This is confirmed in Figs. 7 and 8 in which the statistics of the gameplay scores of 20 separate executions of the two-player DRL algorithm are analyzed.

The fact that the two-player meta-modeling game is able to reach a perfect score in blind prediction indicates that it has successfully reverse engineered the constitutive law. The ability to automatically reverse-engineering a constitutive model could be of potential commercial value, as it allows one to understand attributes of legacy or proprietary software even when only the executable is available. Even in the case when reverse engineering fails to recover the constitutive responses perfectly, the score can indicate how close the DRL-generated model replicates the constitutive law in the legacy or proprietary codes.

Furthermore, the fact that the training is able to recover the model also enables us to use a different architecture for computational mechanics software in which the material model library does not necessarily contain multiple constitutive laws categorized by labels or model names. Instead, any new model in the literature that contains new "action" not available in the previous constitutive law can be decomposed into directed graphs and subsequently be merged with the existing pool of actions such that the modeler agent can have more tools to generate new models that optimize objective functions. Since (1) new actions will only be picked by the modeler agent if they can help it achieve a higher score, and (2) should this happens, the improvement in prediction quality is quantified by the increase of the score, the meta-modeling game can be used as a tool to evaluate the true

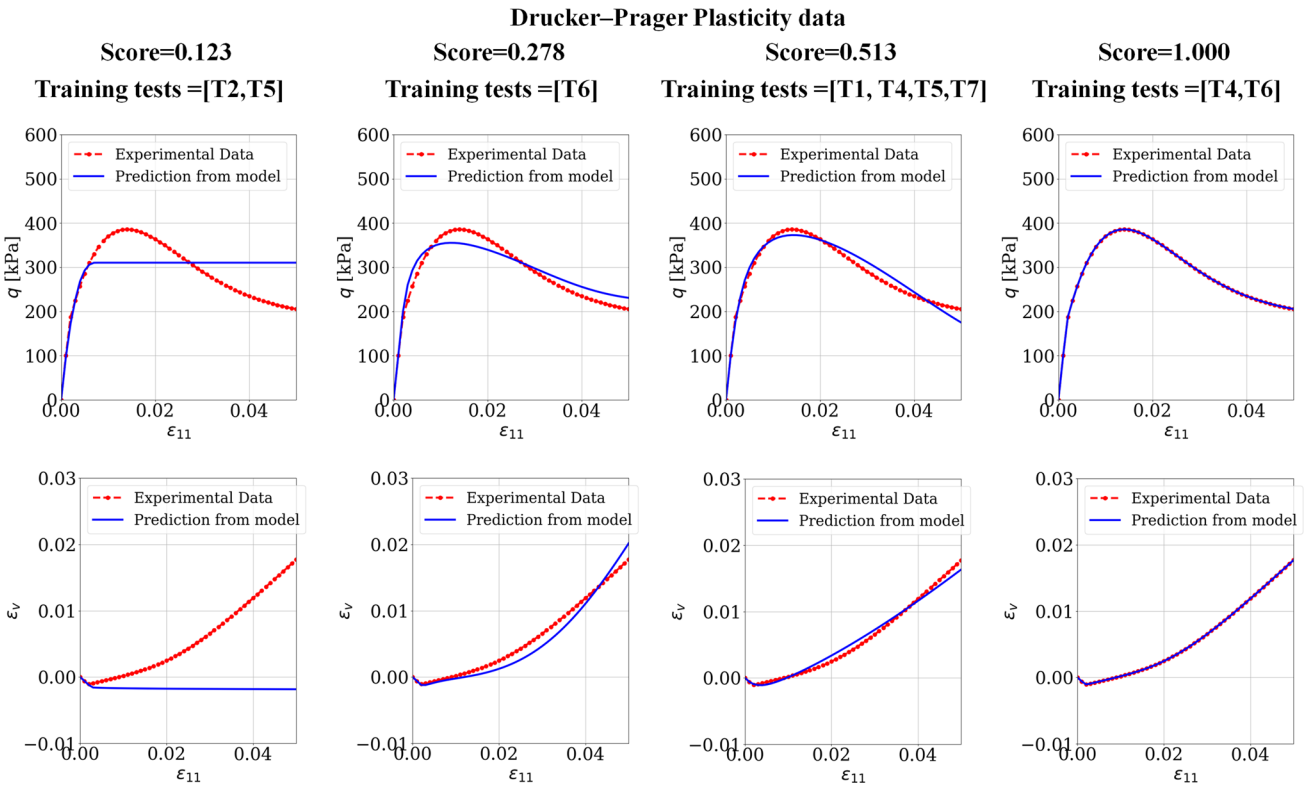


Fig. 5 Knowledge of elasto-plastic models learned by deep reinforcement learning in Numerical Experiment 1 using synthetic data from Drucker–Pager plasticity. Four representative games played during the DRL iterations and their prediction accuracy against synthetic data are presented

benefit of any new action that departs from the state-of-the-art.

5.2 Numerical experiments 2: testing the ability of AI for forward predictions

In this numerical experiment, we examine the ability of the proposed meta-modeling agents to (1) generate the knowledge and model represented by the directed graph from given data, (2) decide the set of experiments that aids data-driven discovery and (3) terminate the learning process when further experiments no longer benefit predictions.

5.2.1 Data generation from discrete element simulations

In this test, we consider an idealized situation in which the data is generated from discrete element simulations for granular materials [17,37,96]. While the constitutive responses from the discrete element simulations may contain fluctuation, we do not introduce any contaminated noise on purpose to test how the meta-modeling procedure might be affected by noise. While this could be addressed using dropout layers as shown in [94], a comprehensive study on learning with noisy data is out of the scope in this study but will be considered in the future.

The data for calibration and evaluation of prediction accuracy of the deep-reinforcement-learned constitutive models are generated by numerical simulations on a representative volume element (RVE) of densely-packed spherical DEM particles. The open-source discrete element simulation software YADE for DEM is used by the experimentalist agent to generate data, including the homogenized stress and strain measures and the geometrical and microstructural attributes such as coordination number, fabric tensor, porosity [76,79]. The discrete element particles in the RVE have radii between 1 ± 0.3 mm with a uniform distribution. The Cundall's elastic-frictional contact model ([17]) is used for the inter-particle constitutive law. The material parameters are: interparticle elastic modulus $E_{eq} = 0.5$ GPa, ratio between shear and normal stiffness $k_s/k_n = 0.3$, frictional angle $\varphi = 30°$, density $\rho = 2600\,kg/m^3$, Cundall damping coefficient $\alpha_{damp} = 0.6$.

The test data constitute of triaxial tests on DEM samples with different initial confining pressure and void ratio $\dot{\sigma}_{33} = \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0, b = \frac{\sigma_{22}-\sigma_{33}}{\sigma_{11}-\sigma_{33}}$.

T1: $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -300\,kPa, e_0 = 0.539.$
T2: $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -400\,kPa, e_0 = 0.536.$
T3: $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -500\,kPa, e_0 = 0.534.$
T4: $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -300\,kPa, e_0 = 0.539.$
T5: $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -400\,kPa, e_0 = 0.536.$

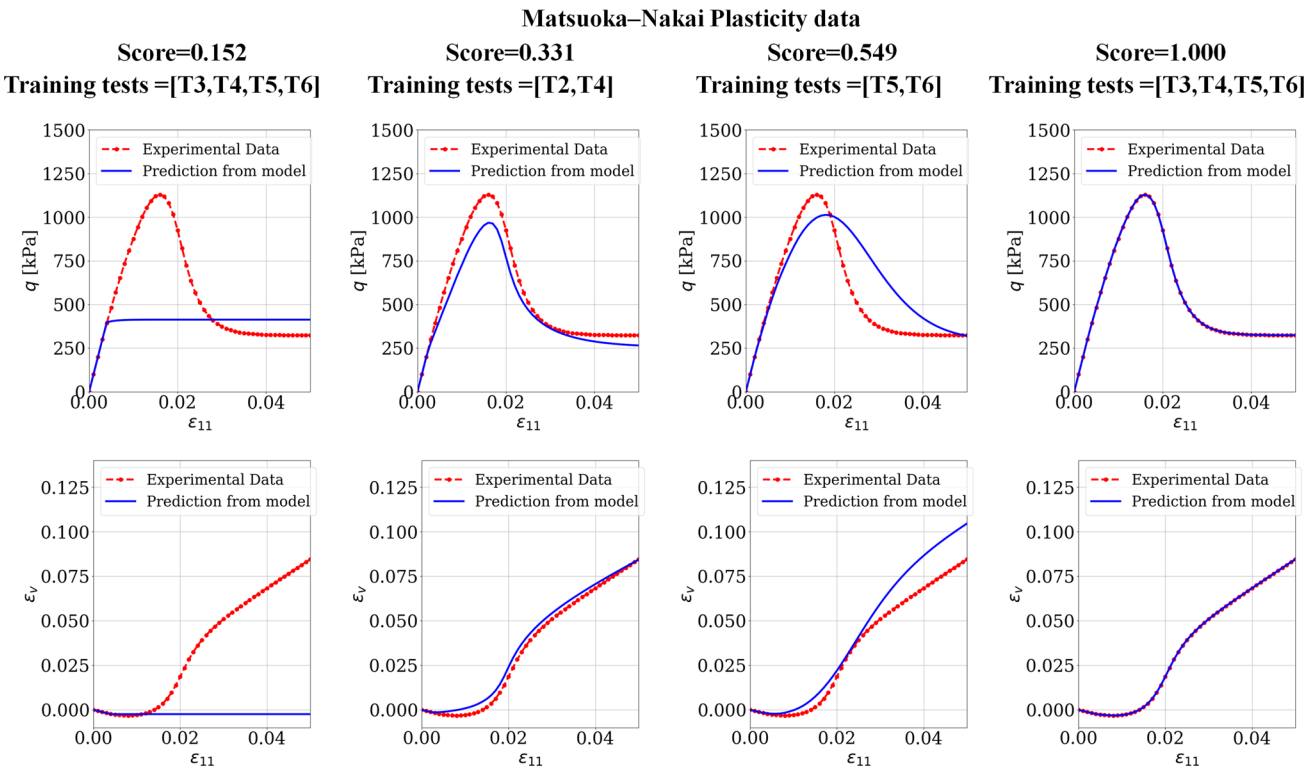


Fig. 6 Knowledge of elasto-plastic models learned by deep reinforcement learning in Numerical Experiment 1 using synthetic data from Matsuoka–Nakai plasticity. Four representative games played during the DRL iterations and their prediction accuracy against synthetic data are presented

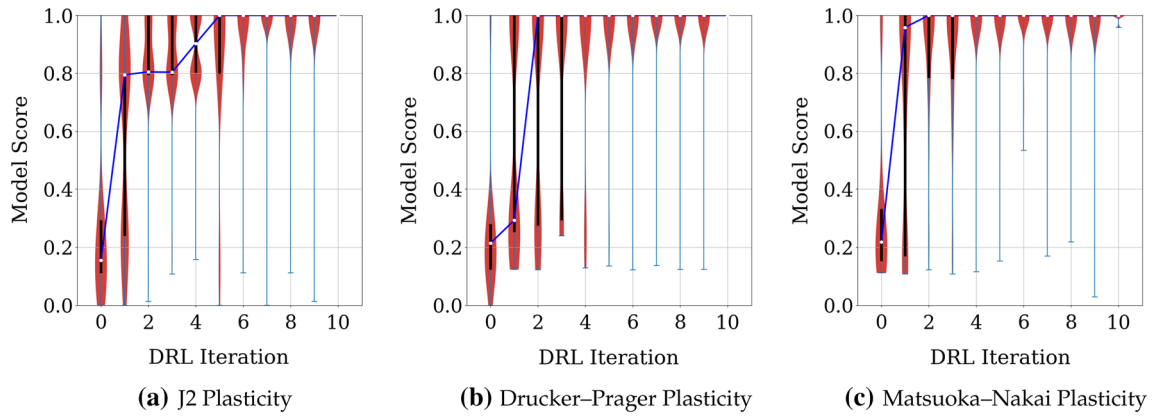
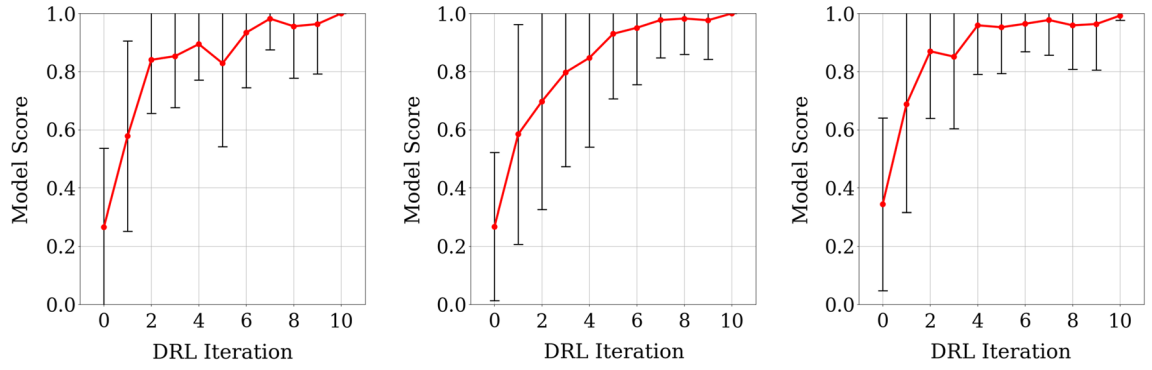


Fig. 7 Violin plots of the density distributions of model scores in each DRL iteration in Numerical Experiment 1. Statistics of the model scores in deep reinforcement learning iterations from 20 separate runs of the DRL procedure for Numerical Experiment 1. Each DRL procedure contains ten iterations 0–9 of "exploration and exploitation" (by setting the temperature parameter $\tau = 1.0$) and a final iteration 10 of "competitive gameplay" ($\tau = 0.01$). Each iteration consists of 30 games. The shaded area represents the density distribution of scores. The white point represents the median. The thick black bar represents the inter-quartile range between 25% quantile and 75% quantile. The maximum and minimum scores played in each iteration are marked by horizontal lines

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



(a) J2 Plasticity **(b)** Drucker–Prager Plasticity **(c)** Matsuoka–Nakai Plasticity

Fig. 8 Mean value (red dots) and ± standard deviation (error bars) of model score in each DRL iteration in Numerical Experiment 1. (Color figure online)

T16: $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -300\,kPa, e_0 = 0.539.$
T17: $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -400\,kPa, e_0 = 0.536.$
T18: $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -500\,kPa, e_0 = 0.534.$
T19: $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -300\,kPa, e_0 = 0.539.$
T20: $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -400\,kPa, e_0 = 0.536.$
T21: $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -500\,kPa, e_0 = 0.534.$
T22: $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -300\,kPa, e_0 = 0.539.$
T23: $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -400\,kPa, e_0 = 0.536.$
T24: $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -500\,kPa, e_0 = 0.534.$
T25: $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -300\,kPa, e_0 = 0.539.$
T26: $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -400\,kPa, e_0 = 0.536.$
T27: $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -500\,kPa, e_0 = 0.534.$
T28: $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -300\,kPa, e_0 = 0.539.$
T29: $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -400\,kPa, e_0 = 0.536.$
T30: $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -500\,kPa, e_0 = 0.534.$

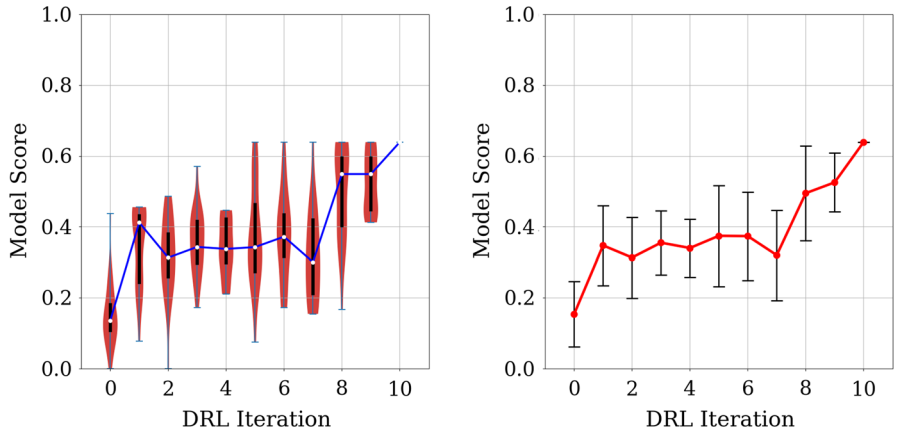
The candidate tests for the calibration data generation include $\mathbf{T}_c^0 = \{T1, T2, T3, \ldots, T11, T12\}$ and the validation tests are $\mathbf{T}_v^0 = \{T13, T14, T15, \ldots, T19, T30\}$. As explained in Sect. 3.1, the tests not selected in the final calibration set by the experimentalist agent will be moved to the final validation set to evaluate the blind prediction performance. The parameters for the DRL procedure are $numIters = 10, numEpisodes = 30, numMCTSSims = 300$.

5.2.2 Statistics of game scores via DRL iterations

The statistics of the gameplay results from 5 separate runs of the DRL procedure are presented in Fig. 9. We observe efficient improvements in the generated elasto-plastic models over the DRL training iterations with the discrete element simulation data.

Figure 10 presents the example model predictions and calibration tests during the DRL improvement of the experimentalist and modeler agents. The final converged calibration test set chosen by the AI experimentalist after the DRL procedure consists of the triaxial extension and compression tests with $b = 0$ and $b = 0.5$ under initial pressures of -300 kPa and -500 kPa. Accordingly, the final converged elasto-plastic model generated by the AI modeler after the DRL procedure is composed of the non-linear elasticity of Eq. (24), the loading direction defined as Eq. (35), the plastic flow direction defined as Eq. (40), and the hardening modulus defined as Eq. (42). The resultant model is a generalized plasticity model (without explicitly defined yield surface and plastic potential) combined with the critical state plasticity theory (dependence on the p, q, θ stress invariants and the void ratio e). Figure 11 presents five representative examples of blind predictions of this selected model and the selected calibration data. This optimal model for the given action space is generated from data obtained from 9 experiments in the following order: [T1, T3, T4, T5, T7, T9, T10, T11, T12].

One interesting aspect revealed in this numerical experiment is the potential of using the meta-modeling game as a tool to evaluate and analyze of relative policy values of the ingredients of constitutive laws in a prediction task. For instance, this numerical experiment reveals that the optimal configuration of the constitutive model for predicting the behavior of monotonic loading triaxial compression test should not contain any neural network edge (Eq. (44), (45) in Sect. 3.2) This could be attributed to the facts that the training data of the loading directions, plastic flow directions and hardening moduli from the DEM experimental data contain high-frequency fluctuations and that our testing data, which are used to evaluate the forward prediction performance, contain only monotonic stress paths. Since the high-frequency fluctuation makes the neural network easily to exhibit overfitting responses, and the relatively simple stress paths make it less advantageous to use a high-dimensional universal approximator like a neural network in any component of the



(a) Violin plots of the density distribution of model scores in each DRL iteration

(b) Mean value and ± standard deviation of model scores in each DRL iteration

Fig. 9 Statistics of the model scores in deep reinforcement learning iterations from 5 separate runs of the DRL procedure for Numerical Experiment 2. Each DRL procedure contains ten iterations 0–9 of "exploration and exploitation" (by setting the temperature parameter $\tau = 1.0$) and a final iteration 10 of "competitive gameplay" ($\tau = 0.01$). Each iteration consists of 30 gameplay episodes. **a** Violin Plot of model scores played in each DRL iteration. The shaded area represents the density distribution of scores. The white point represents the median. The thick black bar represents the interquartile range between 25% quantile and 75% quantile. The maximum and minimum scores played in each iteration are marked by horizontal lines. **b** Mean model score in each iteration and the error bars mark ± standard deviation. (Color figure online)

constitutive models, the edges that map input from the output vertices through mathematical expressions are revealed to have higher policy values as the game progresses and ultimately become the selected models.

Note that this result is in sharp contrast with the meta-modeling game results of the traction-separation law in which the neural network edges become dominant and yield consistently good forward predictions [94,95]. Comparing the choices the agents made in the two games reveal that the autonomous agents are capable of adjusting their decisions based on the availability of the data and the type of the forward prediction tasks. In other words, the agents are able to make judgments such that it employs edges that contain low-dimensional mathematical expression when the regularization (avoiding the curse of high dimensionality) is more critical than high-dimensionality afforded by the large numbers of neural network nodes (in this case), but also able to select the high-dimensional neural network options when the advantages of the options outweigh the drawbacks (in the traction-separation law game in [95]). Note that this optimal configuration sought by the meta-modeling game is sensitive to the available actions. For instance, the improvements of the neural network could be achieved by introducing techniques to filter out the noisy data and employing advanced neural networks with noise-resistant architecture [78]. These changes can impose adjustments in the policy values and therefore affect the optimal configuration. The incorporation of de-noising mechanisms and the investigation of the influence of data quality on the meta-modeling game framework will be conducted in the future study.

This **automated** strategy change by the AI agents is significant as it demonstrates that the agent system is able to adapt to the environment (availability of calibration data and the types of testing data) to make rational choices like a human modeler should when given different prediction tasks of different complexities.

5.2.3 Post-game performance analysis

Another important implication of the meta-modeling game is its ability to quantitatively analyze the performance of families of models currently (or historically, if possible to be inferred from reverse engineering) available in the literature for an intended prediction task. Table 1 shows the post-game analysis of the performance of the 112 models automatically generated from the two-player game. The resultant models are grouped into five different classes based on the types of the edges used in the game. The interesting aspect of the data in Table 1 is that it provides users a quantitative measure that configurations based on generalized plasticity and critical state outperform all the other 90 configurations. This result is consistent with the convention understanding from soil mechanics in which the classical critical state plasticity theory and the resultant plastic dilatancy/contraction predictions is regarded as the key ingredient for predictive constitutive laws. Examinations on models in Class 1 also reveals that three-invariant generalized plasticity with critical state perform the best in the blind predictions, especially

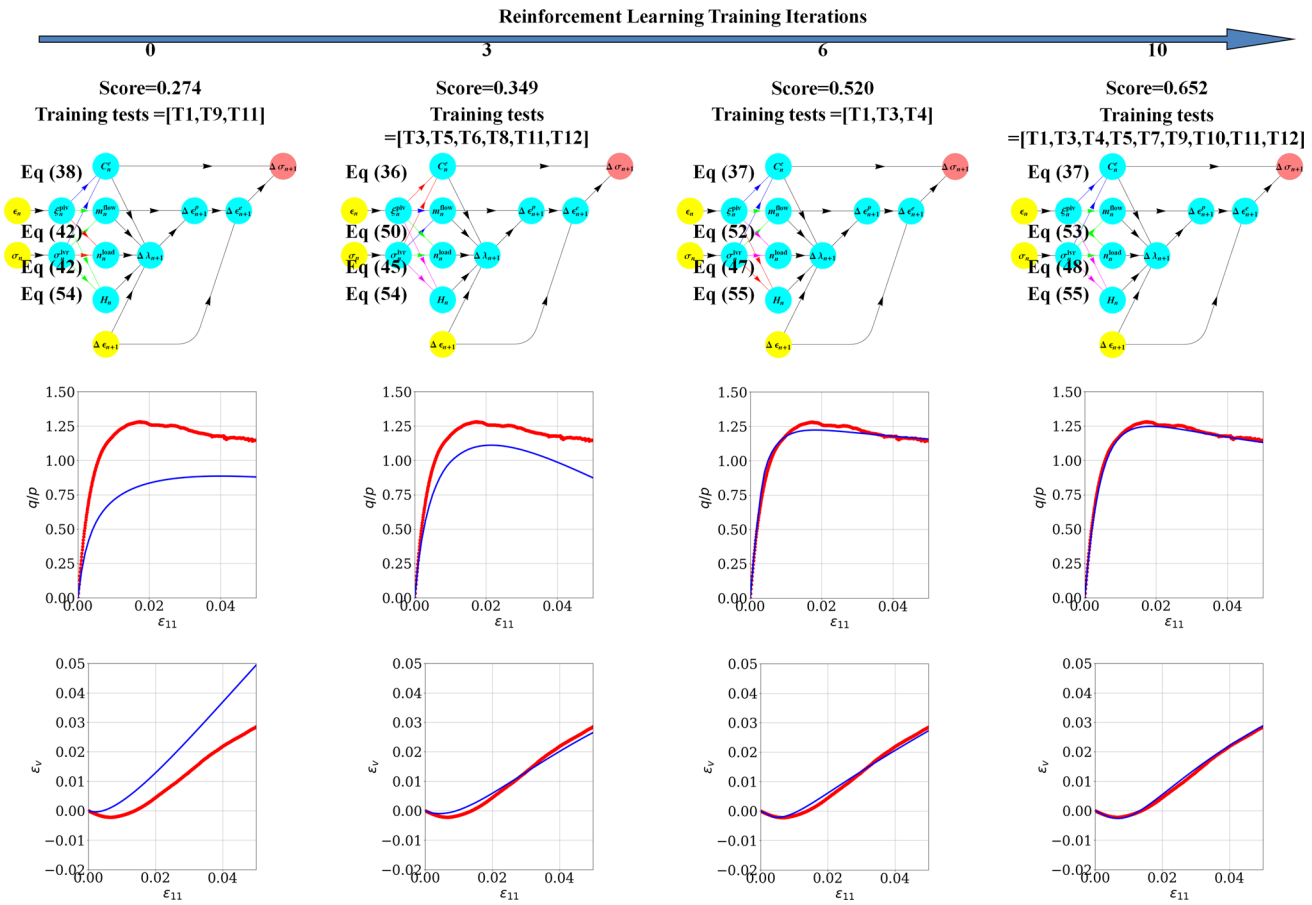


Fig. 10 Knowledge of elasto-plastic models learned by deep reinforcement learning in Numerical Experiment 2 using data from drained triaxial tests. Four representative games played during the DRL iterations and their prediction accuracy against synthetic data are presented. The color edges illustrate different labeled edges selected in the constitutive models. The labels are represented by equation numbers and these equations are detailed in *Game Choices* in Sect. 3.2

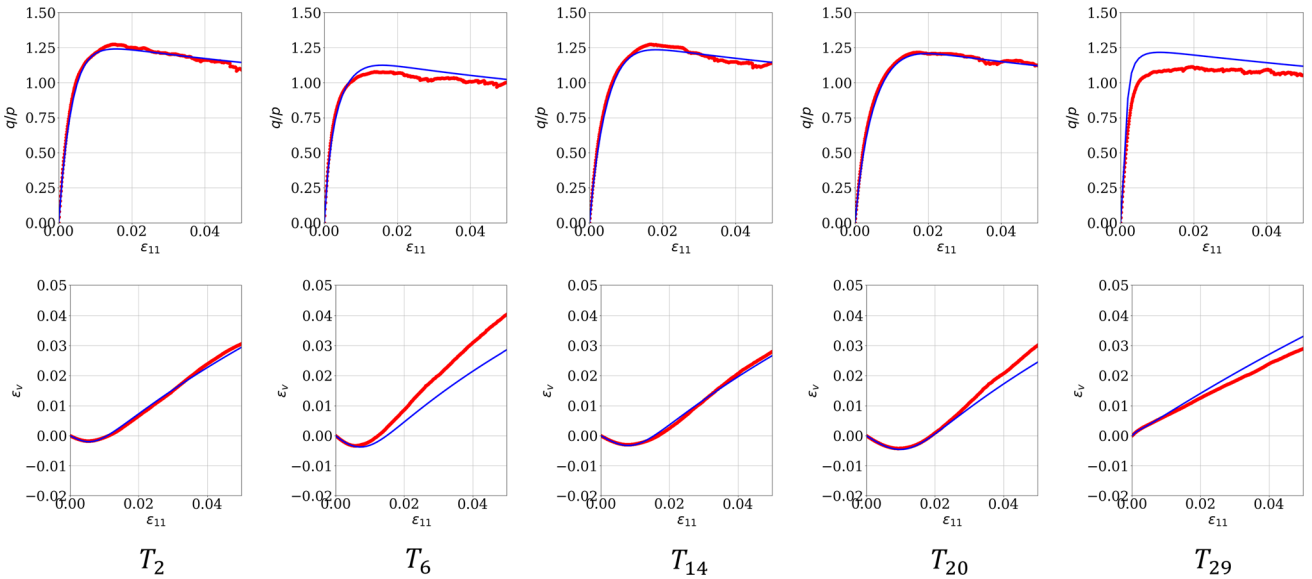


Fig. 11 Five examples of **blind** predictions from the optimal digraph configuration (The 4th digraph in Fig. 10) against data from the tests

Table 1 Five classes of the constitutive models generated during the deep reinforcement learning

Model class	Number of models	Mean score	Standard deviation	Generalized plasticity 'GP'	Critical state 'CS'	Classical pressure dependent elasto-plasticity 'DP'	Others 'O'
1	22	0.603	0.054	✓	✓		
2	25	0.565	0.051	✓			
3	13	0.295	0.028		✓	✓	
4	19	0.450	0.086			✓	
5	33	0.163	0.063				✓

when the material states of the granular materials in the calibration tests (e.g. confining pressure, initial void ratio, stress path) are significantly different than the ones in blind predictions.

However, comparisons of the results in Classes 1, 2,3 and 4 shown in Fig. 12 reveal a somewhat surprising conclusion in which the generalized plasticity seems to be consistently the more important ingredient than the critical state theory for yielding predictive models. Although it is important to stress that this conclusion must be interpreted with respect to the types and amount of data available and the intended prediction task, this result does provide further evidence to support the speculations that the generalized plasticity, if calibrated properly, does likely to improve the accuracy of blind predictions of the responses of granular materials in the monotonic triaxial compression tests [43,63,68,103].

In conclusion, this numerical experiment shows that the meta-modeling game can provide three important types of knowledge, the knowledge on the hierarchy of information flow, the estimation on the amount of data required to reach the state-of-the-art performance for a given action space and specified objective, and the relative values/benefits/importance of each model/theoretical/data-driven components revealed in the post-game analysis.

Remark 2 Note that applying the meta-modeling game to predicting responses of granular materials under different water drainage conditions may likely yield a very different conclusion where machine learning edges could be more widely used in the optimal configuration. This is because of the lack of a constitutive model that can quantitatively capture the constitutive responses of granular materials in drained and undrained conditions [23,51,64,80,104]. The creation of models for more generic purposes and the estimation of the trusted range of application are both important issues, which will be considered in future studies but are out of the scope of this paper.

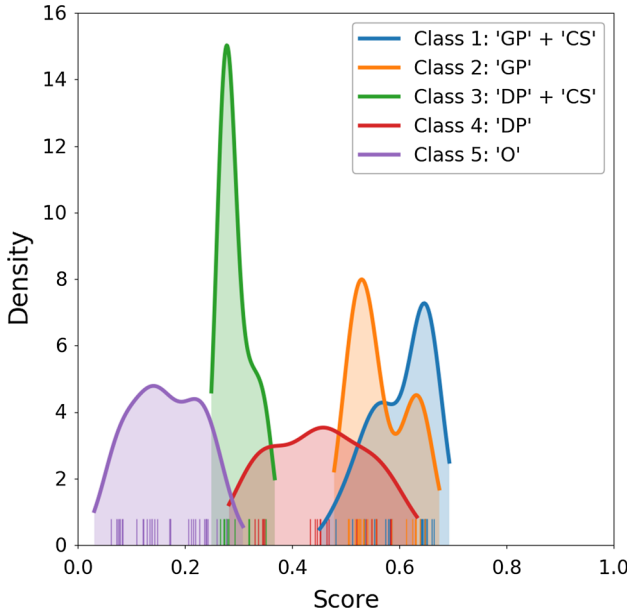


Fig. 12 Distribution of the scores of the models generated during the deep reinforcement learning. The models are grouped into five families (see Table 1). The curves present the Gaussian kernel density estimation of the model score distributions (The estimated function $f_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x-x_i}{h})$ for score data $(x_1, x_2, \ldots, x_n)$, $K(x)$ is the standard normal distribution function, h is the bandwidth parameter determined by Scott's rule $h = \frac{3.5\hat{\sigma}}{n^{1/3}}$, where $\hat{\sigma}$ is the standard deviation)

5.3 Numerical experiment 3: AI-generated material models in finite element simulations

To demonstrate the applicability of the AI-generated models from the plays of the data collection/meta-modeling game presented in Numerical Experiment 2, we conduct finite element simulations of a plane strain compression test on a rectangular specimen in which the AI-generated model is used to provide the incremental constitutive update. The numerical specimen is assumed to be in quasi-static condition and the Galerkin form of the balance of mass is solved incrementally with an implicit solver. The tangent of the residual is obtained via a perturbation method. The geometry, mesh

and the boundary conditions of the simulations are given in Fig. 13. The specimen is initially consolidated to isotropic pressure of $p_0 = -400\,\text{kPa}$, and have a uniform initial void ratio of 0.536. The specimen is compressed from the top surface, while the constant pressure p_0 are maintained on the lateral surfaces. Slight imperfection is introduced at the middle of the specimen to trigger heterogeneous deformation and shear bands. Three simulations are performed with the material properties given by the three example models generated during the DRL in Numerical Experiment 2 (1st, 3rd and 4th digraphs in Fig. 10).

The finite element implementation of the AI generated digraph-based model is simple and convenient. All modeling choices (Sect. 2) and the general purpose integration scheme (Algorithm 3) are already implemented in a single material model class. The FEM program is free to switch to other models simply by loading the digraphs and the corresponding calibrated parameters from the gameplay into this material class. The local distribution of the deviatoric strain ϵ_s and the volumetric strain ϵ_v in the specimen from the three models are compared in Figs. 14 and 15, respectively. The global differential stress - axial strain and volumetric strain - axial strain curves are compared in Fig. 16. The results demonstrate that all the local constitutive models, regardless of the quality, can all be implemented in the finite element solver. As mentioned previously this meta-modeling game can be easily incorporated in a new finite element solver architecture in which material library commonly used in the current paradigm is replaced by one single labeled directed multigraph and the conventional material identification process is replaced by the meta-modeling game such that both the optimal **combination of model components and material parameters** are simultaneously selected. Furthermore, the results also indicate that the qualities of the constitutive laws are continuously improved in each iteration of the meta-modeling game. In particular, we see that the correct type of shear band for dense granular assembles (dilatant shear band) is reproduced in the numerical specimens after 5th iterations (cf. [3,80]), and the shear band mode converges in the 8th iteration.

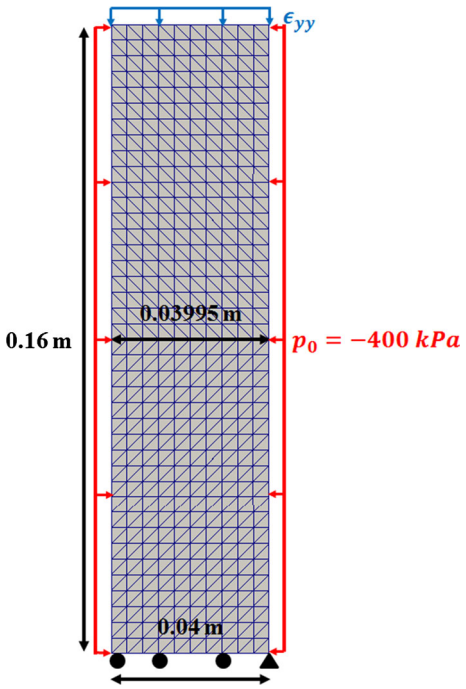
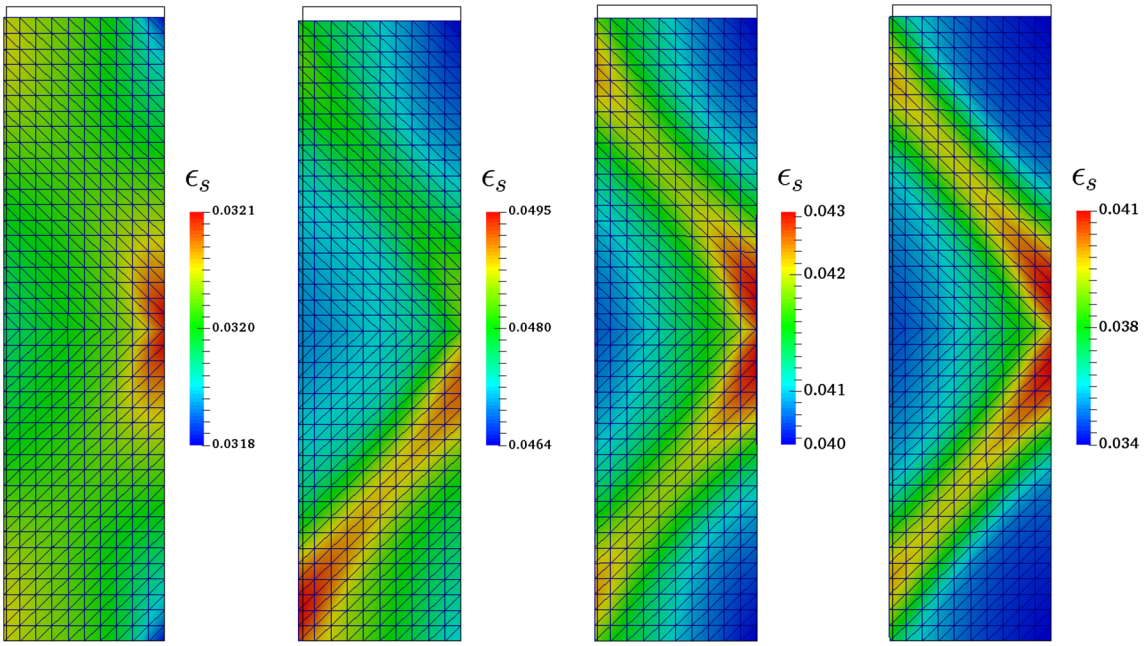


Fig. 13 Description of the geometry, mesh, boundary, and loading conditions of the plane strain compression problem

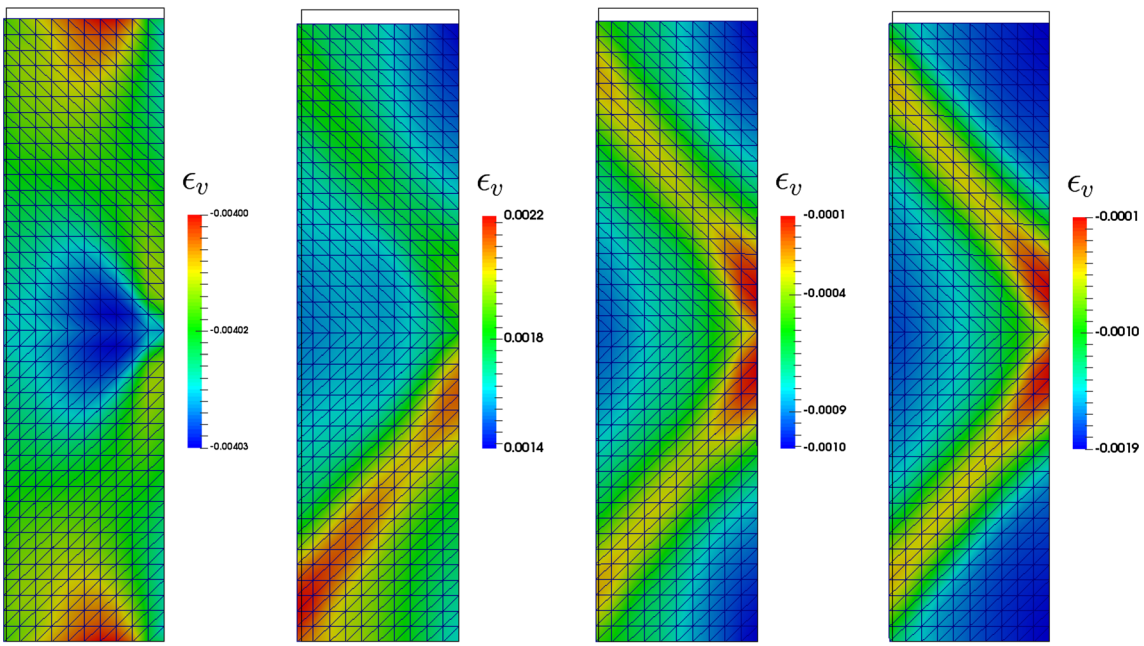
6 Efficiency compared to the brute force approach

Consider the case in which reinforcement learning is not used. Instead, one simply generates all the possible directed graph from the labeled directed graph. In this case, the data agent must generate all possible sequences of calibration tests from the 12 candidates in the calibration test set (T1, T2, T3, …, T11, T12), so there are $2^{12} - 1 = 4095$ possible (without considering the orders of tests) subsets of calibration test combinations. Note that the *order* of the tests provided to the modeler agent actually influences the Q value estimation of the modeler agent and hence this number is a lower bound of the total number of possible test sequences. Meanwhile, the modeler agent must select a model (directed graph) from the following possibilities (all specified in Sect. 3.2.6). In this small meta-modeling game, there are 3 choices for the elasticity law, 11 choices for loading direction definition, 8 choices of plastic flow direction, 4 choices of hardening law. The total number of model combinations is therefore 3*11*8*4 = 1056. Each subset of tests chosen by the data agent is used to calibrate a model chosen by the modeler agent, so the brutal force evaluations in total would be 4095*1056 = 4,324,320. In the DRL approach, there are 10 training iterations, each iteration has 30 gameplay episodes, and each gameplay episode has 300 MCTS (Monte Carlo Tree Search). So in the DRL there are in total 10*30*300 = 90,000 evaluations. The percentage of DRL evaluations versus brute force evaluations is 90,000/4,324,320 = 2.08% to obtain the optimal model score of 0.652. Whether the DRL-generated model is the ultimate optimal model among all the possibilities is unknown in this game (unless all the configurations have been tried out). However, it is possible to conduct benchmark experiments for a smaller game that has limited among of configurations.



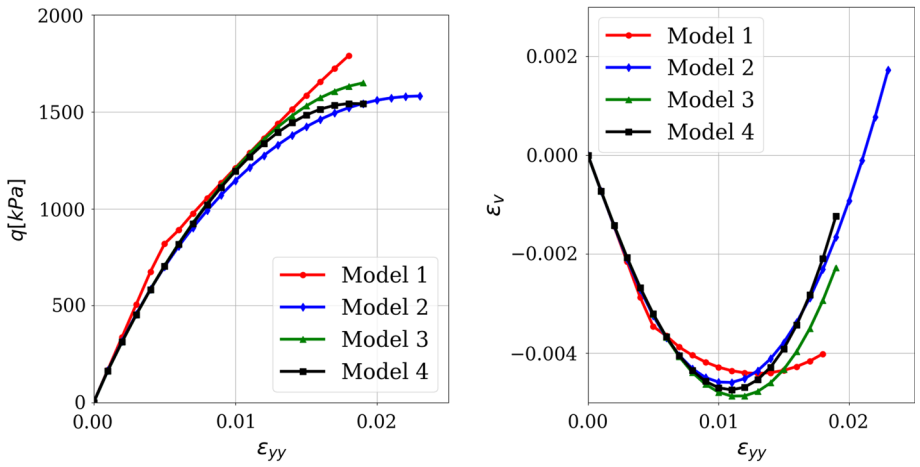
(a) Model 1 **(b)** Model 2 **(c)** Model 3 **(d)** Model 4

Fig. 14 Distribution of local deviatoric strain ϵ_s within the specimen at the end of the plane strain compression loadings. The finite element solutions using three models generated during the deep reinforcement learning of the meta-modeling game are compared (Model 1 is generated in the 1st DRL iteration in Fig. 10, Model 2 in the 5th iteration, Model 3 in the 8th iteration, Model 4 in the 10th iteration)



(a) Model 1 **(b)** Model 2 **(c)** Model 3 **(d)** Model 4

Fig. 15 Distribution of local volumetric strain ϵ_v within the specimen at the end of the plane strain compression loadings. The finite element solutions using three models generated during the deep reinforcement learning of the meta-modeling game are compared (Model 1 is generated in the 1st DRL iteration in Fig. 10, Model 2 in the 5th iteration, Model 3 in the 8th iteration, Model 4 in the 10th iteration)



(a) Global diffrential stress q **(b)** Global volumetric strain ϵ_v

Fig. 16 Evolution of the global differential stress ($q = |\sigma_{yy} - p_0|$, $p_0 = -400\,kPa$, and σ_{yy} is the stress on the top surface) and the global volumetric strain ϵ_v of the specimen with respect to the axial strain ϵ_{yy} computed using three models generated during the deep reinforcement learning of the meta-modeling game (Model 1 is generated in the 1st DRL iteration in Fig. 10, Model 2 in the 5th iteration, Model 3 in the 8th iteration, Model 4 in the 10th iteration). Each simulation terminates when the finite element solutions start to diverge

7 Conclusion and future perspectives

We introduce a new multi-agent meta-modeling game in which the experimental task, i.e. the generation of data, and the modeling task, the interpretation of data, are handled by two artificial intelligence agents. Mincing the collaboration of a pair of experimentalist and modeler collaborating to derive, implement, calibrate and validate a model to explain a path-dependent process, these two agents interact with each other sequentially and exchange information until either the model and data they reach the objectives or when further action does not generate a further reward. The major contribution of this research is as follows. First, we introduce the idea of using labeled directed multi-graph to mathematically represent the action space of the modeling agent. This action space can be expanded by adding plausible actions invented by previous human modelers or by generated new actions from deep neural networks or other machine learning methods. This invention therefore enables us to idealize the process of writing constitutive models as a continuous decision-making process in an action space of very high dimensions such that a pre-defined objective function can be maximized. As shown previously in work such as AlphaGo [71,72], using deep neural networks in a deep reinforcement learning framework to search proper actions from a very large number of possible moves is shown to achieve superior performance. To the best knowledge of the authors, this is the first time the ideas of using deep reinforcement learning applied on generating the knowledge graph and constitutive laws for history-dependent responses of materials.

The introductions of the graph, directed graph and labeled directed multigraph in the meta-modeling game enables us to recast the scientific process as a combinatorial optimization problem. Coupled with a reinforcement learning algorithm, the search for the optimal sequence of decision leads to a meta-modeling game closely resembles a more human-like iterative cyclical scientific process through which information is continuously gathered, hypotheses are continuously tested and the plausible understanding is continually revised. The major elements of scientific methods used by human, including characterization (observation and measurement stored in vertices, definition stored in edges), hypotheses (selection of a particular form of edges and edge sets), predictions (the information flow from root to leave of the directed graph obtained from the meta-modeling game) and experiments are all incorporated and automated. This new approach produces a forecast engine that can make predictions, but more importantly has the ability to generate human-interpretable knowledge on the relationships among different measurable physical quantities. This feature is significantly unique among other neural network approaches which often produce black-box models with no easy way to interpret the rationale of the predictions. It should be pointed out that models generated from the meta-modeling game do not discriminate the types of the edges used. They can be any operator that links the input to output, including but not limited to regression, support vector machine, neural network, mathematical expression or a bootstrapped version of them. These edges are only being formed by the AI when they are estimated to have higher policy value according to a specific objective function.

By introducing a gateway to merge existing and new models and introduce a seamless integration of data generation and data-driven discovery. Since the meta-modeling game stops when further action does not yield reward, this framework enables one to determine the best **configuration** of model one can possibly obtain within a well-defined action space for a given set of data. As shown in Sect. 5.2, this ability is not only important in making better predictions, but also help us identify the limitation of the action space. Given that modern constitutive laws have become increasingly complex and are often combined products of multiple material theo-

ries, concepts and assumptions created by different schools or theoretical backgrounds, the quantifiable policy values of the edges in the edge label set, if used properly, may enable us to pin down the relative values of each **component** of the constitutive laws while avoiding any potential implicit bias. As a result, the cooperative game enables us to make forward predictions while controlling the cost of generating the experimental data. Unlike other AI field which is largely driven by the exponential growth of available data, extracting an adequate amount of reliable experimental data remains a challenging task for the field of mechanics. The cooperative game designed in this paper does not only provide a tool to optimize the collaborations of the AI agents, but also shed lights on how to make productive scientific discovery through emulating the research progress in a setting where data generation can be costly.

In this work, we assume that the data obtained from experiments are perfect and without any significant noise. Furthermore, the meta-modeling game is also operated in a setting where the vertex set and the corresponding label are fixed. Future work will consider how to introduce quantifiable assurance of the meta-modeling game, incorporate sensitivity analysis in the validation and predictions, and quantify different types of uncertainties. For instance, one trains Bayesian neural network to generate edges that deliver not only deterministic predictions but also perform variational inference. By quantifying the sensitivity of the predictions, one may explore the weakness of the existing action space for both the modeler and experimentalist agents and use this knowledge to generate new actions. Work in this area is currently in progress.

Example 2 Game Action for traction-separation Laws. Given an 8-tuple $\mathbb{G} = (\mathbb{L}_{\mathbb{V}}, \mathbb{L}_{\mathbb{E}}, \mathbb{V}, \mathbb{E}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{n}_{\boldsymbol{V}}, \boldsymbol{n}_{\boldsymbol{E}})$ with elements defined in (66), (67), (72), (71). Find the subgraph $\mathbb{G}'$ of $\mathbb{G}$ such that this subgraph becomes the directed acyclic graph that maximizes the blind prediction accuracy defined by an objection function. □

Acknowledgements The work of KW and WCS is supported by the Earth Materials and Processes program from the US Army Research Office under Grant Contract W911NF-18-2-0306, the Dynamic Materials and Interactions Program from the Air Force Office of Scientific Research under Grant Contract FA9550-17-1-0169, the nuclear energy university program from Department of Energy under Grant Contract DE-NE0008534, the Mechanics of Material program at National Science Foundation under Grant Contract CMMI-1462760, and the Columbia SEAS Interdisciplinary Research Seed Grant. The work of QD is supported in part by NSF CCF-1704833, DMS-1719699, DMR-1534910, and ARO MURI W911NF-15-1-0562. These supports are gratefully acknowledged. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the sponsors, including the 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.

A Appendix: Traction-separation example

In this appendix, we provide an additional simple example for traction-separation law to (1) demonstrate how to selection components from a combinations of existing hand-crafted and machine-generated operators/mappings, (2) provide one possible example to form the the directed labeled multi-graph from existing mappings in the plasticity literature and (3) show the inclusive nature of the multi-graph approach by demonstrating how to merge multiple sub-graph into a multi-graph.

Example 1 Traction-separation Law. Given a pre-defined objective function, assume that the only known theoretical traction-separation model incorporated in the labeled directed multigraph are the Tvergaard model (cf. [90]) and the Ortiz–Pandolfi model (cf. [61]). In addition, we also consider using a neural network that incorporates porosity to predict traction-separation relations. Define the labeled directed multi-graph that provides all the options available.

First, we convert the traction-separation laws into directed graphs where the relative displacement vector is the input and the traction is the output. Notice that both Tvergaard [90] and Pandolfi et al. [61] are effective displacement models where an effective displacement $\overline{\Delta}$ is used as additional input to predict the traction. In [90],

$$T_n = \frac{\overline{T}(\overline{\Delta})}{\overline{\Delta}} \frac{\Delta_n}{\delta_n}, \tag{57}$$

$$T_t = \frac{\overline{T}(\overline{\Delta})}{\overline{\Delta}} \alpha \frac{\Delta_n}{\delta_t} \tag{58}$$

and the effective displacement and effective traction are scalars defined as,

$$\overline{\Delta} = \sqrt{(\Delta_n/\delta_n)^2 + (\Delta_t/\delta_t)^2}, \tag{59}$$

$$\overline{T}(\overline{\Delta}) = \frac{27}{4}\sigma_{\max}\overline{\Delta}(1 - 2\overline{\Delta} + \overline{\Delta}^2), \tag{60}$$

where δ_n and δ_t are the characteristic length corresponding to the fracture energy and cohesive strength of the normal and tangential opening modes, α is a non-dimensional material parameter. As pointed out in [62], the traction-separation model in [61] can be expressed in the forms of Eqs. (57) and (58) with the alternative definition of effective displacement and traction separation law, i.e.,

$$\overline{\Delta} = \tilde{\Delta}/\delta_n\ ,\ \tilde{\Delta} = \sqrt{\Delta_n^2 + \beta^2\Delta_t^2} \tag{61}$$

$$\overline{T}(\overline{\Delta}) = k\overline{\Delta} + c \tag{62}$$

where k is typically negative and c is the effective cohesive strength. Finally, we consider a neural network model

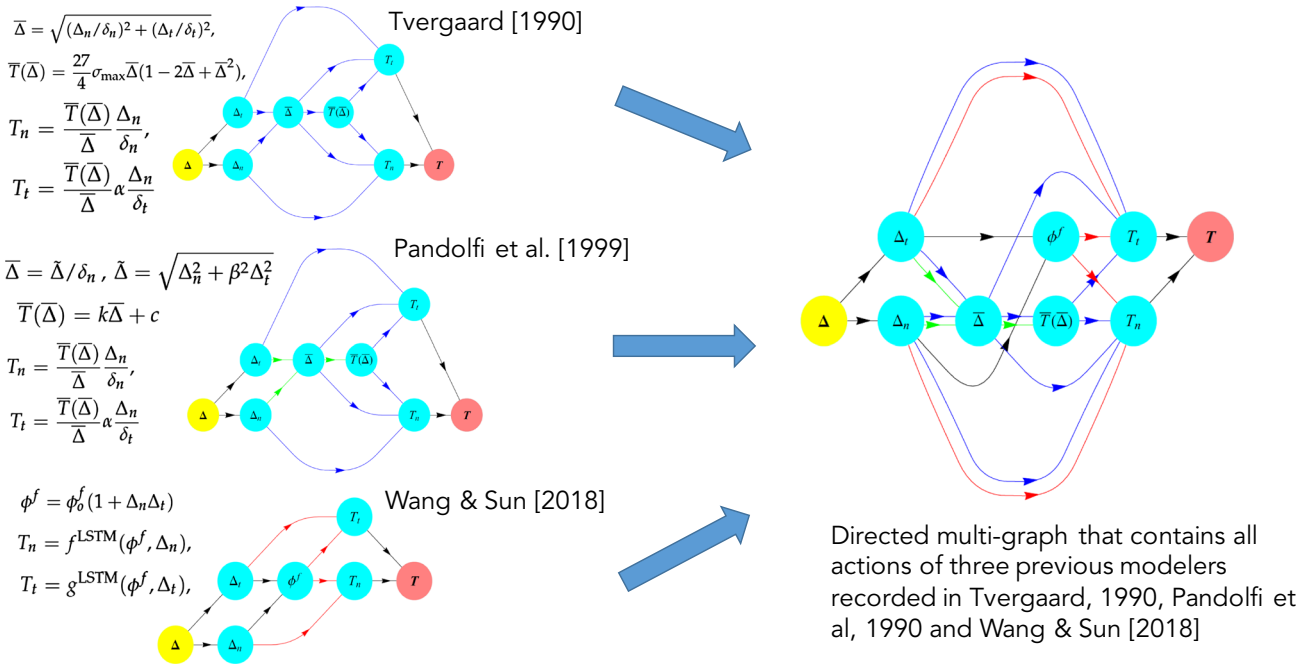


Fig. 17 The generation of directed multi-graph by expanding action space using previous models

in which the traction depends on the porosity ϕ^f [16,82,92], i.e.,

$$T_n = f^{\text{LSTM}}(\phi^f, \Delta_n), \tag{63}$$
$$T_t = g^{\text{LSTM}}(\phi^f, \Delta_t), \tag{64}$$

where the exact expression of the function f^{LSTM} and g^{LSTM} are determined by adjusting the weight of the neurons in the recurrent neural network [36,94]. Assuming that the solid constituent is incompressible, the porosity reads,

$$\phi^f = \phi_o^f (1 + \Delta_n \Delta_t) \tag{65}$$

The multi-graph that combines all the possible choices of the three traction separation laws can therefore be defined by multi-graph statement with the following sets,

$$\mathbb{V} = \{\Delta_n, \Delta_t, T_n, T_t, \overline{\Delta}, \overline{T}, \phi^f\} \tag{66}$$
$$\mathbb{E} = \mathbb{E}_1 \cup \mathbb{E}_2 \cup \mathbb{E}_3 \tag{67}$$
$$\mathbb{E}_1 = \{\Delta_n \to \overline{\Delta}, \Delta_t \to \overline{\Delta}, \Delta_n \to \phi^f, \Delta_t \to \phi^f, \Delta_n \to T_n, \Delta_t \to T_t\} \tag{68}$$
$$\mathbb{E}_2 = \{\overline{\Delta} \to \overline{T}, \phi^f \to T_n, \phi^f \to T_t, \Delta_n \to T_n\} \tag{69}$$
$$\mathbb{E}_3 = \{\overline{T} \to T_n, \overline{T} \to T_t\} \tag{70}$$
$$\mathbb{L}_{\mathbb{V}} = \{\text{normal disp., tan. disp., normal traction, tan., traction eff. disp., eff. traction, porosity}\} \tag{71}$$
$$\mathbb{L}_{\mathbb{E}} = \{\text{Eq.(57), Eq.(58), Eq.(59), Eq.(60), Eq.(61), Eq.(62), Eq.(63), Eq.(64), Eq.(65)}\} \tag{72}$$

Since $\boldsymbol{n}_{\mathbb{V}}$ is a bijective mapping, the labeling of the vertices is trivial. The rest of the mappings, i.e. $\boldsymbol{s}$, $\boldsymbol{t}$ and $\boldsymbol{n}_{\mathbb{E}}$ can be visualized in a labeled directed multigraph as shown in Fig. 17. Essentially, the process of creating the directed multigraph is to mathematically represent all the possible options modelers can have when they are tasked to create a constitutive model for a data set. □

References

1. Andrade JE, Borja RI (2006) Capturing strain localization in dense sands with random density. Int J Numer Methods Eng 67(11):1531–1564
2. Asaro RJ (1983) Crystal plasticity. J Appl Mech 50(4b):921–934
3. Aydin A, Borja RI, Eichhubl P (2006) Geological and mathematical framework for failure modes in granular rock. J Struct Geol 28(1):83–98
4. Bang-Jensen J, Gutin GZ (2008) Digraphs: theory, algorithms and applications. Springer, Berlin
5. Bardet JP, Choucair W (1991) A linearized integration technique for incremental constitutive equations. Int J Numer Anal Methods Geomech 15(1):1–19
6. Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, Tacchetti A, Raposo D, Santoro A, Faulkner R et al (2018) Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261
7. Been K, Jefferies MG, Hachey J (1991) Critical state of sands. Geotechnique 41(3):365–381
8. Bonabeau E (2002) Agent-based modeling: methods and techniques for simulating human systems. Proc Natl Acad Sci 99(suppl 3):7280–7287
9. Borja RI (2013) Plasticity: modeling and computation. Springer, Berlin

10. Borja RI, Sama KM, Sanz PF (2003) On the numerical integration of three-invariant elastoplastic constitutive models. Comput Methods Appl Mech Eng 192(9–10):1227–1258
11. Boyce BL, Kramer SLB, Fang HE, Cordova TE, Neilsen MK, Dion K, Kaczmarowski AK, Karasz E, Xue L, Gross AJ (2014) The sandia fracture challenge: blind round robin predictions of ductile tearing. Int J Fract 186(1–2):5–68
12. Casagrande A (1976) Liquefaction and cyclic deformation of sands—a critical review. Harvard soil mechanics series (88). Harvard University, Cambridge
13. Chomsky N (2014) Aspects of the theory of syntax, vol 11. MIT press
14. Choo J, Sun WC (2018a) Coupled phase-field and plasticity modeling of geological materials: from brittle fracture to ductile flow. Comput Methods Appl Mech Eng 330:1–32
15. Choo J, Sun WC (2018b) Cracking and damage from crystallization in pores: coupled chemo-hydro-mechanics and phase-field modeling. Comput Methods Appl Mech Eng 335:347–379
16. Coussy O (2004) Poromechanics. Wiley, New York
17. Cundall PA, Strack ODL (1979) A discrete numerical model for granular assemblies. Geotechnique 29(1):47–65
18. Dabrowski R, Stencel K, Timoszuk G (2011) Software is a directed multigraph. In: European conference on software architecture. Springer, pp 360–369
19. Dafalias YF, Manzari MT (2004) Simple plasticity sand model accounting for fabric change effects. J Eng Mech 130(6):622–634
20. de Borst R, Heeres OM (2002) A unified approach to the implicit integration of standard, non-standard and viscous plasticity models. Int J Numer Anal Methods Geomech 26(11):1059–1070
21. Foerster J, Assael IA, de Freitas N, Whiteson S (2016) Learning to communicate with deep multi-agent reinforcement learning. In: 30th conference on neural information processing systems (NIPS 2016). Advances in neural information processing systems, Barcelona, Spain, pp 2137–2145
22. Furukawa T, Yagawa G (1998) Implicit constitutive modelling for viscoplasticity using neural networks. Int J Numer Methods Eng 43(2):195–219
23. Gens A, Potts DM (1988) Critical state models in computational geomechanics. Eng Comput 5(3):178–197
24. Ghaboussi J, Garrett JH Jr, Wu X (1991) Knowledge-based modeling of material behavior with neural networks. J Eng Mech 117(1):132–153
25. Ghaboussi J, Pecknold DA, Zhang M, Haj-Ali RM (1998) Autoprogressive training of neural network constitutive models. Int J Numer Methods Eng 42(1):105–126
26. Ghavamzadeh M, Mannor S, Pineau J, Tamar A (2015) Bayesian reinforcement learning: a survey. Found Trends® Mach Learn 8(5–6):359–483
27. Graham RL, Knuth DE, Patashnik O, Liu S (1989) Concrete mathematics: a foundation for computer science. Comput Phys 3(5):106–107
28. Hibbitt, Karlsson, Sorensen (2001) ABAQUS/standard user's manual, vol 1. Hibbitt, Karlsson & Sorensen, Pawtucket
29. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366
30. Humboldt W (1999) On language: on the diversity of human language construction and its influence on the mental development of the human species. Cambridge University Press
31. Ibañez R, Abisset-Chavanne E, Aguado JV, Gonzalez D, Cueto E, Chinesta F (2018) A manifold learning approach to data-driven computational elasticity and inelasticity. Arch Comput Methods Eng 25(1):47–57
32. Jefferies MG (1993) Nor-sand: a simle critical state model for sand. Géotechnique 43(1):91–103
33. Kendall MG et al (1946) The advanced theory of statistics. , 5th edn. Charles Griffin & Company, London
34. Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. Comput Methods Appl Mech Eng 304:81–101
35. Kirchdoerfer T, Ortiz M (2017) Data driven computing with noisy material data sets. Comput Methods Appl Mech Eng 326:622–641
36. Koeppe A, Bamer F, Padilla CAH, Markert B (2017) Neural network representation of a phase-field model for brittle fracture. PAMM 17(1):253–254
37. Kuhn MR, Sun WC, Wang Q (2015) Stress-induced anisotropy in granular materials: fabric, stiffness, and permeability. Acta Geotech 10(4):399–419
38. Lake Brenden M, Ullman Tomer D, Tenenbaum Joshua B, Gershman Samuel J (2017) Building machines that learn and think like people. Behav Brain Sci 40:e253
39. Lange M (2012) What makes a scientific explanation distinctively mathematical? Br J Philos Sci 64(3):485–511
40. Lefik M, Schrefler BA (2002) Artificial neural network for parameter identifications for an elasto-plastic model of superconducting cable under cyclic loading. Comput Struct 80(22):1699–1713
41. Lefik M, Schrefler BA (2003) 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
42. Li XS, Dafalias YF (2011) Anisotropic critical state theory: role of fabric. J Eng Mech 138(3):263–275
43. Ling HI, Liu H (2003) Pressure-level dependency and densification behavior of sand through generalized plasticity model. J Eng Mech 129(8):851–860
44. Ling HI, Yang S (2006) Unified sand model based on the critical state and generalized plasticity. J Eng Mech 132(12):1380–1391
45. Liu Y, Sun WC, Fish J (2016) Determining material parameters for critical state plasticity models based on multilevel extended digital database. J Appl Mech 83(1):011003
46. Liu Y, Sun WC, Yuan Z, Fish J (2016) A nonlocal multiscale discrete-continuum model for predicting mechanical behavior of granular materials. Int J Numer Methods Eng 106(2):129–160
47. Liu Z, Fleming M, Liu WK (2018) Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials. Comput Methods Appl Mech Eng 330:547–577
48. Lubliner J, Auricchio F (1996) Generalized plasticity and shape-memory alloys. Int J Solids Struct 33(7):991–1003
49. Malcher L, Pires FMA, de Sá JMAC, Andrade FXC (2009) Numerical integration algorithm of a new model for metal plasticity and fracture including pressure and lode angle dependence. Int J Mater Form 2(1):443–446
50. Malmgren RD, Ottino JM, Amaral LAN (2010) The role of mentorship in protégé performance. Nature 465(7298):622
51. Manzari MT, Dafalias YF (1997) A critical state two-surface plasticity model for sands. Geotechnique 47(2):255–272
52. Miehe C, Schröder J (2001) A comparative study of stress update algorithms for rate-independent and rate-dependent crystal plasticity. Int J Numer Methods Eng 50(2):273–298
53. Mira P, Tonni L, Pastor M, Merodo JAF (2009) A generalized midpoint algorithm for the integration of a generalized plasticity model for sands. Int J Numer Methods Eng 77(9):1201–1223
54. Mooney MA, Finno RJ, Viggiani MG (1998) A unique critical state for sand? J Geotech Geoenviron Eng 124(11):1100–1108
55. Munafò MR, Nosek BA, Bishop DVM, Button KS, Chambers CD, du Sert NP, Simonsohn U, Wagenmakers E-J, Ware JJ, Ioannidis JPA (2017) A manifesto for reproducible science. Nat Hum Behav 1:0021
56. Na SH, Sun WC (2017) Computational thermo-hydro-mechanics for multiphase freezing and thawing porous media in the finite deformation range. Comput Methods Appl Mech Eng 318:667–700

57. Na SH, Sun WC (2018) Computational thermomechanics of crystalline rock, part I: a combined multi-phase-field/crystal plasticity approach for single crystal simulations. Comput Methods Appl Mech Eng 338:657–691
58. Olivier A, Smyth AW (2018) A marginalized unscented kalman filter for efficient parameter estimation with applications to finite element models. Comput Methods Appl Mech Eng 339:615–643
59. Ortiz M, Pandolfi A (1999) Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. Int J Numer Methods Eng 44(9):1267–1282
60. Pack K, Luo M, Wierzbicki T (2014) Sandia fracture challenge: blind prediction and full calibration to enhance fracture predictability. Int J Fract 186(1–2):155–175
61. Pandolfi ANNA, Krysl P, Ortiz M (1999) Finite element simulation of ring expansion and fragmentation: the capturing of length and time scales through cohesive models of fracture. Int J Fract 95(1–4):279–297
62. Park K, Paulino GH (2011) Cohesive zone models: a critical review of traction-separation relationships across fracture surfaces. Appl Mech Rev 64(6):060802
63. Pastor M, Zienkiewicz OC, Chan AHC (1990) Generalized plasticity and the modelling of soil behaviour. Int J Numer Anal Methods Geomech 14(3):151–190
64. Pestana JM, Whittle AJ, Salvati LA (2002) Evaluation of a constitutive model for clays and sands: part I—sand behaviour. Int J Numer Anal Methods Geomech 26(11):1097–1121
65. Raileanu R, Denton E, Szlam A, Fergus R (2018) Modeling others using oneself in multi-agent reinforcement learning. arXiv preprint arXiv:1802.09640
66. Rutqvist J, Ijiri Y, Yamamoto H (2011) Implementation of the barcelona basic model into tough-flac for simulations of the geomechanical behavior of unsaturated soils. Comput Geosci 37(6):751–762
67. Salinger AG, Bartlett RA, Bradley AM, Chen Q, Demeshko IP, Gao X, Hansen GA, Mota A, Muller RP, Nielsen E et al (2016) Albany: using component-based design to develop a flexible, generic multiphysics analysis code. Int J Multiscale Comput Eng 14(4):415–438
68. Sánchez M, Gens A, Guimarães LN, Olivella S (2005) A double structure generalized plasticity model for expansive materials. Int J Numer Anal Methods Geomech 29(8):751–787
69. Schofield A, Wroth P (1968) Critical state soil mechanics, vol 310. McGraw-Hill, London
70. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T et al (2017) Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815
71. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T et al (2017) Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815
72. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A et al (2017c) Mastering the game of go without human knowledge. Nature 550(7676):354
73. Simo JC, Hughes TJR (2006) Computational inelasticity, vol 7. Springer, Berlin
74. Sloan SW (1987) Substepping schemes for the numerical integration of elastoplastic stress–strain relations. Int J Numer Methods Eng 24(5):893–911
75. Sloan SW, Abbo AJ, Sheng D (2001) Refined explicit integration of elastoplastic models with automatic error control. Eng Comput 18(1/2):121–194
76. Šmilauer V, Catalano E, Chareyre B, Dorofeenko S, Duriez J, Gladky A, Kozicki J, Modenese C, Scholtès L, Sibille L et al (2010) Yade reference documentation. Yade Doc 474(1):1–161
77. Smith J, Xiong W, Yan W, Lin S, Cheng P, Kafka OL, Wagner GJ, Cao J, Liu WK (2016) Linking process, structure, property, and performance for metal-based additive manufacturing: computational approaches with experimental support. Comput Mech 57(4):583–610
78. Sun Q, Tao Y, Du Q (2018) Stochastic training of residual networks: a differential equation viewpoint. arXiv preprint-arXiv:1812.00174
79. Sun WC, Kuhn MR, Rudnicki JW et al (2014) A micromechanical analysis on permeability evolutions of a dilatant shear band. In: *48th US rock mechanics/geomechanics symposium*. American Rock Mechanics Association
80. Sun WC (2013) A unified method to predict diffuse and localized instabilities in sands. Geomech Geoeng 8(2):65–75
81. Sun WC (2015) A stabilized finite element formulation for monolithic thermo-hydro-mechanical simulations at finite strain. Int J Numer Methods Eng 103(11):798–839
82. Sun WC, Kuhn MR, Rudnicki JW (2013) A multiscale DEM-LBM analysis on permeability evolutions inside a dilatant shear band. Acta Geotech 8(5):465–480
83. Sun WC, Ostien JT, Salinger AG (2013) A stabilized assumed deformation gradient finite element formulation for strongly coupled poromechanical simulations at finite strain. Int J Numer Anal Methods Geomech 37(16):2755–2788
84. Tampuu A, Matiisen T, Kodelja D, Kuzovkin I, Korjus K, Aru J, Aru J, Vicente R (2017) Multiagent cooperation and competition with deep reinforcement learning. PLoS ONE 12(4):e0172395
85. Tan M (1993) Multi-agent reinforcement learning: independent vs. cooperative agents. In: Proceedings of the tenth international conference on machine learning, pp 330–337
86. Tang S, Zhang L, Liu WK (2018) From virtual clustering analysis to self-consistent clustering analysis: a mathematical study. Comput Mech 62(6):1443–1460
87. Truesdell C (1959) The rational mechanics of materials—past, present, future. Appl Mech Rev 12:75–80
88. Truesdell C, Noll W (2004) The non-linear field theories of mechanics. Springer, Berlin, Heidelberg, pp 1–579
89. Tu X, Andrade JE, Chen Q (2009) Return mapping for nonsmooth and multiscale elastoplasticity. Comput Methods Appl Mech Eng 198(30–32):2286–2296
90. Tvergaard V (1990) Effect of fibre debonding in a whisker-reinforced metal. Mater Sci Eng A 125(2):203–213
91. Ulven OI, Sun WC (2018) Capturing the two-way hydromechanical coupling effect on fluid-driven fracture in a dual-graph lattice beam model. Int J Numer Anal Methods Geomech 42(5):736–767
92. Wang K, Sun WC (2016) A semi-implicit discrete-continuum coupling method for porous media based on the effective stress principle at finite strain. Comput Methods Appl Mech Eng 304:546–583
93. Wang K, Sun WC (2017) Data-driven discrete-continuum method for partially saturated micro-polar porous media. In: *Poromechanics VI*, pp 571–578
94. Wang K, Sun WC (2018) A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. Comput Methods Appl Mech Eng 334:337–380
95. Wang K, Sun WC (2019) Meta-modeling game for deriving theory-consistent, microstructure-based traction-separation laws via deep reinforcement learning. Comput Methods Appl Mech Eng 346:216–241
96. Wang K, Sun WC (2019) 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
97. Wang K, Sun W, Salager S, Na S, Khaddour G (2016) Identifying material parameters for a micro-polar plasticity model via

X-ray micro-CT images: lessons learned from the curve-fitting exercises. Int J Multiscale Comput Eng 14(4):389–413
98. West DB et al (2001) Introduction to graph theory, vol 2. Prentice Hall, Upper Saddle River
99. Wollny I, Sun WC, Kaliske M (2017) A hierarchical sequential ale poromechanics model for tire–soil–water interaction on fluid-infiltrated roads. Int J Numer Methods Eng 112(8):909–938
100. Wood DM (1990) Soil behaviour and critical state soil mechanics. Cambridge University Press, Cambridge
101. Xin H, Sun WC, Fish J (2017) Discrete element simulations of powder-bed sintering-based additive manufacturing. Int J Mech Sci 149:373–392
102. Zhao J, Guo N (2013) Unique critical state characteristics in granular media considering fabric anisotropy. Géotechnique 63(8):695
103. Zienkiewicz OC, Mroz Z (1984) Generalized plasticity formulation and applications to geomechanics. Mech Eng Mater 44(3):655–680
104. Zienkiewicz Olgierd C, Chan AHC, Pastor M, Schrefler BA, Shiomi T (1999) Computational geomechanics. Citeseer, New York
105. Zohdi TI (2013) Rapid simulation of laser processing of discrete particulate materials. Arch Comput Methods Eng 20(4):309–325