From multiscale modeling to metamodeling of geomechanics problems

Kun Wang

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2019

ABSTRACT

From multiscale modeling to metamodeling of geomechanics problems

Kun Wang

In numerical simulations of geomechanics problems, a grand challenge consists of overcoming the difficulties in making accurate and robust predictions by revealing the true mechanisms in particle interactions, fluid flow inside pore spaces, and hydromechanical coupling effect between the solid and fluid constituents, from microscale to mesoscale, and to macroscale. While simulation tools incorporating subscale physics can provide detailed insights and accurate material properties to macroscale simulations via computational homogenizations, these numerical simulations are often too computational demanding to be directly used across multiple scales. Recent breakthroughs of Artificial Intelligence (AI) via machine learning have great potential to overcome these barriers, as evidenced by their great success in many applications such as image recognition, natural language processing, and strategy exploration in games. The AI can achieve super-human performance level in a large number of applications, and accomplish tasks that were thought to be not feasible due to the limitations of human and previous computer algorithms. Yet, machine learning approaches can also suffer from overfitting, lack of interpretability, and lack of reliability. Thus the application of machine learning into generation of accurate and reliable surrogate constitutive models for geomaterials with multiscale and multiphysics is not trivial. For this purpose, we propose to establish an integrated modeling process for automatic designing, training, validating, and falsifying of constitutive models, or ”metamodeling”. This dissertation focuses on our efforts in laying down step-by-step the necessary theoretical and technical foundations for the multiscale metamodeling framework.

The first step is to develop multiscale hydromechanical homogenization frameworks for both bulk granular materials and granular interfaces, with their behaviors homoge-

nized from subscale microstructural simulations. For efficient simulations of field-scale geomechanics problems across more than two scales, we develop a hybrid data-driven method designed to capture the multiscale hydro-mechanical coupling effect of porous media with pores of various different sizes. By using sub-scale simulations to generate database to train material models, an offline homogenization procedure is used to replace the up-scaling procedure to generate path-dependent cohesive laws for localized physical discontinuities at both grain and specimen scales.

To enable AI in taking over the trial-and-error tasks in the constitutive modeling process, we introduce a novel “metamodeling” framework that employs both graph theory and deep reinforcement learning (DRL) to generate accurate, physics compatible and interpretable surrogate machine learning models. The process of writing constitutive models is simplified as a sequence of forming graph edges with the goal of maximizing the model score (a function of accuracy, robustness and forward prediction quality). By using neural networks to estimate policies and state values, the computer agent is able to efficiently self-improve the constitutive models generated through self-playing.

To overcome the obstacle of limited information in geomechanics, we improve the efficiency in utilization of experimental data by a multi-agent cooperative metamodeling framework to provide guidance on database generation and constitutive modeling at the same time. The modeler agent in the framework focuses on evaluating all modeling options (from domain experts’ knowledge or machine learning) in a directed multigraph of elasto-plasticity theory, and finding the optimal path that links the source of the directed graph (e.g., strain history) to the target (e.g., stress). Meanwhile, the data agent focuses on collecting data from real or virtual experiments, interacts with the modeler agent sequentially and generates the database for model calibration to optimize the prediction accuracy. Finally, we design a non-cooperative meta-modeling framework that focuses on automatically developing strategies that simultaneously generate experimental data to calibrate model parameters and explore weakness of a known constitutive model until the strengths and weaknesses of the constitutive law on the application range can be identified through competition. These tasks are enabled by a zero-sum reward system of the metamodeling game and robust adversarial reinforcement learning techniques.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to express my most gratitude and appreciation to my advisor Professor Steve WaiChing Sun for his continuous support, kind patience and inspirational encouragement throughout my graduate study at Columbia University. As a tremendous mentor, his guidance helped me in all the time of research and writing of this thesis. It is my fortune to have the opportunity to work with Professor Sun as his graduate student.

I am grateful to Professor Qiang Du, Professor Jeffrey W. Kysar, Professor Haim Waisman and Professor Marco Giometto, for agreeing to serve on my dissertation committee. Their valuable suggestions, remarks and comments in the completion of my dissertation are greatly appreciated.

Many thanks to my friends and colleagues at Columbia: SeonHong Na, Eric Bryant, Nikolaus Vlassis, Hyoung Suk Suh, Ran Ma, Chuanqi Liu, Yousef Heider, Yang Jiao, Zifeng Yuan and many others. They made my life at Columbia eventful and memorable.

To my family

Chapter 1

Introduction

Many engineering applications and geological processes involve localized failures in geomaterials across multiple length scales (from millimeters to kilometers). For example, the overall structural response of macroscale failures, such as hydraulic fracturing, fault reactivation, and damage from underground explosions, depends on the underlying microscale failures such as micro-slip-planes, microfractured damage zones, microvoid collapse, and grain crushing. Understanding the multiscale hydro-mechanical responses of these failures is of ultimate importance for applications such as geological CO_2 storage, nuclear waste disposal, and earthquake rupture dynamics. Advanced high-fidelity experimental and mesoscale modeling techniques can provide a large amount of microstructural information. However, extracting the insightful and interpretable information from this 'big data' to efficiently describe the underlying (grain-scale) mechanisms of larger-scale failures is difficult due to the complexity of hydro-mechanical coupling and microfracture induced anisotropy. Hence the demand for modeling approaches that can efficiently extract and interpret information from the collected data. Novel machine learning techniques such as reinforcement learning and convolutional neural networks have found great success across multiple disciplines such as image recognition, natural language processing, and strategy exploration in games, but are just now being explored by the computational geomechanics community. The data-driven computational geomechanics requires overcoming the major obstacles of overfitting, lack of interpretability, and lack of reliability in machine learning. This dissertation focuses on our efforts in laying down step-by-step the necessary theoretical and technical foundations for developing an innovative framework of designing and validating accurate surrogate constitutive models for multiscale and multiphysics of geomaterials, which is named ”metamodeling”.

The first step consists of establishing multiscale homogenization frameworks for porous media. We develop multiscale multiphysics finite element modeling frameworks for both bulk granular materials (**Chapter 2**) and granular interfaces (**Chapter 3**), with their hydromechanical behaviors homogenized from subscale microstructural simulations. This approach allows one to bypass the need of deriving multi-physical phenomenological laws for complex loading paths and enables the capturing of the evolving anisotropy of the contact fabrics of grains and the permeabilities of macro-pores and micro-pores.

For efficient simulations of field-scale geomechanics problems across more than two scales, we develop a hybrid data-driven method designed to capture the multiscale hydromechanical coupling effect of porous media with pores of various different sizes (**Chapter 4**). At each scale, data-driven models generated from supervised machine learning are hybridized with classical constitutive laws in a directed graph that represents the numerical models. By using sub-scale simulations to generate database to train material models, an offline homogenization procedure is used to replace the up-scaling procedure to generate cohesive laws for localized physical discontinuities at both grain and specimen scales. Through a proper homogenization procedure that preserves spatial length scales, the proposed method enables field-scale simulations to gather insights from meso-scale and grain-scale micro-structural attributes. This method is proven to be much more computationally efficient than the classical DEM-FEM approach while at the same time more robust and flexible than the classical surrogate modeling approach.

For automatic generation of optimal surrogate models of geomaterials based on collected data, we introduce a novel "meta-modeling" framework that employs both graph theory and deep reinforcement learning (DRL) to generate constitutive models (**Chapter 5**). The constitutive models are conceptualized as information flow in directed graphs. The process of writing constitutive models is simplified as a sequence of forming graph edges with the goal of maximizing the model score (a function of accuracy, robustness and forward prediction quality). Thus meta-modeling can be formulated as a Markov decision process with well-defined states, actions, rules, and rewards. By using neural networks to estimate policies and state values, the computer agent is able to efficiently self-improve the constitutive models generated through self-playing, in the same way AlphaGo Zero (the algorithm that outplayed the world champion in the game of Go) improves its gameplay. Our numerical examples show that this automated meta-modeling framework not only produces models which outperform existing cohesive models on benchmark traction–separation data, but is also capable of detecting hidden mechanisms among micro-structural features and incorporating them in constitutive models to improve the forward prediction accuracy.

To overcome the obstacle of limited information in geomechanics, a multi-agent cooperative metamodeling framework is designed to provide guidance on database generation and constitutive modeling at the same time (**Chapter 6**). The modeler agent in the framework focuses on evaluating all modeling options (from domain experts' knowledge or machine learning) in a directed multigraph of elasto-plasticity theory, and finding the optimal path that links the source of the directed graph (e.g., strain history) to the target (e.g., stress). Meanwhile, the data agent, who focuses on collecting data from real or virtual experiments, interacts with the modeler agent sequentially and generate the database for model calibration to optimize the prediction accuracy. This treatment enables us to emulate an idealized scientific collaboration between experimentalists and modelers to derive, implement, calibrate and validate a constitutive model for the complex responses of geomaterials.

Finally, to automatically assess the strength, weakness, and robustness of geomechanics models, we design a non-cooperative meta-modeling framework that focuses on developing strategies that simultaneously generate experimental data to calibrate model parameters and explore weakness of a known constitutive model until the strengths and weaknesses of the constitutive law on the application range can be identified through competition (**Chapter 7**). These tasks are enabled by the recent development of robust adversarial reinforcement learning techniques.

As for notations and symbols, bold-faced letters denote tensors; the symbol ′·′ 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. $\boldsymbol{C} : \boldsymbol{\epsilon}^{\boldsymbol{e}} = C_{ijkl} \epsilon^e_{kl}$); 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 $(\boldsymbol{I}^4_{\mathrm{sym}})_{ijkl} = \frac{1}{2}(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{kj})$, 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.

Chapter 2

Multiscale discrete-continuum coupling for saturated porous media

This chapter is reproduced from the published paper: K. Wang, W.C. Sun, A semi-implicit discrete-continuum coupling method for porous media based on the effective stress principle at finite strain, *Computer Methods in Applied Mechanics and Engineering*, 304(1):546-583, 2016.

2.1 Introduction

A two-phase fluid-infiltrating porous solid is made of a solid matrix and a pore space saturated by fluid. When subjected to external loading, the mechanical responses of the porous solid strongly depend on whether and how pore fluid diffuse inside the pore space. The classical approach to model the fluid-solid interaction in a porous solid is to consider it as a mixture continuum in the macroscopic scale. At each continuum material point, a fraction of volume is occupied by one or multiple types of fluid, while the rest of volume is occupied by the solid constituent. A governing equation can then be derived from balance principles of the mixture [223, 19, 232, 31]. One key ingredient for the success of this continuum approach is the effective stress principle, which postulates that the external loading imposed on porous solid is partially carried by the solid skeleton and partially supported by the fluid [224, 19, 183]. By assuming that the total stress is a linear combination of the effective stress of the solid skeleton and the pore pressure of interstitial fluid, analytical and numerical solutions can be sought once a proper set of constitutive laws is identified to relate effective stress with strain and internal variables, and Darcy's velocity with pore pressure can be identified even though effective stress cannot be measured directly [223, 258, 138]. In recent years, the advancement of computational resources has led to the development of numerous finite element models that employ the effective stress principle [176, 199, 25, 6]. Nevertheless, modeling the complex path-dependent responses for geomaterials remains a big challenge [52]. This difficulty is partly due to the need to incorporate a large amount of internal variables and material parameters, which makes the calibration more difficult. Another difficulty is due to the weak underpinning of the phenomenological approach to replicate anisotropy caused by changes of micro-structures and fabric [124].

A conceptually simple but computational expensive remedy to resolve this issue is to explicitly model the microscopic fluid-solid interaction. In fact, this approach has been widely used to study sedimentation problems. Previous work, such as [91, 137, 82, 92, 184, 17], has obtained various degree of successes in simulating fluidized granular beds by establishing information exchange mechanism among discrete element model and fluid solvers. For a subset of two-phase problems in which the length scale of interest is larger than the grain diameter and the fluid flow is laminar, the pore-scale interstitial fluid motion is often not resolved but instead modeled via a locally averaged Navier-Stokes equations

(LANS) that couples with DEM via a parametric drag force [50, 184]. By assuming that a weak separation of scale is valid, this meso-scale approach essentially couples the large scale Navier-Stokes fluid motion with grain-scale DEM model that captures the granular flow nature via interface force. While this method is found to be very efficient for mixing problem, coupling the macroscopic flow at meso-scale via force-based interaction is not without limitations. First, the simulated hydro-mechanical coupling effect is highly sensitive to the fluid drag force model chosen to replicate the particle-fluid interaction. This can lead to complications for calibration and material identification, as the expressions of these fluid drag forces are often empirically correlated by the local porosity, Reynolds number and other factors such as the diameter of the particles, [268]. Furthermore, the meso-scale fluid-particle simulations still require significant computational resources when a large number of particles are involved.

The purpose of this study is to propose a new multiscale hydro-mechanical model that (1) provides the physical underpinning from discrete element simulations, (2) resolves the problems associated with the phenomenological nature of drag force, and (3) improves the efficiency of large-scale problems. Our target is a sub-class of problems in which the solid skeleton is composed of particulate assemblies in solid state (rather than granular flow) and the porous space is fully saturated with a single type of pore fluid in laminar regime. As in the previous work for particle-fluid system [50, 184], we also adopt the assumption that a weak separation of scale exists between the motion of solid particles and that of the pore fluid. Our major departure is the way we employ this weak separation of the pore fluid to establish hydro-mechanical coupling across length scales. Instead of using the phenomenological drag force model to establish coupling, we use the effective stress principle to partition the macroscopic total stress as the sum of effective stress, which comes from microscopic DEM simulation, and the fluid contribution, which comes from the Biot's coefficient inferred from DEM assemblies and the pore pressure updated from a total Lagrangian poromechanics finite element solver.

The coupled transient problem requires a time integration scheme to advance numerical solution from known solid displacement $\boldsymbol{u}_n$ and pore-fluid pressure p^f_n at time t_n to unknowns $\boldsymbol{u}_{n+1}$ and p^f_{n+1} at the next time step $t_{n+1} = t_n + \Delta t$. Explicit integration scheme has been employed in multiscale dynamic analysis of soils [159]. This method is simple in the sense that it advances solutions without solving system of equations. However, it often requires small time steps in order to achieve numerical stability, and when coupling with DEM solver, the condition is even more stringent. Another approach, the implicit scheme, has the possibility of attaining unconditional stability, but the linearization of variational equations, equation solving and iterations require much computational cost per time step. To make a trade-off, [99] and [177] suggest the usage of an implicit-explicit predictor/multicorrector scheme in nonlinear hydro-mechanical transient problem. Our main contribution in this study is the extension of this method to multiscale coupling problems. We suggest a distinct treatment of the elastic and plastic component of material stiffness homogenized from DEM microstructures. Accordingly, an information exchange scheme is established between the FEM and DEM solvers.

The rest of this chapter is organized as follows. In section 2, we first describe the homogenization theory for saturated porous media serving as the framework for micro-macro transitions. Then, the discrete-continuum coupling model in the finite deformation range is presented in Section 3. The details of the multiscale semi-implicit method are provided in Section 4, with an emphasis placed on how the material properties homogenized from DEM are employed in the semi-implicit FEM-mixed-DEM solution scheme.

Selected problems in geomechanics are simulated via the proposed method to study its performance and their results are presented in Section 5. Finally, concluding remarks are given in Section 6.

2.2 Homogenization theory for porous media

In this section, we describe the homogenization theory we adopt to establish the DEM-mixed-FEM coupling model for fully saturated porous media. Previous work for dry granular materials, such as [143, 144, 88], has demonstrated that a hierarchical discrete-continuum coupling model can be established by using grain-scale simulations to provide Gauss point stress update for finite element simulations in a fully implicit scheme. Nevertheless, the extension of this idea for partially or fully saturated porous media has not been explored, to the best knowledge of the authors.

In this work, we hypothesize that the pore-fluid flow inside the pores is in the laminar regime and is dominated by viscous forces such that Darcy's law is valid at the representative elementary volume level [210, 209, 214]. Provided that this assumption is valid, we define the pore pressure field only at the macroscopic level and neglect local fluctuation of the pore pressure at the pore- and grain-scale.

On the other hand, we abandon the usage of macroscopic constitutive law to replicate the constitutive responses of the solid constituent. Instead, we apply the effective stress principle [223, 87, 86] and thus allow the change of the macroscopic effective stress as a direct consequence of the compression, deformation and shear resistance of the solid constituent inferred from grain-scale simulations. As a result, the effective stress can be obtained from homogenizing the forces and branch vectors of the force network formed by the solid particles or aggregates, while the total stress becomes a partition of the homogenized effective stress from the microscopic granular assemblies, and the pore pressure from the macroscopic mixture continuum.

2.2.1 Dual-scale effective stress principle

In this study, we make assumptions that (1) a separation of scale exists and that (2) a representative volume element (RVE) can be clearly defined. Strictly speaking, the assumption (2) is true if the unit cell has a periodic microstructure or when the volume is sufficiently large such that it possesses statistically homogeneous and ergodic properties [80].

With the aforementioned assumptions in mind, we consider a homogenized macroscopic solid skeleton continuum $\mathcal{B}^s \subset \mathbb{R}^3$ whose displacement field is C^0 continuous. Each position of the macroscopic solid body in the reference configuration, i.e., $\boldsymbol{X} = \boldsymbol{X}^s \in \mathcal{B}^s_0$, is associated with a micro-structure of the RVE size. Let us denote the trajectories of the macroscopic solid skeleton and the fluid constituent in the saturated two-phase porous medium from the reference configuration to the current solid configuration as,

$$\boldsymbol{x} = \boldsymbol{\varphi}^s(\boldsymbol{X}, t) \ ; \ \boldsymbol{x} = \boldsymbol{\varphi}^f(\boldsymbol{X}^f, t) \tag{2.1}$$

Unless the porous medium is locally undrained, the solid and fluid constituents are not bundled to move along the same trajectory, i.e., $\boldsymbol{\varphi}^s(\cdot, t) \neq \boldsymbol{\varphi}^f(\cdot, t)$. If we choose to follow the macroscopic solid skeleton trajectory to formulate the macroscopic balance principles, then the control volumes are attached to solid skeleton only, and the pore fluid motion is described by relative movement between the fluid constituent and the solid matrix, as shown in Fig. 2.1. The deformation gradient of the macroscopic solid constituent $\boldsymbol{F}$ can therefore be written as,

$$\boldsymbol{F} = \frac{\partial \boldsymbol{\varphi}(\boldsymbol{X}^s, t)}{\partial \boldsymbol{X}^s} = \frac{\partial \boldsymbol{\varphi}(\boldsymbol{X}, t)}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} \tag{2.2}$$

in which we omit the superscript s when quantities are referred to solid phase. Now, following [143], we associate each point in the current configuration $\boldsymbol{x}$ with an aggregate of N particles inside the representative volume $\mathcal{V}$. Furthermore, we introduce a local coordinate system for the RVE in which the position vector $\boldsymbol{y} \in \mathbb{R}^3$ becomes $\mathbf{0}$ at the geometric centroid of the RVE. The locations of the centroids of the N particles expressed using the local coordinate system read, i.e.,

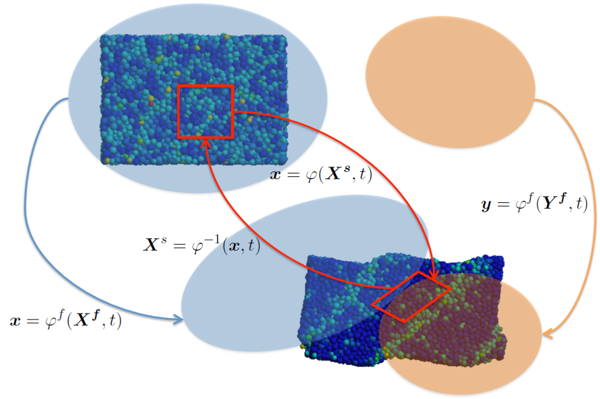

$\boldsymbol{x} = \varphi(\boldsymbol{X}^s, t)$
$\boldsymbol{y} = \varphi^f(\boldsymbol{Y}^f, t)$
$\boldsymbol{X}^s = \varphi^{-1}(\boldsymbol{x}, t)$
$\boldsymbol{x} = \varphi^f(\boldsymbol{X}^f, t)$

Figure 2.1: Trajectories of the solid and fluid constituents $\boldsymbol{\varphi}^s = \boldsymbol{\varphi}$ and $\boldsymbol{\varphi}^f$. The motion $\boldsymbol{\varphi}$ conserves all the mass of the solid constituent, while the fluid may enter or leave the body of the solid constituent. Figure reproduced from [215]

.

$$\boldsymbol{y}_p \in \mathcal{V}, \; p = 1, 2, ...N. \tag{2.3}$$

where $\boldsymbol{y}_p$ is the local position vector of the center of the p-th particle in the microstructure and $\boldsymbol{x} + \boldsymbol{y}_p$ is the same position expressed in the macroscopic current coordinate system. Particles inside the RVE may make contacts to each other. The local position vector of each contact between each particle-pair $\boldsymbol{y}_c$ can be written as,

$$\boldsymbol{y}_c \in \mathcal{V}, \; c = 1, 2, ...N_c. \tag{2.4}$$

Both the positions of the particles $\boldsymbol{y}_p$ and that of the contacts $\boldsymbol{y}_c$ are governed by contact law and the equilibrium equations. Previous works, such as [50, 73, 184], have found success in explicitly modeling the pore-scale grain-fluid interaction. Nevertheless, such grain-fluid interaction simulations do impose a very high computational demand due to the fact that the fluid flow typically requires at least an order more of degree of freedoms to resolve the flow in the void space among particles. However, for seepage flow that is within the laminar regime where Darcy's law applies, the new insight obtained from the costly simulations will be limited. As a result, this discrete-continuum coupling model does not explicitly model the pore-scale solid-fluid interaction. Instead, we rely on the hypothesis that effective stress principle is valid for the specific boundary value problems we considered. In particular, we make the following assumptions:

- The void space is always fully saturated with one type of fluid and there is no capillary effect that leads to apparent cohesion of the solid skeleton.
- The flow in the void space remains Darcian at the macroscopic level.
- All particles in the granular assemblies are in contact with the neighboring particles.
- Fluidization, suffusion and erosion do not occur.
- Grain crushing does not occur.
- There is no mass exchange between the fluid and solid constituents.

As a result, we may express the total macroscopic Cauchy stress as a function of homogenized Cauchy effective stress inferred from DEM and the macroscopic pore pressure obtained from the mixed finite element, i.e.

$$\boldsymbol{\sigma}(\boldsymbol{x}, t) = < \boldsymbol{\sigma}'(\boldsymbol{x}, t) >_{\mathrm{RVE}} - B(\boldsymbol{x}, t) p^f(\boldsymbol{x}, \boldsymbol{t}) \boldsymbol{I} \tag{2.5}$$

where

$$< \sigma'(\boldsymbol{x},t) >_{\text{RVE}} = \frac{1}{2\text{V}_{\text{RVE}}} \sum_{c}^{Nc} (\boldsymbol{f}^c \otimes \boldsymbol{l}^c + \boldsymbol{l}^c \otimes \boldsymbol{f}^c) \tag{2.6}$$

$\boldsymbol{f}^c$ is the contact force and $\boldsymbol{l}^c$ is the branch vector, the vector that connects the centroids of two grains forming the contact [46, 11, 214], at the grain contact $\boldsymbol{x} + \boldsymbol{y}_c \in \mathbb{R}^3$. V_{RVE} is the volume of the RVE and N_c is the total number of particles in the RVE. Meanwhile, the Biot's coefficient B reads,

$$B(\boldsymbol{x}, \boldsymbol{t}) = 1 - \frac{K_T^{\text{DEM}}(\boldsymbol{x}, t)}{K_s} \tag{2.7}$$

with $K_T^{\text{DEM}}(\boldsymbol{x}, t)$ and K_s being the effective tangential bulk modulus of the solid matrix inferred from DEM, and the bulk modulus of the solid grain respectively [156, 199]. Notice that, in the geotechnical engineering and geomechanics literature, such as [153, 114], it is common to impose incompressible volumetric constraint on dry DEM assembly to simulate undrained condition at meso-scale. This treatment can be considered as a special case of (2.7) when the bulk modulus of the solid grain is significantly higher than that of the skeleton such that the Biot's coefficient is approximately equal to one.

2.2.2 Micro-macro-transition for solid skeleton

In this study, we consider the class of two-phase porous media of which the solid skeleton is composed of particles. These particles can be cohesion-less or cohesive, but the assemblies they formed are assumed to be of particulate nature and hence suitable for DEM simulations. [49].

In our implementation, the DEM simulations are conducted via YADE (Yet Another Dynamic Engine [202]), an open source code base for discontinua. These grain-scale DEM simulations are used as a replacement to the macroscopic constitutive laws that relate strain measure with effective stress measure for each RVE associated with a Gauss point in the macroscopic mixed finite element. In particular, a velocity gradient is prescribed to move the frame of the unit cell and the DEM will seek for the static equilibrium state via dynamics relaxation method. After static equilibrium is achieved, the internal forces and branch vectors are used to compute the homogenized effective Cauchy stress via the micro-macro transition theory [143, 144, 252].

The Hill-Mandel micro-heterogeneity condition demands that the power at the microscopic scale must be equal to the the rate of work done measured by the macroscopic effective stress and strain rate measures. For the solid constituent of the two-phase porous media, this condition can be expressed in terms of any power-conjugate effective stress and strain rate pair , such as $(\boldsymbol{P}', \dot{\boldsymbol{F}})$ and $(\boldsymbol{S}', \dot{\boldsymbol{E}})$ and $(\boldsymbol{\sigma}', \boldsymbol{D})$ [25, 6]. For instance, the condition can be written in terms of the effective stress and rate of deformation of the solid skeleton, i.e.,

$$< \boldsymbol{\sigma}' >_{\text{RVE}} : < \boldsymbol{D} >_{\text{RVE}} = < \boldsymbol{\sigma}' : \boldsymbol{D} >_{\text{RVE}} \tag{2.8}$$

where $\boldsymbol{D}$ is the rate of deformation, i.e., the symmetric part of the velocity gradient tensor,

$$< \boldsymbol{D} >_{\text{RVE}} = \frac{1}{2}(< \boldsymbol{L} >_{\text{RVE}} + < \boldsymbol{L}^{\text{T}} >_{\text{RVE}}) \ ; \ \boldsymbol{L} = \boldsymbol{\nabla}^{x} \boldsymbol{v} \tag{2.9}$$

and $< \boldsymbol{\sigma}' >_{\text{RVE}}$ is defined previously in (2.6). Previous studies, such as, [143, 252, 144, 68], have established that the linear deformation, periodic, and uniform traction are three boundary conditions that satisfy the Hill-Mandel micro-heterogeneity condition. In our implementation, we apply the periodic boundary condition to obtain the effective stress measure, because the periodic boundary condition may yield responses that are softer than those obtained from the linear deformation BC but stiffer than those obtained from the uniform traction BC. In particular, the periodic boundary condition enforces two constraints:

(1) the periodicity of the deformation, i.e.,

$$[[\boldsymbol{y}_b]] = < \boldsymbol{F} >_{\mathrm{RVE}} [[\boldsymbol{Y}_b]] \text{ and } [[\boldsymbol{R}_b]] = \boldsymbol{0} \tag{2.10}$$

where $[[\cdot]]$ denotes the jump across boundaries, $\boldsymbol{y}_b$ and $\boldsymbol{Y}_b$ represent the position vectors of the particles at the boundary of the reference and current configurations, $R_b \in SO(3)$ represents the rotation tensor of particles at the boundary, and (2) the anti-periodicity of the force $\boldsymbol{f}_b$ and moment on the boundary of the RVE, i.e.,

$$[[\boldsymbol{f}_b]] = \boldsymbol{0} \text{ and } [[(\boldsymbol{y}_c - \boldsymbol{y}_b) \times \boldsymbol{f}_b]] = \boldsymbol{0} \tag{2.11}$$

In YADE, the DEM code we employed for grain-scale simulations, the deformation of an RVE is driven by a periodic cell box in which the macroscopic velocity gradient of the unit cell $< \boldsymbol{L} >_{\mathrm{RVE}}$ can both be measured and prescribed.

2.3 Multiscale DEM-mixed-FEM hydro-mechanical model

The differential equations governing the isothermal saturated porous media in large deformation are derived based on the mixture theory, in which solid matrix and pore fluid are treated together as a multiphase continuum [176, 25, 6, 48, 215, 139]. The solid and fluid constituents may simultaneously occupy fractions of the volume of the same material point. The physical quantities of the mixture, such as density and total stress, are spatially homogenized from its components. For example, the averaged density of the fluid saturated soil mixture is defined as:

$$\rho = \rho^s + \rho^f = (1 - \phi)\rho_s + \phi\rho_f \tag{2.12}$$

where ρ^α is the partial mass density of the α constituent and ρ_α is the intrinsic mass density of the α constituent, with ϕ being the porosity.

2.3.1 Balance of linear momentum

For the balance of linear momentum law in finite strain, we adopt the total Lagrangian formulation and choose the total second Piola-Kirchhoff stress (PK2) $\boldsymbol{S}$ as the stress measure. The inertial effect is neglected. The equation takes the form:

$$\nabla^{\boldsymbol{X}} \cdot (\boldsymbol{F}\boldsymbol{S}) + J(\rho^s + \rho^f)\boldsymbol{g} = \boldsymbol{0} \tag{2.13}$$

where the Jacobian $J = \det(\boldsymbol{F})$. The principle of effective stress postulates that the total Cauchy stress $\boldsymbol{\sigma}$ can be decomposed into an effective stress due to the solid skeleton deformation and an isotropic pore pressure (p^f) stress. The effective stress principle in terms of PK2 writes:

$$\boldsymbol{S} = \boldsymbol{S'}^{\mathrm{DEM}} - J\boldsymbol{F}^{-1}B^{\mathrm{DEM}}p^f\boldsymbol{I}\boldsymbol{F}^{-T} \tag{2.14}$$

where

$$\boldsymbol{S'}^{\mathrm{DEM}} = J\boldsymbol{F}^{-1}\boldsymbol{\sigma'}^{\mathrm{DEM}}\boldsymbol{F}^{-T} = J\boldsymbol{F}^{-1}(\frac{1}{\mathrm{V}_{\mathrm{RVE}}}\sum_{i}^{Nc}\boldsymbol{f}\otimes\boldsymbol{l})\boldsymbol{F}^{-T} \tag{2.15}$$

Thus the balance of linear momentum becomes:

$$\nabla^{\boldsymbol{X}} \cdot (\boldsymbol{F}\boldsymbol{S'}^{\mathrm{DEM}} - JB^{\mathrm{DEM}}p^f\boldsymbol{F}^{-T}) + J(\rho^s + \rho^f)\boldsymbol{g} = \boldsymbol{0} \tag{2.16}$$

2.3.2 Balance of fluid mass

The simplified u-p formulation in finite strain requires another equation illustrating the balance of mass for pore fluid constituent:

$$\frac{D\rho^f}{Dt} = -\nabla^{\boldsymbol{X}} \cdot (J\boldsymbol{F}^{-\boldsymbol{1}}[\phi^{\mathrm{DEM}}\rho_f(\boldsymbol{v}^f - \boldsymbol{v})]) \tag{2.17}$$

where $\frac{D[\cdot]}{Dt} = \dot{[\cdot]}$ is the material time derivative with respect to the velocity of solid skeleton $\boldsymbol{v}$.

We make isothermal and barotropic assumptions and suppose that $p^f << K_s$ and that $\frac{DB^{\mathrm{DEM}}}{Dt} \sim 0$. After simplifications [215], the balance of mass becomes:

$$\frac{B^{\mathrm{DEM}}}{J}\frac{DJ}{Dt} + \frac{1}{M^{\mathrm{DEM}}}\frac{Dp^f}{Dt} + \nabla^{\boldsymbol{X}} \cdot \left(\frac{1}{\rho_f}(J\boldsymbol{F}^{-\mathbf{1}}[\phi^{\mathrm{DEM}}\rho_f(\boldsymbol{v}^{\boldsymbol{f}} - \boldsymbol{v})])\right) = 0 \tag{2.18}$$

where

$$M^{\mathrm{DEM}} = \frac{K_s K_f}{K_f(B^{\mathrm{DEM}} - \phi^{\mathrm{DEM}}) + K_s \phi^{\mathrm{DEM}}} \tag{2.19}$$

is the Biot's modulus [156], with K_f being the bulk modulus of pore fluid.

In this chapter, Darcy's constitutive law relating the relative flow and the pore pressure is employed, neglecting the inertial effect:

$$\boldsymbol{Q} = \boldsymbol{K}^{\mathrm{DEM}} \cdot (-\nabla^{\boldsymbol{X}} p^f + \rho_f \boldsymbol{F}^{\mathrm{T}} \cdot \boldsymbol{g}) \tag{2.20}$$

where the pull-back permeability tensor $\boldsymbol{K}^{\mathrm{DEM}}$ is defined as

$$\boldsymbol{K}^{\mathrm{DEM}} = J\boldsymbol{F}^{-1} \cdot \boldsymbol{k}^{\mathrm{DEM}} \cdot \boldsymbol{F}^{\text{-T}} \tag{2.21}$$

Assume that the effective permeability tensor $\boldsymbol{k}^{\mathrm{DEM}}$ is isotropic, i.e.,

$$\boldsymbol{k}^{\mathrm{DEM}} = k^{\mathrm{DEM}}\boldsymbol{I} \tag{2.22}$$

where k^{DEM} is the scalar effective permeability in unit of $\frac{m^2}{Pa \cdot s}$. It is updated from porosity of DEM RVEs according to the Kozeny-Carmen equation.

2.3.3 Weak form

To construct the macroscopic hydro-mechanical boundary-value problem, consider a reference domain $\mathcal{B}$ with its boundary $\partial\mathcal{B}$ composed of Dirichlet boundaries (solid displacement $\partial\mathcal{B}_u$, pore pressure $\partial\mathcal{B}_p$) and Von Neumann boundaries (solid traction $\partial\mathcal{B}_t$, fluid flux

$\partial\mathcal{B}_q$) satisfying

$$\begin{cases} \partial\mathcal{B} = \overline{\partial\mathcal{B}_{\boldsymbol{u}} \cup \partial\mathcal{B}_{\boldsymbol{t}}} = \overline{\partial\mathcal{B}_p \cup \partial\mathcal{B}_q} \\ \emptyset = \partial\mathcal{B}_{\boldsymbol{u}} \cap \partial\mathcal{B}_{\boldsymbol{t}} = \partial\mathcal{B}_p \cap \partial\mathcal{B}_q \end{cases} \tag{2.23}$$

The prescribed boundary conditions are

$$\begin{cases} \boldsymbol{u} = \overline{\boldsymbol{u}} \text{ on } \partial\mathcal{B}_{\boldsymbol{u}} \\ \boldsymbol{P}\cdot\boldsymbol{N} = (\boldsymbol{F}\cdot\boldsymbol{S})\cdot\boldsymbol{N} = \overline{\boldsymbol{t}} \text{ on } \partial\mathcal{B}_{\boldsymbol{t}} \\ p^f = \overline{p} \text{ on } \partial\mathcal{B}_p \\ -\boldsymbol{N}\cdot\boldsymbol{Q} = \overline{Q} \text{ on } \partial\mathcal{B}_Q \end{cases} \tag{2.24}$$

where $\boldsymbol{N}$ is outward unit normal on undeformed surface $\partial\mathcal{B}$.

For model closure, the initial conditions are imposed as

$$p^f = p^f_0,\ \boldsymbol{u} = \boldsymbol{u_0} \text{ at } t = t_0 \tag{2.25}$$

Following the standard procedures of the variational formulation, we obtain finally the weak form of the balance of linear momentum and mass

$$\begin{aligned} G \quad &: \quad V_{\boldsymbol{u}} \times V_p \times V_\eta \to \mathbb{R} \\ G(\boldsymbol{u}, p^f, \boldsymbol{\eta}) \quad &= \quad \int_{\mathcal{B}} \nabla^{\boldsymbol{X}} \boldsymbol{\eta} : (\boldsymbol{F}\cdot\boldsymbol{S'}^{\text{DEM}} - JBp^f\boldsymbol{F}^{\text{-T}})\ \mathrm{dV} - \int_{\mathcal{B}} J(\rho^f + \rho^s)\boldsymbol{\eta}\cdot\boldsymbol{g}\mathrm{dV} \\ &\quad - \int_{\partial\mathcal{B}_t} \boldsymbol{\eta}\cdot\overline{\boldsymbol{t}}\ \mathrm{d}\Gamma = 0 \end{aligned} \tag{2.26}$$

$$\begin{aligned} H \quad &: \quad V_{\boldsymbol{u}} \times V_p \times V_\psi \to \mathbb{R} \\ H(\boldsymbol{u}, p^f, \psi) \quad &= \quad \int_{\mathcal{B}} \psi \frac{B^{\text{DEM}}}{J}\dot{J}\ \mathrm{dV} + \int_{\mathcal{B}} \psi \frac{1}{M^{\text{DEM}}}\dot{p^f}\ \mathrm{dV} \\ &\quad - \int_{\mathcal{B}} \nabla^{\boldsymbol{X}} \psi \cdot [\boldsymbol{K}^{\text{DEM}} \cdot (-\nabla^{\boldsymbol{X}} p^f + \rho_f \boldsymbol{F}^{\text{T}}\cdot\boldsymbol{g})]\ \mathrm{dV} \\ &\quad - \int_{\partial B_Q} \psi\overline{\boldsymbol{Q}}\ \mathrm{d}\Gamma = 0 \end{aligned} \tag{2.27}$$

The first integral of $H(\boldsymbol{u}, p^f, \psi)$ can be related to the solid velocity field $\dot{\boldsymbol{u}}$ using the equations $\dot{J} = J\, \nabla^x \cdot \dot{\boldsymbol{u}}$ and $\nabla^x \cdot \dot{\boldsymbol{u}} = \nabla^X \dot{\boldsymbol{u}} : \boldsymbol{F}^{\text{-T}}$ [25]:

$$\int_{\mathcal{B}} \psi \frac{B^{\text{DEM}}}{J} \dot{J}\, \text{dV} = \int_{\mathcal{B}} \psi B^{\text{DEM}}\, \nabla^x \cdot \dot{\boldsymbol{u}}\, \text{dV} = \int_{\mathcal{B}} \psi B^{\text{DEM}} \boldsymbol{F}^{\text{-T}} : \nabla^X \dot{\boldsymbol{u}}\, \text{dV} \tag{2.28}$$

The displacement and pore pressure trial spaces for the weak form are defined as

$$V_{\boldsymbol{u}} = \{\boldsymbol{u} : \mathcal{B} \rightarrow \mathbb{R}^3 | \boldsymbol{u} \in [H^1(\mathcal{B})]^3, \boldsymbol{u}|_{\partial B_u} = \overline{\boldsymbol{u}}\} \tag{2.29}$$

$$V_p = \{p^f : \mathcal{B} \rightarrow \mathbb{R} | p^f \in H^1(\mathcal{B}), p^f|_{\partial B_p} = \overline{p}\} \tag{2.30}$$

and the corresponding admissible spaces of variations are defined as

$$V_{\boldsymbol{\eta}} = \{\boldsymbol{\eta} : \mathcal{B} \rightarrow \mathbb{R}^3 | \boldsymbol{\eta} \in [H^1(\mathcal{B})]^3, \boldsymbol{\eta}|_{\partial B_u} = \boldsymbol{0}\} \tag{2.31}$$

$$V_\psi = \{\psi : \mathcal{B} \rightarrow \mathbb{R} | \psi \in H^1(\mathcal{B}), \psi|_{\partial B_p} = 0\} \tag{2.32}$$

H^1 denotes the Sobolev space of degree one, which is the space of square integrable function whose weak derivative up to order 1 are also square integrable (cf. [98, 32]).

2.3.4 Finite element spatial discretization

The spatially discretized equations can be derived following the standard Galerkin procedure. Shape functions $\boldsymbol{N_u}(\boldsymbol{X})$ and $\boldsymbol{N_p}(\boldsymbol{X})$ are used for approximation of solid motion $\boldsymbol{u}$, $\dot{\boldsymbol{u}}$ and pore pressure p^f, $\dot{p^f}$, respectively:

$$\begin{cases} \boldsymbol{u} = \boldsymbol{N_u}\bar{\boldsymbol{u}}, \;\; \dot{\boldsymbol{u}} = \boldsymbol{N_u}\dot{\bar{\boldsymbol{u}}}, \;\; \eta = \boldsymbol{N_u}\bar{\boldsymbol{\eta}} \\ p^f = \boldsymbol{N_p}\bar{\boldsymbol{p}^f}, \;\; \dot{p^f} = \boldsymbol{N_p}\dot{\bar{\boldsymbol{p}}}^f, \;\; \psi = \boldsymbol{N_p}\bar{\boldsymbol{\psi}} \end{cases} \tag{2.33}$$

with $\bar{\boldsymbol{u}}$ being the nodal solid displacement vector, $\bar{\boldsymbol{p}}^f$ being the nodal pore pressure vector, $\dot{\bar{\boldsymbol{u}}}$, $\dot{\bar{\boldsymbol{p}}}^f$ being their time derivatives, and $\bar{\boldsymbol{\eta}}$, $\bar{\boldsymbol{\psi}}$ being their variations.

The adopted eight-node hexahedral element interpolates the displacement and pore pressure field with the same order. As a result, this combination does not inherently satisfy the inf-sup condition [254, 215, 207]. Therefore a stabilization procedure is necessary. In this study, the fluid pressure Laplacian scheme is applied. This scheme consists of adding the following stabilization term to the balance of mass equation (2.27) :

$$\int_{\mathcal{B}} \nabla^{\boldsymbol{X}} \psi \; \alpha_{stab} \;\; \nabla^{\boldsymbol{X}} \dot{p^f} \mathrm{dV} \tag{2.34}$$

with α_{stab} a scale factor depending on element size and material properties of the porous media. For detailed formulations, please refer to [233, 215].

We obtain the finite element equations for balance of linear momentum and balance of mass as:

$$\begin{cases} G(\boldsymbol{u}, p^f, \boldsymbol{\eta}) = 0 \\ \\ H(\boldsymbol{u}, p^f, \psi) = 0 \end{cases} \Longrightarrow \begin{cases} \boldsymbol{F^s_{int}}(\bar{\boldsymbol{u}}) - \boldsymbol{K^{up}} \bar{\boldsymbol{p}}^f - \boldsymbol{G^1} = \boldsymbol{F^1_{ext}} \\ \\ \boldsymbol{C_1} \dot{\bar{\boldsymbol{u}}} + (\boldsymbol{C_2} + \boldsymbol{C_{stab}}) \dot{\bar{\boldsymbol{p}}}^f + \boldsymbol{K^p} \bar{\boldsymbol{p}}^f - \boldsymbol{G^2} = \boldsymbol{F^2_{ext}} \end{cases} \tag{2.35}$$

with expressions for each term:

$$\begin{cases} \boldsymbol{F^s_{int}}(\bar{\boldsymbol{u}}) = \int_{\mathcal{B}} (\nabla^{\boldsymbol{X}} \boldsymbol{N_u})^{\mathrm{T}} : (\boldsymbol{F} \cdot \boldsymbol{S'}^{\mathrm{DEM}}) \mathrm{dV} \\ \boldsymbol{K^{up}} = \int_{\mathcal{B}} J B^{\mathrm{DEM}} (\nabla^{\boldsymbol{X}} \boldsymbol{N_u})^{\mathrm{T}} : \boldsymbol{F}^{\text{-T}} \cdot \boldsymbol{N_p} \mathrm{dV} \\ \boldsymbol{C_1} = \int_{\mathcal{B}} B^{\mathrm{DEM}} \boldsymbol{N_p^{\mathrm{T}}} \boldsymbol{F}^{\text{-T}} : (\nabla^{\boldsymbol{X}} \boldsymbol{N_u}) \mathrm{dV} \\ \boldsymbol{C_2} = \int_{\mathcal{B}} \frac{1}{M^{\mathrm{DEM}}} \boldsymbol{N_p^{\mathrm{T}}} \boldsymbol{N_p} \mathrm{dV} \\ \boldsymbol{C_{stab}} = \int_{\mathcal{B}} (\nabla^{\boldsymbol{X}} \boldsymbol{N_p})^{\mathrm{T}} \alpha_{stab} (\nabla^{\boldsymbol{X}} \boldsymbol{N_p}) \mathrm{dV} \\ \boldsymbol{K^p} = \int_{\mathcal{B}} (\nabla^{\boldsymbol{X}} \boldsymbol{N_p})^{\mathrm{T}} (J \boldsymbol{F}^{-1} \cdot \boldsymbol{k}^{\mathrm{DEM}} \cdot \boldsymbol{F}^{\text{-T}}) (\nabla^{\boldsymbol{X}} \boldsymbol{N_p}) \mathrm{dV} \\ \boldsymbol{G^1} = \int_{\mathcal{B}} J (\rho^f + \rho^s) \boldsymbol{N_u}^{\mathrm{T}} \boldsymbol{g} \mathrm{dV} \\ \boldsymbol{G^2} = \int_{\mathcal{B}} (\nabla^{\boldsymbol{X}} \boldsymbol{N_p})^{\mathrm{T}} (J \boldsymbol{F}^{-1} \cdot \boldsymbol{k}^{\mathrm{DEM}} \cdot \boldsymbol{F}^{\text{-T}}) \rho_f \boldsymbol{F}^{\mathrm{T}} \cdot \boldsymbol{g} \mathrm{dV} \\ \boldsymbol{F^1_{ext}} = \int_{\partial \mathcal{B}_t} \boldsymbol{N_u}^{\mathrm{T}} \bar{\boldsymbol{t}} \mathrm{d}\Gamma \\ \boldsymbol{F^2_{ext}} = \int_{\partial B_Q} \boldsymbol{N_p}^{\mathrm{T}} \overline{\boldsymbol{Q}} \mathrm{d}\Gamma \end{cases} \tag{2.36}$$

The non-linear equation system (2.35) can be rewritten in a compact form:

$$\boldsymbol{M^*} \cdot \boldsymbol{v} + \boldsymbol{F^{int}}(\boldsymbol{d}) - \boldsymbol{G}(\boldsymbol{d}) = \boldsymbol{F^{ext}} \tag{2.37}$$

where $\boldsymbol{M}^* = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{C_1} & (\boldsymbol{C_2} + \boldsymbol{C_{stab}}) \end{bmatrix}$, $\boldsymbol{v} = \begin{Bmatrix} \dot{\bar{\boldsymbol{u}}} \\ \dot{\bar{\boldsymbol{p}}}^f \end{Bmatrix}$, $\boldsymbol{F}^{int} = \begin{Bmatrix} \boldsymbol{F}^s_{int}(\bar{\boldsymbol{u}}) - \boldsymbol{K}^{up}\bar{\boldsymbol{p}}^f \\ \boldsymbol{K}^p\bar{\boldsymbol{p}}^f \end{Bmatrix}$, $\boldsymbol{d} = \begin{Bmatrix} \bar{\boldsymbol{u}} \\ \bar{\boldsymbol{p}^f} \end{Bmatrix}$,
$\boldsymbol{G} = \begin{Bmatrix} \boldsymbol{G}^1 \\ \boldsymbol{G}^2 \end{Bmatrix}$ and $\boldsymbol{F}^{ext} = \begin{Bmatrix} \boldsymbol{F}^1_{ext} \\ \boldsymbol{F}^2_{ext} \end{Bmatrix}$.

2.3.5 Consistent linearization

The semi-implicit solution scheme requires the expression of the tangential stiffness of the implicit contribution. Thus, we perform the consistent linearization of the weak forms (2.26) and (2.27) in the reference configuration [29, 189]. For the balance of linear momentum equation, the consistent linearization reads,

$$\begin{aligned} \delta G(\boldsymbol{u}, p^f, \boldsymbol{\eta}) = {} & \overbrace{\int_{\mathcal{B}} (\boldsymbol{F}^{\mathrm{T}} \cdot \nabla^{\boldsymbol{X}} \boldsymbol{\eta}) : (\boldsymbol{C}^{SE})^{\mathrm{DEM}} : \boldsymbol{\delta E} \, \mathrm{dV}}^{\bar{\boldsymbol{\eta}}^{\mathrm{T}} \boldsymbol{K}^s \delta \bar{\boldsymbol{u}}} + \overbrace{\int_{\mathcal{B}} \boldsymbol{S'}^{\mathrm{DEM}} : (\nabla^{\boldsymbol{X}} \boldsymbol{\delta u})^{\mathrm{T}} \cdot \nabla^{\boldsymbol{X}} \boldsymbol{\eta} \, \mathrm{dV}}^{\bar{\boldsymbol{\eta}}^{\mathrm{T}} \boldsymbol{K}^{geo}_{S'} \delta \bar{\boldsymbol{u}}} \\ & - \overbrace{\int_{\mathcal{B}} \nabla^{\boldsymbol{X}} \boldsymbol{\eta} : \delta(JB^{\mathrm{DEM}} \boldsymbol{F}^{\text{-T}}) p^f \, \mathrm{dV}}^{\bar{\boldsymbol{\eta}}^{\mathrm{T}} \boldsymbol{K}^{geo}_{p^f} \delta \bar{\boldsymbol{u}}} - \overbrace{\int_{\mathcal{B}} JB^{\mathrm{DEM}} \, \nabla^{\boldsymbol{X}} \boldsymbol{\eta} : \boldsymbol{F}^{\text{-T}} \delta p^f \, \mathrm{dV}}^{\bar{\boldsymbol{\eta}}^{\mathrm{T}} \boldsymbol{K}^{up} \delta \bar{\boldsymbol{p}}^f} \\ & - \overbrace{\int_{\mathcal{B}} \rho_f \, \nabla^{\boldsymbol{X}} \cdot (J\boldsymbol{F}^{\text{-1}} \cdot \boldsymbol{\delta u}) \boldsymbol{\eta} \cdot \boldsymbol{g} \, \mathrm{dV}}^{\bar{\boldsymbol{\eta}}^{\mathrm{T}} \delta \boldsymbol{G}^1} - \overbrace{\int_{\partial \mathcal{B}_t} \boldsymbol{\eta} \cdot \boldsymbol{\delta \bar{t}} \, \mathrm{d}\Gamma}^{\bar{\boldsymbol{\eta}}^{\mathrm{T}} \delta \boldsymbol{F}^1_{ext}} = 0 \end{aligned} \tag{2.38}$$

where $C^{SE}_{IJKL} = \frac{\partial S'_{IJ}}{\partial E_{KL}}$ is the material tangential stiffness. $\boldsymbol{\delta E}$ is the variation of the Green-Lagrange strain tensor and $\boldsymbol{\delta E} = \frac{1}{2}[(\nabla^{\boldsymbol{X}} \boldsymbol{\delta u})^{\mathrm{T}} \boldsymbol{F} + \boldsymbol{F}^{\mathrm{T}} (\nabla^{\boldsymbol{X}} \boldsymbol{\delta u})]$. $\boldsymbol{K}^{geo}_{S'}$ and $\boldsymbol{K}^{geo}_{p^f}$ are the initial stress and initial pore pressure contributions to the geometrical stiffness. For the balance of mass equation, the corresponding linearization term reads,

$$
\begin{aligned}
\delta H(\boldsymbol{u}, p^f, \psi) = & \overbrace{\int_{\mathcal{B}} \psi \delta (B^{\mathrm{DEM}} \boldsymbol{F}^{\text{-T}}) : \nabla^{\boldsymbol{X}}\, \dot{\boldsymbol{u}}\, \mathrm{dV}}^{\bar{\boldsymbol{\psi}}^{\mathrm{T}} \delta \boldsymbol{C}_{\boldsymbol{1}} \dot{\bar{\boldsymbol{u}}}} + \overbrace{\int_{\mathcal{B}} \psi B^{\mathrm{DEM}} \boldsymbol{F}^{\text{-T}} : \nabla^{\boldsymbol{X}}\, \boldsymbol{\delta} \dot{\boldsymbol{u}}\, \mathrm{dV}}^{\bar{\boldsymbol{\psi}}^{\mathrm{T}} \boldsymbol{C}_{\boldsymbol{1}} \delta \dot{\bar{\boldsymbol{u}}}} \\
& + \overbrace{\int_{\mathcal{B}} \psi \delta (\frac{1}{M^{\mathrm{DEM}}}) \dot{p^f}\, \mathrm{dV}}^{\bar{\boldsymbol{\psi}}^{\mathrm{T}} \delta \boldsymbol{C}_{\boldsymbol{2}} \dot{\bar{\boldsymbol{p}}}^{f}} + \overbrace{\int_{\mathcal{B}} \psi \frac{1}{M^{\mathrm{DEM}}} \delta \dot{p^f}\, \mathrm{dV}}^{\bar{\boldsymbol{\psi}}^{\mathrm{T}} \boldsymbol{C}_{\boldsymbol{2}} \delta \dot{\bar{\boldsymbol{p}}}^{f}} + \overbrace{\int_{\mathcal{B}} \nabla^{\boldsymbol{X}}\, \psi\, \alpha_{stab}\, \nabla^{\boldsymbol{X}}\, \delta \dot{p^f} \mathrm{dV}}^{\bar{\boldsymbol{\psi}}^{\mathrm{T}} \boldsymbol{C}_{\boldsymbol{stab}} \delta \dot{\bar{\boldsymbol{p}}}^{f}} \\
& + \overbrace{\int_{\mathcal{B}} \nabla^{\boldsymbol{X}}\, \psi \cdot \boldsymbol{K}^{\mathrm{DEM}} \cdot \nabla^{\boldsymbol{X}}\, \delta p^f\, \mathrm{dV}}^{\bar{\boldsymbol{\psi}}^{\mathrm{T}} \boldsymbol{K}^{\boldsymbol{p}} \delta \bar{\boldsymbol{p}}^{f}} + \overbrace{\int_{\mathcal{B}} \nabla^{\boldsymbol{X}}\, \psi \cdot \delta \boldsymbol{K}^{\mathrm{DEM}} \cdot \nabla^{\boldsymbol{X}}\, p^f\, \mathrm{dV}}^{\bar{\boldsymbol{\psi}}^{\mathrm{T}} \boldsymbol{K}_{\boldsymbol{1}}^{\boldsymbol{p}} \delta \bar{\boldsymbol{u}}} \\
& - \overbrace{\int_{\mathcal{B}} \nabla^{\boldsymbol{X}}\, \psi \cdot \delta (\boldsymbol{K}^{\mathrm{DEM}} \cdot \rho_f \boldsymbol{F}^{\mathrm{T}}) \cdot \boldsymbol{g}\, \mathrm{dV}}^{\bar{\boldsymbol{\psi}}^{\mathrm{T}} \delta \boldsymbol{G}^{\boldsymbol{2}}} - \overbrace{\int_{\partial B_Q} \psi \delta \overline{\boldsymbol{Q}}\, \mathrm{d}\Gamma}^{\bar{\boldsymbol{\psi}}^{\mathrm{T}} \delta \boldsymbol{F}_{\boldsymbol{ext}}^{\boldsymbol{2}}} = 0
\end{aligned}
\tag{2.39}
$$

where $\boldsymbol{K}_{\boldsymbol{1}}^{\boldsymbol{p}}$ is the geometrical term related to the permeability k.

The proposed semi-implicit scheme splits $G(\boldsymbol{u}, p^f, \boldsymbol{\eta})$ and $H(\boldsymbol{u}, p^f, \psi)$ into implicitly treated parts and explicitly treated parts, thus only a subset of the linearization terms in (2.38) and (2.39) will be used. The implicit-explicit split will be explained in the next section.

2.4 Semi-implicit multiscale time integrator

While both implicit and explicit time integrators have been used in DEM-FEM coupling models for dry granular materials [143, 88, 132], the extension of these algorithms to multiphysics hydro-mechanical problem is not straightforward. The key difference is that the pore-fluid diffusion is transient and hence the initial boundary value problem is elliptic.

While it is possible to add the inertial terms and update the macroscopic displacement and pore pressure explicitly via a dynamics relaxation procedure, this strategy is impractical due to the small critical time step size of the explicit scheme as pointed out by [177]. Another possible approach is to solve the macroscopic problem in a fully implicit, unconditionally stable scheme. The drawback of this approach is that it requires additional CPU time to compute the elasto-plastic tangential stiffness from DEM. Unlike a conventional constitutive model (in which an analytical expression of the tangential stiffness is often available and hence easy to implement), the tangential stiffness inferred from DEM must be obtained numerically via perturbation methods [88, 33]. For three-dimensional simulations, this means that additional 36 to 81 simulations are required to obtain the tangential stiffness, depending on which energy-conjugate stress-strain pair is used in the formulation. This is a sizable burden given the fact that a converged update may require tens of iterations.

To avoid this additional computational cost, we adopt the implicit-explicit predictor-multicorrector scheme originally proposed in [99] and [177] and apply it to the finite strain DEM-mixed-FEM model. In [177], the internal force is split into two components, one treated implicitly and another treated explicitly. We adopt this idea here by treating the elasto-plastic force from DEM explicitly, and the other internal forces implicitly, in a fashion similar to the unconditionally stable Yanenko operator splitting (i.e. $\mathcal{L} = \mathcal{L}_{exp} + \mathcal{L}_{imp}$, c.f. [262]).

The implicit time integration based on the generalized trapezoidal rule consists of satisfying the equation (2.37) at time t_{n+1}:

$$\boldsymbol{M}^{*}_{n+1} \cdot \boldsymbol{v}_{n+1} + \boldsymbol{F}^{int}(\boldsymbol{d}_{n+1}) - \boldsymbol{G}(\boldsymbol{d}_{n+1}) = \boldsymbol{F}^{ext}_{n+1} \tag{2.40}$$

with the solution

$$\boldsymbol{d}_{n+1} = \tilde{\boldsymbol{d}} + \alpha \Delta t \boldsymbol{v}_{n+1} \tag{2.41}$$

where

$$\tilde{\boldsymbol{d}} = \boldsymbol{d}_{\boldsymbol{n}} + (1-\alpha)\Delta t \boldsymbol{v}_{\boldsymbol{n}}. \tag{2.42}$$

The notation is as follows: the subscripts n and $n+1$ denote that the variables are evaluated at time t_n and t_{n+1}, respectively; Δt is the time step; α is the integration parameter. The quantity $\tilde{\boldsymbol{d}}$ is referred to as the predicted solution.

Similar to the scheme of [177], the semi-implicit predictor-corrector scheme is performed by evaluating a portion of the left hand side forces of (2.40) explicitly using the predicted solution $\tilde{\boldsymbol{d}}$ and $\boldsymbol{v}_{\boldsymbol{n}}$, and treating the remaining portion implicitly with the solution $\boldsymbol{d}_{\boldsymbol{n+1}}$ and $\boldsymbol{v}_{\boldsymbol{n+1}}$:

$$\boldsymbol{F}_{\text{IMP}}(\boldsymbol{v}_{\boldsymbol{n+1}}, \boldsymbol{d}_{\boldsymbol{n+1}}) + \boldsymbol{F}_{\text{EXP}}(\boldsymbol{v}_{\boldsymbol{n}}, \tilde{\boldsymbol{d}}) = \boldsymbol{F}^{\boldsymbol{ext}}_{\boldsymbol{n+1}} \tag{2.43}$$

where

$$\begin{cases} \boldsymbol{F}_{\text{IMP}} = \{\boldsymbol{M}^* \cdot \boldsymbol{v}\}^{\text{implicit}}_{\boldsymbol{n+1}} + \left\{\boldsymbol{F}^{\boldsymbol{int}}(\boldsymbol{d}_{\boldsymbol{n+1}})\right\}^{\text{implicit}} \\ \boldsymbol{F}_{\text{EXP}} = \{\boldsymbol{M}^* \cdot \boldsymbol{v}\}^{\text{explicit}}_{\boldsymbol{n}} + \left\{\boldsymbol{F}^{\boldsymbol{int}}(\tilde{\boldsymbol{d}})\right\}^{\text{explicit}} - \boldsymbol{G}(\tilde{\boldsymbol{d}}) \end{cases} \tag{2.44}$$

To obtain the macroscopic displacement and pore pressure at time t_{n+1} from the nonlinear equation system (2.43), Newton-Raphson iteration method is employed. Let us denote the corrected solutions as $\boldsymbol{d}^{\boldsymbol{j}}_{\boldsymbol{n+1}}$ and $\boldsymbol{v}^{\boldsymbol{j}}_{\boldsymbol{n+1}}$, at the time step $n+1$ and jth iteration, i.e.,

$$\boldsymbol{d}^{\boldsymbol{j}}_{\boldsymbol{n+1}} = \tilde{\boldsymbol{d}} + \alpha \Delta t \boldsymbol{v}^{\boldsymbol{j}}_{\boldsymbol{n+1}} \tag{2.45}$$

with $\boldsymbol{v}^{\boldsymbol{0}}_{\boldsymbol{n+1}} = \boldsymbol{0}$. The relationship of their increments is thus:

$$\Delta \boldsymbol{d}^{\boldsymbol{j}}_{\boldsymbol{n+1}} = \alpha \Delta t \Delta \boldsymbol{v}^{\boldsymbol{j}}_{\boldsymbol{n+1}} \tag{2.46}$$

The equation (2.43) in terms of these iterative solutions is written as:

$$\boldsymbol{F}_{\text{IMP}}(\boldsymbol{v}^{\boldsymbol{j+1}}_{\boldsymbol{n+1}}, \boldsymbol{d}^{\boldsymbol{j+1}}_{\boldsymbol{n+1}}) + \boldsymbol{F}_{\text{EXP}}(\boldsymbol{v}^{\boldsymbol{j}}_{\boldsymbol{n+1}}, \boldsymbol{d}^{\boldsymbol{j}}_{\boldsymbol{n+1}}) = \boldsymbol{F}^{\boldsymbol{ext}}_{\boldsymbol{n+1}} \tag{2.47}$$

The consistent linearization of the implicit part $\boldsymbol{F}_{\text{IMP}}$ is required to solve (2.47). The resulting tangential stiffness matrix depends on what force terms are included in $\{\boldsymbol{M}^* \cdot \boldsymbol{v}\}^{\text{implicit}}$ and $\left\{\boldsymbol{F}^{\boldsymbol{int}}\right\}^{\text{implicit}}$.

For the implicit-explicit split of the nonlinear rate of change term $\boldsymbol{M}^* \cdot \boldsymbol{v}$, note that, from (2.38) and (2.39), its variation contains two components:

$$\frac{\partial(\boldsymbol{M}^* \cdot \boldsymbol{v})}{\partial \boldsymbol{v}} \cdot \delta \boldsymbol{v} = (\boldsymbol{M}^* + \frac{\partial \boldsymbol{M}^*}{\partial \boldsymbol{v}} \cdot \boldsymbol{v}) \cdot \delta \boldsymbol{v} \tag{2.48}$$

In the proposed scheme, the rate of change term is split in a way that only $\boldsymbol{M}^*$ is treated implicitly:

$$\frac{\partial(\boldsymbol{M}^* \cdot \boldsymbol{v})}{\partial \boldsymbol{v}} = \left\{\frac{\partial \boldsymbol{M}^* \cdot \boldsymbol{v}}{\partial \boldsymbol{v}}\right\}^{\text{implicit}} + \left\{\frac{\partial \boldsymbol{M}^* \cdot \boldsymbol{v}}{\partial \boldsymbol{v}}\right\}^{\text{explicit}} \quad ; \quad \begin{cases} \left\{\dfrac{\partial \boldsymbol{M}^* \cdot \boldsymbol{v}}{\partial \boldsymbol{v}}\right\}^{\text{implicit}} & = \boldsymbol{M}^* \\ \left\{\dfrac{\partial \boldsymbol{M}^* \cdot \boldsymbol{v}}{\partial \boldsymbol{v}}\right\}^{\text{explicit}} & = \dfrac{\partial \boldsymbol{M}^*}{\partial \boldsymbol{v}} \cdot \boldsymbol{v} \end{cases} \tag{2.49}$$

From the above implicit-explicit split, the first order linearization form of $\boldsymbol{F}_{\text{IMP}}$ in (2.47) reads:

$$\begin{aligned} \boldsymbol{F}_{\text{IMP}}(\boldsymbol{v}_{n+1}^{j+1}, \boldsymbol{d}_{n+1}^{j+1}) &\simeq \boldsymbol{F}_{\text{IMP}}(\boldsymbol{v}_{n+1}^{j}, \boldsymbol{d}_{n+1}^{j}) + \boldsymbol{M}^* \cdot \Delta \boldsymbol{v}_{n+1}^{j+1} + \boldsymbol{K}_{\boldsymbol{T}}^{\text{implicit}} \cdot \Delta \boldsymbol{d}_{n+1}^{j+1} \\ &\simeq \boldsymbol{F}_{\text{IMP}}(\boldsymbol{v}_{n+1}^{j}, \boldsymbol{d}_{n+1}^{j}) + \underbrace{[\boldsymbol{M}^* + \alpha \Delta t \boldsymbol{K}_{\boldsymbol{T}}^{\text{implicit}}]}_{\boldsymbol{M}^{**}} \cdot \Delta \boldsymbol{v}_{n+1}^{j+1} \end{aligned} \tag{2.50}$$

where

$$\boldsymbol{K}_{\boldsymbol{T}}{}^{\text{implicit}} \cdot \Delta \boldsymbol{d}_{n+1}^{j+1} = \frac{d}{d\beta} \left\{\boldsymbol{F}^{\boldsymbol{int}}(\boldsymbol{d}_{n+1}^{j} + \beta \Delta \boldsymbol{d}_{n+1}^{j+1})\right\}^{\text{implicit}} |_{\beta=0} \tag{2.51}$$

is the directional derivative of $\left\{\boldsymbol{F}^{\boldsymbol{int}}\right\}^{\text{implicit}}$ at $\boldsymbol{d}_{n+1}^{j}$ in the direction of $\Delta \boldsymbol{d}_{n+1}^{j+1}$.

For construction of $\boldsymbol{K}_{\boldsymbol{T}}{}^{\text{implicit}}$, firstly, a complete linearization of the internal force $\boldsymbol{F}^{\boldsymbol{int}}$ results in the following form of tangential stiffness matrix, according to (2.38) and (2.39):

$$\boldsymbol{K}_{\boldsymbol{T}} = \frac{\partial \boldsymbol{F}^{\boldsymbol{int}}}{\partial \boldsymbol{d}} = \begin{bmatrix} (\underbrace{\boldsymbol{K}^{e} - \boldsymbol{K}^{ep}}_{\boldsymbol{K}^{s}} + \boldsymbol{K}^{geo}) & -\boldsymbol{K}^{up} \\ \boldsymbol{K}_{\boldsymbol{1}}^{p} & \boldsymbol{K}^{p} \end{bmatrix} \tag{2.52}$$

where $\boldsymbol{K^e}$ is the elastic contribution and $\boldsymbol{K^{ep}}$ is the non-linear elastic-plastic contribution to the material tangential stiffness $\boldsymbol{K^s}$. $\boldsymbol{K^{geo}}$ represents the sum of the geometrical stiffness $\boldsymbol{K^{geo}_{S'}}$ and $\boldsymbol{K^{geo}_{p^f}}$.

Since computation of the homogenized $\boldsymbol{K^s}$ from DEM RVEs produces considerable computational cost, in the proposed multiscale solution scheme, we choose to treat $\boldsymbol{K^e}$ implicitly and $\boldsymbol{K^{ep}}$ explicitly. $\boldsymbol{K^e}$ is thus evaluated at the initial time step using the elastic properties (bulk modulus K^{DEM}_{bulk} and shear modulus G^{DEM}_{shear}) homogenized from the initial RVEs. $\boldsymbol{K^{up}}$ and $\boldsymbol{K^p}$ are included in the implicit part of the tangential stiffness matrix. The geometrical terms $\boldsymbol{K^{geo}}$ and $\boldsymbol{K^p_1}$ are treated explicitly. With these considerations and (2.52), the resulting operator split writes:

$$\begin{cases} \boldsymbol{K_T} = \boldsymbol{K_T}^{\text{implicit}} + \boldsymbol{K_T}^{\text{explicit}} \\ \boldsymbol{K_T}^{\text{implicit}} = \dfrac{\partial \left\{ \boldsymbol{F^{int}} \right\}^{\text{implicit}}}{\partial \boldsymbol{d}} = \begin{bmatrix} \boldsymbol{K^e} & -\boldsymbol{K^{up}} \\ \boldsymbol{0} & \boldsymbol{K^p} \end{bmatrix} \\ \boldsymbol{K_T}^{\text{explicit}} = \begin{bmatrix} -\boldsymbol{K^{ep}} + \boldsymbol{K^{geo}} & \boldsymbol{0} \\ \boldsymbol{K^p_1} & \boldsymbol{0} \end{bmatrix} \end{cases} \tag{2.53}$$

From equations (2.44), (2.47), (2.49), (2.50) and (2.53), we obtain equation (2.54), which represents the iteration equation of the semi-implicit predictor-multicorrector scheme:

$$\boldsymbol{M}^{**} \cdot \Delta \boldsymbol{v}^{j+1}_{n+1} = \Delta \boldsymbol{F}^{j}_{\boldsymbol{n+1}} = \boldsymbol{F}^{\boldsymbol{ext}}_{\boldsymbol{n+1}} - \boldsymbol{M}^{*} \cdot \boldsymbol{v}^{j}_{n+1} - \boldsymbol{F^{int}}(\boldsymbol{d}^{j}_{\boldsymbol{n+1}}) - \boldsymbol{G}(\boldsymbol{d}^{j}_{\boldsymbol{n+1}}) \tag{2.54}$$

where the internal force $\boldsymbol{F^{int}}(\boldsymbol{d}^j_{\boldsymbol{n+1}})$ has two contributions: the PK2 effective stresses which are homogenized from DEM RVEs and the macroscopic internal force from FEM, i.e.,

$$\boldsymbol{F^{int}}(\boldsymbol{d}^j_{\boldsymbol{n+1}}) = \begin{Bmatrix} \boldsymbol{f^{int}}(\boldsymbol{u}^j_{\boldsymbol{n+1}}) \\ \boldsymbol{0} \end{Bmatrix}^{\text{DEM}} + \left\{ \boldsymbol{F^{int}}(\boldsymbol{d}^j_{\boldsymbol{n+1}}) \right\}^{\text{FEM}} \tag{2.55}$$

ϕ^{DEM}, B^{DEM}, M^{DEM} and k^{DEM} are homogenized at each time step to construct the tangential matrix $\boldsymbol{M}^{**}$. The convergence is achieved when $\frac{||\Delta \boldsymbol{F}^j_{\boldsymbol{n+1}}||}{||\Delta \boldsymbol{F}^{\boldsymbol{0}}_{\boldsymbol{n+1}}||} \leq$ TOL [177]. In the numerical examples TOL is equal to 10^{-4}. To recapitulate and illustrate the multiscale semi-implicit scheme, we present a flowchart as shown in Fig.2.2.

2.5 Numerical Examples

The objective of this section is to demonstrate the versatility and accuracy of the proposed method in both the small and finite deformation ranges. The numerical examples in this section are the representative problems commonly encountered in geotechnical engineering. The first example is a globally undrained shear test which examines how granular motion altered by fluid seepage within a soil specimen affects the macroscopic responses. In the second example, we simulate the responses of a cylindrical DEM-FEM model with drained condition subjected to triaxial compression loading with both quarter- and full domains and found that the quarter simulation may suppress the non-symmetric bifurcation mode that leads to shear band. The analysis on fabric tensor also reveals that the fabric and deviatoric stress tensors are almost co-axial inside the shear band, but they are not co-axial in the host matrix.

2.5.1 Globally undrained shear test of dense and loose assemblies

For the first example we employ our multiscale scheme to perform shear tests on both dense and loose granular assemblies. The macroscopic geometry and boundary conditions are illustrated on a sample discretized by coarse mesh ($1\times5\times5$ in X,Y,Z directions) as Fig. 2.3. We also use a medium fine mesh ($1\times8\times8$) and a fine mesh ($1\times10\times10$) to investigate the mesh dependency issue of the proposed scheme. All results in this section are computed from the fine mesh model, if not specified. The nodes on the bottom boundary are fixed in all directions and those on the upper boundary are translated identically towards the

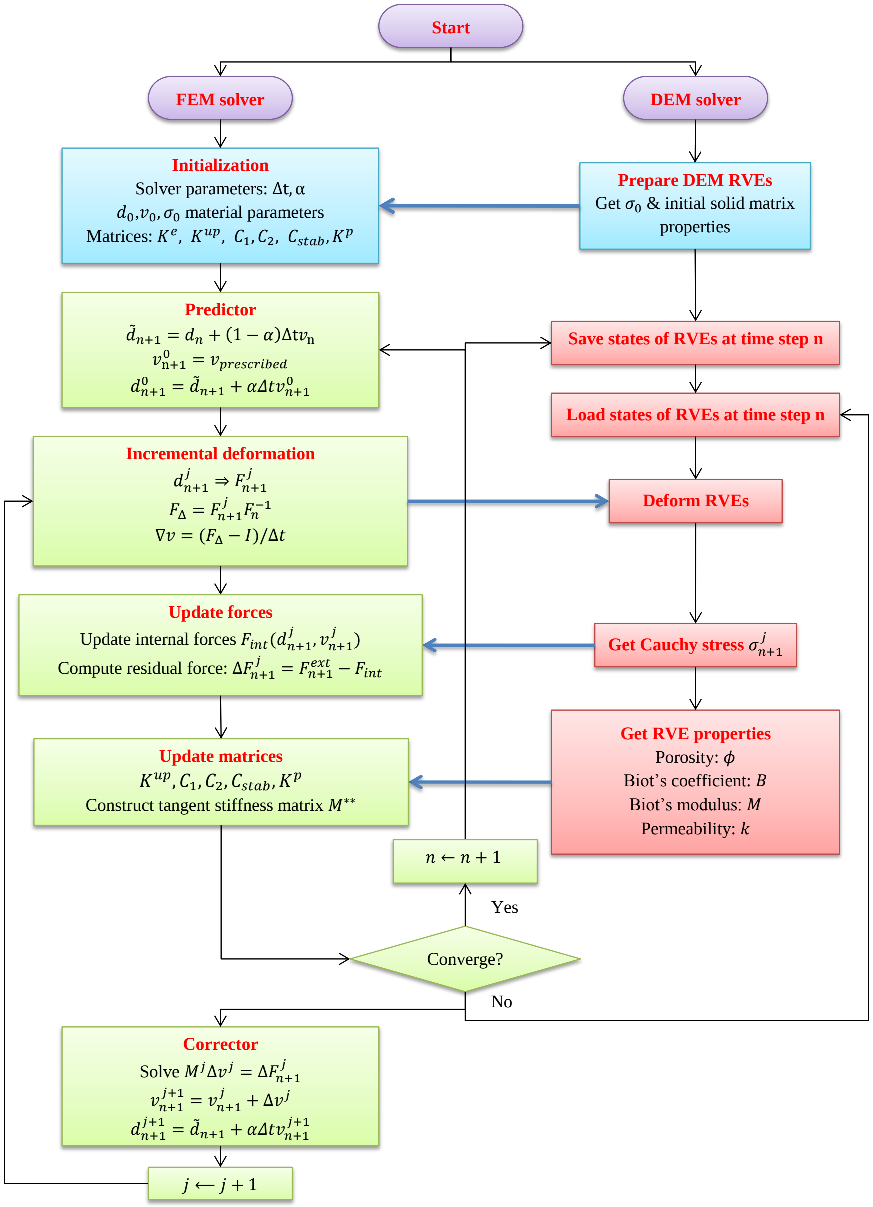

Start
FEM solver
DEM solver
Initialization
Solver parameters: $\Delta t, \alpha$
d_0, v_0, σ_0 material parameters
Matrices: K^e, K^{up}, C_1, C_2, C_{stab}, K^p
Prepare DEM RVEs
Get σ_0 & initial solid matrix properties
Predictor
$\tilde{d}_{n+1} = d_n + (1-\alpha)\Delta t v_n$
$v^0_{n+1} = v_{prescribed}$
$d^0_{n+1} = \tilde{d}_{n+1} + \alpha \Delta t v^0_{n+1}$
Save states of RVEs at time step n
Load states of RVEs at time step n
Incremental deformation
$d^j_{n+1} \Rightarrow F^j_{n+1}$
$F_\Delta = F^j_{n+1} F^{-1}_n$
$\nabla v = (F_\Delta - I)/\Delta t$
Deform RVEs
Update forces
Update internal forces $F_{int}(d^j_{n+1}, v^j_{n+1})$
Compute residual force: $\Delta F^j_{n+1} = F^{ext}_{n+1} - F_{int}$
Get Cauchy stress σ^j_{n+1}
Update matrices
$K^{up}, C_1, C_2, C_{stab}, K^p$
Construct tangent stiffness matrix M^{**}
Get RVE properties
Porosity: ϕ
Biot's coefficient: B
Biot's modulus: M
Permeability: k
$n \leftarrow n+1$
Yes
Converge?
No
Corrector
Solve $M^j \Delta v^j = \Delta F^j_{n+1}$
$v^{j+1}_{n+1} = v^j_{n+1} + \Delta v^j$
$d^{j+1}_{n+1} = \tilde{d}_{n+1} + \alpha \Delta t v^{j+1}_{n+1}$
$j \leftarrow j+1$

Figure 2.2: Flowchart of the multiscale semi-implicit scheme. Blue blocks represent initialization steps of the solution scheme ; Green blocks refer to FEM solver steps and red blocks refer to DEM solver steps; Blue arrows indicate the information flow between the two solvers.

positive y axis at a constant rate. They are maintained at a constant vertical stress $\sigma_z = 100kPa$ by a horizontal rigid layer (not shown). This constraint is imposed in the model by the Lagrange multiplier method. The lateral surfaces are constrained by frictionless rigid walls (not shown). All surfaces are impervious. The gravitational effect is not considered in this study. For coupled microscopic DEM models, periodic unit cells composed of uniform spheres are prepared by an isotropic compression engine in YADE up to $\sigma_{iso} = 100kPa$ with initial porosity of 0.375 and 0.427 for dense and loose assemblies respectively, and then are assigned identically to all the integration points of the FEM model before shearing.

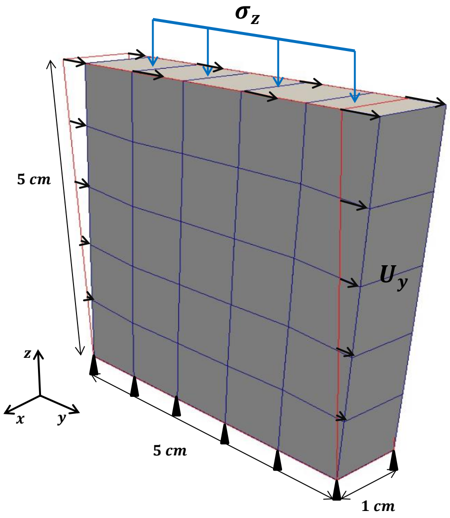


Figure 2.3: Geometry and boundary conditions for globally undrained shear test

The finite strain formulation is first adopted to study the hydro-mechanical coupling effect during the shearing of the dense and loose samples with undrained boundaries. The material parameters used in the simulations allowing hydraulic diffusion within the specimen are presented in Table 2.1. They are categorized into micromechanical material

Parameter	Value
Solid grain normal stiffness k_n	$2.2 \times 10^6\ N/m$
Solid grain tangential stiffness k_s	$1.9 \times 10^6\ N/m$
Solid grain friction angle β	30°
Solid grain bulk modulus K_s	0.33 GPa
Porosity ϕ	dense: 0.375, loose: 0.427
Biot's coefficient B	dense: 0.976, loose: 0.983
Biot's Modulus M	dense: 180 Mpa, loose: 168 Mpa
Fluid bulk modulus K_f	0.1 GPa
Initial permeability k	$1 \times 10^{-9}\ m^2/(Pa \cdot s)$
Solid density ρ_s	$2700\ kg/m^3$
Fluid density ρ_f	$1000\ kg/m^3$

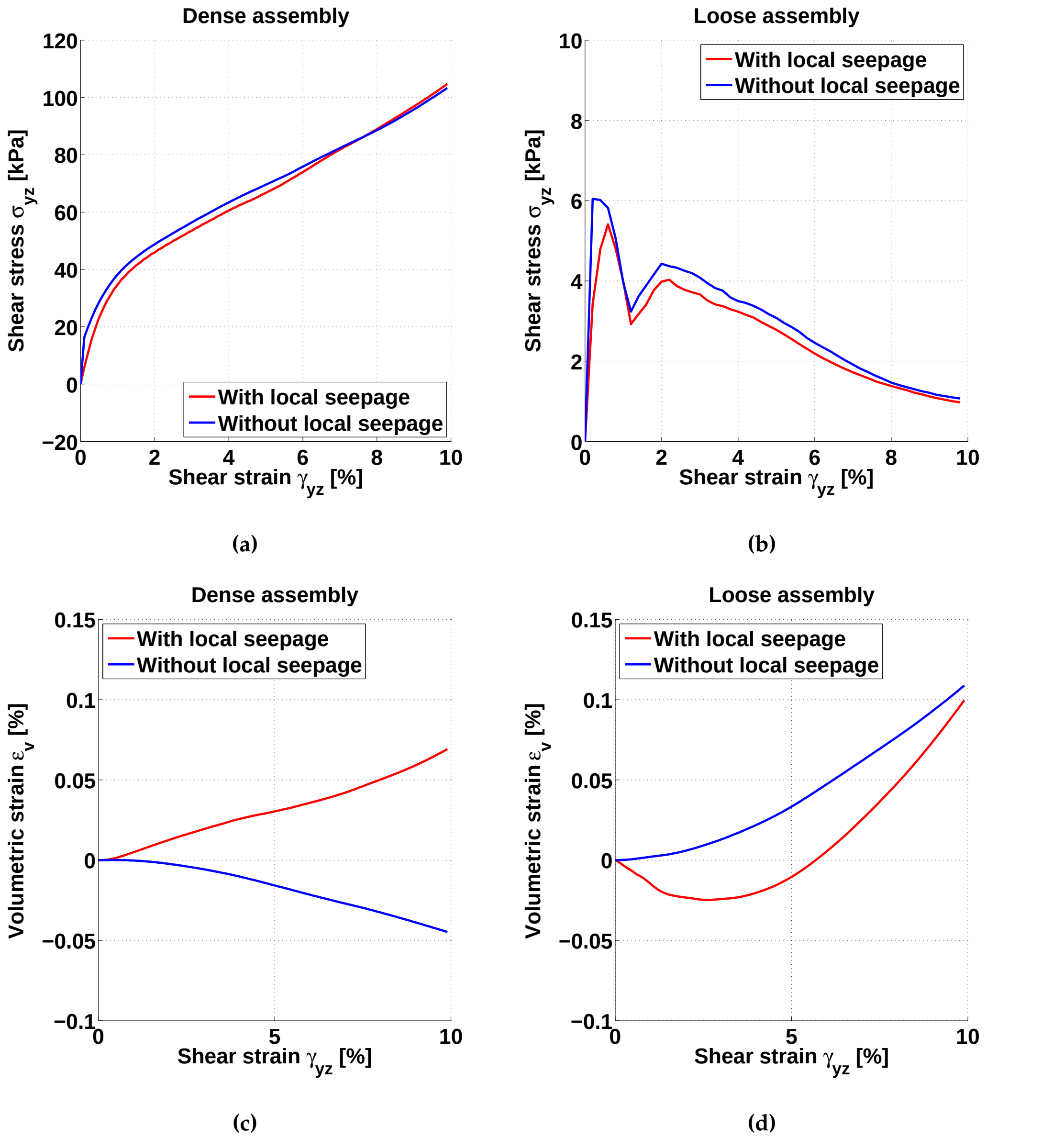
Table 2.1: Material parameters in globally undrained shear problem

parameters used in DEM solver, poro and poro-plasticity parameters derived from DEM RVEs and macroscopic properties set in FEM. Note that the permeability k is updated with porosity of RVEs using the Kozeny-Carman relation during the simulation. To prevent local seepage of water within the samples, the permeability k is set to 0 $m^2/(Pa \cdot s)$.

Fig. 2.4 represents the global shear stress and volumetric strain behavior of shear simulations with and without local seepage of water. The strain hardening behavior of undrained dense granular assemblies (left column) and strain softening behavior of loose granular assemblies (right column) are recovered [264]. In both assemblies, when local seepage is prohibited, the shear stress immediately rises when the shearing begins and the saturated porous media behaves stiffer than the samples with local seepage. Note that the sudden drop in Fig. 2.4(b) is due to the unstable solid matrix of loosely confined DEM unit cell. The volumetric strain of the dense sample with seepage monotonically increases. This phenomenon is attributed to the rearrangement of solid matrix as the grains tend to rise over adjacent grains when they are driven by shear forces. In absence of local diffusion, the dense sample experiences a reduction of volume instead, suggesting that the compression of overall solid matrix predominates the above phenomenon. As for loose samples, however, the volumetric behavior is opposite. When local diffusion of water is prohibited, the pore collapse and densification of local regions within specimen could occur, resulting in a compression at early stage of shearing before the dilatancy phenomenon. The curve of no-local-seepage case shows that the dilatancy phenomenon prevails all along the shearing. In all cases, the volume changes are beneath 0.12%, confirming that the samples are indeed sheared under globally undrained condition.

We examine the mesh dependency by three aforementioned mesh densities adopted in simulations of dense assembly with local seepage. The effect is presented via plots of global $\sigma_{yz} - \gamma_{yz}$ and $\varepsilon_v - \gamma_{yz}$ responses as Fig. 2.5. For stress response, discrepancy between medium and fine meshes is not significant, but coarse mesh apparently yields stiffer solution after 2% shear strain and the maximum deviation is about 7.6% with respect to the fine mesh solution. The differences between ε_v curves are less significant and do not exceed 4% of the fine mesh solution. Thus, our choice of the fine mesh to conduct numerical experiments is acceptable.

We next display the difference between finite and small strain multiscale schemes for simulations of dense granular sample in both local diffusion conditions in Fig. 2.6. According to the global shear responses, the small strain and finite strain yield consistent solutions within 2% shear strain. Then the discrepancy gradually emerges and the introduction of geometrical non-linearity renders the sample stiffer. Finite strain solutions exhibit less volume changes in both cases. Moreover, geometrical non-linear term even



(a)

(b)

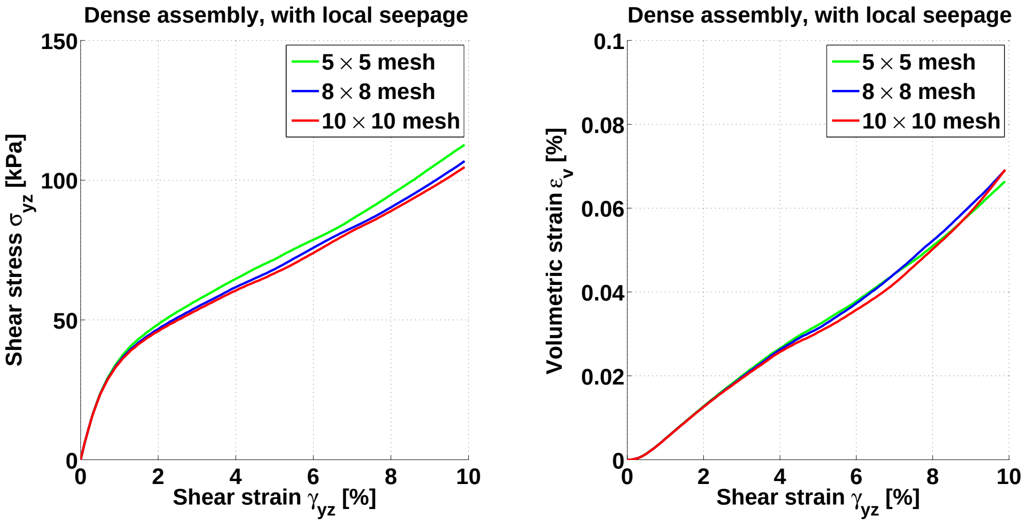
(c)

(d)

Figure 2.4: Comparison of global shear stress and volumetric strain behavior between globally undrained dense and loose assemblies with and without local diffusion

alters the dilatancy behavior: the sample is computed to be compressed when no local seepage of water is allowed, while the small strain solution conserves the dilatant trend.

We also assess the local diffusion effect via color maps of pore pressure developed during the deformation, as shown in Fig. 2.7. The dense sample with local seepage has



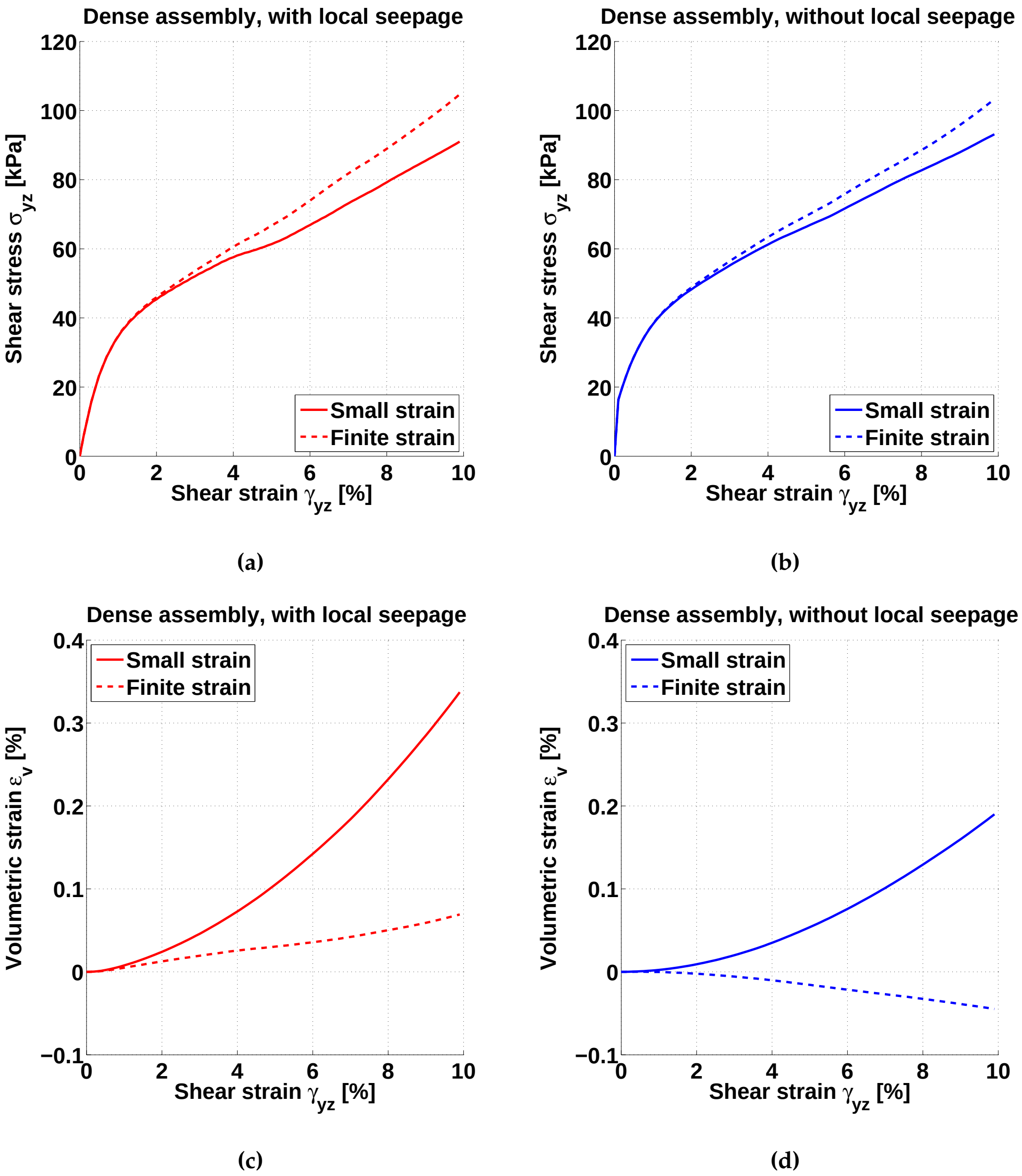
(a)

(b)

Figure 2.5: Comparison of global shear stress and global volumetric strain behavior between coarse mesh (1×5×5), medium mesh (1×8×8), fine mesh (1×10×10), finite strain formulation

developed negative pore pressure and the pressure distribution is nearly uniform, since fluid flow could take place inside the specimen to dissipate pressure difference between neighboring pores. Without local seepage of water, the pore pressure is concentrated to four corners of the sample, with the upper left and bottom right corners compressed (positive pressure) and the other two dilated (negative pressure). Furthermore, these corners have maximum pressure gradient $||\nabla p^f||$.

The multiscale nature of our method offers more insight into the local states of granular sample. With the granular material behavior homogenized from responses of RVEs, the grain displacements, the effective stress paths (shear stress $q = \sigma_1 - \sigma_3$ vs. effective mean stress $p' = \frac{\sigma_1+\sigma_2+\sigma_3}{3}$) and the volumetric strain paths (ε_v vs. p') in each DEM unit cell are directly accessible. As an example, the local distribution of q at the end of shearing for globally undrained yet locally diffused dense sample (2.8) shows a concentration



(a)

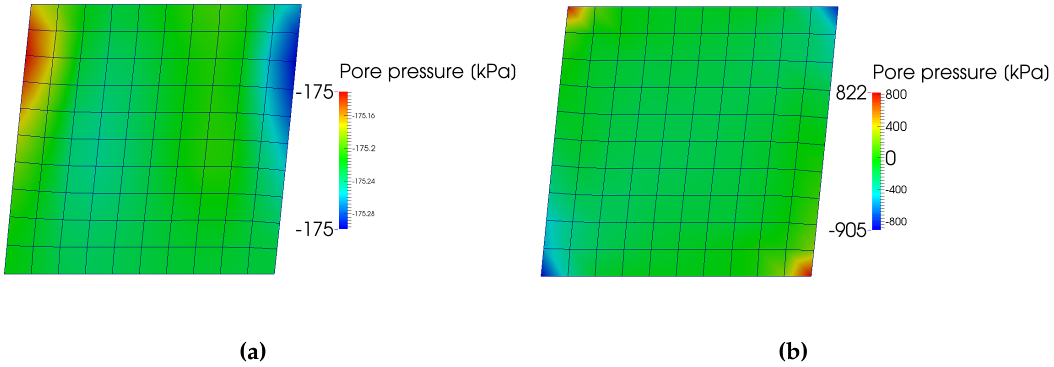
(b)

(c)

(d)

Figure 2.6: Comparison of global shear stress and global volumetric strain behavior between small strain and finite strain formulation. Left: globally undrained with local diffusion condition, Right: globally undrained but without local diffusion condition

of shear stress in upper left and bottom right corners, while the corners correspondent to the other diagonal sustain comparably very little shear stress. The deformed configuration of spheres in three representative RVEs are colored according to the dimensionless dis-



(a) **(b)**

Figure 2.7: Comparison of pore pressure at 10% shear strain between (a) dense sample with local seepage and (b) dense sample without local seepage

placement magnitude $\frac{||\boldsymbol{u}||_2}{\text{inital size of unit cell}}$ compared to initial RVE configuration. We present stress paths of these three RVEs providing evidence that strain-softening (Fig. 2.9(a)), limited strain-softening (Fig. 2.9(b)) and strain-hardening (Fig. 2.9(c)) could locally occur in a dense sample which globally behaves in a strain-hardening manner. A critical state line $q = \eta p'$ is drawn for three stress paths and the value of slope η is identified as 1.16. η and the Mohr-Coulomb friction angle β' is computed to be 29.1° by the following relation for cohesionless soil [258]:

$$\sin \beta' = \frac{3\eta}{6+\eta} \tag{2.56}$$

, which is close to the inter-particle friction angle $\beta = 30°$. Paths of ε_v further demonstrate that large local volume change up to 5.5% is possible even globally the sample is only dilated about 0.07%. According to these figures, the small strain and finite strain shear responses are almost identical. The stress path curves exhibit little difference. However, geometrical non-linearity has more significant effect on volumetric strain path. A major remark is that, inside the strain-softening spot as 2.9(d), the small strain solution has large fluctuation when the mean effective stress is very small, because DEM assemblies are highly unstable with nearly zero confining stress. On the contrary, finite strain scheme avoids this unstable regime and yield smooth solutions.

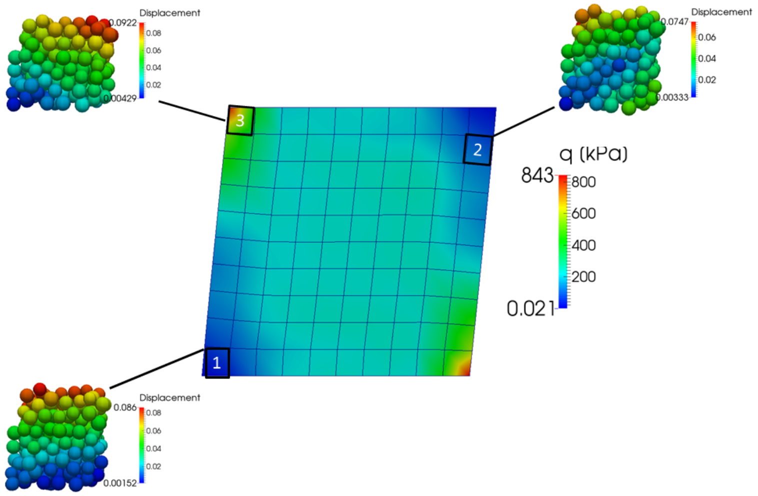
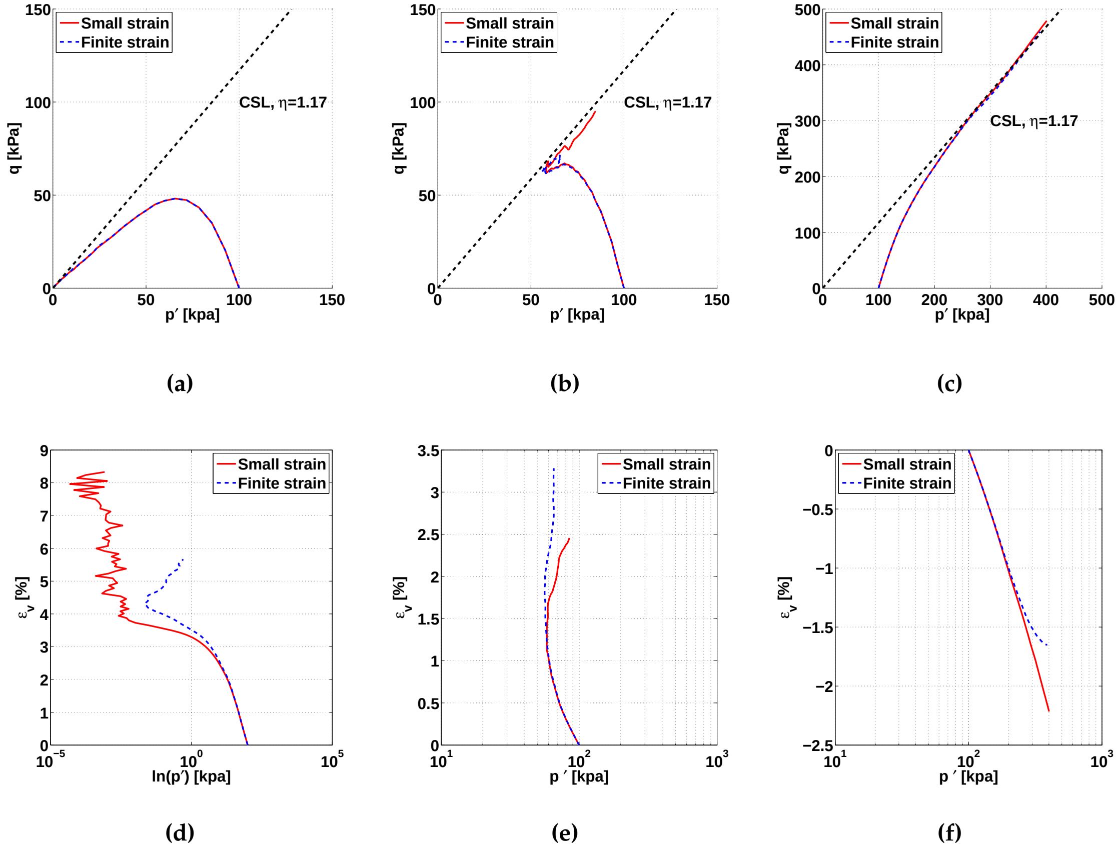


Figure 2.8: Spatial distribution of shear stress q at 10% shear strain for globally undrained dense sample allowing seepage within the specimen, attached with displacement magnitude of grains in unit cells (normalized by the initial cell size)

Lastly, we investigate the rate-dependent shearing behavior using the proposed coupling scheme. A faster shearing of saturated granular sample influences its mechanical response mainly by speeding up the solid matrix re-arrangement and also by allowing less fluid diffusion inside the sample between loading steps. The former effect leads to swelling of the sample, while the latter renders the specimen more locally undrained. Fig.2.10 illustrates the combined effect of these two mechanisms on a dense sample with local seepage. The evolution of shear stress and volumetric strain with shearing rates of 0.1% and 0.5% per second are compared. When shearing is completed, shear stress sustained by the sample increases about 4.6% under higher shearing rate. The rate effect on volumetric strain is more prominent, by the fact that the sample experiences more volume expansion of about



(a) (b) (c)

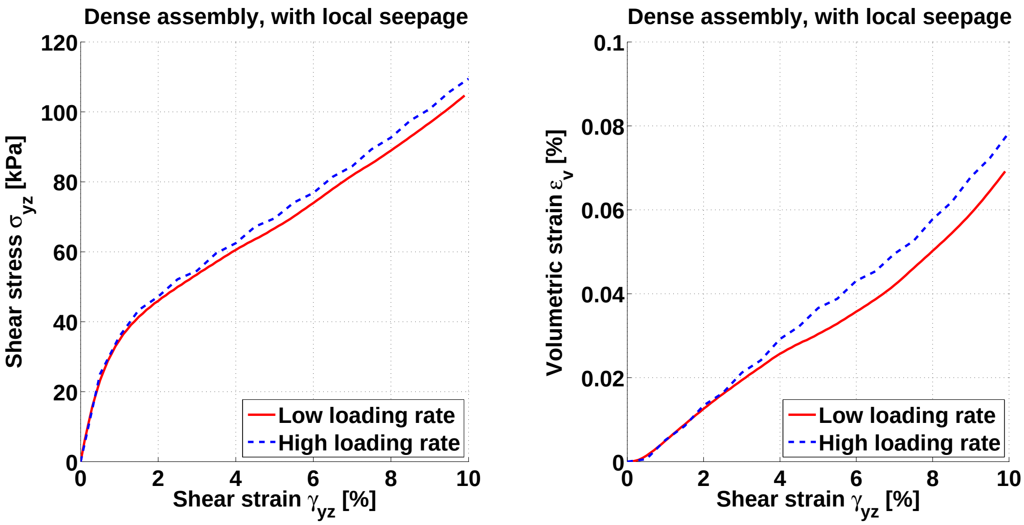
(d) (e) (f)

Figure 2.9: Shear stress vs. effective mean stress at different locations indexed as Fig. 2.8: (b) stress path at point 1 (c) stress path at point 2 (d) stress path at point 3; Volumetric strain vs. effective mean stress at different locations: (e) volume path at point 1 (f) volume path at point 2 (g) volume path at point 3

13.5% at the end.

2.5.2 Globally drained triaxial compression test

The second example consists of the globally drained triaxial compression test on an isotropically consolidated cylindrical specimen. This example demonstrates the applicability of the proposed multiscale finite strain scheme on 3D problems. In this numerical example, we analyze (1) the difference between quarter-domain and full-domain simulations for ma-



(a)

(b)

Figure 2.10: Comparison of global shear stress and global volumetric strain behavior between low loading rate (0.1% shear strain per second) and high loading rate (0.5% shear strain per second), finite strain formulation

terial subjected to axial-symmetrical loading, (2) the consequence of the build-up of excess pore pressure due to a high loading rate and (3) the evolution of the fabric tensor inside and outside the shear band and the implications on the critical state of the materials. As a result, water is allowed to flow through the bottom and the top of the specimen. However, triaxial compression simulation is intentionally not conducted under a fully drained condition at a material point level. Instead, the rate dependence of the constitutive responses introduced via the hydro-mechanical coupling effect is studied to quantify what is the acceptable range of the prescribed loading rate that can prevent significant amount of excess pore pressure.

In addition, microscopic information such as the Biot's coefficient, Biot's modulus and micro-structure fabric are provided to highlight the advantage of the DEM-FEM coupled model. The convergence profile of this simulation is also presented. In an experimental setting, the drained triaxial test is performed on a cylindrical water-saturated soil specimen, laterally enveloped by rubber membrane and drained through top and bottom surfaces. One of the idealized 3D numerical model constitutes only a quarter of the cylinder by assuming the rotational symmetry. The constant confining pressure is directly applied on the lateral surface, neglecting the effect of rubber membrane. The quasi-static compression is achieved by gradually increasing the axial strain ε_z at the rate of 0.05% per second. The lateral surface is impermeable and a no-flux boundary condition is imposed, while the pore water pressure on both top and bottom surfaces are constrained to be 0. Another simulation is triaxial compression of the full cylindrical domain. Similar confining pressure and pore pressure boundary conditions are applied. The middle point of the bottom surface is fixed to prohibit rigid body translation. The geometry, mesh and boundary conditions of the quarter-/full-domain simulations are illustrated in Fig. 2.11. The DEM assembly adopted in these simulations is identical to the dense sample in the previous section. The fluid bulk modulus in this example is 2.2 GPa.

Fig. 2.12 compares the global shear stress and volumetric strain behavior from quarter-domain and full domain simulations. The shear stress curve obtained from full-domain simulation exhibits less peak stress and more significant softening than quarter-domain simulation. The volumetric strain curves, however, only show notable difference after the axial strain approaches 7%. This discrepancy may be attributed to the strain localization in full-domain simulation, as shown by the distribution of deviatoric strain and porosity in Fig. 2.13. A dilatant shear band is developed inside the cylindrical specimen, while in the quarter-domain, the deformation is nearly homogeneous. This difference is more profound given the fact that the proposed model also incorporates the geometrical effect at the finite strain range. Results from this set of simulations show that the quarter-domain

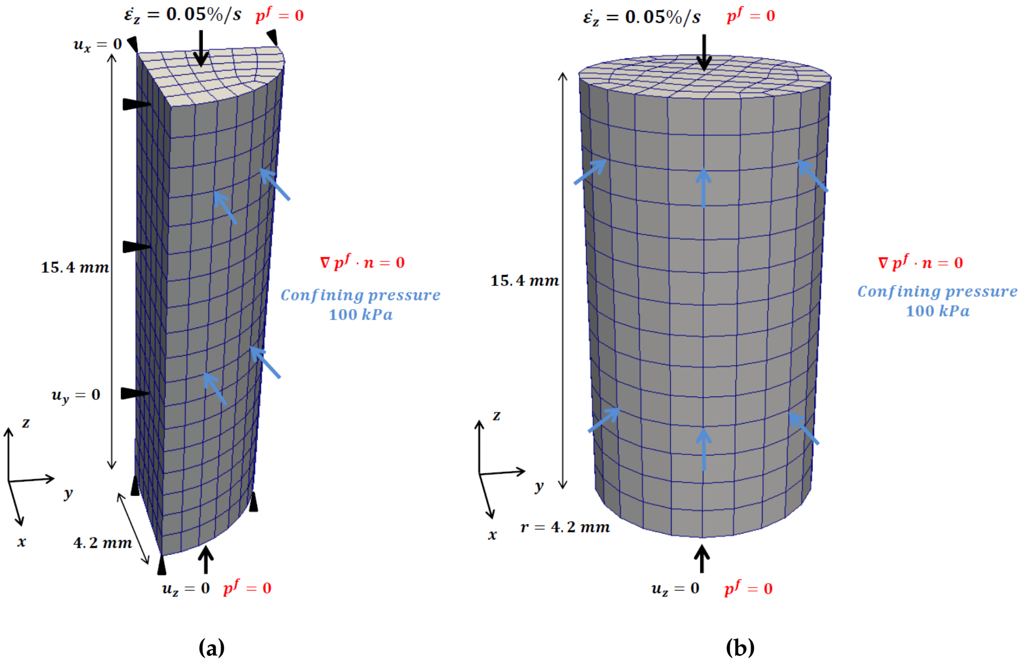
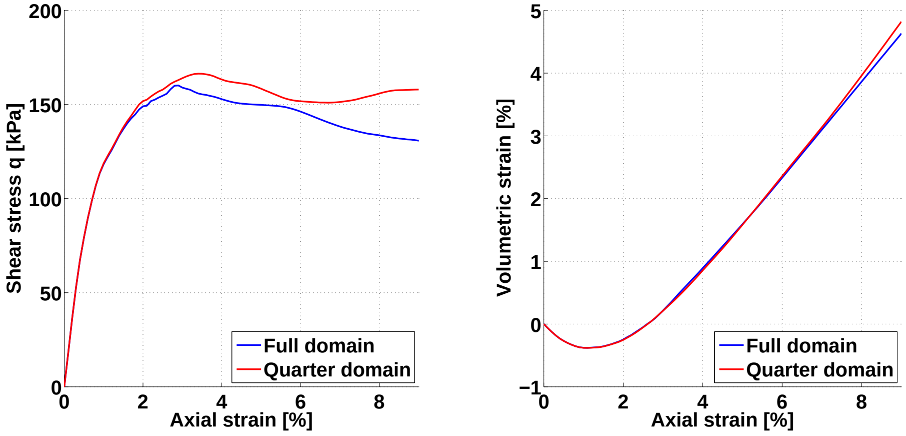


Figure 2.11: Geometry, mesh and boundary conditions for globally drained triaxial compression test. (a) Quarter-domain simulation. (b) Full-domain simulation

simulation is insufficient to capture the deformed configuration when bifurcation occurs. While the assumption of axial-symmetry is valid before the onset of strain localization, enforcing axial-symmetry via reduced domain and additional essential boundary condition may eliminate the bifurcation mode(s) that is not axial-symmetric.

An additional full-domain simulation is performed at a strain rate ten times slower: $\dot{\epsilon}_z = 0.005\%$ per second. The global shear stress and volumetric strain behavior are compared for the two loading rates in Fig. 2.14. The specimen under higher strain rate can sustain higher shear stress, but the strain rate has very little influence on volumetric strain behavior. The evolution of pore pressure at the center of the cylindrical specimen in two cases are also shown in Fig. 2.15. At a high strain rate, the pore water does not have time to fully diffuse through local pores and reach steady state. As a result, excess pore pressure



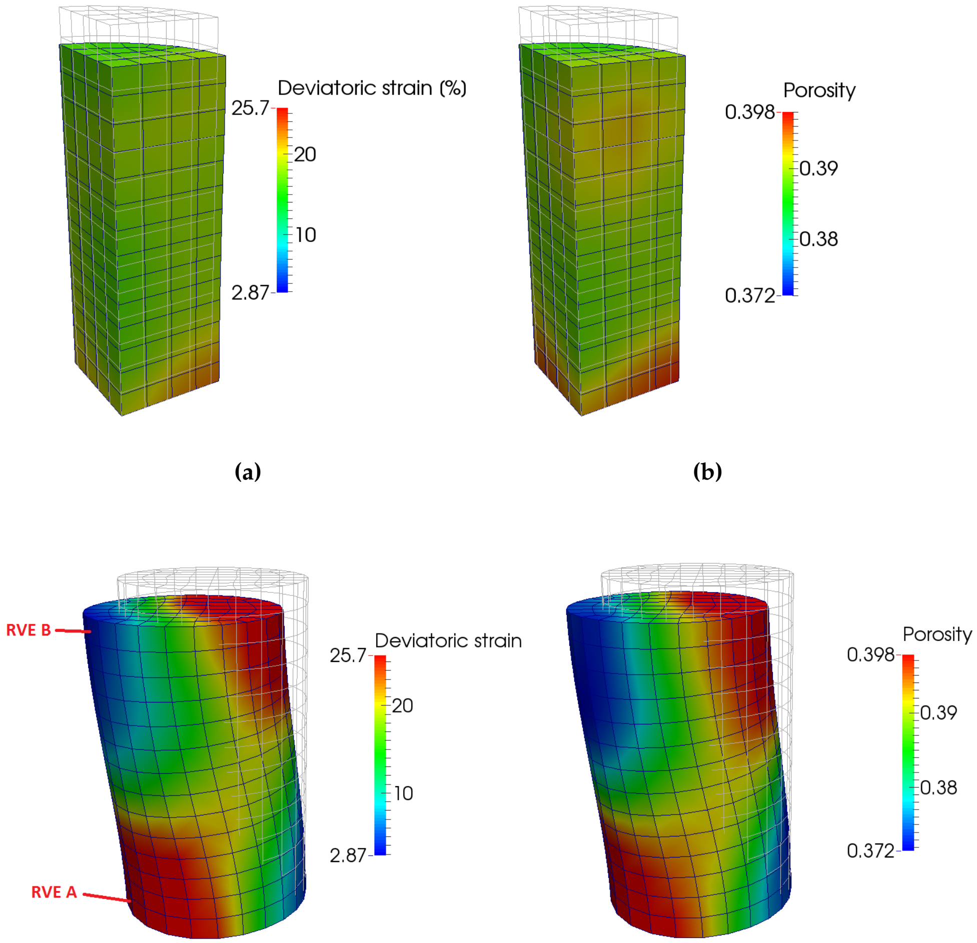
(a)

(b)

Figure 2.12: Global shear stress and volumetric strain behavior in globally drained triaxial compression test. Comparison of quarter-domain and full-domain simulations

builds up to about 5 kPa while the specimen shrinks. The pressure then decreases and becomes negative when the specimen dilates. In the low-strain-rate case, the magnitude of pore pressure is about five times smaller while the trend looks similar of the high-strain-rate counterpart.

One of the advantages of substituting macroscopic phenomenological constitutive model with DEM simulations for the poromechanics problem is that the macroscopic poroelasticity properties, such as Biot's coefficient B, Biot's modulus M and effective permeability k could be inferred and updated from DEM at each Gauss point. As a result, the spatial variability of these poro-elasticity parameters triggered by material bifurcation or non-homogeneous loading can be properly captured. As an example, we monitor the evolution of these poro-elasticity parameters against axial strain ε_z for a RVE inside the shear band (RVE A, shown in Fig. 2.13(c)) and another RVE outside the shear band (RVE B, shown in Fig. 2.13(c)) in the $\dot{\epsilon_z} = 0.05\%$-per-second, full-domain simulation (Fig. 2.16). The

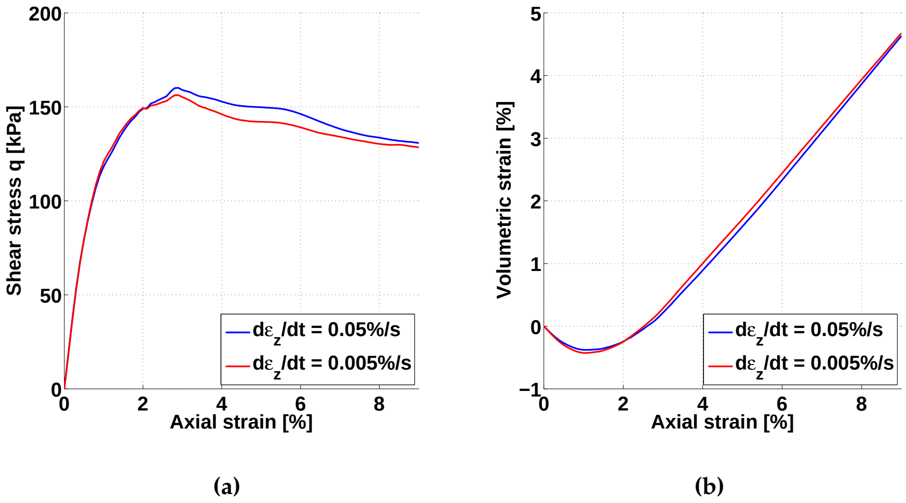


(a) (b)

(c) (d)

Figure 2.13: Distribution of deviatoric strain and porosity in globally drained triaxial compression test at 9% axial strain. Comparison of quarter-domain and full-domain simulations.

evolution of the Biot's coefficient B shown in Fig. 2.16)(a) suggests that the effective bulk modulus of the solid skeleton (K_T^{DEM}) first increases and then decreases presumably due to the porosity changes in both RVEs A and B. The Biot's modulus M, which is related to the Biot's coefficient B and porosity ϕ, exhibits an initial reduction and largely increases after about $\varepsilon_z = 2\%$ for RVE A. For RVE B, M stays at a constant value. The effective



(a)

(b)

Figure 2.14: Global shear stress and volumetric strain behavior in globally drained triaxial compression test. Comparison of two loading rate.

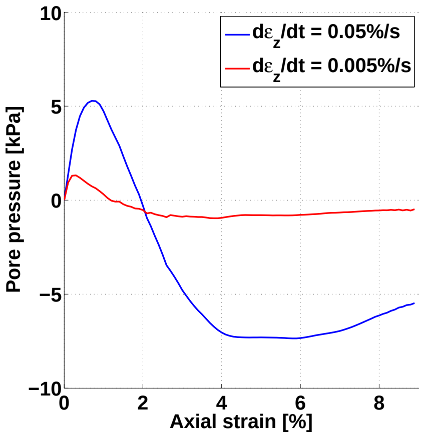
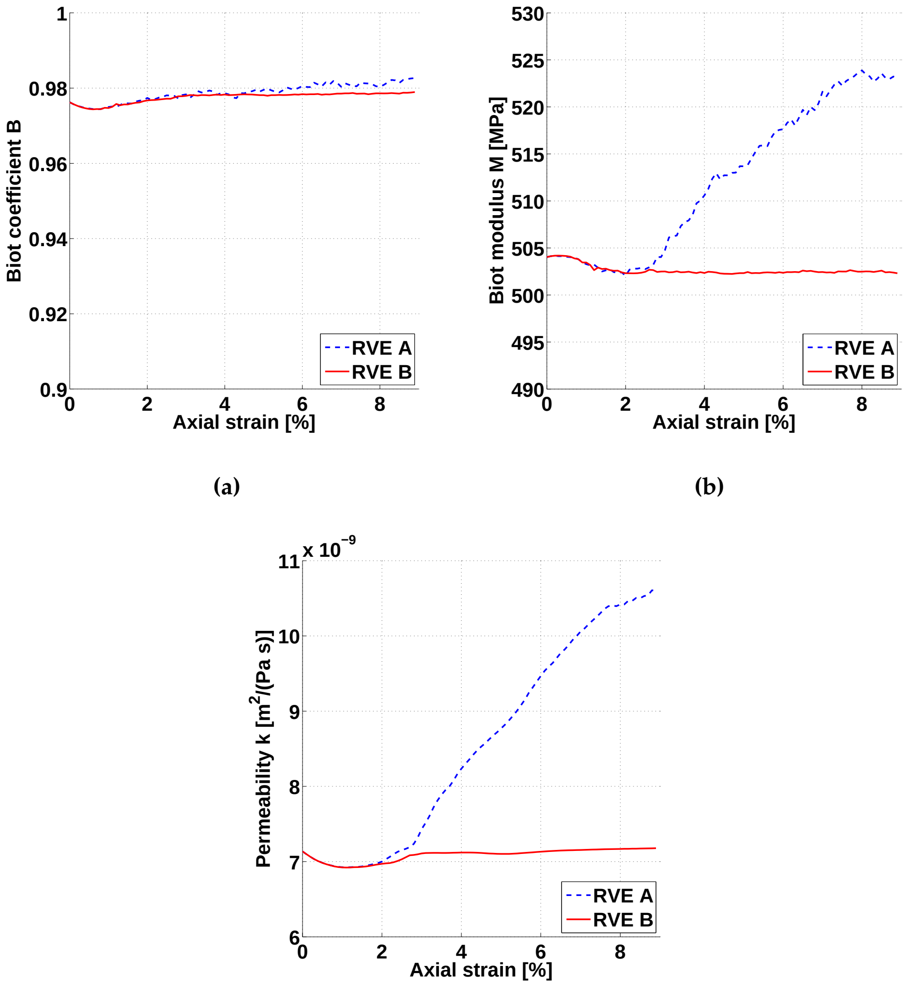


Figure 2.15: Evolution of pore pressure at the center of the cylindrical specimen during triaxial compression test subjected to two loading rate.

permeability k also evolves with the porosity according to the Kozeny-Carmen relation.

Another advantage of the multiscale scheme is the accessibility to evolution of microstructures during deformations. To demonstrate this, we perform a simple microstructural analysis in which the Anisotropic Critical State Theory (ACST) introduced by [126, 267,



(a)

(b)

(c)

Figure 2.16: Evolution of (a) Biot's coefficient, (b) Biot's modulus and (c) effective permeability for RVE A (inside shear band, Fig. 2.13(c)) and RVE B (outside shear band, Fig. 2.13(c)).

124] is adopted to analyze the fabric of the fluid-saturated granular assemblies at the finite strain range. The fabric anisotropy of two RVEs, one taken inside the shear band (RVE A) and another one in the host matrix (RVE B) are analyzed and compared against each other. The fabric tensor $\boldsymbol{G}_{\text{fabric}}$ is contact-normal-based and is computed from a DEM RVE via

[124]

$$G_{\text{fabric } ij} = \frac{1}{N_c} \sum_{c \in N_c} n_i^c n_j^c \tag{2.57}$$

where $\boldsymbol{n}^c$ is the unit vector of contact normal and N_c is the number of contacts inside the RVE. The tensor $\boldsymbol{F}_{\text{fabric}}$ characterizes the fabric anisotropy of the RVE and is written as [267]

$$F_{\text{fabric } ij} = \frac{15}{2}(G_{\text{fabric } ij} - \frac{1}{3}\delta_{ij}) \tag{2.58}$$

where δ_{ij} is the Kronecker delta. Its norm F_{fabric} and direction $\boldsymbol{n}_F$ are defined by

$$\boldsymbol{F}_{\text{fabric}} = F_{\text{fabric}}\boldsymbol{n}_F, \quad F_{\text{fabric}} = \sqrt{\boldsymbol{F}_{\text{fabric}} : \boldsymbol{F}_{\text{fabric}}} \tag{2.59}$$

To analyze whether and how fabric evolves differently inside shear band and the host matrix, we compute the normalized fabric anisotropy variable (FAV) $A = \boldsymbol{n}_F : \boldsymbol{n}_s$ (a measure introduced in [126, 267] that quantifies the relative orientation of the tensor $\boldsymbol{F}_{\text{fabric}}$ and the deviatoric stress tensor $\boldsymbol{s}$) for RVE A (inside shear band) and RVE B (outside shear band). The evolution of deviatoric stress q and porosity against axial strain ε_z are also monitored to measure how close the materials in the two RVEs reach the critical state according to the anisotropic critical state theory, i.e.,

$$\eta = \eta_c, \; e = e_c = \hat{e}_c(p) \text{ and } A = A_c = 1 \tag{2.60}$$

where η is the ratio between the effective mean pressure p' and the deviatoric stress q and e is the void ratio. η_c, $e_c = \hat{e}_c(p)$ and $A_c = 1$ are critical state values of the stress ratio, void ratio and fabric anisotropy variable (cf. [126, 124]).

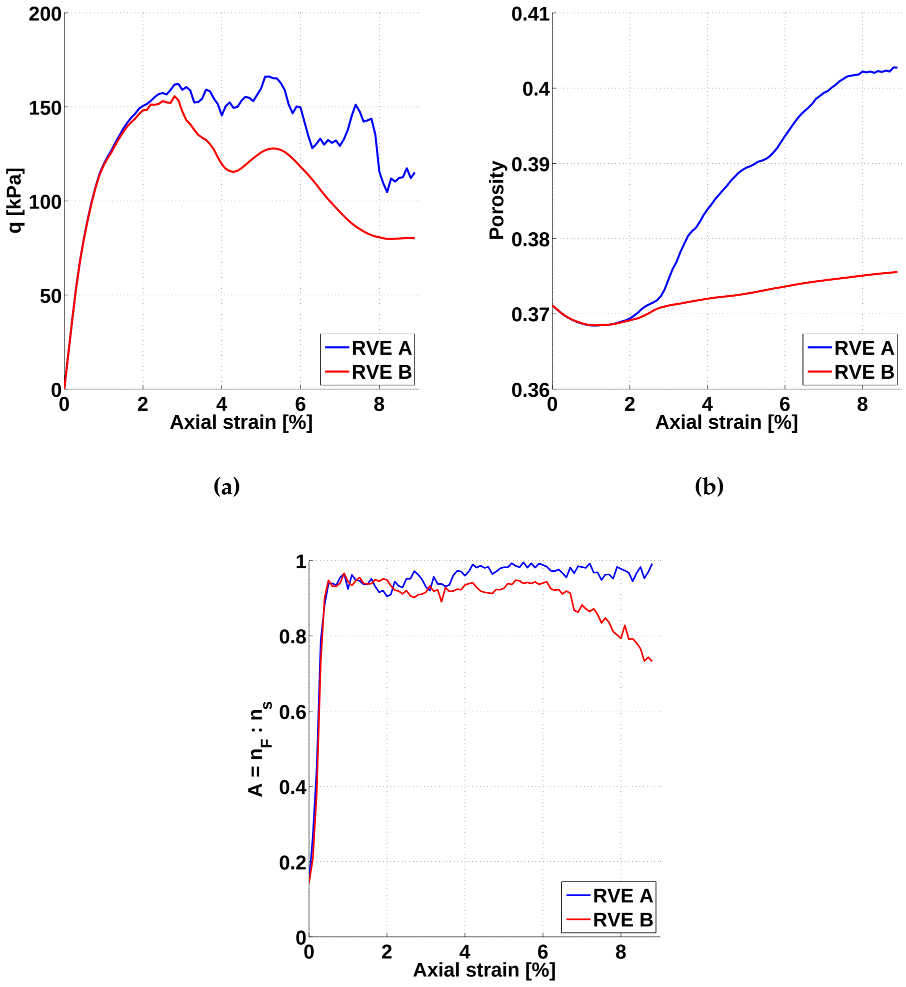
The results are summarized in Fig. 2.17. The stress-strain response shown in Fig. 2.17(a) indicates that RVE A becomes unstable after the peak shear stress and experiences significant dilation until the critical state indicated by the plateau in the porosity curve. The normalized FAV of RVE A rises to about 0.96 quickly upon subjected to the triaxial loading. Then, normalized FAV stay close to 1, which indicates that the fabric and stress directions in RVE A is nearly coaxial, as the RVE A approaches the critical state.

On the other hand, RVE B, which lies outside the shear band, experiences slightly more softening, but the dilatancy is much less than RVE A. The FAV curve of RVE B deviates from the curve of RVE A after axial strain of 2% and exhibits opposite trend that the fabric and stress directions loss coaxiality. This observation suggests that the critical states are not achieved simultaneously within an specimen that forms deformation band.

To demonstrate the performance of the multiscale semi-implicit scheme, the convergence rate of the quarter-domain simulation is illustrated in Fig. 2.18 as an example. At different strain levels, the convergence curves show linear profiles in the logarithm-scale plot. The first step converges the fastest since the RVEs are linear elastic at $\varepsilon_z = 0.1\%$. The number of iterations required for convergence increases to 11 when the global shear stress reaches the peak (about $\varepsilon_z = 2\%$). In the softening stage, the explicitly treated the elastic-plastic contribution K^{ep} to the material tangential stiffness becomes more significant. Therefore the convergence rate is further reduced and each time step requires about 20 iterations.

2.6 Conclusions

In this work, we present a finite strain dual-scale hydro-mechanical model that couples grain-scale granular simulations with a macroscopic poro-plasticity model at low Reynolds number. Using effective stress principle, the macroscopic total stress is partitioned into effective stress, which is homogenized from grain-scale simulations and macroscopic pore pressure, which is updated from macroscopic simulation. To improve computational efficiency, we adopt a semi-implicit predictor-multicorrector scheme that splits the inter-



(a)

(b)

(c)

Figure 2.17: Evolution of (a) deviatoric stress q (b) porosity (c) $A = \boldsymbol{n}_F : \boldsymbol{n}_s$ (relative orientation between anisotropic fabric and deviatoric stress directions) during triaxial compression test ($\dot{\epsilon}_z = 0.05\%/s$) for RVE A (inside shear band, Fig. 2.13(c)) and RVE B (outside shear band, Fig. 2.13(c)).

nal force into macroscopic and microscopic components. By updating the macroscopic poro-elastic contribution (FEM) implicitly and the microscopic counterpart (DEM) explicitly, we establish a multiscale scheme that is unconditionally stable and therefore allows simulations to advance in time steps large enough for practical applications. Multiscale

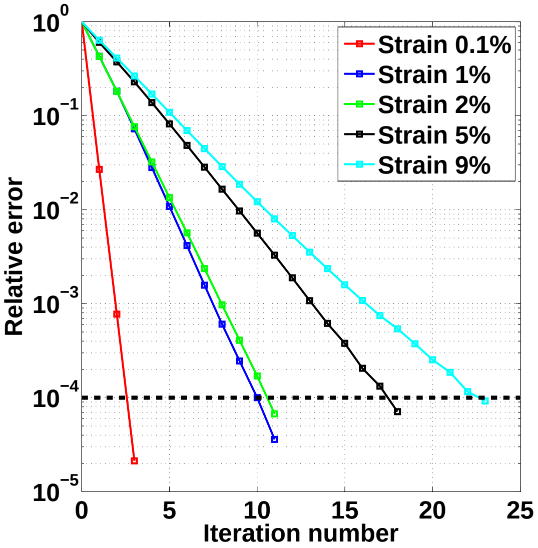


Figure 2.18: Convergence profiles of the triaxial compression test at different axial strain levels. The relative error is defined as $\frac{||\Delta F^i||}{||\Delta F^{(i=0)}||}$, where $\boldsymbol{\Delta F^i}$ is the residual force at the iteration step i. The convergence is reached when the error falls below 10^{-4}.

simulations at specimen-scale are conducted to showcase the potentials of the proposed method to solve a wide spectrum of problems across spatial length scales. To the best of our knowledge, this is the first time a hierarchical multiscale coupling scheme is established to resolve finite strain poro-plasticity problem.

Chapter 3

Multiscale LBM-DEM-FEM coupling for dual-permeability fissured porous media

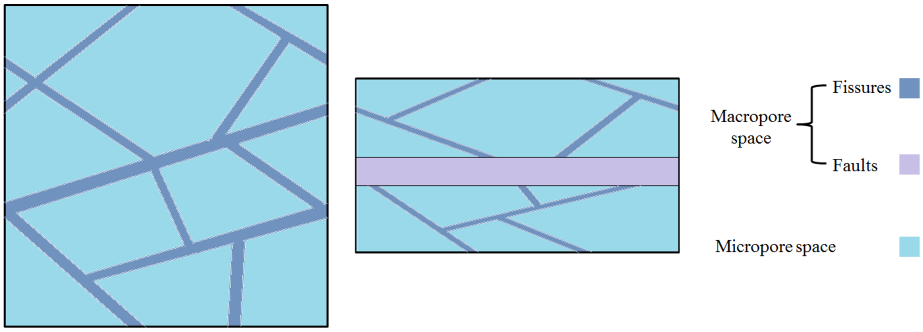
This chapter is reproduced from the published paper: K. Wang, W.C. Sun, An updated Lagrangian LBM-DEM-FEM coupling model for dual-permeability porous media with embedded discontinuities, *Computer Methods in Applied Mechanics and Engineering*, 344:276-305, 2019.

3.1 Introduction

The geological complexity of many geo-systems, such as fractured reservoirs and faults often makes the single-permeability Darcian model inadequate to replicate the complex hydraulic behaviors [81]. This complexity is linked to the wide spectrum of pore sizes. The pores among particles, inside individual grains and crystalline planes and those formed by the dissolution and cavities are often of orders of difference in sizes. In the idealized cases where pore space distribution of a porous medium is, roughly speaking, bi-modal (e.g. fractured reservoir composed of sandstone) a dual-porosity dual-permeability model can be used to approximate the hydraulic behavior [76, 179, 113, 26, 45]. For a fissured porous medium exhibiting a major fault, the fissures and the fault can be considered as macropores, while the pores in the intact solid matrix are considered as the micropores (Figure. 3.1). Then the macroscopic flow in this double porosity medium is the overlapping of both flows inside the individual systems of macropores and micropores, as well as the fluid transfer between the two systems of pores. Should mechanical forces or perturbations in pore pressure field take place, the major fault which is previously stable is put at the risk of being reactivated. The evolution of the microstructure of the fault results in the change of the effective macropore permeability and hence the fluid flow in macropores.

Strong discontinuities such as grain boundaries, flaws, cracks, joints, and faults are very common across multiple length scales in geological materials. Understanding the hydro-mechanical responses of the interfaces is important for numerous engineering applications, such as oil exploration, geothermal applications, geological disposal of nuclear waste and CO2. The presence of strong discontinuities is important due to (1) their significant roles in altering the mechanical responses of the host system (e.g. strain localization, crack bands), (2) inducing anisotropic changes in the flow characteristics as they function as flow barrier (e.g. compaction band) or channels (e.g. tensile cracks), as well as (3) changing the hydro-mechanical coupling mechanisms under different loading conditions [169, 209, 210, 255, 246].

A common approach to capture the hydro-mechanical responses is to use phenomenological laws designed for these interfaces [261, 20, 57]. Within a finite element model, the interfaces are then either represented by surface elements inserted in between volume elements or enriched basis designed to captures the kinematics of the embedded strong



1. REV of micropores and the net of fissures, with two-dimensional periodicity

2. REV of a major fault and its neighborhood (micropores and the net of fissures), with one-dimensional periodicity along the fault

Figure 3.1: Schematic representation of representative elementary volumes (REVs) of a fissured porous medium with pre-existing fault through double porosity model. REV 1 is a multi-dimensional periodic cell of standard double-porosity medium outside the major fault, with the net of fissures regarded as macropores and the voids in intact solid matrix as micropores. REV 2 is a cell of double-porosity medium of a fault neighborhood, with the major fault (induced by cracking, shear band formation, etc, and will be reactivated under mechanical forces or changes in pore pressure) regarded as macropore spaces in addition to the net of fissures. REV 2 has a one-dimensional periodicity along the fault. The formulation of multiscale finite element with embedded strong discontinuity in this chapter makes use of the second REV.

discontinuities (e.g. assumed strain or extended finite element) within an element (e.g. [148, 60, 37, 38, 23, 129]). Regardless of the techniques used to represent the strong discontinuities, the quality of the simulations strongly depends on the cohesive zone model chosen to represent the interfaces. However, as pointed out by [95] and exemplified in [165], many cohesive zone models are highly idealized constitutive responses where softening regimes are often curve-fitted via simple mathematical expressions (e.g. cubic polynomial, smoothed trapezoidal, exponential, and bilinear) to yield the right amount of fracture energy and stiffness. As a result, capturing the mixed-mode and cyclic responses remains a difficult task. Furthermore, the mixed-mode traction-separation is often an extension of the

Mode I separation law where one simply uses an effective separation (defined as a function of normal and tangential displacement jumps) to determine the normal and tangential traction [235, 164]. In the case where effective separation is defined as a weighted norm of the normal and tangential component of the displacement jumps, the difference between Mode I crack and anti-crack cannot be captured. [165, 246]. This assessment has been also discussed in [16] in which the complications due to the fact that unstable materials cannot maintain a homogeneous state even when the perturbation is small. This limitation is more profound when the interfaces possess multiscale microstructures where (1) meso-scale features such as voids and inclusions of sizes spanning multiple orders of magnitudes (e.g. the dual-porosity materials); (2) the microstructural length scale is comparable to the thickness of the interfaces (e.g. granular materials) and (3) when mixed-mode separation and closure happen cyclically.

The objective of this chapter is to fill the alluded knowledge gaps via multiscale techniques designed specifically to capture the hydro-mechanical responses of interfaces in fluid-infiltrating porous materials in the finite deformation range. In particular, we introduce the computational homogenization procedure for dual-porosity porous layer based on the generalized effective stress principle. Extending the previous work in [132, 243, 242], we establish micro-macro transitions to generate the proper hydro-mechanical responses from representative volume element (RVE) simulations as a replacement for the phenomenological traction-separation law and cubic laws for interfaces. Furthermore, as spatial averaging effective permeability of fractured porous system often leads to erroneous predictions on the flow characteristics, the interactions of the pore-fluid in the embedded strong discontinuities and that in the host porous matrix is captured via a dual-permeability system. As the porous media with strong discontinuities is viewed as a dual-permeability system at the macroscopic scale, a homogenization procedure is used to obtain the macro-pore, micro-pore and interfacial effective permeabilities [120, 5]. By considering the Hill-Mandel lemma for Darcy's flow in the macro- and micro-pore systems, admissible boundary conditions are defined such that both the macro- and micro-permeabilities can be estimated. Numerical examples are used to demonstrate the robustness and capacity of the multiscale poromechanics models. To the best knowledge of both authors, this is the first time a hierarchical discrete-continuum model has been established for embedded strong discontinuities in dual-permeability system undergoing large deformation.

The remainder of this chapter is organized as follows. We first explain the kinematics of the dual-porosity, dual-permeability system with strong discontinuities. We then explain the various choices of RVE simulations with different boundary conditions that provide macroscopic hydro-mechanical responses for the interfaces. Following the description of the RVE simulations, we then describe the macroscopic finite element formulation with embedded discontinuity for dual-permeability systems. The implementation techniques are then highlighted, and numerical examples are presented.

3.2 Problem Statement

This section provides a brief account of the theoretical basis of the multi-scale coupling model designed for a dual-porosity, dual-permeability system with embedded strong discontinuities. While there are previous work dedicated to model embedded strong discontinuities of porous media in the small and finite deformation ranges (e.g. [204, 118, 38, 150, 149, 211, 154, 54, 178, 55]), there has not yet been any attempt to introduce bridging-scale methods for interfaces composed of dual-porosity, dual-permeability materials. Nor is there any work aimed at introducing constitutive responses via discrete-continuum coupling to simultaneously replace phenomenological cohesive zone law and anisotropic permeability model for interfaces in the finite deformation range. In this section, we first briefly review the kinematics of the hydro-mechanical interfaces. Following this, a brief description of the boundary value problem for the dual-permeability system in finite deformation range and that of the corresponding finite element formulation of interfaces are given.

3.2.1 Large-scale dual-permeability hydro-mechanical problem

Consider a saturated porous medium with highly localized deformation zones (e.g. cracks, faults.) occupying a spatial domain $\mathcal{B} \subset \mathbb{R}^{n_{sd}}$, where n_{sd} =1, 2, or 3 stands for the number of spatial dimensions. The boundary of the body in the reference configuration is denoted as $\partial\mathcal{B} \subset \mathbb{R}^{n_{sd}-1}$. The porous solid is treated as a double-porosity mixture and the two dominant pore scales are the macropores M (in this case the pores of fissures, shear bands, cracks) and the micropores m, (micro-pores in the solid matrix). A solid skeleton material point at point $\boldsymbol{X}$ of the reference configuration may move due to rigid body motion and/or deformation. Here we assume that this material point is defined in a macroscopic sense such that the material point is associated with a representative elementary volume whose space is partially occupied by the solid constituent and the fluid constituent in the macro- and micro-pores. For convenience, the microstructural attributes of this representative elementary volume are often neglected and the porous medium is therefore regarded as a multiphase effective medium. If no crack growth or healing occurs, the location of this material point in the current configuration can be determined via the mapping $\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t)$. The displacement is $\boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{x}(\boldsymbol{X}, t) - \boldsymbol{X} = \boldsymbol{\varphi}(\boldsymbol{X}, t) - \boldsymbol{X}$. The macroscopic deformation of the effective medium is therefore characterized by $\boldsymbol{F} = \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{X}} = \nabla^{X} \boldsymbol{\varphi}$. The elementary volumes of the total mixture, solid, macropores, micropores and void are denoted as dV, dV_s, dV_M, dV_m, $dV_v = dV - dV_s = dV_M + dV_m$, respectively. The pore fractions for macropores (ψ^M) and micropores (ψ^m) are defined as,

$$\psi^M(\boldsymbol{X}, t) = \psi = \frac{dV_M}{dV_v}, \quad \psi^m(\boldsymbol{X}, t) = 1 - \psi = \frac{dV_m}{dV_v}. \tag{3.1}$$

Meanwhile, the volume fractions for solid, macropores and micropores can be expressed as a function of porosity and pore fractions, i.e.,

$$\begin{aligned}
\phi(\boldsymbol{X}, t) &= \frac{dV_v}{dV}, \\
\phi^s(\boldsymbol{X}, t) &= \frac{dV_s}{dV} = 1 - \phi, \\
\phi^M(\boldsymbol{X}, t) &= \frac{dV_M}{dV} = \phi\psi, \\
\phi^m(\boldsymbol{X}, t) &= \frac{dV_m}{dV} = \phi(1 - \psi).
\end{aligned} \tag{3.2}$$

The partial mass densities of each constituent can be determined using the volume fractions and the intrinsic mass density of solid ρ_s and fluid ρ_f:

$$\begin{aligned}
\rho^s &= \phi^s \rho_s = (1 - \phi)\rho_s, \\
\rho^M &= \phi^M \rho_f = \phi\psi\rho_f, \\
\rho^m &= \phi^m \rho_f = \phi(1 - \psi)\rho_f.
\end{aligned} \tag{3.3}$$

The pull-back total mass density of the mixture is given by

$$\rho_0 = \rho_0^s + \rho_0^M + \rho_0^m = J\rho^s + J\rho^M + J\rho^m, \tag{3.4}$$

where $J = \det(\boldsymbol{F})$ is the Jacobian of the solid motion.

Let us denote the (solid) material time derivative following the solid skeleton trajectory as $\dot{(\bullet)} = \frac{\partial (\bullet)}{\partial t} + \nabla^x (\bullet) \cdot \boldsymbol{v}$. Assuming incompressible solid and fluid constituents and no mass exchange between solid and fluid [26]:

$$\begin{aligned}
\dot{\rho}_0^M &= \dot{\overline{J\phi\psi\rho_f}} = \dot{J}\psi\rho_f + J\phi\dot{\psi}\rho_f = \rho_f J\psi\,\nabla^{\boldsymbol{x}}\cdot\boldsymbol{v} + \rho_f J\phi\dot{\psi},\\
\dot{\rho}_0^m &= \dot{\overline{J\phi(1-\psi)\rho_f}} = \dot{J}(1-\psi)\rho_f - J\phi\dot{\psi}\rho_f = \rho_f J(1-\psi)\,\nabla^{\boldsymbol{x}}\cdot\boldsymbol{v} - \rho_f J\phi\dot{\psi},
\end{aligned} \tag{3.5}$$

using the identities $\dot{\overline{J\phi}} = \dot{J} - \dot{\overline{J\phi^s}} = \dot{J}$ and $\dot{J} = J\,\nabla^{\boldsymbol{x}}\cdot\boldsymbol{v}$ [203].

The relative fluid mass fluxes $\boldsymbol{q}_M$, $\boldsymbol{q}_m$, and relative fluid velocities $\widetilde{\boldsymbol{v}}_M$, $\widetilde{\boldsymbol{v}}_m$ take the form

$$\begin{aligned}
\boldsymbol{q}_M &= \rho^M\widetilde{\boldsymbol{v}}_M &&= \rho_f\phi\psi\,\widetilde{\boldsymbol{v}}_M &&= \rho_f\phi\psi\,(\boldsymbol{v}_M - \boldsymbol{v}),\\
\boldsymbol{q}_m &= \rho^m\widetilde{\boldsymbol{v}}_m &&= \rho_f\phi(1-\psi)\,\widetilde{\boldsymbol{v}}_m &&= \rho_f\phi(1-\psi)\,(\boldsymbol{v}_m - \boldsymbol{v}),
\end{aligned} \tag{3.6}$$

where $\boldsymbol{v}_M$ and $\boldsymbol{v}_m$ are the fluid velocities in the macropores and micropores, respectively.

The relative fluid mass fluxes are related to pore pressures via Darcy's law:

$$\begin{aligned}
\boldsymbol{q}_M &= -\rho_f\frac{\boldsymbol{k}_M}{\mu_f}\cdot(\nabla^{\boldsymbol{x}} p_M - \rho_f\boldsymbol{g}),\\
\boldsymbol{q}_m &= -\rho_f\frac{\boldsymbol{k}_m}{\mu_f}\cdot(\nabla^{\boldsymbol{x}} p_m - \rho_f\boldsymbol{g}),
\end{aligned} \tag{3.7}$$

where p_M and p_m are Cauchy macropore pressure and Cauchy micropore pressure, respectively. $\boldsymbol{k}_M$ and $\boldsymbol{k}_m$ are intrinsic permeability tensors for macro-scale pore and micro-scale pore. μ_f is the dynamic viscosity of the fluid. $\boldsymbol{g}$ is the gravity acceleration vector.

The pull-back mass fluxes are obtained by Piola transforms

$$\boldsymbol{Q}_M = J\boldsymbol{F}^{-1}\cdot\boldsymbol{q}_M,\quad \boldsymbol{Q}_m = J\boldsymbol{F}^{-1}\cdot\boldsymbol{q}_m. \tag{3.8}$$

The fluid in macropores can diffuse from or into micropores. The fluid mass transfer between the two scales is characterized by the coefficient [45]

$$c_0 = Jc = J\frac{\bar{\alpha}}{\mu_f}(p_M - p_m). \tag{3.9}$$

$\bar{\alpha}$ is a material parameter for macro-micro-pore interface permeability.

To construct the strong form of the problem, the boundary $\partial\mathcal{B}$ having unit normal $\boldsymbol{N}$ at $\boldsymbol{X} \in \partial\mathcal{B}$ admits the decomposition

$$\begin{cases} \partial\mathcal{B} = \overline{\partial\mathcal{B}_u \cup \partial\mathcal{B}_t} = \overline{\partial\mathcal{B}_{p_M} \cup \partial\mathcal{B}_{q_M}} = \overline{\partial\mathcal{B}_{p_m} \cup \partial\mathcal{B}_{q_m}} \\ \emptyset = \partial\mathcal{B}_u \cap \partial\mathcal{B}_t = \partial\mathcal{B}_{p_M} \cap \partial\mathcal{B}_{q_m} = \partial\mathcal{B}_{p_M} \cap \partial\mathcal{B}_{q_m} \end{cases}, \tag{3.10}$$

where $\partial\mathcal{B}_u$, $\partial\mathcal{B}_{p_M}$ and $\partial\mathcal{B}_{p_m}$ are Dirichlet boundaries with prescribed solid displacement, macropore pressure and micropore pressure, respectively. $\partial\mathcal{B}_t$, $\partial\mathcal{B}_{q_M}$ and $\partial\mathcal{B}_{q_m}$ are Neumann boundaries with prescribed tractions, macropore flux and micropore flux, respectively.

Assuming quasi-static case and incompressible solid and fluid constituent, the Lagrangian strong form of the large-scale dual-permeability hydro-mechanical problem reads: find the displacement $\boldsymbol{u} : \mathcal{B} \to \mathbb{R}^{n_{sd}}$, the Cauchy macropore pressure $p_M : \mathcal{B} \to \mathbb{R}$ and the Cauchy micropore pressure $p_m : \mathcal{B} \to \mathbb{R}$ such that the balance of linear momentum, the balance of mass in macropores and micropores, the boundary conditions are satisfied:

$$\begin{cases} \nabla^{\boldsymbol{X}} \cdot \boldsymbol{P} + \rho_0 \boldsymbol{g} = c_0(\widetilde{\boldsymbol{v}}_m - \widetilde{\boldsymbol{v}}_M) & \text{on } \mathcal{B}, \\ \dot{\rho}_0^M + \nabla^{\boldsymbol{X}} \cdot \boldsymbol{Q}_M = -c_0 & \text{on } \mathcal{B}, \\ \dot{\rho}_0^m + \nabla^{\boldsymbol{X}} \cdot \boldsymbol{Q}_m = c_0 & \text{on } \mathcal{B}, \\ \boldsymbol{u} = \hat{\boldsymbol{u}} & \text{on } \partial\mathcal{B}_{\boldsymbol{u}} \\ \boldsymbol{P} \cdot \boldsymbol{N} = \hat{\boldsymbol{t}} & \text{on } \partial\mathcal{B}_{\boldsymbol{t}}, \\ p_M = \hat{p}_M & \text{on } \partial\mathcal{B}_{p_M}, \\ \boldsymbol{Q}_M \cdot \boldsymbol{N} = -\hat{Q}_M & \text{on } \partial\mathcal{B}_{q_M}, \\ p_m = \hat{p}_m & \text{on } \partial\mathcal{B}_{p_m}, \\ \boldsymbol{Q}_m \cdot \boldsymbol{N} = -\hat{Q}_m & \text{on } \partial\mathcal{B}_{q_m}, \end{cases} \tag{3.11}$$

where $\boldsymbol{P}$ is the first Piola-Kirchhoff stress and its relation to Kirchhoff stress tensor $\boldsymbol{\tau}$ and

Cauchy stress $\boldsymbol{\sigma}$ are:

$$\boldsymbol{\tau} = J\boldsymbol{\sigma} = \boldsymbol{P} \cdot \boldsymbol{F}^{\mathrm{T}}. \tag{3.12}$$

The total Kirchhoff stress tensor permits the following decomposition in terms of effective Kirchhoff stress $\boldsymbol{\tau}'$ and Cauchy pore pressures p_M and p_m, based on the effective stress principle,

$$\boldsymbol{\tau} = \boldsymbol{\tau}' - Jp^{\mathrm{avg}}\boldsymbol{I} = \boldsymbol{\tau}' - J[\psi p_M + (1-\psi)p_m]\boldsymbol{I}. \tag{3.13}$$

The initial conditions are imposed as

$$\boldsymbol{u}(\boldsymbol{X}) = \boldsymbol{u_0}(\boldsymbol{X}),\ \ p_M(\boldsymbol{X}) = p_{M_0}(\boldsymbol{X}),\ \ p_m(\boldsymbol{X}) = p_{m_0}(\boldsymbol{X}) \text{ for all } \boldsymbol{X} \in \mathcal{B} \text{ at } t = t_0. \tag{3.14}$$

3.2.2 Kinematics of embedded strong discontinuities in dual-permeability porous media

Consider a material point $\boldsymbol{X}$ in the dual-permeability porous media $\mathcal{B}$ and an associated local neighborhood $\mathcal{B}_X \subset \mathcal{B}$ embedded with strong discontinuity (fracture, shear band, fault, etc.). Denote the surface of discontinuity as Γ and the local domain $\mathcal{B}_X$ is thus divided by Γ into sub-domain pair $\mathcal{B}_X = \mathcal{B}_X^+ \bigcup \mathcal{B}_X^-$. The motion of the particles within $\mathcal{B}_X$ is described by local displacement field $\boldsymbol{u}_\Gamma = \boldsymbol{x}_\Gamma - \boldsymbol{X} = \boldsymbol{\varphi}_\Gamma(\boldsymbol{X}) - \boldsymbol{X}$. Assume the following relation to the large-scale (or conformal) displacement field $\boldsymbol{u}$,

$$\boldsymbol{u}_\Gamma = \boldsymbol{u} + [\![\boldsymbol{u}]\!](H_\Gamma - f_\Gamma), \tag{3.15}$$

where $[\![\boldsymbol{u}]\!]$ is the displacement jump across the interface Γ, H_Γ is the Heaviside step function across Γ and f_Γ is a smooth ramp function in $\mathcal{B}_X$ [22]. It is also useful to define the continuous part $\bar{\boldsymbol{u}}$ of motion $\boldsymbol{u}_\Gamma$ as

$$\bar{\boldsymbol{u}} = \boldsymbol{u} - [\![\boldsymbol{u}]\!]\, f_\Gamma. \tag{3.16}$$

The large-scale (or conformal) and continuous deformation gradients are defined as,

$$\boldsymbol{F} = \boldsymbol{I} + \nabla^{\boldsymbol{X}} \boldsymbol{u}, \quad \overline{\boldsymbol{F}} = \boldsymbol{I} + \nabla^{\boldsymbol{X}} \overline{\boldsymbol{u}}. \tag{3.17}$$

The local deformation gradient is given by, assuming relative uniformity of $[\![\boldsymbol{u}]\!]$ along Γ such that $\nabla^{\boldsymbol{X}} [\![\boldsymbol{u}]\!] \to \boldsymbol{0}$ in $\mathcal{B}_X$,

$$\begin{aligned} \boldsymbol{F}_\Gamma &= \boldsymbol{I} + \nabla^{\boldsymbol{X}} \boldsymbol{u}_\Gamma \\ &= \boldsymbol{F} - [\![\boldsymbol{u}]\!] \otimes \nabla^{\boldsymbol{X}} f_\Gamma + ([\![\boldsymbol{u}]\!] \otimes \boldsymbol{N})\delta_\Gamma \\ &= \overline{\boldsymbol{F}} + ([\![\boldsymbol{u}]\!] \otimes \boldsymbol{N})\delta_\Gamma, \end{aligned} \tag{3.18}$$

where the equation $\nabla^{\boldsymbol{X}} H_\Gamma = \delta_\Gamma \boldsymbol{N}$ is employed. δ_Γ is the Dirac delta function across Γ and $\boldsymbol{N}$ is the unit normal of Γ pointing from $\mathcal{B}_X^-$ to $\mathcal{B}_X^+$.

From Eq. 3.18, following [8, 38, 9], the local deformation gradient allows a multiplicative decomposition into two parts:

$$\boldsymbol{F}_\Gamma = \overline{\boldsymbol{F}} \cdot \widetilde{\boldsymbol{F}} = \overline{\boldsymbol{F}} \cdot (\boldsymbol{I} + ([\![\boldsymbol{U}]\!] \otimes \boldsymbol{N})\delta_\Gamma) \tag{3.19}$$

where $[\![\boldsymbol{U}]\!] = \overline{\boldsymbol{F}}^{-1} \cdot [\![\boldsymbol{u}]\!]$ is the material displacement jump across Γ.

The presence of displacement jump in the solid phase results in the discontinuity of the fluid flux across the interface. The localized fluid flow model developed by [37, 38] states that the fluid flux vector field $\boldsymbol{Q}_\Gamma$ in the local neighborhood $\mathcal{B}_X$ is composed of a regular flow field $\boldsymbol{Q}$ and a local flux jump $[\![\boldsymbol{Q}]\!]$:

$$\boldsymbol{Q}_\Gamma = \boldsymbol{Q} + [\![\boldsymbol{Q}]\!](H_\Gamma - f_\Gamma). \tag{3.20}$$

The same ramp function f_Γ as in the displacement field is employed, but a different ramp function can also be chosen. The rate of local fluid content M_Γ (fluid mass increment per unit reference volume of porous solid) is thus obtained by

$$\dot{M}_\Gamma = -\nabla^{\boldsymbol{X}} \cdot \boldsymbol{Q}_\Gamma = -\nabla^{\boldsymbol{X}} \cdot \boldsymbol{Q} + [\![\boldsymbol{Q}]\!] \cdot \nabla^{\boldsymbol{X}} f_\Gamma - ([\![\boldsymbol{Q}]\!] \cdot \boldsymbol{N})\delta_\Gamma, \tag{3.21}$$

with the assumption of $\nabla^{X} \cdot [\![\boldsymbol{Q}]\!] \rightarrow 0$ in $\mathcal{B}_X$. In this model, the mass flux could be discontinuous across the interface, while the pressure field remains continuous. This assumption is justified by the experimental findings that in ordinary soil mechanics testing situations or for quasi-static loading conditions, no pore-water pressure shocks can develop across shear band boundaries [237]. The assumption is also a necessary condition for the existence of second order derivative of pore pressures as shown in Equations (3.7), (3.21). This assumption allows for storage and fluid flow within the discontinuity, but the interface permeability must be infinitely large. Moreover, since there is no independent pressure inside the interface, the model also lacks the capability to simulate the pressurized crack [56].

To circumvent this issue described in [56], we introduce a simplified effective-medium approach in which pores inside the major fault (strong discontinuity), the nearby net of fissures and the intact host continuum are idealized as two porous systems of distinct pore sizes. The pore spaces inside the strong discontinuities and fissures constitute the macropore system, whereas the pore spaces inside the intact continuum are considered as the micropore system. Those porous systems may exchange fluid mass. Assuming that the separation of scales applies and there exists an appropriate length scale such that the fissured porous media can be treated as the superimposition of three continua, the solid skeleton, the macro-fluid continuum and the micro-fluid continuum at the representative elementary volume level. Then the pore pressure and fluid flux of each pore system could be distinctive until both porous systems reach steady state. In this treatment, the detailed fluctuation of the micro- and macro-pore fluid flow below the scale of the RVE is ignored. Nevertheless, this treatment also enables us to capture the transient fluid responses across distinct time and spatial scales. This capacity is particularly important to deal with dual-permeability media with significant permeability differences and/or when the mass exchange is slow compared to the flux in either porous system (e.g. the pressurized crack).

Assuming that the homogenization procedure is valid even if an embedded strong discontinuity exists, then the pore pressure and flux fields can be defined at the macroscopic continuum scale such that p_M and $\boldsymbol{Q}_M$ are the macroscopic macropore pressure and flux of the effective porous system that represents all the macropore space inside the net of fissures and the major fault. Meanwhile, p_m and $\boldsymbol{Q}_m$ are the pressure and flux of the effective porous system that represents the micropore space inside the intact solid skeleton. Furthermore, the transfer flow at the continuum scale can be idealized as the flow between the effective porous system that represents the pore space inside intact solid skeleton and the counterpart that simultaneously represents both the pore space of the fault and the fissures.

p_M, p_m and $\boldsymbol{Q}_M$, $\boldsymbol{Q}_m$ are continuous fields obtained from separate balance equations of fluid mass in both pore-scales, interconnected by the fluid mass transfer c_0 between the two pores (Eq. 3.11). For finite elements with embedded strong discontinuity, standard integrations are employed for all pressure and fluid flow in macropores and micropores. Note that this treatment employed in the proposed framework is not the only feasible approach. In the case where sub-scale fluctuations of pore pressure and flux are important, one may consider the localized fluid flux formulation previously established in [38].

3.2.3 Condition of traction continuity

The solution of the local displacement jump field $[\![\boldsymbol{u}]\!]$ requires a traction continuity equation relating the nominal stress field $\boldsymbol{P}$ in $\mathcal{B}_X \setminus \Gamma$ and the nominal traction $\boldsymbol{T}_\Gamma$ in Γ driving the mechanical inelastic effects inside the strong discontinuity. The weak form writes, for all variations $\delta \llbracket \boldsymbol{u} \rrbracket$, [7]

$$-\frac{1}{V_{\mathcal{B}_X}} \int_{\mathcal{B}_X} \delta \llbracket \boldsymbol{u} \rrbracket \cdot \boldsymbol{P} \boldsymbol{N} \, dV + \frac{1}{L_\Gamma} \int_\Gamma \delta \llbracket \boldsymbol{u} \rrbracket \cdot \boldsymbol{T}_\Gamma \, d\Gamma = 0, \tag{3.22}$$

where $V_{\mathcal{B}_X} = \text{measure}(\mathcal{B}_X)$ and $L_\Gamma = \text{measure}(\Gamma)$. Since the fluid pressure in both pore-scales are continuous across the strong discontinuity, $\llbracket p_M \rrbracket = 0$, $\llbracket p_m \rrbracket = 0$, Eq. 3.22 can be written in terms of effective stress and traction:

$$-\frac{1}{V_{\mathcal{B}_X}} \int_{\mathcal{B}_X} \delta \llbracket \boldsymbol{u} \rrbracket \cdot \boldsymbol{P}' \boldsymbol{N} \, dV + \frac{1}{L_\Gamma} \int_\Gamma \delta \llbracket \boldsymbol{u} \rrbracket \cdot \boldsymbol{T}'_\Gamma \, d\Gamma = 0. \tag{3.23}$$

In the limit $\frac{V_{\mathcal{B}_X}}{L_\Gamma} \to 0$, the local equilibrium equation writes:

$$\boldsymbol{T}'_\Gamma = (\boldsymbol{P}' \cdot \boldsymbol{N})|_\Gamma. \tag{3.24}$$

The effective stress measure $\boldsymbol{P}'$ in the host continuum just outside the strong discontinuity is determined by the continuous part of the total deformation gradient, i.e., $\boldsymbol{P}' = \boldsymbol{P}'(\overline{\boldsymbol{F}}) = \boldsymbol{P}'(\boldsymbol{F}, \llbracket \boldsymbol{u} \rrbracket)$. The effective traction measure is given by the displacement jump, i.e., $\boldsymbol{T}'_\Gamma = \boldsymbol{T}'_\Gamma(\llbracket \boldsymbol{u} \rrbracket)$. These two constitutive laws for the host continuum and for the fracture are presented in the subsequent section.

3.2.4 Constitutive equations

Inelastic dissipation mainly occurs inside the strong discontinuities of a solid body under external loading, while the host matrix outside these localized zones remains intact. Thus, to reduce computational cost, the multi-scale approach is only adopted in the vicinity of strong discontinuities, and we assume a Neo-Hookean hyperelastic constitutive model is sufficient to replicate the constitutive responses of the host matrix. For a RVE associated with a material point undergoing a deformation characterized by the Left Cauchy-Green deformation tensor $\boldsymbol{b} = \boldsymbol{F} \cdot \boldsymbol{F}^T = \overline{\boldsymbol{F}} \cdot \overline{\boldsymbol{F}}^T$, the effective Kirchhoff stress is given by [16],

$$\boldsymbol{\tau}' = \lambda_0 \, \ln J \, \boldsymbol{I} + \mu_0 (\boldsymbol{b} - \boldsymbol{I}), \tag{3.25}$$

where λ_0 and μ_0 are the Lamé constants from linear elasticity.

The spatial elasticity tensor $\boldsymbol{\mathcal{C}}^e$ is given by

$$\boldsymbol{\mathcal{C}}^e = \lambda_0 \boldsymbol{I} \otimes \boldsymbol{I} + 2(\mu_0 - \lambda_0 \ln J)\boldsymbol{I}^4_{\text{sym}}. \tag{3.26}$$

$\boldsymbol{\mathcal{C}}^e$ relates the Lie derivative of the effective stress tensor to the velocity gradient, i.e., $\mathscr{L}_v \boldsymbol{\tau}' = \boldsymbol{\mathcal{C}}^e : \boldsymbol{d} = \boldsymbol{\mathcal{C}}^e : \frac{1}{2}(\nabla^x \boldsymbol{v} + \boldsymbol{v}\, \nabla^x) = \boldsymbol{\mathcal{C}}^e : \nabla^x \boldsymbol{v}$, after noting the minor symmetry of $\boldsymbol{\mathcal{C}}^e$. Then the spatial tensor $\boldsymbol{\alpha}^e$ with respect to the rate of $\boldsymbol{\tau}'$ is given by

$$\dot{\boldsymbol{\tau}}' = \boldsymbol{\alpha}^e : \nabla^x \boldsymbol{v}, \quad \boldsymbol{\alpha}^e = \boldsymbol{\mathcal{C}}^e + \boldsymbol{\tau}' \oplus \boldsymbol{I} + \boldsymbol{\tau}' \ominus \boldsymbol{I}. \tag{3.27}$$

The material parameters for the model can be determined by numerical experiments on an initial micro-scale representative volume element (RVE) composed of discrete particles: a uniaxial tension/compression test to get the P-wave modulus M and a simple shear test to get the shear modulus G. Then $\lambda_0 = M - 2G$ and $\mu_0 = G$.

As for the traction-separation relation $\boldsymbol{T}'_\Gamma = \boldsymbol{T}'_\Gamma([\![\boldsymbol{u}]\!])$ for the strong discontinuity, it is homogenized from the micro-scale RVE, and the approach is detailed in the subsequent section.

The material properties for the dual-porosity hydraulic model are obtained as follows. The permeability tensor $\boldsymbol{k_M}$ in macropores (fractures) in Eq. (3.7) is given by machine learning model trained with permeability data from Lattice-Boltzmann simulation on micro-scale RVE, as explained in Section 3.3. Since the microscale RVE only represents the medium inside the strong discontinuity, the hydraulic properties for the host medium and mass transfer between the two media remain updated from phenomenological laws. The permeability tensor $\boldsymbol{k_m}$ in micropores (host matrix) is assumed isotropic and its evolution against the micro-porosity follows the Kozeny–Carman relation,

$$\boldsymbol{k_m} = k_m \boldsymbol{I},\ k_m = k_{m0} \frac{\phi_m^3/(1-\phi_m)^2}{\phi_{m0}^3/(1-\phi_{m0})^2}, \tag{3.28}$$

where k_{m0} is the initial permeability for the solid matrix having initial micro-porosity of ϕ_{m0}. The pore fraction ψ for macropores (the ratio between the pore volumes of fissures and the total porous continuum) is assumed to be constant during the deformation of the continuum. Thus the current value of macroporosity is obtained as, assuming incompressible solid phase, $\phi^M = \psi\phi = \psi[1-(1-\phi_0)J^{-1}]$, and $\phi^m = (1-\psi)\phi = (1-\psi)[1-(1-\phi_0)J^{-1}]$.

The mass transfer coefficient $\bar{\alpha}$ in Eq. (3.9) is a dimensionless parameter that depends on the permeability of the interface between macropores and micropores $\bar{k}$, as well as characteristic length of the macropores spacing and solid matrix geometry [39, 43, 26]. The interface permeability $\bar{k}$ is assumed to equal to the micropore permeability k_m, following the same assumption as [123, 43]. If the effect of the geometry of the strong discontinuity is considered in the mass transfer term, $\bar{\alpha}$ should become a tensor $\bar{\boldsymbol{\alpha}}$ instead of a scalar to take into account the preferential mass transfer direction normal to the interface, in addition to the assumed isotropic mass transfer between the net of fissure and the micropore space. Hence Equation (3.9) is modified to a tensor form,

$$c_0 = Jc = \frac{J}{3\mu_f} \bar{\boldsymbol{\alpha}} : (p_M \boldsymbol{I} - p_m \boldsymbol{I}). \tag{3.29}$$

where $\bar{\boldsymbol{\alpha}}$ is now a homogenized mass transfer coefficient tensor including the structural information of the interfaces between macropores and micropores. A simple form of $\bar{\boldsymbol{\alpha}}$ can be defined as

$$\bar{\boldsymbol{\alpha}} = \frac{dV_{fissure}}{dV_M} \bar{\alpha} \boldsymbol{I} + \frac{dV_{SD}}{dV_M} \bar{\alpha} \boldsymbol{n} \otimes \boldsymbol{n}, \tag{3.30}$$

where $dV_{fissure}$ and dV_{SD} are elementary volumes of the net of fissures and the strong discontinuity in the macropore space dV_M, respectively.

3.3 Computational homogenization for strong discontinuity

Here we present the procedure to obtain the hydro-mechanical constitutive updates for embedded strong discontinuity from microscale simulations on RVEs nested inside the material interfaces. The computational homogenization schemes of single-physics material layers have been explored in a number of previous studies [94, 47, 30, 247]. For instance, [94] have introduced a procedure to generate an effective cohesive zone law for a single interface from microscale RVE. In those studies, FE^2 simulations with interface elements are used as the test bed. [47, 30] establish a multi-scale approach for RVE (or Microstructural Volume Element as introduced in the literature) having localized zones and proposed a new generalized periodic boundary condition. The overall macro-homogeneous deformation is applied to the MVE and the stress and displacement jump are homogenized. The local equation to be solved is the consistency between the macro displacement jump and the homogenized displacement jump in the RVE, instead of the traction continuity equation. [229, 228] proposed multiscale model at regular points (MMRp) and singular points (MMSp). It has been successfully used in enhanced strain finite element simulations [158]. In this study, the RVEs of discrete elements describe the underlying microstructures inside the discontinuity interface. Based on the effective stress principle, the mechanical and hydraulic constitutive laws are obtained **separately** from two types microscale simulations, i.e. the grain-scale DEM simulation and the pore-scale LBM simulation, as explained in [214] and [243]. In other words, the effective traction and the interfacial permeability (and hence the interfacial Darcy's velocity) are both obtained from the same deformed configuration. However, the deformed configuration is not obtained from LBM-DEM simulations but from DEM simulations that generate the admissible boundary conditions by assuming the validity of the effective stress principle. The major advantage of this approach is two-fold. First, the calculations of the interfacial permeability are much faster. This is due to the fact that the de-coupled permeability calculation can be conducted offline such that the trained and validated neural network can be used to replace the costly LB simulations). The second advantage is the simplicity. As the effective stress approach does not require the introduction of particle-scale hydro-mechanical force and any treatment to update the fluid-solid boundary at pore scale. Nevertheless, it should be noted that the validity of this split approach is designed for the case in which the effective stress principle is applicable for the dual-permeability system. In many situations that involve particle erosion [74, 230], soil liquefaction [64], or solid-fluid mixture with non-Darcy flow or high Reynold's number, such a simplification may lead to significant errors. In such cases, one must derive the corresponding Hill-Mandel condition for the multi-physical poromechanics problems to obtain the admissible boundary conditions and apply them to the DEM-LBM model or use direct numerical simulation (DNS) to capture the multi-physical problems. Such an extension will be considered in the future study but is out of the scope of this work.

The Hill-Mandel condition and the corresponding computational homogenization procedure that calculates the homogenized effective traction and interfacial permeability measures in the finite deformation enhanced strain formulation are detailed in the following sub-sections.

3.3.1 Online incremental homogenized mechanical responses for strong discontinuities

The homogenization procedure of mechanical constitutive law is an extension of the approach described in [94] to particle assembly using the theory in [143, 144]. Consider a cubic assembly of discrete particles representing the granular material inside the strong discontinuity (Fig. 3.2). The body force is negligible at micro-scale. This RVE of domain Ω_μ and boundary $\partial\Omega_\mu$ has an initial height of h^0_μ and is associated with a coordinate system with basis vectors $\boldsymbol{M}_\mu$ and $\boldsymbol{N}_\mu$. Choose the geometric center as the origin and place the RVE in alignment with the normal and tangential directions of the strong discontinuity Γ in the reference configuration ($\boldsymbol{N}_\mu = \boldsymbol{N}$, $\boldsymbol{M}_\mu = \boldsymbol{M}$). The current position $\boldsymbol{x}^c_\mu$ of a center of a particle is related to its position $\boldsymbol{X}^c_\mu$ in the reference configuration via the deformation map $\boldsymbol{\varphi}_\mu$. The local deformation gradient $\boldsymbol{F}_\mu = \frac{\delta \boldsymbol{\varphi}_\mu}{\delta \boldsymbol{X}^c_\mu}$. The volume average of $\boldsymbol{F}_\mu$ is given as:

$$\langle \boldsymbol{F}_\mu \rangle = \frac{1}{V_0} \int_{\Omega_\mu} \boldsymbol{F}_\mu \, d\Omega_\mu = \frac{1}{V_0} \sum_i^{N_{bound}} (\boldsymbol{x}^c_\mu)_i \otimes \boldsymbol{A}^c_i, \tag{3.31}$$

where V_0 is the initial volume of the RVE. $\boldsymbol{A}^c_i$is the surface vector of $\partial\Omega_\mu$ associated with the particle i and N_{bound} is the number of particles on $\partial\Omega_\mu$. Assuming rigid particles, the motion of a particle material point can be decomposed to the motion of the particle center and the particle rotation, i.e.,

$$\boldsymbol{x}_\mu = \boldsymbol{x}^c_\mu + \boldsymbol{R}_\mu \cdot (\boldsymbol{X}_\mu - \boldsymbol{X}^c_\mu); \quad \boldsymbol{x}^c_\mu = \langle \boldsymbol{F}_\mu \rangle \cdot \boldsymbol{X}^c_\mu + \boldsymbol{w}_c, \tag{3.32}$$

where $\boldsymbol{w}_c$ is the particle center displacement fluctuation and $\boldsymbol{R}_\mu \in SO(3)$ describes the particle rotation.

The overall effective Piola stress is given by the volume average

$$\langle \boldsymbol{P}'_\mu \rangle = \frac{1}{V_0} \int_{\Omega_\mu} \boldsymbol{P}'_\mu \, d\Omega_\mu = \frac{1}{V_0} \sum_{cont}^{Ncont} \boldsymbol{f}^{cont}_\mu \otimes \boldsymbol{L}^{cont}_\mu = \frac{1}{V_0} \sum_i^{N_{bound}} (\boldsymbol{f}^{ext}_\mu)_i \otimes (\boldsymbol{X}^c_\mu)_i, \tag{3.33}$$

where $\boldsymbol{f}^{cont}_{\mu}$ is the contact force at the grain contact $\boldsymbol{x}^{cont}_{\mu}$. $\boldsymbol{L}^{cont}_{\mu}$ is the initial branch vector, the vector that connects the centroids of two grains forming the contact. N_{cont} is the total number of particles contacts in the RVE. $(\boldsymbol{f}^{ext}_{\mu})_i$ is the external support force acting on the boundary particle i. The transition between the summation involving contact forces and the summation involving external support forces is ensured by the equilibrium of the RVE of particles.

The volume average of the virtual power in the RVE is given by

$$\langle \boldsymbol{P}'_{\mu} : \dot{\boldsymbol{F}}_{\mu} \rangle = \frac{1}{V_0} \int_{\Omega_{\mu}} \boldsymbol{P}'_{\mu} : \dot{\boldsymbol{F}}_{\mu} \, d\Omega_{\mu} = \frac{1}{V_0} \sum_{i}^{N_{bound}} (\boldsymbol{f}^{ext}_{\mu})_i \cdot (\dot{\boldsymbol{x}}^{c}_{\mu})_i. \tag{3.34}$$

The Hill-Mandel micro-heterogeneity condition requires the volume average of the virtual power in the RVE to equal the virtual power done by the volume averages of power-conjugate stress and deformation measures:

$$\langle \boldsymbol{P}'_{\mu} : \dot{\boldsymbol{F}}_{\mu} \rangle = \langle \boldsymbol{P}'_{\mu} \rangle : \langle \dot{\boldsymbol{F}}_{\mu} \rangle. \tag{3.35}$$

Since the constitutive behavior of the RVE is homogenized to a traction-separation law on the interface, the Hill-Mandel condition is recast into the form involving power-conjugate effective traction and displacement jump measures

$$h_0 \langle \boldsymbol{P}'_{\mu} : \dot{\boldsymbol{F}}_{\mu} \rangle = \langle \boldsymbol{T}'_{\Gamma} \rangle \cdot \mathscr{L}_v \, [\![\boldsymbol{u}]\!] = \langle \mathscr{T}_{\Gamma} \rangle \cdot [\![\dot{\boldsymbol{U}}]\!]. \tag{3.36}$$

For the transition between the macro-scale kinematics of the strong discontinuity and the deformation of the micro-scale RVE, the volume average of deformation gradient is defined as

$$\langle \boldsymbol{F}_{\mu} \rangle = \boldsymbol{I} + \frac{1}{h^0_{\mu}} [\![\boldsymbol{u}]\!] \otimes \boldsymbol{N}. \tag{3.37}$$

The effective nominal traction $\langle \boldsymbol{T}'_{\Gamma} \rangle$ averaged in the RVE representing the interface is given by:

$$\langle \boldsymbol{T}'_{\Gamma} \rangle = \langle \boldsymbol{P}'_{\mu} \rangle \cdot \boldsymbol{N}. \tag{3.38}$$

Among the admissible boundary conditions fulfilling the Hill-Mandel micro-heterogeneity condition, we adopt the periodic boundary conditions, where for a pair of particles on opposite boundaries $\partial\mathcal{V}^{+}$ and $\partial\mathcal{V}^{-}$, the periodicity enforces the periodicity of fluctuations and rotations

$$\boldsymbol{w}_c^{-} = \boldsymbol{w}_c^{+}, \quad \boldsymbol{R}_{\mu}^{-} = \boldsymbol{R}_{\mu}^{+}, \tag{3.39}$$

and the anti-periodiocity of support forces and couples

$$\boldsymbol{a}_c^{-} = -\boldsymbol{a}_c^{+}, \quad \boldsymbol{m}_c^{-} = -\boldsymbol{m}_c^{+}, \tag{3.40}$$

where $\boldsymbol{a}_c$ is the opposite of the resultant force on the boundary particle exerted by other particles, $\boldsymbol{m}_c$ is the opposite of the resultant couple about the center $\boldsymbol{X}_c$ on the boundary particle.

3.3.2 Offline incremental data-driven hydraulic responses for strong discontinuities

The homogenization procedure used to obtain the effective permeability from a microstructure RVE has been previously studied in [62, 161, 209, 214]. Here we apply the same procedure to obtain the homogenized effective permeability of the embedded strong discontinuities. Assume that the separation of the spatial length scale is valid, one may use the Hill-Mandel lemma corresponding to Darcy's flow problem to determine the admissible boundary condition for the flow problems. Recall that the Hill-Mandel lemma requires that

$$\langle \nabla^{x} p_M \cdot \boldsymbol{q}_M \rangle_x = \langle \nabla^{x} p_M \rangle_x \cdot \langle \boldsymbol{q}_M \rangle_x \tag{3.41}$$

where $\langle \cdot \rangle_x$ is the spatial volume averaged operator.

As shown in [62] and [161], this can lead to a number of admissible boundary conditions. For instance, one may either prescribe flux or pore pressure gradient in two opposite faces of the RVEs. One interesting aspect found in previous works (cf. [62, 209, 214, 115]) is that the choice of the boundary condition does not affect the effective permeability once the size of the RVE is sufficiently large. As mentioned previously in Section 3.2.2, we follow the treatment in [56] and assume that there is no pore pressure jump across the interface, whereas discontinuous mass flux is admissible.

The effective permeability tensor of a RVE can be determined via inverse fluid flow problem performed on the deformed RVE subjected to prescribed loading paths. The Eulerian fluid flux vector $\boldsymbol{q}$ within the RVE is computed when subjected to Eulerian pressure gradient $\nabla^{x} p$, and the macro-pore effective permeability $\boldsymbol{k}^{M}_{\mathrm{RVE}}$ is determined by Darcy's law

$$\boldsymbol{q}_M = -\frac{1}{\mu} \boldsymbol{k}^{M}_{\mathrm{RVE}} \nabla^{x} p_M. \tag{3.42}$$

μ is the dynamic viscosity of the fluid. We assume that the normal and tangential directions of the interface are also the principal directions of the macro-pore effective permeability tensors. Thus, we need only two hydraulic simulations to determine the permeability values normal and tangential to the interface, denoted as k^M_n and k^M_m, respectively. Thus the permeability tensor is expressed as

$$\boldsymbol{k}^{M}_{\mathrm{RVE}} = k^M_n \boldsymbol{n} \otimes \boldsymbol{n} + k^M_m \boldsymbol{m} \otimes \boldsymbol{m}, \tag{3.43}$$

where $\boldsymbol{n} = \overline{\boldsymbol{F}} \cdot \boldsymbol{N}$ and $\boldsymbol{m} = \overline{\boldsymbol{F}}^{-T} \cdot \boldsymbol{M}$. We choose the lattice Boltzmann (LB) method to solve the inverse fluid flow problem. For brevity, we omit the description of the LB method. Interested readers are referred to [210, 214] and [115] for details. The LB code used in this study is a C++ open source code called Palabos [59]. The procedure to obtain the two normal and tangential components is as follows. We first record the positions of all grains in the deformed microstructural assembly at different strain levels. As the size of each grain is known, the configuration of the pore space can be reconstructed and subsequently converted into binary images (cf. [214]). Then, pore pressure difference is imposed on two opposite sides orthogonal to the flow direction and no-flow boundary conditions are applied on the four remaining side faces. This setting leads to a macroscopic pressure gradient. As the lattice Boltzmann flow simulation reaches steady state, the resultant fluid flow velocity is computed and the permeability value is derived via Darcy's law (Fig. 3.2). Fig. 3.4 illustrates an example computation of permeabilities from LBM. The RVE is subjected to various displacement loading paths with loading-unloading cycles. The evolution of normal and tangential permeabilities predicted by the neural network are presented and are compared to the empirical Kozeny-Carman equation.

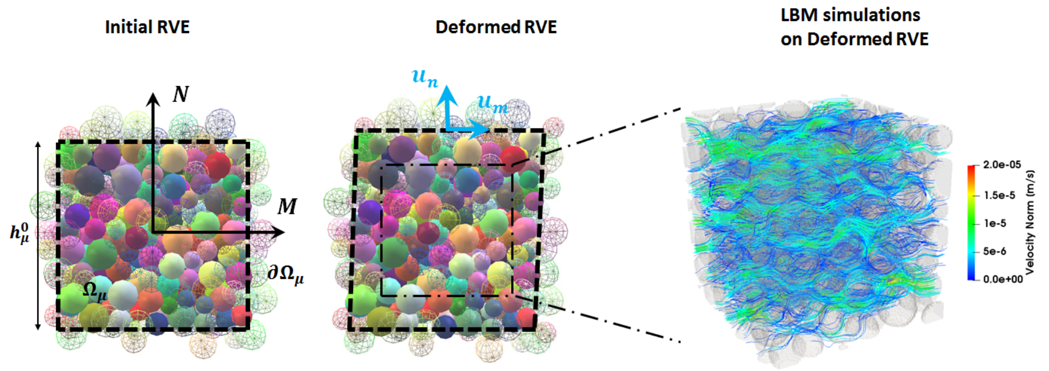


Figure 3.2: Initial and deformed configurations of the particle assembly representing the granular materials inside strong discontinuity. The effective permeabilities in the normal and tangential directions are determined by Lattice-Boltzmann simulations on representative volume of current particle assembly.

The numerical solutions of Stokes equations using Lattice-Boltzmann method yield accurate results, especially in the low Reynold number regime, but require significant computational resources to resolve the flow field at pore space. To achieve a reasonable accuracy, the number of degree of freedoms required to obtain the effective permeability is at least a few orders more than those used in discrete element simulations [214] Thus, querying the effective permeability tensor from LBM simulations from each RVE for all incremental steps during a multiscale simulation is computationally expensive. In this work, we resort to a deep learning approach to predict the effective permeability for each incremental step. The design, training, and testing of the LSTM network on path-dependent material constitutive laws are detailed in a separate and dedicated work (cf. [247]). For completeness, a brief overview is provided. First, a database containing the prescribed displacement jump loading paths, porosity and associated computed permeabilities is established by running multiple LBM simulations on deformed discrete element RVEs. Then, a recurrent neural network consisting of Long-Short-Term-Memory (LSTM) layers (see Figure 4.6) is trained using the database generated by LBM simulations [96, 245].

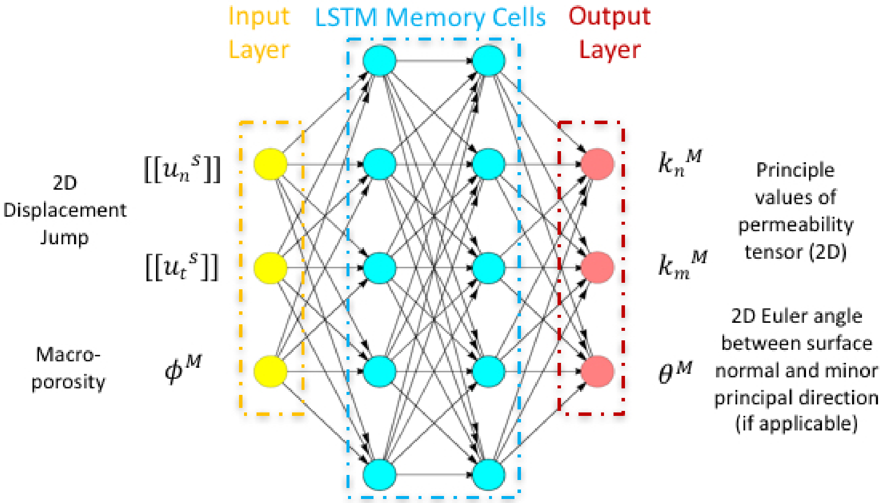
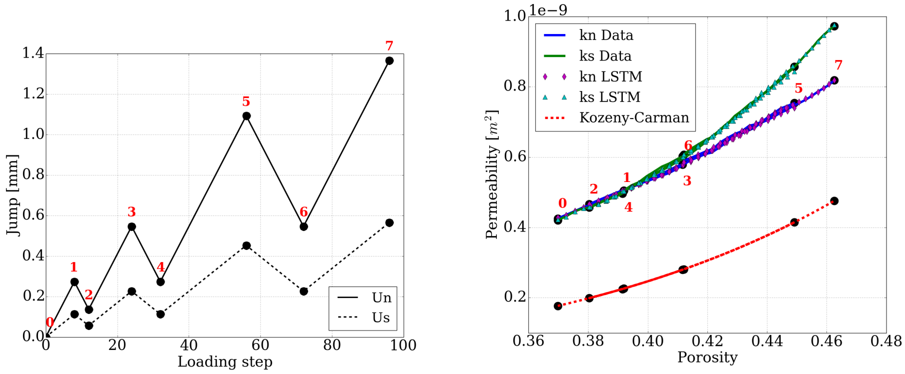


Figure 3.3: The recurrent neural network used to predict the permeability of the interface.

Finally, in each incremental update of the multiscale strong discontinuity simulation, the updated effective permeability components are generated by propagating signals from the input layer of the recurrent neural network to the output layers. In this particular case, the current displacement jumps and porosity are used as the input and the principal values and the spectral directions of the effective permeability tensor are the output of the recurrent neural network. One important upshot of this approach is that the querying time is largely reduced, as the deep learning permeability model typically requires only few seconds to make predictions.



(a) Displacement jump loading path

(b) Evolution of permeabilities along loading path

Figure 3.4: Example of permeability data generated from LBM simulations on RVEs undergoing loading-unloading sequences. (a) loading path of the normal Un and tangential displacement jumps Us. (b) Comparison between the normal kn and tangential ks permeability data from LBM simulations and the permeability components from predictions of LSTM neural network model. The calculation from empirical Kozeny-Carman equation $k = \frac{d_{50}^2}{180} \frac{\phi^3}{(1-\phi)^2}$ $(d_{50} = 1mm)$ is shown for comparison.

Remarks on the computation time Each LBM simulation for determining the permeability of a DEM assembly costs a CPU time of 5 minutes. If the LBM simulation is used online with the FEM-DEM simulation, suppose there exist 10 000 integration points in a finite element mesh, for each iteration within each time step, the total CPU time spent in updating the permeability will be 50 000 minutes. If the data-driven approach is adopted, the LBM simulations are conducted offline to generate the database for the permeability of DEM assembly. The total CPU time is 5 000 minutes, when the overall size of the training and testing data is 1 000. The training and testing of a LSTM neural network model is 250 minutes. The CPU time for a LSTM neural network to predict permeability online with the FEM-DEM simulation is less than 1 minute for 10 000 integration points. Suppose a simulation consists of 100 time steps and each time step requires on average 10 iterations to converge. The comparison of CPU time spent on permeability calculation between the online approach and the data-driven approach is 50 000 000 (5 000 * 100 * 10) minutes against 6 250 (5 000 + 250 + 1*100*10) minutes. This justifies the advantage of artificial neural network over the online LBM simulations in saving the computation time.

3.4 Numerical Example: Reactivation of faults

This example analyzes the slip of a pre-existing and formerly stable fault in saturated soil triggered by the injection of water at a nearby location. The idealized problem geometry and boundary conditions are shown in Fig. 3.5. The dimensions of the 2D field of saturated porous media are 10 m x 10 m. The domain is constrained in the x-direction on the left boundary and in the y-direction on the bottom boundary. A foundation has been constructed on top of the domain, generating a uniform loading pressure of 10 MPa. A lateral confining pressure of 5 MPa is applied on the right boundary for the frictional porous media to sustain the vertical load. There exists a 45-degree fault under the foundation. The entire system is stable and has been in equilibrium for a long time since the construction of the foundation, thus the excess pore pressures in both fractures and host matrix are zero. The initial effective stress of the porous solid is hence

$$\sigma'_{\text{Init}} = \begin{bmatrix} -5 & 0 \\ 0 & -10 \end{bmatrix}_{xy} \text{MPa}, \tag{3.44}$$

where the subscript xy refers to the coordinate system $\{x, y\}$ depicted in Fig. 3.5.

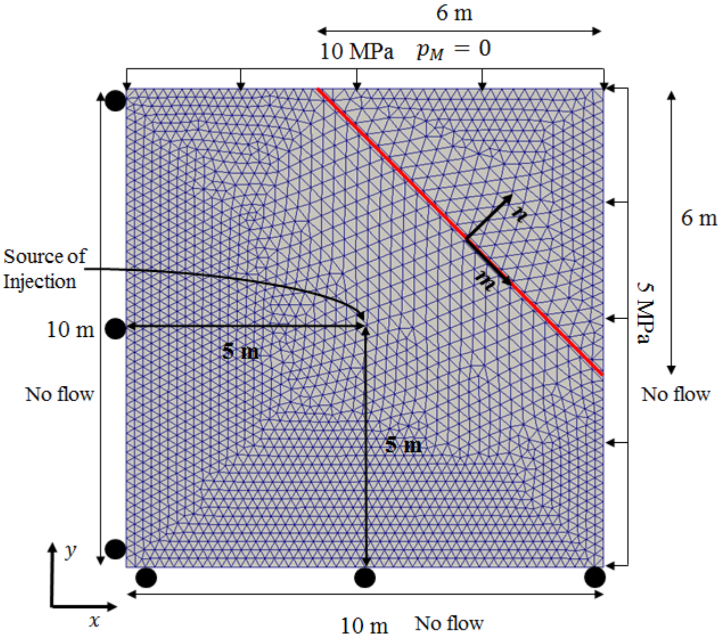


Figure 3.5: Geometry of fault reactivation problem and boundary conditions. Red line represents the pre-existing fault.

The DEM RVEs characterizing the traction-separation law of the fault are placed in alignment with the strong discontinuity. They must be in the initial stress state consistent to the macroscopic boundary conditions. From the initial stress state of the macro-scale problem (Eq. 3.44) and via a coordinate transformation ($\boldsymbol{\sigma}^{mn} = \boldsymbol{R}^{\mathrm{T}} \cdot \boldsymbol{\sigma}^{xy} \cdot \boldsymbol{R}$), the initial stress tensor of the DEM assemblies is expressed as

$$\sigma'_{\text{InitRVE}} = \begin{bmatrix} -7.5 & 2.5 \\ 2.5 & -7.5 \end{bmatrix}_{mn} \text{MPa}, \tag{3.45}$$

where the subscript mn refers to the rotated frame $\{m, n\}$ for the fault depicted in Fig. 3.5. The initial DEM RVEs in this stress state provide the correct amount of initial shear and normal tractions along the strong discontinuity.

In this example, the particle contact model for DEM is frictional and without cohesion. The normal and tangential permeabilities are obtained from machine learning models trained with LBM simulation data. The bulk material is idealized as isotropic hyperelastic material. The permeability tensors in macro- and micro-pores of the bulk are assumed isotropic and evolve according to the Kozeny-Carman equation. The material parameters used in the numerical example are summarized in Table 3.1.

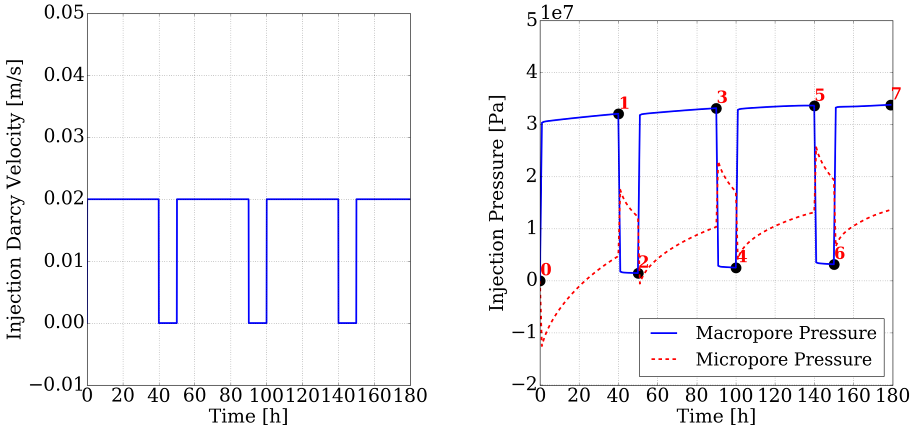
Water is injected to the macropore space (pre-existing fractures) of the field through the source S located at the center of the domain. The macropore pressure is zero on the top surface and the other three surfaces are no-flow boundaries. There is no drainage boundary for micropore pressure. This flow boundary condition is to suppress spurious micropore pressure oscillations near the drainage boundary [43]. The prescribed time history of Darcy velocity at the source is shown in Fig. 3.6. The injection profile is composed of injection-pause cycles, in which water supply is provided for 40 hours under a constant rate of 0.02 m/s, followed by a pause for 10 hours before the next cycle of injection. From the simulation results, the time history of the pore pressure in both scales at the source S is presented in Fig. 3.6. Upon injection or pause, the macropore injection pressure jumps up or plunges immediately, while the micropore pressure at the injection point has the opposite behavior. This is caused by the low mass transfer permeability between the macropores and micropores. Then in the transient regime, when fluid gradually diffuses into the micropores by mass transfer, micropore pressure slowly approaches the macropore pressure. The two pressures will eventually be identical when the diffusion between

Scale & Model	Parameter	Value
Grain-scale DEM	Particle Young's modulus E	0.5 GPa
Grain-scale DEM	Particle Poisson's ratio ν	0.3
Grain-scale DEM	Particle Friction Angle φ	$\frac{\pi}{6}$
Grain-scale DEM	Particle density	2600 kg/m^3
Grain-scale DEM	Particle mean diameter	0.5 mm
Grain-scale DEM-LBM	Initial intrinsic permeability	$9e^{-14}$ m^2
Macro-scale FEM	Young's modulus E	0.2 GPa
Macro-scale FEM	Poisson's ratio ν	0.2
Macro-scale FEM	Porosity of macropore ϕ_M	0.05
Macro-scale FEM	Porosity of micropore ϕ_m	0.1
Macro-scale FEM	Intrinsic permeability of macropore k_M	$1e^{-14}$ m^2
Macro-scale FEM	Intrinsic permeability of micropore k_m	$1e^{-17}$ m^2
Macro-scale FEM	Parameter of mass transfer $\overline{\alpha}$	$\rho_f * k_m$
Macro-scale FEM	Dynamic viscosity μ	$1e^{-3}$ Pa $\cdot$ s

Table 3.1: Material parameters for the grain- and macro-scale poromechanics problem with embedded strong discontinuities across length scales. The parameters for the simple frictional DEM model are defined in [240].

pores reaches equilibrium.

The macropore and micropore pressure field at time 40 h, 100 h and 180 h are presented in Fig. 3.7. The pressure plume is initially of the shape of a circle and then expands as the increasing amount of water are being injected through the source. The pore pressure drops when the injection pauses, but the plume is still expanding, driven by the excess pore pressure that has not been entirely diffused. When the injection is resumed, the pore pressure rises again. The presence of the fault with higher permeability disturbs the pressure plume. The fluid flows more quickly to the top surface through the channel inside the



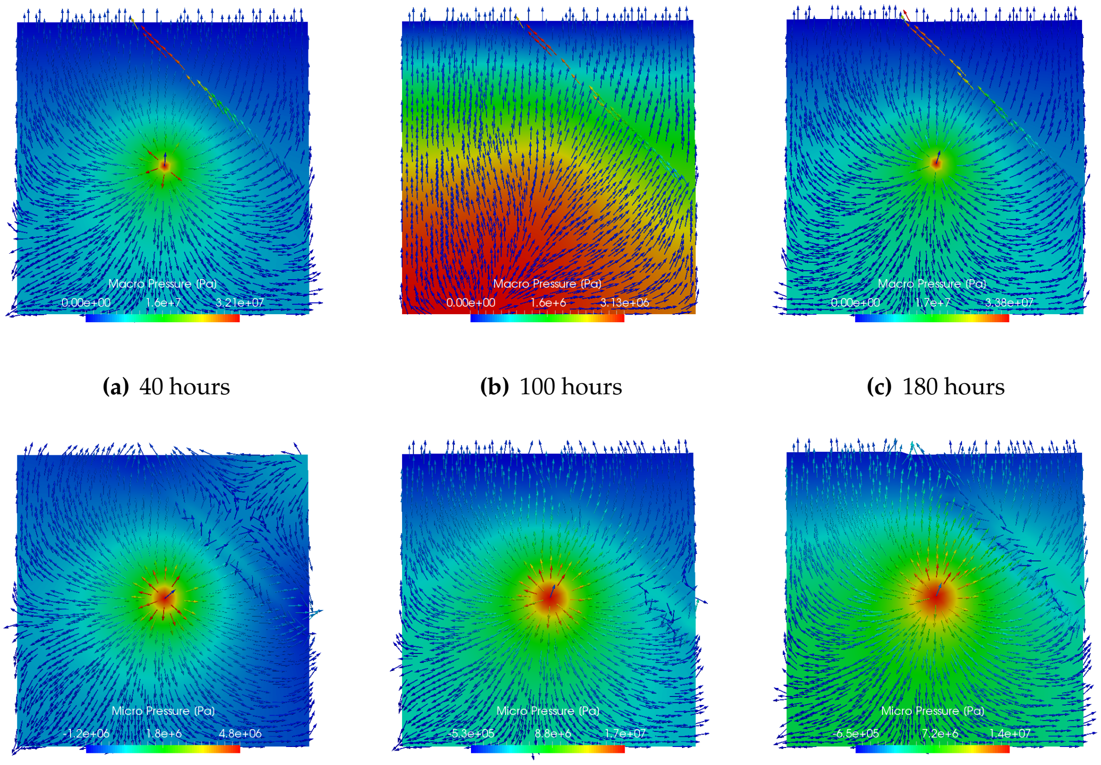
(a)

(b)

Figure 3.6: Water supply in the fault reactivation problem. (a) Time history of the prescribed injection velocity in macropore at the source point. (b) Computed responses of injection pressure in macropore and micropore at the source point. The numbers mark the sequence of injection-pause cycles.

fault. As for the micropore pressure field, it has a similar but delayed evolution behavior, due to the time required for the fluid transfer between macropores and micropores. The difference between macropore and micropore pressure is due to the different permeability in macropores and micropores for the fluid to diffuse in the macro-scale field, and also the low transfer permeability between pores.

Due to the fully coupled nature of the problem, the mechanical responses of the porous solid, especially the displacement jump and traction at the strong discontinuity, strongly depend on how pore fluid diffuses inside the pore space. The evolution of macro-scale mean effective stress field during the fluid injection cycles is shown in Fig. 4.28. The increase in the mean effective stress is due to the increase in excess pore pressure, in agreement to the effective stress principle Eq. 3.13. This results in a reduction in the normal compression traction. As the fault is frictional, this reduction in normal compression also



(a) 40 hours

(b) 100 hours

(c) 180 hours

(d) 40 hours

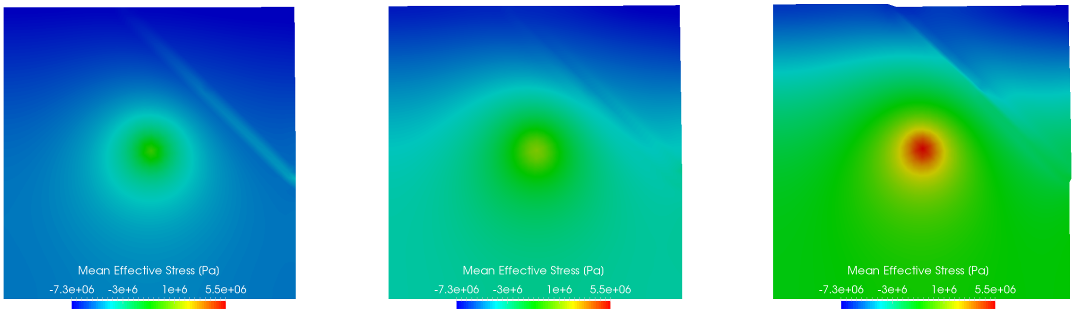
(e) 100 hours

(f) 180 hours

Figure 3.7: Evolution of macropore pressure (a-c) and micropore pressure (d-f) field. Arrows indicate the fluid flux vector field in macropores (a-c) and in micropores (d-f). The non-zero components normal to the impervious boundaries are due to the inaccuracy of the nodal projection of the flow vector field evaluated at quadrature points.

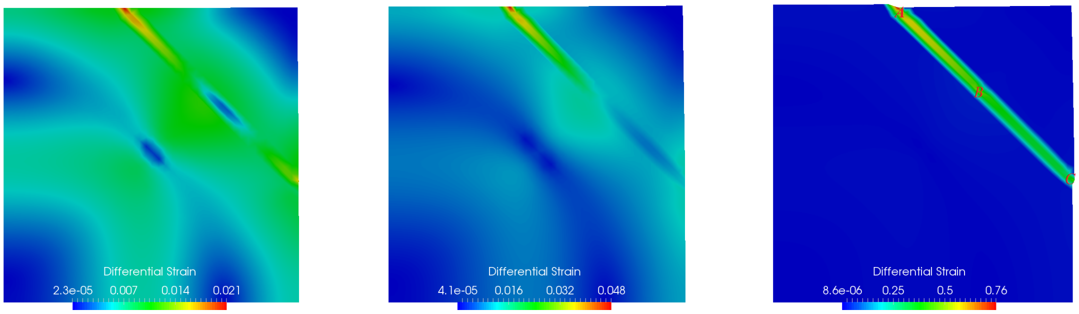
reduce the shear strength and ultimately leads to the reactivation of the fault. The slip can be clearly observed from the changes in deviatoric strain field illustrated in 3.9. The deviatoric strain gradually increases and concentrates inside the fault zone. This simulation result suggests the hazardous effect of injecting water to the underground, as a fast fluid flow may trigger the slip of a nearby pre-existing fault, leading to the failure of the foundation.

The local responses to the fluid injection-pause cycles, including the spatial displace-



(a) 40 hours **(b)** 100 hours **(c)** 180 hours

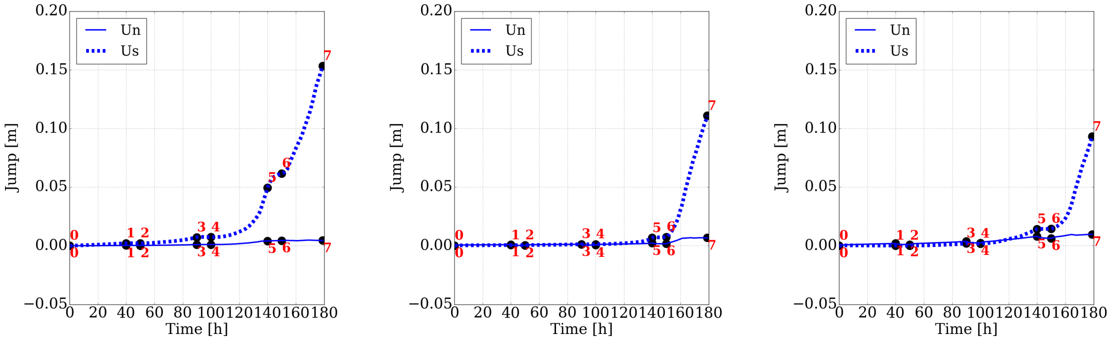
Figure 3.8: Evolution of the mean effective stress field in the macro-scale simulation.



(a) 40 hours **(b)** 100 hours **(c)** 180 hours

Figure 3.9: Evolution of the differential strain field in the macro-scale simulation.

ment jump, effective nominal traction and spatial macropore permeability, are illustrated in Fig. 3.10, Fig. 3.11 and 3.12 respectively for three locations A, B, C in the fault indicated in Fig. 3.9. The plots clearly illustrate the failure of the fault system by the opening and sliding of the local microstructures, caused by reductions in both normal and tangential tractions. These results demonstrate the capacity of our proposed multiscale model in capturing the complex mechanical and hydraulic behaviors of the interfacial materials. This is an improvement over the phenomenological traction-separation laws where idealized tensile and shear (linear or exponential) behavior is often adopted [165].

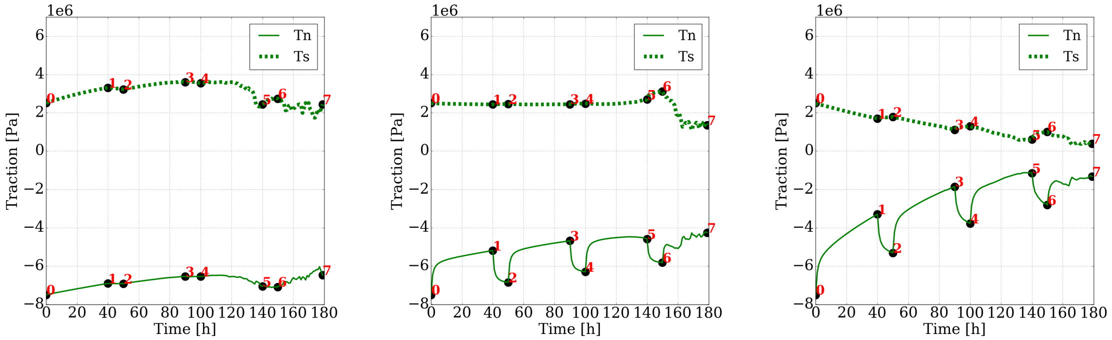


(a) RVE A

(b) RVE B

(c) RVE C

Figure 3.10: History of normal Un and tangential Us components of the displacement jump $[\![\boldsymbol{u}]\!]$ for local RVEs A, B and C (locations shown in Fig. 3.9). The numbers mark the sequence of injection-pause cycles (Fig. 3.6).



(a) RVE A

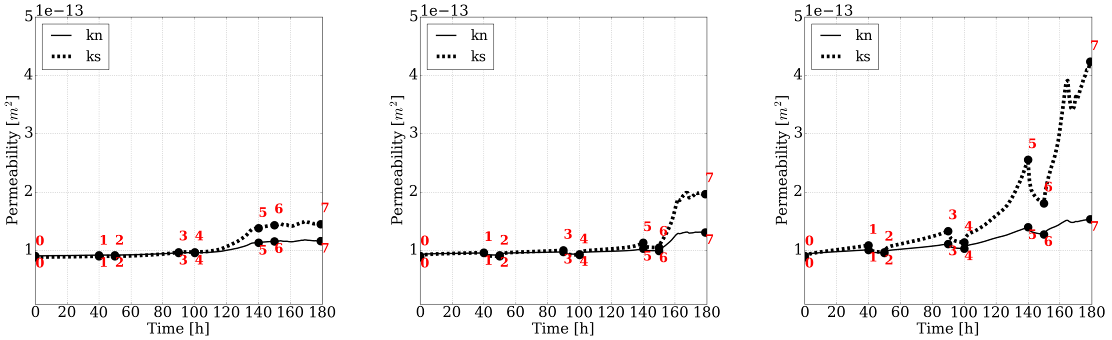
(b) RVE B

(c) RVE C

Figure 3.11: History of normal Tn and tangential Ts components of the effective nominal traction $\boldsymbol{T}'$ for local RVEs A, B and C (locations shown in Fig. 3.9). The numbers mark the sequence of injection-pause cycles (Fig. 3.6).

3.5 Conclusions

In this work, we present, for the first time, a multiscale coupling model that captures the hydro-mechanical responses of dual-permeability porous media with strong discontinuities in the finite deformation range. The traction-separation law is homogenized from



(a) RVE A **(b)** RVE B **(c)** RVE C

Figure 3.12: History of normal kn and tangential ks components of the macropore permeability $\boldsymbol{k}_{\text{RVE}}$ for local RVEs A, B and C (locations shown in Fig. 3.9). The numbers mark the sequence of injection-pause cycles (Fig. 3.6).

DEM RVEs located in the strong discontinuity, and the interfacial permeability is given by a data-driven model trained with Lattice-Boltzmann simulations on deformed RVEs. An enhanced-strain dual-porosity formulation suitable for incorporating the homogenized constitutive laws is derived. The proposed semi-data-driven multiscale framework is capable of simulating complex and fully coupled geomechanics problems with pre-existing and non-propagating fractures. This is demonstrated by a field-scale problem that showcase the failure of a fault system induced by the underground injection of fluid.

Chapter 4

Multiscale multi-permeability poroplasticity model via recursive homogenizations and deep learning

This chapter is reproduced from the published paper: K. Wang, W.C. Sun, A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning, *Computer Methods in Applied Mechanics and Engineering*, 334(1):337-380, 2018.

4.1 Introduction

In the classical hierarchical multiscale framework, such as FEM^2 (cf. [71, 68]) and DEM-FEM ([132, 244, 242]), the multiscale simulations are conducted by replacing constitutive law with representative elementary volume (REV) simulations that provide the incremental constitutive updates at each integration point. This method is typically much more cost efficient than the direct numerical simulation, as the micro-mechanical simulations are confined in the REV domain rather than conducted in the entire physical domain. However, the computational cost is typically much higher than the conventional constitutive law driven method [132, 89].

This computational cost is perhaps feasible for two-scale coupling simulations, but it may become a severe computational barrier if one attempts to link simulations across more than one scale. The remedy to this issue can be classified as two approaches – the usage of surrogate model [108, 131] or the usage of reduced-order modeling [69]. In the former case where surrogate model is used, the smaller scale simulations will often be used to generate a database aimed to record the homogenized responses of the representative elementary volume. This database can also be experimental data or a combination of both ″real″ experimental data and the ″virtual″ simulation data. This database is then split into two *mutually exclusive* subsets – One used to calibrate and identify material parameters via inverse problems; another one used for validation and performance assessment of the numerical models [131]. Nevertheless, the primary drawback of the surrogate-based approach, in particular in the cases where phenomenological models are used as surrogate model, is that the accuracy and efficiency are highly dependent on the quality of the surrogate models that replace the direct numerical simulations (DNS). Furthermore, this approach often requires multiple surrogate models for multiphysics problem that might not be consistent with each other. This issue is particularly common for poromechanics problems (e.g. [192]) where the usage of kinematic hardening plasticity model coupling with isotropic permeability model often leads to the discrepancy that is hard to detect. Even worse, the introduction of multiple material parameters may make it easier to complete curve-fitting for a model that lacks prediction capacity otherwise.

Here we limit our focus on a hierarchical multiscale coupling approach in which recurrent neural network trained by a supervised deep learning is used as surrogate model to deliver constitutive responses, from solution database [221, 102]. Our major point of departure is the use of a graph-based concept previously presented in [170], [208] and [188] to design the information flow from smaller to larger scales and the use of recurrent neural network to automatically generate **one** surrogate model that provide the updates for both the effective stress and permeability tensors in the macroscopic and microscopic pore space.

This directed graph represents the hierarchy of information processed in a computational model that utilizes a combination of classical and data-driven models. In the directed graph representation, physical quantities are viewed as vertices, while the relations among physical quantities are considered as edges that link those building blocks together to form a computational model. In the case where the directed graph represents the hierarchy of information of the initial boundary value problem, the most upstream vertices (also called the root in graph theory literature) would be the governing equations (e.g. balance principles, phase field evolution equations, etc), while the downstream vertices (also called the leaves in graph theory literature) are the unknown variables (e.g. displacement, pore pressure, temperature). Close examination of the information flow may help us distinguish the edges into 1. definitions (e.g. relation between deformation gradient and displacement field, relation between Biot's coefficient and bulk moduli) 2. universal principles (e.g. effective stress principle – relationship among total stress, effective stress and the pore pressure of the macro- and micro-pores, balance principles) and 3. material laws, phenomenological relations or empirical rules (e.g. Darcy's law, water retention curve, stress-strain relation), as shown in Figure 4.1.

Unlike the model-free approach in which the entire computational model is replaced by neural network (cf. the model-free approach e.g. [84]), our approach is to keep the edges identified either as definitions or universal principles, but replace the edges that are

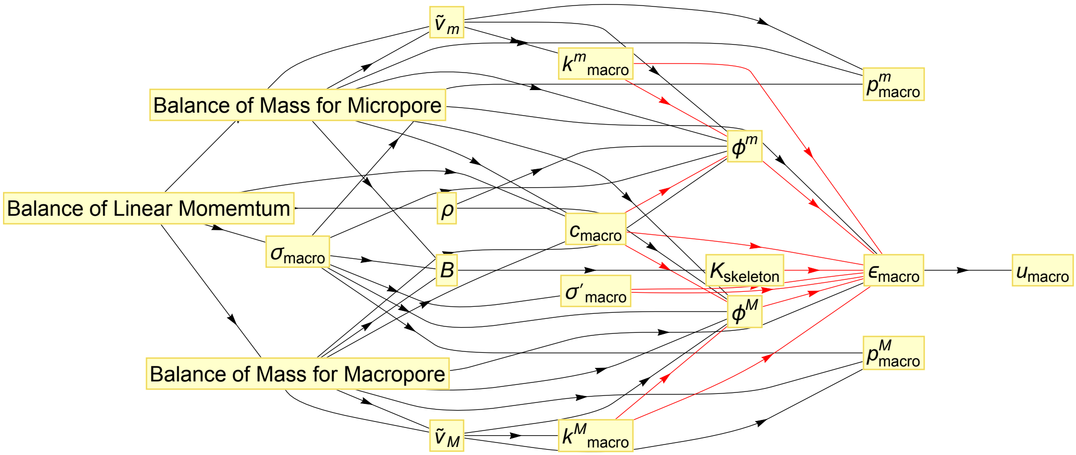


Figure 4.1: Directed graph representing the information flow of the multi-scale multi-physics dual-permeability poromechanics problem for fluid-infiltrating media. Red arrow represents either a phenomenological relation or an operator that is defined not by definition, universal law or first principle.

commonly linked together by phenomenological models with data-driven model trained from deep learning. To do so, we first identify the subgraph (the graph formed by a subset of the vertices and edges of the graph that presents the computer model) in which the vertices are only connected by material laws. In this subgraph, we again identify the leaves and roots.

This information in return gives us an idea about what we should ”learn” and what should be in the input and output in the supervised machine learning setting. In the dual-permeability poromechanics problem, which we selected as test bed, we assume that the effective stress principle is valid for the dual-permeability system [19, 222, 39, 48]. Furthermore, since we use a recurrent neural network for the supervised training, the time history of strain is not explicitly expressed as an additional vertex in the directed graph,

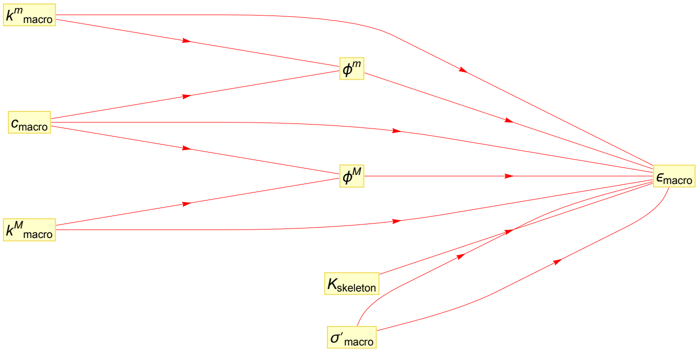


Figure 4.2: Sub-graph of the multi-scale multi-physics poromechanics problem for fluid-infiltrating media. Red arrow represents either a phenomenological relation or an operator that is defined not based on first principles.

as previously done in [122] where feed-forward neural network is used. Rather, they are taken into account as internal state by default such that the path-dependent behavior of the dual-permeability porous material can be replicated. In the dual-permeability problem, we identify that macroscopic strain, $\boldsymbol{\epsilon}_{\text{marco}}$ is the root and the effective stress $\boldsymbol{\sigma}'_{\text{marco}}$, permeability of the macroscopic and microscope pores, $\boldsymbol{k}^M_{\text{macro}}$, $\boldsymbol{k}^m_{\text{macro}}$, the mass exchange rate c_{macro}, and the bulk modulus of solid skeleton K_{skeleton}, as shown in Figure 4.2. Once the input and output are determined, the rest of the task is to determine the appropriate model that gives us the output prediction when a specific input is given. Notice that it is also possible that the ”material laws” of a multiphysical problem may also lead to multiple sub-graphs that share no vertex. In such sense, the procedure described above still applies, but the machine learning for each sub-graph will be independent to each other.

Another important observation is that it is not necessary to completely replace the sub-graph with data-driven model. For instance, one may use the conventional material law to connect the strain and strain history with stress but use experimental data to generate a data-driven model that predicts the permeability from strain history. The optimization of the choice of the edges for the hybrid approach, especially when it is subjected to noisy data is an important topic but is out of the scope of this study.

4.2 Preparation of databases for offline hierarchical supervised machine learning

In the previous section, we discuss the anatomy of the mathematical model represented in a directed graph, and the method to identify the components to insert data-driven model properly in a multi-physical problem without altering the hierarchy and connectivity of the physical quantities. In this section, our goal is to focus on how to use sub-scale data to enhance the predictions via an offline hierarchical bridging scale method. In particular, we will review the difference of online and offline hierarchical multiscale approaches and procedure of generating pre-computation databases for fast or real-time multiscale simulations.

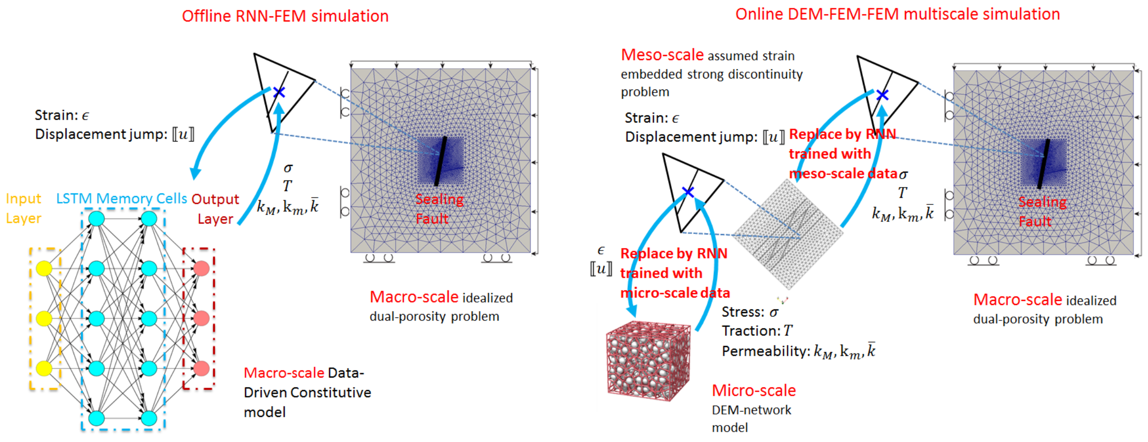

Offline RNN-FEM simulation
Strain: ϵ
Displacement jump: $[\![u]\!]$
σ
T
$k_M, k_m, \bar{k}$
Input Layer
LSTM Memory Cells
Output Layer
Sealing Fault
Macro-scale idealized dual-porosity problem
Macro-scale Data-Driven Constitutive model
Online DEM-FEM-FEM multiscale simulation
Meso-scale assumed strain embedded strong discontinuity problem
Strain: ϵ
Displacement jump: $[\![u]\!]$
Replace by RNN trained with meso-scale data
σ
T
$k_M, k_m, \bar{k}$
ϵ
$[\![u]\!]$
Replace by RNN trained with micro-scale data
Stress: σ
Traction: T
Permeability: $k_M, k_m, \bar{k}$
Micro-scale DEM-network model
Sealing Fault
Macro-scale idealized dual-porosity problem

Figure 4.3: Comparison between off-line pre-trained multiscale ANN-FEM simulations and online hierarchical multiscale simulations.

In an online hierarchical coupling approach in which simulations of different scales are linked, we might define macroscopic problems and microscopic problems and consider different technique to link the two problems in a hierarchical manner. For instance, the mechanical and hydro-mechanical responses of granular materials can be replicated by a DEM-FEM coupled model in which the macroscopic material laws are homogenized from discrete element simulations. By associating each integration point with an RVE, the coupling of the micro-problem (DEM) and macro-problem (FEM) is established by replacing macroscopic phenomenological relations with DEM simulations for each incremental time step. [142, 155, 88, 132, 242, 244]. This approach is nevertheless very expensive, as each constitutive update at each integration point would require an individual sub-scale DEM simulation performed on a different RVE at each incremental step. While parallel implementation is efficient for the hierarchical DEM-FEM coupling approach (as the DEM simulations can be embarrassingly parallel), the total number of required DEM simulations still grows proportionally with the mesh size used in the macroscopic problems. This cost becomes more profound when information across more than two scales become important, as shown in the dual-porosity, dual-permeability poromechanics problem illustrated in Figure 4.3.

On the other hand, the offline hierarchical coupling method does not directly use the DEM simulations **during** the macroscopic simulations. Rather, it involves an additional step in which the sub-scale simulations are used to generate a database. This database is then used to calibrate a surrogate model that is sufficiently efficient for macroscopic boundary value problems. The surrogate model can be simply a phenomenological model [108, 131], a reduced-order sub-scale model [68, 265, 266] or a data-driven model [111]. The offline hierarchical technique, if conducted properly, has at least two advantages. First, it costs less as the offline techniques as it does not need on-the-fly sub-scale simulations. Second, the offline treatment provides an opportunity for one to combine real and virtual data together to improve the accuracy of numerical simulations.

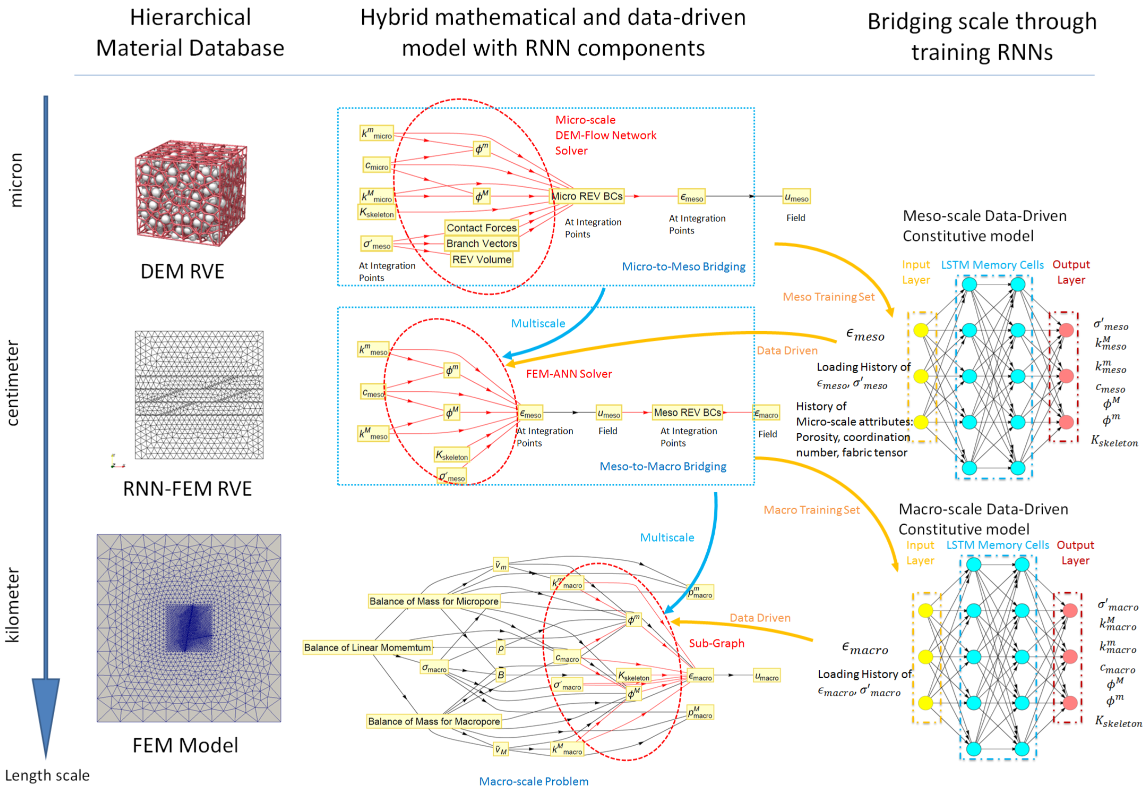


Figure 4.4: Hierarchy of a multi-scale multi-physics poromechanics problem for fluid-infiltrating media. Black arrow represents a definition or a ”universal principle”; red arrow represents either a phenomenological relation or an operator that is defined not based on first principles.

As shown in Figure 4.3, we will leverage these advantages to conduct a simulation that links the hydro-mechanical simulations of fractured porous media across three scales. In particular, our objective is to introduce a recursive training procedure where the database generated from small-scale simulations would be used to train a meso-scale RNN data-driven model, while the meso-scale RNN-FEM model will be used to generate another database to train the macroscopic data-driven model used for field scale simulations, as shown in Figure 4.4. Here we first assume that the principle of separation of scale is valid such that,

$$l_{\text{micro}} \leq l_{\text{meso}} \leq l_{\text{macro}}. \tag{4.1}$$

Previously, [244] has established a finite strain DEM-FEM coupling model to simulate two-phase poromechanics problem. Here we extend this work and focus on the case where intense localization due to damage or fractures occurs across the micro-, meso- and macroscopic scales. As a result, we require two sets of numerical simulations to generate the appropriate database to first link micro-mechanical DEM simulations to the meso-scale RVE, then again link the meso-scale DEM-FEM simulation to macroscopic scale field problem.

The first micro-mechanical simulation database consists of simulation results obtained from a DEM-network model in which DEM assemblies are subjected to different loading paths. The constitutive laws (traction-separation law and anisotropic permeability of macroscopic pore space) obtained from homogenizing the DEM responses are used as the data set for training and validating the neural network models. Here we assume that the mesoscale model employs a finite element discretization with displacement, pore pressure and their corresponding jumps as unknown in each incremental time step. Applying the effective stress principle, we postulate that there exists an effective stress such that it solely depends on the deformation and deformation history of the solid skeleton [222, 19].

In the infinitesimal regime, the Hill-Mandel micro-heterogeneity condition requires the volume average of the virtual power in the RVE to equal the virtual power done by the volume averages of power-conjugate stress and deformation measures. In terms of stress

$\boldsymbol{\sigma}'$ and infinitesimal strain $\boldsymbol{\epsilon}$:

$$\langle \boldsymbol{\sigma}' \rangle : \langle \dot{\boldsymbol{\epsilon}} \rangle = \langle \boldsymbol{\sigma}' : \dot{\boldsymbol{\epsilon}} \rangle \tag{4.2}$$

The traction $\langle \boldsymbol{T}'_{\Gamma} \rangle$ averaged in the RVE representing the interface is given by,

$$\langle \boldsymbol{T}'_{\Gamma} \rangle = \langle \boldsymbol{\sigma}' \rangle \cdot \boldsymbol{n}, \tag{4.3}$$

where $\boldsymbol{n}$ is a unit vector normal to the interface and $\boldsymbol{m}$ is a unit vector tangential to the interface. The average infinitesimal strain is defined in terms of $[\![\boldsymbol{u}]\!]$ and the initial height of the RVE h_0:

$$\langle \boldsymbol{\epsilon} \rangle = \operatorname{sym}(\frac{1}{h_0} [\![\boldsymbol{u}]\!] \otimes \boldsymbol{n}). \tag{4.4}$$

Thus, the Hill-Mandel lemma in the interface in terms of $\langle \boldsymbol{T}'_{\Gamma} \rangle$ and $[\![\boldsymbol{u}]\!]$ is given by:

$$h_0 \langle \boldsymbol{\sigma}' : \dot{\boldsymbol{\epsilon}} \rangle = \langle \boldsymbol{T}'_{\Gamma} \rangle \cdot [\![\dot{\boldsymbol{u}}]\!]. \tag{4.5}$$

In the infinitesimal regime, the time derivative of displacement jump reads,

$$[\![\dot{\boldsymbol{u}}]\!] = [\![\dot{u}]\!]_n \boldsymbol{n} + [\![\dot{u}]\!]_m \boldsymbol{m}. \tag{4.6}$$

The Hill-Mandel lemma for the interface therefore can be rewritten as,

$$h_0 \langle \boldsymbol{\sigma}' : \dot{\boldsymbol{\epsilon}} \rangle = \quad = \langle T_n \rangle [\![\dot{u}]\!]_n + \langle T_m \rangle [\![\dot{u}]\!]_m. \tag{4.7}$$

According to Eq. (4.3), effective traction in the normal and tangential direction can be written as,

$$\begin{aligned} \langle T_n \rangle &= \langle \boldsymbol{\sigma}' \rangle \cdot \boldsymbol{n} \cdot \boldsymbol{n} \\ \langle T_m \rangle &= \langle \boldsymbol{\sigma}' \rangle \cdot \boldsymbol{n} \cdot \boldsymbol{m} \end{aligned} \tag{4.8}$$

where the overall effective stress is given by,

$$\langle \boldsymbol{\sigma}' \rangle = \frac{1}{V_0} \sum_{c}^{Nc} \boldsymbol{f}^c \otimes \boldsymbol{l}^c \tag{4.9}$$

where V_0 is the initial volume of the RVE. $\boldsymbol{f}^c$ is the contact force at the grain contact $\boldsymbol{x} + \boldsymbol{y}_c \in \mathbb{R}^3$. $\boldsymbol{l}^c$ is the branch vector, the vector that connects the centroids of two grains forming the contact. N_c is the total number of particles in the RVE. Among the admissible boundary conditions fulfilling the Hill-Mandel micro-heterogeneity condition, we adopt the periodic boundary conditions [144], where the motion of a particle on the boundary of the RVE is characterized by, assuming rigid particles,

$$\boldsymbol{x}(\boldsymbol{X}) = \langle \boldsymbol{F} \rangle \cdot \boldsymbol{X}_c + \boldsymbol{w}_c + \boldsymbol{R} \cdot (\boldsymbol{X} - \boldsymbol{X}_c) \tag{4.10}$$

where $\boldsymbol{X}_c$ is the initial position of particle center, $\boldsymbol{w}_c$ is the displacement fluctuation and $\boldsymbol{R}$ describes the particle rotation. For a pair of particles on opposite boundaries $\partial\mathcal{V}^+$ and $\partial\mathcal{V}^-$, the periodicity enforces the periodicity of fluctuations and rotations

$$\boldsymbol{w}_c^- = \boldsymbol{w}_c^+, \quad \boldsymbol{R}^- = \boldsymbol{R}^+, \tag{4.11}$$

and the anti-periodicity of support forces and couples

$$\boldsymbol{a}_c^- = -\boldsymbol{a}_c^+, \quad \boldsymbol{m}_c^- = -\boldsymbol{m}_c^+, \tag{4.12}$$

where $\boldsymbol{a}_c$ is the opposite of the resultant force on the boundary particle exerted by other particles, $\boldsymbol{m}_c$ is the opposite of the resultant couple about the center $\boldsymbol{X}_c$ on the boundary particle.

The theoretical basis and the calculation of homogenized permeability has been previously studied in [62, 161, 209]. Assuming that the DEM assembly is used to model the strong discontinuity which often becomes flow conduit or flow barrier, we may again use a Hill-Mandel lemma corresponding to the Darcy's law to determine the effective permeability of the assembly, i.e., [62],

$$\langle \nabla p \cdot \boldsymbol{q} \rangle = \langle \nabla p \rangle \cdot \langle \boldsymbol{q} \rangle \tag{4.13}$$

Eq. 4.13 can be satisfied by the Dirichlet boundary condition in which the pore pressure difference across two opposite face is prescribed and Darcy's velocity is determined. As a result, the effective permeability tensor of a RVE can be determined via numerical flow experiment on the RVE. The fluid flux vector $\boldsymbol{q}$ within the RVE is computed when subjected to pressure gradient ∇p, and the permeability $\boldsymbol{k}_{\text{RVE}}$ is determined by Darcy's law

$$\boldsymbol{q} = -\frac{1}{\mu} \boldsymbol{k}_{\text{RVE}} \nabla p. \tag{4.14}$$

Among the solution strategies, the numerical solution of Stokes equations using finite element or Lattice-Boltzmann method yield accurate results but require large computational resources. This work resorts to the pore network flow model which simplifies the pores as nodes interconnected by edges allowing fluid flow [36, 40]. The first step of the homogenization procedure of permeability is the domain decomposition of the DEM sphere packing. This is achieved by well developed Delaunay triangulation and dual Voronoi graph algorithms [63, 40]. Using the particle centers as the triangulation nodes, the deformed micro-scale domain Ω_μ is decomposed into cells $\Omega_\mu = \bigcup_{i=1}^{N_t} \Omega_\mu^i$. Ω_μ^i is triangle in 2D analysis and tetrahedron in 3D analysis. The 2D concepts are adopted in the following descriptions. Each triangular cell Ω_μ^i encloses a pore space of volume V_v^i between three particles. The remaining solid space is the intersection of Ω_μ^i with the three particles and has the volume V_s^i. The dual domain decomposition of Ω_μ into Voronoi cells generates the pore network in the DEM assembly. Each node is regarded as the center of the pore space in a triangular cell Ω_μ^i and each edge serves as the flow pipe connecting two pore space centers.

The next step is to define the local conductance of each edge ("pipe") in the flow network. Consider an edge connecting the pore i and j of cells Ω^i_μ and Ω^j_μ, respectively. Suppose that the flux in this pipe is q^{ij} when i at position $\boldsymbol{x}^i$ has the pressure p^i and j at position $\boldsymbol{j}^i$ has the pressure p^j, and that the length of the pipe is $L^{ij} = ||\boldsymbol{x}^i - \boldsymbol{x}^j||_2$, the local conductance g^{ij} relates these quantities by,

$$\boldsymbol{q}^{ij} \cdot (\boldsymbol{x}^i - \boldsymbol{x}^j) = q^{ij} = \frac{g^{ij}}{\mu} \frac{p^i - p^j}{L^{ij}}. \tag{4.15}$$

Extensive studies have been conducted on defining g^{ij} in pore network models [175, 93]. This work adopts the definition in [40], in which a new method of determining hydraulic radius HR^{ij} of a cross-section of complex geometry is proposed. The local conductance admits the expression

$$g^{ij} = \alpha \; A^{ij} \; (HR^{ij})^2. \tag{4.16}$$

where α is a non-dimensional factor reflecting the pore throat shape ($\alpha = 0.5$ in this work), A^{ij} is the cross-sectional area of the throat. A domain Θ^{ij} around the throat between two pore spaces is defined based on the triangular cells and the dual Voronoi cells. Its volume is ϕ^{ij} and the total area of its boundaries is γ^{ij}. The hydraulic radius is given by

$$HR^{ij} = \frac{\phi^{ij}}{\gamma^{ij}}. \tag{4.17}$$

In the numerical example of this work, we adopt the micro-scale DEM assembly as the hydro-mechanical constitutive model for the sealing fault, which has an intrinsic permeability in the order of 10^{-14} m^2. This low level of permeability is observed in porous media that often has a porosity of 0.1-0.2. However, in the DEM numerical assembly, where mono-disperse spheres are adopted to represent the particles, this porosity is not possible to achieve under confining stress in the order of Mega Pascals. The lowest porosity of the

RVE is still higher than 0.3. This discrepancy comes from the poor representation of the actual shape of grains in real soils by the idealized spheres. Realistic and accurate porosity will be reached by using numerical particles with complex shapes, yet this requires more powerful DEM simulation tools and is out of the scope of this study. Hence the permeability obtained from the flow network simulation on the micro-scale DEM assembly is artificially scaled to the order of 10^{-14} m^2. Our focus lies on the path-dependent changes in permeabilities in the normal and tangential directions of the strong discontinuity, following the displacement jumps applied to the interface.

Once the DEM assembly generates a sufficiently large database, the database can be used to train the data-driven model. Typically one would like to test a large variety of different loading paths such that different responses (torsion, shear, stretch, compression, loading & unloading) can be anticipated. The exact content of the database is often determined after a trial-and-error procedure. The size of database strongly depends on the exact configuration, type and the training process used for the neural network. Understandably, it could be counterproductive to generate a large database for a small neural network. On the other hand, it also does not make sense to have a very deep and complex neural network design while the data available for the supervised training and validation are limited. The detailed description on the design of the neural network will be presented in the next section.

One may think of the trained meso-scale data-driven model as a representation of the data or as a surrogate model. However, the advantage of the neural network as surrogate model is that one may easily adjust the neural network configuration, whereas changing from one surrogate model to another often require a substantial amount of work to identify material parameters.

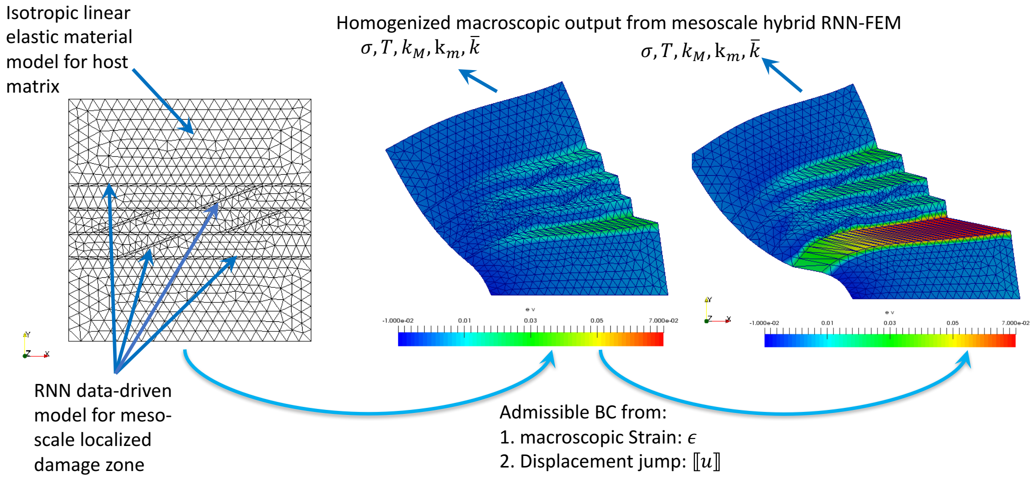

Isotropic linear elastic material model for host matrix
Homogenized macroscopic output from mesoscale hybrid RNN-FEM
$\sigma, T, k_M, \mathrm{k}_m, \bar{k}$
$\sigma, T, k_M, \mathrm{k}_m, \bar{k}$
e v
-1.000e-02
0.01
0.03
0.05
7.000e-02
e v
-1.000e-02
0.01
0.03
0.05
7.000e-02
Y
Z
X
RNN data-driven model for meso-scale localized damage zone
Admissible BC from:
1. macroscopic Strain: ϵ
2. Displacement jump: $[\![u]\!]$

Figure 4.5: The generation of database for macroscopic data-driven model using hybrid neural-network / finite element model. In the meso-scale simulations, we consider the localized damage zone as a dual-porosity material in which the data-driven model provides the traction-separation law and the flow prediction normal and orthogonal to the interface while the responses of the host matrix is captured by a simple elastic materials. The meso-scale RVEs are then subjected to various loadings, and the responses are recorded and used to train and validate the macroscopic data-driven model. In this figure, displacement field is scaled by a factor of 50.

After the completion of the training of the meso-scale data-driven model used to represent the strong discontinuity at the meso-scale, we then generate another set of RVE that uses the data-driven model as a replacement of the DEM model to capture the traction-separation law of the localized damage zones at the meso-scale. We then subject the meso-scale RVE to various loading paths and obtain the simulated responses from the the hybrid neural network/finite element model as shown in Figure 4.5. If there is no comparable experimental data available, then the simulated responses of the dual-porosity material constitute the database for the data-driven model used in field-scale problems.

4.3 Offline bridging scales via recurrent neural network

In a nutshell, machine learning refers to the ability of a computer to learn without being explicitly programmed. In the field of computational mechanics, machine learning has been widely used for finding the bases of the reduced dimensional space for reduced order modeling, and for replacing constitutive laws with trained artificial neural network. The latter tasks have found a various degree of success in previous work such as [77] and [122]. One key aspect that is critical for the application of the solid mechanics applications is that the machine learning process must be able to generate path dependent responses such that the strain, strain rate and strain history may all affect the resultant stress responses. The ability of replicating history dependent behavior is equally important for capturing the hydraulic responses. For instance, the effective permeability of a porous rock may be influenced by the damage of the host matrix. Furthermore, water retention curve, the relation between degree of saturation and suction, is known to be dependent on the wetting and drying history of the pores. In the poromechanics literature [104, 117], the hydro-mechanical path-dependent behavior has been enforced by additional input in a feed-forward neural network or support vector machine.

4.3.1 Deep learning with recurrent neural network

The recurrent neural network (RNN) is an umbrella term for artificial neural networks with connection topology possesses cycles [135]. In other words, the recurrent neural network considers data existing as sequences and the output of a layer in the previous step is added back as additional input and fed back into the same layer to produce the output (hence the name recurrent neural network).

The existence of the cycles leads to a profound difference, as it resembles how a history-dependent process evolve in time with cause-and-effect relationship (e.g. translating paragraphs of content between different languages, analysis of video surveillance). While classical artificial neural network can be regarded as a mathematical function, the recurrent neural is a dynamical system model of the biological neural networks that *possess memory* and is able to process arbitrary sequences of input and generate arbitrary sequences of output. These important features have made RNN versatile among many applications that require learning from temporal data such as speech recognition, machine translation, quick-type for smartphones and driver-less car technology. In structural engineering, RNN has been used to perform model-free structural analysis in which the structural behavior is predicted without any physical model [84]. Similar approach has been applied in geotechnical engineering in which an RNN is used to replicate stress-strain relation of sand subjected to monotonic triaxial compression loading [269].

In all the application mentioned above, the RNN machine learning procedure is often used to produce data-driven model that completely replace the constitutive models based on human interpreted knowledge. In this work, we take a different approach in which the machine learning is not used to generate model-free prediction but to be used for generating links for bridging simulations across scales in an offline fashion. Furthermore, we also retain the usage of the mechanics principles whenever possible in the multiphysical model conceptualized as a directed graph. Ultimately, the resultant model represents a hybridization of human- and machine-interpreted knowledge that can be used to generate predictions and as forecast engine. In the following subsections, we will describe the specific techniques we used to build the recurrent neural network and how it can be trained using a combination of data generated from experiments and micro-mechanical simulations.

4.3.2 Overcoming gradient vanishing or exploding issues with long short-term memory architecture

Despite the exceptional prediction power of RNN as forecast engine, RNN is known to be relatively vulnerable to the vanishing and exploding gradient problems. While a vanishing gradient may lead to the change of weight of the nodes in the recurrent neural network that has no significant changes on the error measured by the objective function or cost function. By contrary, an exploding gradient may lead to the error very sensitive to any small change of weights in the nodes. Both issues can be resolved for mechanics data-driven model produced by very small architecture with limited number of hidden layers [122, 105]. However, since (1) the usage of RNN leads to a larger number of nodes in the input layers, (2) the multiscale coupling scheme uses additional microstructural attributes as input, and more importantly (3) we intend to study the usage of deep learning which requires a sizable number of hidden layers, the vanishing or exploding gradients could be a significant issue.

Here we take advantages of a technique commonly used in computational linguistics called Long Short Term Memory (LSTM). First introduced by [96], the LSTM's major departure is the use of memory blocks, instead of the classical artificial neurons as nodes for RNN. Within a memory block, a new entity called ”gate” is introduced to control the flow of information and the state of the block, as shown in Figure 4.6.

A LSTM neuron possesses a state of the memory cells at time t C_t. Define x_t as the value of the input sequence at time t, and h_t as the value of the output sequence at time t.

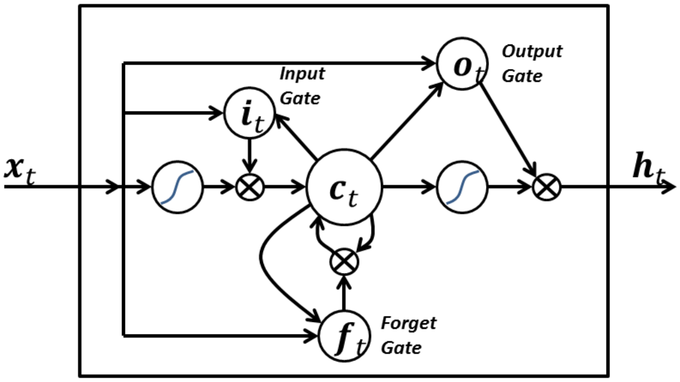


Figure 4.6: A Long-Short-Term-Memory neuron with input, output, and forget gate to process sequence with memory effect.

The signal through the forget gate is given by

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f), \tag{4.18}$$

where σ is the sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$, W_f and U_f are weight matrices, b_f is bias vector for the forget gate.

The new information to be stored in the cell state is given by the signal i_t through the input gate

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i), \tag{4.19}$$

where W_i and U_i are weight matrices, b_i is bias vector for the input gate.

The new candidate value cell state is given by a tanh layer

$$\widetilde{C}_t = \tanh(W_C \cdot x_t + U_C \cdot h_{t-1} + b_C), \tag{4.20}$$

where tanh is the hyperbolic tangent function $\tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$, W_C and U_C are weight matrices, b_C is bias vector.

The old cell state C_{t-1} is updated by the above forget and input information, i.e.,

$$C_t = f_t C_{t-1} + i_t \widetilde{C}_t \tag{4.21}$$

Finally, for the output signal

$$h_t = o_t \tanh(C_t) \tag{4.22}$$

where o_t is the signal through the output gate

$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o), \tag{4.23}$$

where W_o and U_o are weight matrices, b_o is bias vector for the output gate.

To showcase the advantage of using LSTM with micro-structure information as a part of input features, we examine the forward prediction capability of (1) the classical neural network "ANN" (2) LSTM neural network which is specially designed for memorizing sequences "LSTM" and (3) LSTM neural network that also memorizes micro-structure attributes "LSTM Microstructure Data". Numerical simple shear tests with loading-unloading under different confining pressure (σ = 50 MPa, 60 MPa, 70 MPa, 90 MPa and 100 MPa) are conducted on a discrete element assembly and serve as the training data to the three neural network models (Fig 4.7(a)). The "ANN" model design is similar to [77], where the inputs are the confining pressure σ, the shear strains γ_{n-1}, γ_n, γ_{n+1} and the shear stresses τ_{n-2}, τ_{n-1}, τ_n. The subscripts indicate the time steps t_{n-2}, t_{n-1}, t_n, t_{n+1}. The output is the shear stress τ_{n+1} at the next time step t_{n+1}.

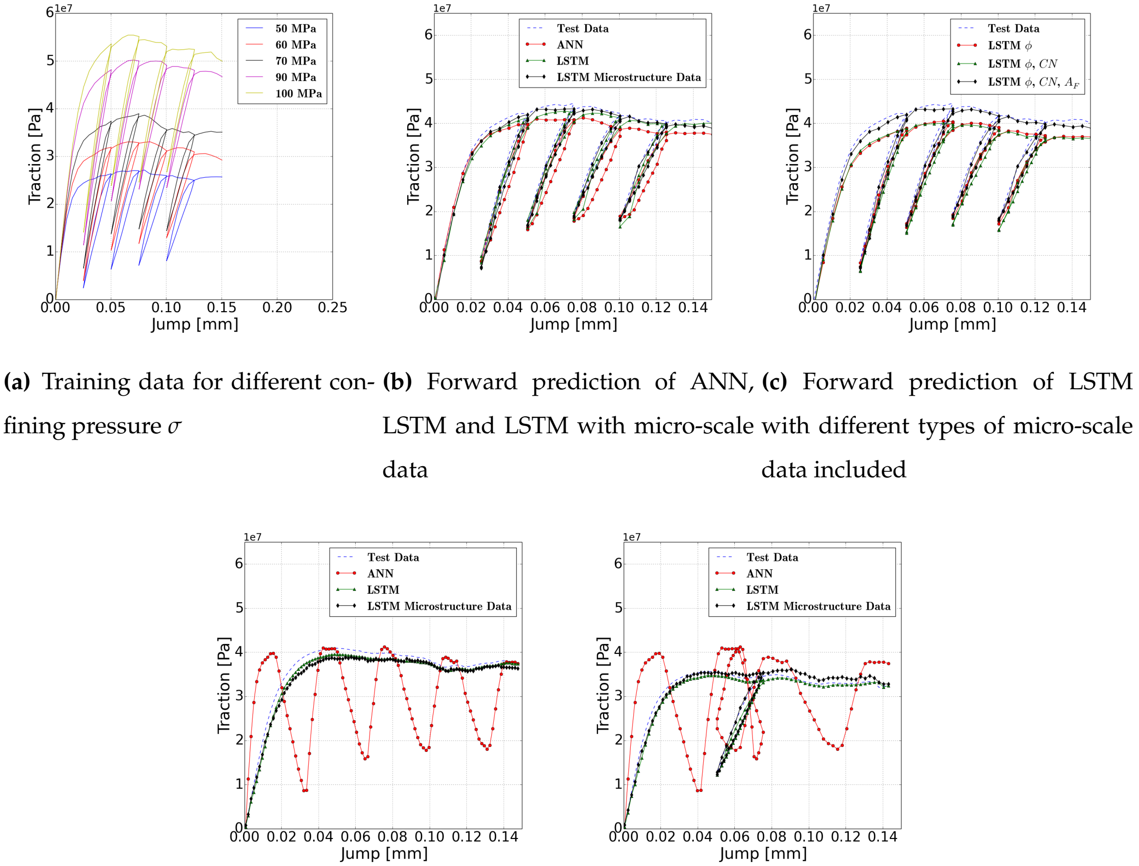
Different from the classical ANN model, LSTM neural network accepts sequences of history values of the physical parameters as inputs. Thus the input features now consist of the confining pressure σ, the sequence of history values of shear strains $[\gamma_{n-1}, \gamma_n, \gamma_{n+1}]$, and the sequence of history values of shear stresses $[\tau_{n-2}, \tau_{n-1}, \tau_n]$. The output features is again the shear stress τ_{n+1}. In addition to the strain and stress history, one key innovation we attempted in this chapter is to incorporate the evolution of microstructural attributes as additional input for the neural network. The micro-structure data adopted in this example are the porosity ϕ of the DEM assembly, the coordination number CN and the fabric tensor $\boldsymbol{A_F}$. $\boldsymbol{A_F} = \frac{1}{N_c}\sum_c \boldsymbol{n}_c \otimes \boldsymbol{n}_c$, where $\boldsymbol{n}_c$ is the normal of contact c. Here our goal is to check whether the incorporation of any of these additional data as input in the RNN network improves the prediction quality. If the answer is positive, it is likely that a human-derived phenomenological model could benefit from the inclusion of these physical quantities.

In a supervised machine learning setting, the LSTM neural network will be adjusted based on the portion of the dataset used for calibration. After the training or back propagation completes, the relationship between these averaged micro-scale attributes and the predicted stress state can be determined. For phenomenological models where history-dependent behavior is encoded in the evolution of internal variables, the influence of micro-structural attributes are often implicitly incorporated (except a few exception such as [52, 72]), this could be a difficult task.

All three neural network models have two hidden layers of 80 nodes, and dropout layers of rate 0.2 are placed after each LSTM layer. The sigmoid activation function is chosen for the output layer. We also set the same mean squared error as the training goal of all three models such that the errors are supposed to reduce to the same level (around 1e-5).

Fig 4.7(b) compares their forward predictions of the loading-unloading behavior under confining pressure of $\sigma = 80$ Mpa (not included in the training set). It is shown that the LSTM model performs better than the ANN model, in regard to the peak stress, softening and unloading-reloading cycles. Also, with micro-scale information, LSTM can yield closer prediction to the test data than LSTM that only process macroscopic strain and stress data.

Furthermore, determining how much and what types of micro-structure data to be in-



(a) Training data for different confining pressure σ

(b) Forward prediction of ANN, LSTM and LSTM with micro-scale data

(c) Forward prediction of LSTM with different types of micro-scale data included

(d) Forward prediction of monolithic loading

(e) Forward prediction of unloading

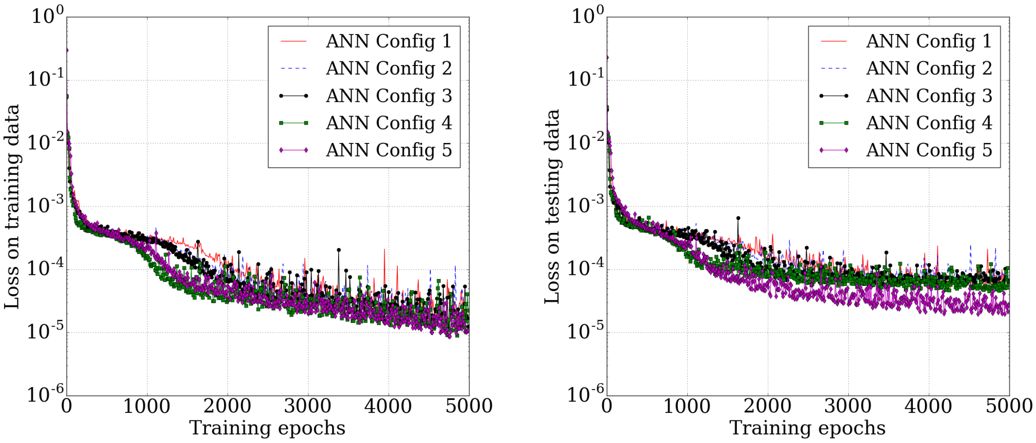
Figure 4.7: Comparison of forward prediction capacity between different configurations of neural network models.

corporated into machine learning model is a challenging task. As an example, we compare three designs of LSTM network (1) LSTM with ϕ (2) LSTM with ϕ, CN and (3) LSTM with ϕ, CN, $\boldsymbol{A_F}$ (Fig 4.7(e)). For (1) and (2), the additional information on average number of interactions per particle does not improve the generalization capability of the LSTM model. The model with the fabric tensor, which describes the matrix of the porous media, gives significantly more accurate results. This is due to the deformation mode of the DEM sample. The micro-structure are heavily distorted during the shearing. This is reflected in the change of principal values and rotation of principal directions of the fabric tensor. ϕ and CN, however, are hard to represent this induced anisotropy in the micro-structure because they are scalars. Thus in this example, the evolution history of the fabric tensor is crucial to the forward prediction capacity of the LSTM network.

Lastly, we study the performance of different LSTM architectures to determine the neural network parameters adopted in this work. Table 4.1 lists 5 neural network configurations that differ in the number of hidden layers, the number of neurons per hidden layer and the activation function for the output layer. The training data and testing data are the previous dataset from the numerical simple shear tests including the micro-scale attributes: porosity, coordination number and fabric tensor. The training data consists of 500 samples and the testing data contains 100 samples. The training phase consists of 5000 epochs and the batch size is 100. The loss function is the standard mean squared error. The value of loss on both training and testing data are recorded during the training epochs and are presented in Fig. 4.8. The performance curves show that all architectures can drive the training error down to the 10^{-5} level and the testing error down to the 10^{-4} level. Configurations 4 and 5 perform better in the training data and Configuration 5 is more accurate in predictions. The discrepancy is not significant. Thus for the neural network architecture used in this work, we choose the second configuration in Table 4.1, which gives good training and prediction performances. The number of LSTM neurons in the network is small so that it will speed up the training and the calculation in the triple-scale online simulations.

ANN Configuration	Number of hidden layers	Number of LSTM neurons per hidden layer	Activation function for output layer
1	2	50	Sigmoid
2	2	80	Sigmoid
3	2	100	Sigmoid
4	3	80	Sigmoid
5	2	80	ReLU

Table 4.1: Different ANN architectures for evaluation of training performance.



(a) Loss on training data

(b) Loss on testing data

Figure 4.8: Comparison of training performance of different ANN architectures in Table 4.1.

4.3.3 Highlights of Implementation

As for the implementation, we have leveraged Keras (cf. [42]), a high-level Python deep learning library, to build the LSTM neural networks and complete the training procedure. This model-level library allows for easy and fast prototyping of machine learning models.

The low-level operations (such as tensor calculus) for machine learning are handled by Tensorflow, an open-source symbolic tensor manipulation library developed by Google, Inc [1], serving as the ”backend engine” of Keras. One upshot of Tensorflow is that it has a flexible architecture based on data flow graphs that enable easy GPU accelerated training of various types of neural networks.

The building and training of the LSTM data-driven model in this chapter contains four steps. Firstly, the data acquired from lower-scale numerical simulations are preprocessed and converted to specific data structure compatible with the LSTM training and validation algorithms. The data of numerical simulations are stored in comma-separated values (CSV) file and are imported by an open-source Python data analysis library Pandas [140]. The data are split to input features and outputs. These data are of different scales: 10^6 for traction, 10^{-2} for jump, 10^{-1} for porosity and 10^{-14} for permeabilities. Thus, each sequence of input and output is re-scaled to be within $[0, 1]$ using the MinMaxScaler class in sklearn.preprocessing toolkit [171]. The input data structure that can be processed by the LSTM model must be an array of dimension 3, where the entries for the first dimension are the samples, the second dimension is the time history steps and the last dimension are the input features.

Secondly the multi-layered neural network is constructed with a few and easy-to-modify lines of codes. Keras offers a simple way to establish neural networks that incorporate input, LSTM, dropout, output layers. Adding or deleting a layer, modifying the number of nodes, changing the activation functions are very convenient thanks to the high-level library of Keras.

Then we are ready to launch the training epochs. We feed the LSTM model with the preprocessed input and output data. The back propagation algorithm will modify the weights of the neural network iteratively and the loss will be reduced to a small number (about 2e-6 in this work). The learning rate can be reduced when the convergence becomes slow. Finally, the performance of the fully-trained LSTM material model is assessed on a set of test data that has not been provided to the model in the training phase.

4.4 Numerical Experiments

In this section, we present a triple-scale simulation which links the grain-scale simulations, the meso-scale assumed strain simulations and the macroscopic fault simulator together, as shown in Figure 4.9.

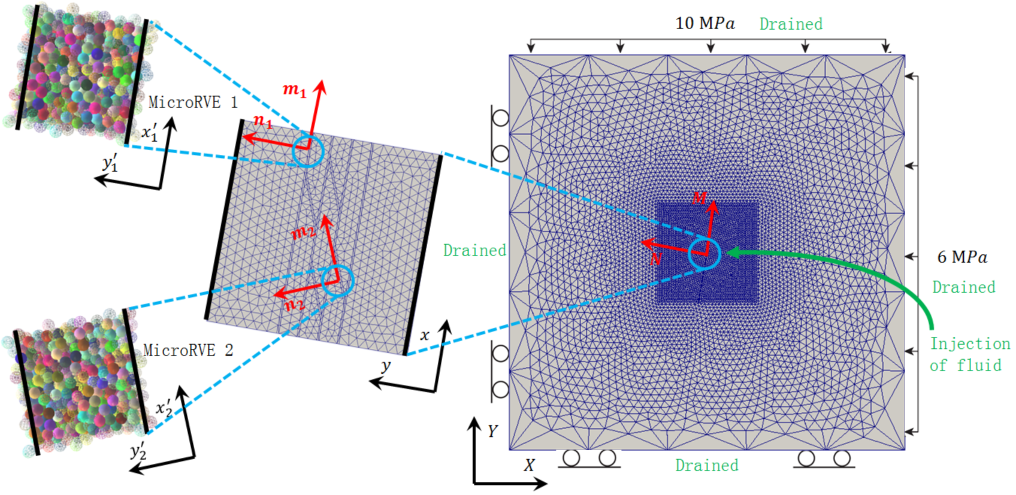


Figure 4.9: Triple-scale data-driven fault reactivation simulations. $\{X, Y\}$ constitutes the coordinate system of the macro-scale problem. $\{M, N\}$ constitutes the local coordinate system of the strong discontinuity in macro-scale. For meso-scale problem, the coordinate system is $\{x, y\}$ (co-axial to $\{M, N\}$), and the internal structure distinguishes two coordinate systems of interface: $\{m_1, n_1\}$ and $\{m_2, n_2\}$. The corresponding coordinate systems for micro-scale RVEs are $\{x'_1, y'_1\}$ and $\{x'_2, y'_2\}$.

Instead of directly replacing phenomenological laws with sub-scale simulations to generate constitutive responses as done in [244], we introduce a data-driven offline coupling method in which numerical results from sub-scale simulations first constitute material databases. These databases are then used to train the recurrent neural network models. Once the training and validation is completed, the neural network is then used to replace the phenomenological traction-separation law and the interface conductivity models of the dual-porosity systems. This process is applied recursively across length scales twice such that the responses of the fault is predicted by a data-driven model trained and validated by data set generated with another set of data-driven models at smaller scale.

As emphasized in [16], macroscopic responses of a material system are often dominated by the evolution of microstructural attributes, especially after the material bifurcation occurs. Yet, the traction-separation law and the conductivity law are often highly simplistic due to the difficulty to propose a proper model that captures the phenomenology. By incorporating the micro-structural information via deep learning, more realistic and complex constitutive laws can be generated automatically such that more accurate simulations of the localized responses can lead to more reliable macroscopic predictions.

In the following simulations, we assume that embedded strong discontinuities are pre-existed and do not propagate. In total, we construct two material databases. One contains the material responses of the DEM-network simulations that replicate the grain-scale interface between two bulk materials. In the second material database, the data are obtained via running RNN-FEM simulations where the neural network trained by the DEM-network database are re-used to model strong discontinuities, while the bulk material is idealized as isotropic elastic material. This recursive training strategy allows one to use machine learning as a mean to incorporate sub-scales information in an offline material. As a result, the triple-scale simulation only requires grain-scale material parameters for the DEM and flow network simulations as well as the material parameters used to model the bulk responses. All the path-dependent behaviors are therefore originated from the meso-scale interfaces and the macroscopic fault. The material parameters used in the numerical example are summarized in Table 4.2.

Scale & Model	Parameter	Value
Grain-scale DEM	Particle Young's modulus E	0.5 GPa
Grain-scale DEM	Particle Poisson's ratio ν	0.3
Grain-scale DEM	Particle Friction Angle	$\frac{\pi}{6}$
Grain-scale DEM	Particle density	2600 kg/m^3
Grain-scale DEM	Particle mean diameter	5 mm
Meso-scale FEM	Young's modulus	0.2 GPa
Meso-scale FEM	Poisson's ratio ν	0.2
Meso-scale FEM	Intrinsic permeability κ	$2e^{-14}$ m^2
Meso-scale FEM	Dynamic viscosity μ	$1e^{-3}$ Pa · s
Marco-scale FEM	Young's modulus	0.2 GPa
Marco-scale FEM	Poisson's ratio ν	0.2
Marco-scale FEM	Porosity of macropore ϕ_M	0.1
Marco-scale FEM	Porosity of micropore ϕ_m	0.2
Marco-scale FEM	Intrinsic permeability of macropore k_M	$1e^{-12}$ m^2
Marco-scale FEM	Intrinsic permeability of micropore k_m	$5e^{-17}$ m^2
Marco-scale FEM	Parameter of mass transfer $\overline{\alpha}$	$\rho_f * k_m$
Marco-scale FEM	Dynamic viscosity μ	$1e^{-3}$ Pa · s

Table 4.2: Material parameters for the grain-, meso- and macro-scale poromechanics problem with embedded strong discontinuities across three length scales.

The initial and boundary conditions of the macroscopic 2D fault reactivation problem is shown in Figure 4.9. Note that this initial boundary value problem is a highly simplified model used for demonstration and testing purpose. A more dedicated case study intended to capture the actual complex operations of fluid injection in an actual field will be conducted in the future but is out of the scope of this study. The size of the macroscopic domain is 1km × 1 km and it is assumed that the field is under plane strain condition. To simulate an anisotropic stress condition, the traction applied on the two opposite faces of the square domain is 10 MPa and 6 MPa accordingly. These values are held constant during the simulations. Meanwhile, we prescribe the Darcy's velocity of the macropores at the injection well about 250 meters from the fault line. To test the capacity of the data-driven model and to generate path-dependent responses at the field-scale level, this Darcy's velocity is not held constant but allowed to change over time, with the initial Darcy's velocity equals to 50m/s.

4.4.1 Training and validation of material laws for meso-scale interface

To train the recurrent neural network such that it can replicate the meso-scale hydro-mechanical responses of the embedded strong discontinuities, we first conduct 21 grain-scale simulations. The time history of the traction, displacement jump, and permeability in the normal and tangential direction, as well as the major and minor principal values of the fabric tensor are recorded. 16 of the simulation results are used as the training data set and rest 5 of them are used as the validation data set.

In each RVE simulation, the displacement boundary conditions are prescribed as shown in Figure 4.10. The DEM RVE is of the size of 10cm X 10cm X 5cm, while the averaged grain diameter is 0.5cm. The micro-scale traction-separation law and the relation between the micro-structure and the permeability tensor on the interface are homogenized from a micro-scale RVE of discrete element particles. A set of displacement jump paths $\{u_n, u_s\}$

are applied to the micro-scale RVE, and the tractions $\{t_n, t_s\}$ are homogenized at each incremental deformation step. Furthermore, at each incremental step, we also construct a flow network inferred from the deformed configuration of the DEM assembly and use an inverse problem to compute the effective permeability in the tangential and normal directions, as shown in Figure 4.10.

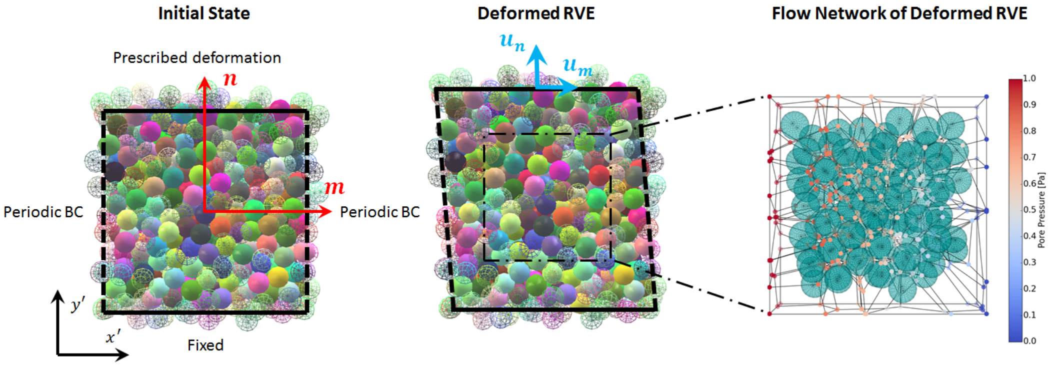


Figure 4.10: Micro-scale RVE. The initial configuration of the granular assembly (LEFT), the deformed configuration of the granular assembly (MIDDLE), and the flow network generated from the deformed configuration used to predict the anisotropic effective permeability (RIGHT).

Before the displacement-driven grain-scale simulation begins, the DEM assembly must be in the stress state consistent to the macroscopic boundary condition. This is achieved by subjecting the DEM assembly with the right amount of shear and normal tractions along the boundaries.

The initial state of the micro-scale RVE is determined by the initial state of the macro-scale problem. The macro-scale fault with the inclination angle of 80° is under a confining pressure of 6 MPa in the X direction and 10 MPa in the Y direction.

$$\sigma^{\text{Init}}_{\text{macro}} = \begin{bmatrix} -6 & 0. \\ 0. & -10 \end{bmatrix}_{XY} \text{MPa}, \tag{4.24}$$

where the subscript XY refers to the frame depicted in Figure 4.9. To introduce the proper initial stress state to the DEM assemblies, we first express this stress tensor in the local frame of the meso-scale RVE as depicted in Figure 4.9 via coordinate transformation, i.e., $\boldsymbol{\sigma}^{xy} = \boldsymbol{R}^{\mathrm{T}} \cdot \boldsymbol{\sigma}^{XY} \cdot \boldsymbol{R}$ such that,

$$\sigma^{\text{Init}}_{\text{meso}} = \begin{bmatrix} -9.88 & -0.68 \\ -0.68 & -6.12 \end{bmatrix}_{xy} \text{MPa}, \tag{4.25}$$

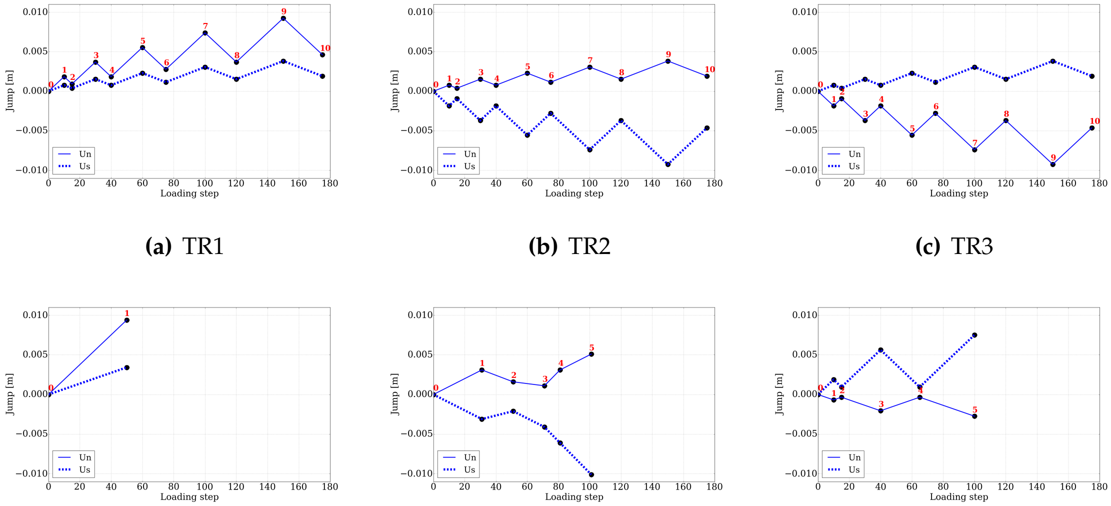
where the subscript xy refers to the rotated frame for the meso-scale RVE. Note that since the DEM assemblies are aligned with the strong discontinuities in the meso-structures, the stress state is re-expressed in the two local coordinate systems such that the correct traction can be applied to the DEM assemblies to generate the correct initial stress state. Recall that the two local coordinate systems are 20 degrees apart from each other (one 10 degrees clockwise to the meso-scale frame, one 10 degrees counterclockwise to the meso-scale frame). As a result, we have,

$$\sigma^{\text{Init}}_{\text{microRVE1}} = \begin{bmatrix} -9.88 & -0.68 \\ -0.68 & -6.12 \end{bmatrix}_{x_1'y_1'} \text{MPa} \; ; \; \sigma^{\text{Init}}_{\text{microRVE2}} = \begin{bmatrix} -9.88 & 0.68 \\ 0.68 & -6.12 \end{bmatrix}_{x_2'y_2'} \text{MPa}. \tag{4.26}$$

Once the initial stress state of the DEM assemblies are set, we then run multiple simulations and collect the results to form the database for supervised machine learning. The choice of loading cases to be included in the training data set is often based on empirical knowledge. In this work, we adopt such design of proportional loading paths: in each loading case the ratio between the normal displacement u_n (along the unit vector $\boldsymbol{n}$ in Fig. 4.10) and the tangential displacement u_s (along the unit vector $\boldsymbol{m}$ in Fig. 4.10) remains a constant. In total the training data set contains 16 ratios: $\frac{u_n}{u_s} = \frac{i\pi}{8}, i = 0, 1, 2, ..., 15$. Similar proportional loading paths have also been used to train constitutive laws for bulk materials in [122]. In order for the data-driven model to learn the path-dependent behavior of the interface, we, for each loading ratio in our cases, prescribe the displacement such that , the norm of the displacement $u = \sqrt{u_n^2 + u_s^2}$ are prescribed with the following loading-unloading sequences: u first increases to 0.2 of the maximum displacement magnitude 0.01 m, then decreases to 0.1 of 0.01 m, and rise again to 0.4, then to 0.2, 0.6, 0.3, 0.8, 0.4, 1.0, 0.5. Note that, this design of training data set is suitable (but not necessarily optimized) for data-driven model used in finite element simulations in which the deformation paths of strong discontinuities are not known a priori. In many simulation cases, the major deformation paths could be anticipated. For instance, in shear band simulations, the shear effects predominate over the opening or closing of the interface. Thus the training data set to be constructed for these simulations should incorporate more shear-dominate loading paths. Nevertheless, the optimization of training data set is a challenging task and will be studied in future work. 5 additional loading paths for testing are also constructed. Some of them are monotonic loading, some have different loading-unloading sequences, and in some cases the ratio $\frac{u_n}{u_s}$ is not constant. Figure 4.11 shows a portion of the loading paths designed for machine learning. For brevity, we did not include all the available simulations in the database in the chapter. Instead, we only show the results of 3 training sets and 3 testing sets, which are denoted as TR1, TR2 and TR3 and TE1, TE2 and TE3 respectively. Nevertheless, the training and validation algorithm as well as the database itself will be made available in an open source repository. The discrepancy between the data from micro-scale DEM simulation X_{DEM} and the results predicted by LSTM neural network X_{LSTM} is quantified by the scaled mean squared error given by

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}[\mathrm{MinMaxScaler}(X_{DEM_i}) - \mathrm{MinMaxScaler}(X_{LSTM_i})]^2, \tag{4.27}$$

where N is the number of data points. X_{DEM} and X_{LSTM} are re-scaled to be within the range $[0, 1]$ using the MinMaxScaler as described in the data preprocessing for machine learning in Section 4.6.



(a) TR1

(b) TR2

(c) TR3

(d) TE1

(e) TE2

(f) TE3

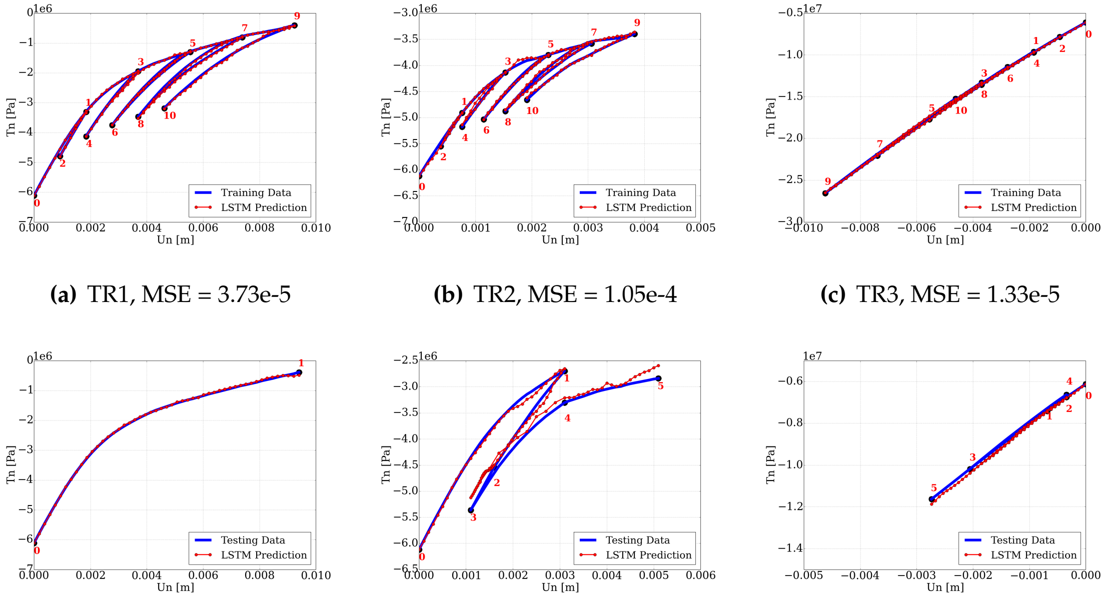
Figure 4.11: Loading path of three selected training cases TR1, TR2, TR3 and three selected testing cases TE1, TE2, TE3 on the micro-scale RVE 1. u_n and u_s are the normal and tangential displacement jumps. The coordinate system is $\{m_1, n_1\}$ (or $\{x'_1, y'_1\}$) depicted in Fig. 4.9. It can be seen that TR1 and TR2 represent tensile-shear loading cases (as u_n is positive) and TR3 represents a compressive-shear loading case (u_n negative). The numbers mark the sequence of loading-unloading cycles.

The physical parameters for the input of the LSTM neural network are the sequence of history values at time [t_{n-1}, t_n, t_{n+1}] of the normal and tangential components of displacement jump, the sequence of history values at time [t_{n-2}, t_{n-1}, t_n] of the normal and tangential components of traction, and the sequence of history values at time [t_{n-2}, t_{n-1}, t_n] of the maximum and minimum principal values of the fabric tensor of the DEM RVE. The outputs of the LSTM neural network are the normal and tangential components of traction at time t_{n+1}, the maximum and minimum principal values of the fabric tensor at time t_{n+1}, and the permeabilities normal and tangential to the strong discontinuity at time t_{n+1}.

Since in many engineering applications, the flow injection rate is transient and changes with time, the data-driven traction-separation laws must be able to capture the resultant combined isotropic and kinematic hardening mechanisms. Figures 4.12 and 4.13 show the comparisons between the DEM simulations and the simulated mechanical responses generated from the recurrent neural network in the normal and tangential directions. Except for the testing case TE2 in which there are notable discrepancy when the thin DEM layers are reloaded, the meo-scale data-driven traction-separation law is able to replicate both the cyclic and monotonic loading responses with negligible errors. Remarkably, this is achieved without using any internal variables to capture the history-dependent effect. Furthermore, we also show that the predicted responses are able to simulate both the damage-plastic flow and the elastic unloading in the cyclic responses. This coupled damage-plastic response is attributed to the evolution of the fabric tensors.

Figures 4.14 and 4.15 show the maximum and minimum eigenvalue of the fabric tensors following the prescribed displacements obtained from DEM and from the RNN predictions. The RNN generated responses are able to deliver very accurate predictions of the fabric tensor evolution. This good match is important for predicting induced anisotropy and may explain why the traction predictions in Figures 4.12 and 4.13 match well with the database.

The predictions of normal and tangential permeabilities following the prescribed displacements are shown in Figures 4.16 and 4.17 respectively. Again, with the help of characteristic microstructure information, the match is satisfying. To sum up, the trained data-driven model is capable of representing the micro-scale DEM-flow network model,



(a) TR1, MSE = 3.73e-5 **(b)** TR2, MSE = 1.05e-4 **(c)** TR3, MSE = 1.33e-5

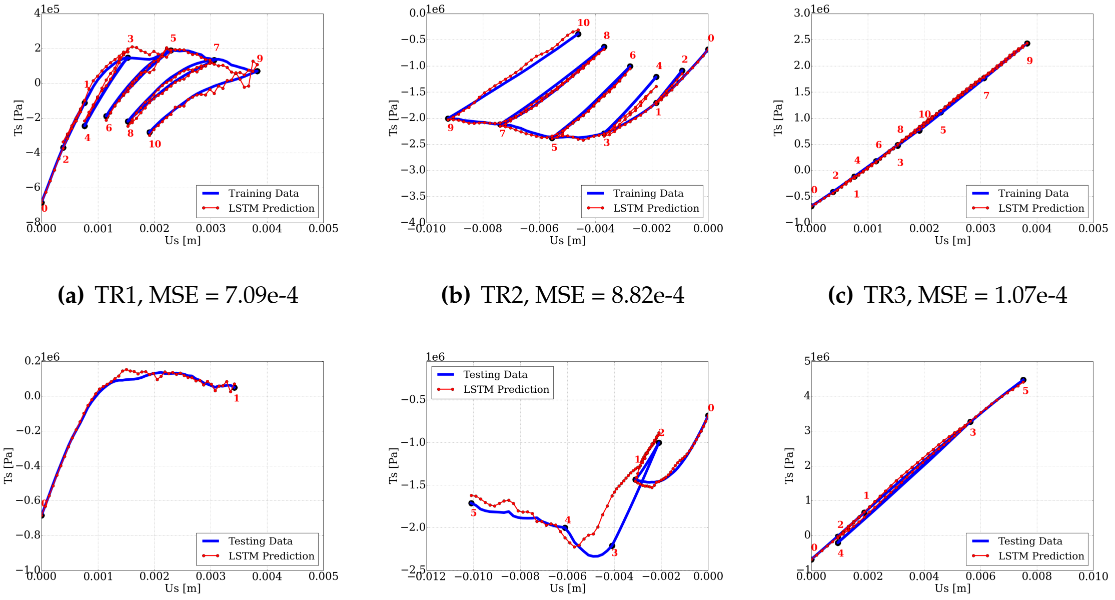
(d) TE1, MSE = 2.62e-5 **(e)** TE2, MSE = 1.21e-3 **(f)** TE3, MSE = 7.11e-4

Figure 4.12: Comparison of the micro-scale DEM simulation data and the trained meso-scale data-driven model. Normal traction against normal displacement jump for the selected training and testing cases. The numbers mark the sequence of loading-unloading cycles. MSE refers to the scaled mean squared error defined in Eq. 4.27.

in terms of traction, permeability, and invariants of fabric tensor. It is ready to be used as constitutive law for the strong discontinuity of the meso-scale RVE in the FEM-LSTM coupled simulations.

4.4.2 Training and validation of material laws for dual-porosity fault

The path-dependent constitutive model governing the displacement jump induced traction and permeability changes in the macroscopic sealing fault is provided by the macro-scale data-driven LSTM model. The data used to train and test this model are generated from multiscale simulations of the meso-scale RVE, where the interface behavior comes from the micro-scale RVE. The multiscale model is FEM-LSTM coupled, using the data-



(a) TR1, MSE = 7.09e-4

(b) TR2, MSE = 8.82e-4

(c) TR3, MSE = 1.07e-4

(d) TE1, MSE = 5.98e-4

(e) TE2, MSE = 1.49e-2

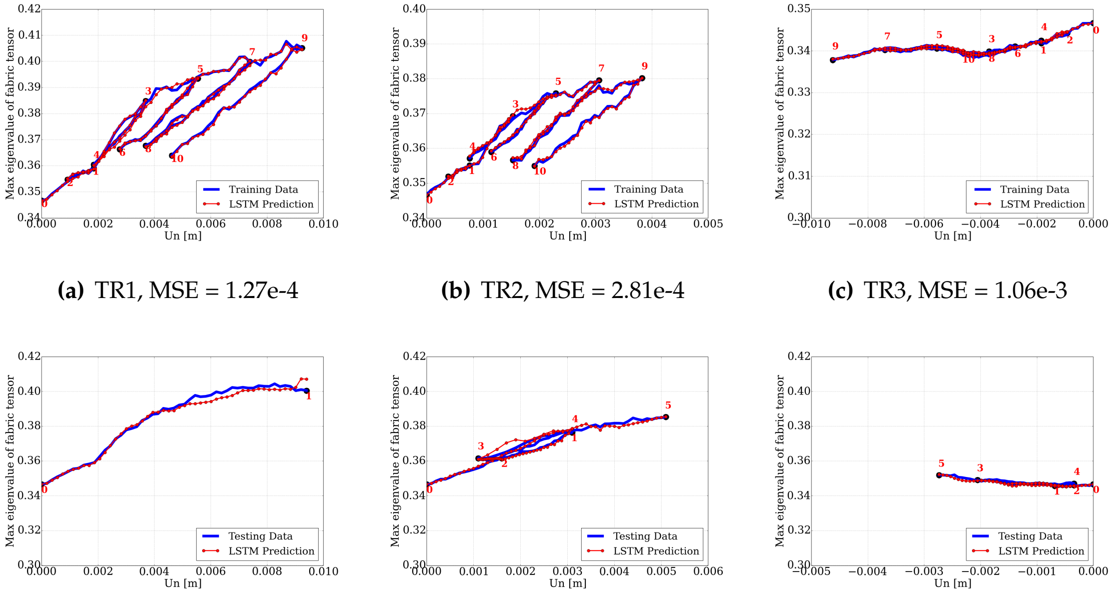
(f) TE3, MSE = 4.66e-4

Figure 4.13: Comparison of the micro-scale DEM simulation data and the trained meso-scale data-driven model. Tangential traction against tangential displacement jump for the selected training and testing cases. The numbers mark the sequence of loading-unloading cycles. MSE refers to the scaled mean squared error defined in Eq. 4.27.

driven model trained in the previous section.

In each RVE simulation, the displacement boundary conditions are prescribed as shown in Figure 4.18. The meso-scale RVE is 2D, and has the size of 1m X 1m. A set of displacement jump paths $\{u_n, u_s\}$ are applied to the meso-scale RVE, and the tractions $\{t_n, t_s\}$ are homogenized at each incremental deformation step. Furthermore, at each incremental step, we also conduct an inverse problem to compute the effective permeability in the tangential and normal directions, as shown in Figure 4.18.

It is important to investigate how well the FEM-LSTM coupled scheme represents the FEM-DEM multiscale scheme, where the interface constitutive law comes from the DEM assembly from the previous section. An example of comparison is presented, where the



(a) TR1, MSE = 1.27e-4

(b) TR2, MSE = 2.81e-4

(c) TR3, MSE = 1.06e-3

(d) TE1, MSE = 1.44e-3

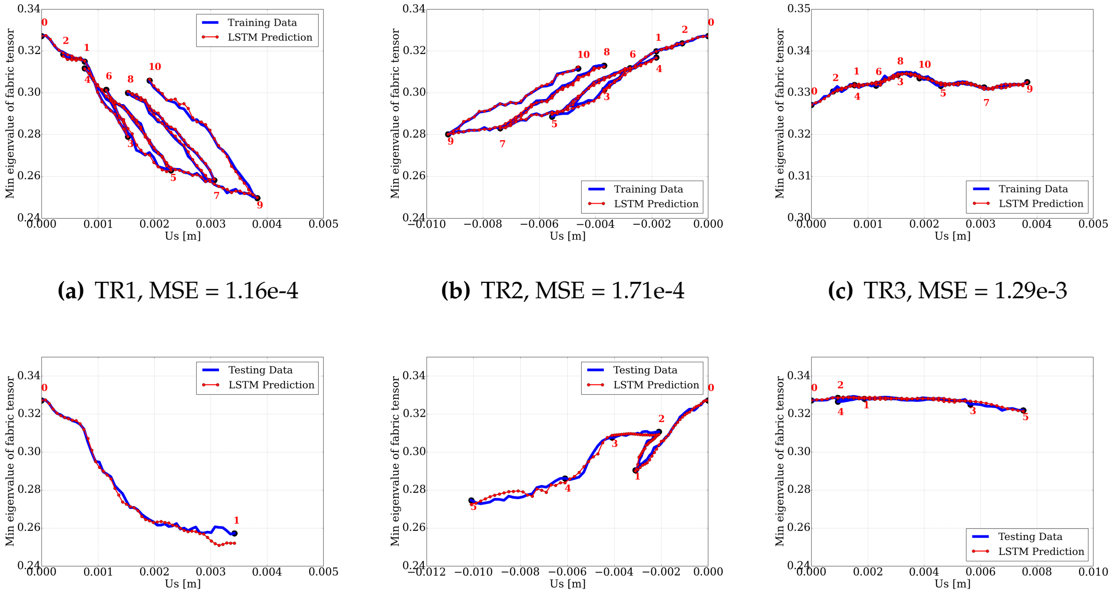
(e) TE2, MSE = 1.26e-3

(f) TE3, MSE = 1.38e-3

Figure 4.14: Comparison of the micro-scale DEM simulation data and the trained meso-scale data-driven model. Maximum eigenvalue of fabric tensor against normal displacement jump for the selected training and testing cases. The numbers mark the sequence of loading-unloading cycles. MSE refers to the scaled mean squared error defined in Eq. 4.27.

meso-scale RVE is subjected to a displacement loading path with unloading. Fig. 4.19 compares the tractions in the normal and tangential directions following the prescribed displacements. The results are close to each other. Thus FEM-LSTM model could approximately represent the FEM-DEM model in the generation of a database. Another alternative is using hybrid database. In other words, a portion of the data is from FEM-LSTM model simulations, while the other potion is from FEM-DEM model. In the extreme case, all data are from FEM-DEM model, then the numerical example in this work will be two-scale data-driven simulation, instead of triple-scale.

We then run multiple meso-scale simulations and collect the results to form the database for supervised machine learning. The design of the training and testing data set is simi-



(a) TR1, MSE = 1.16e-4

(b) TR2, MSE = 1.71e-4

(c) TR3, MSE = 1.29e-3

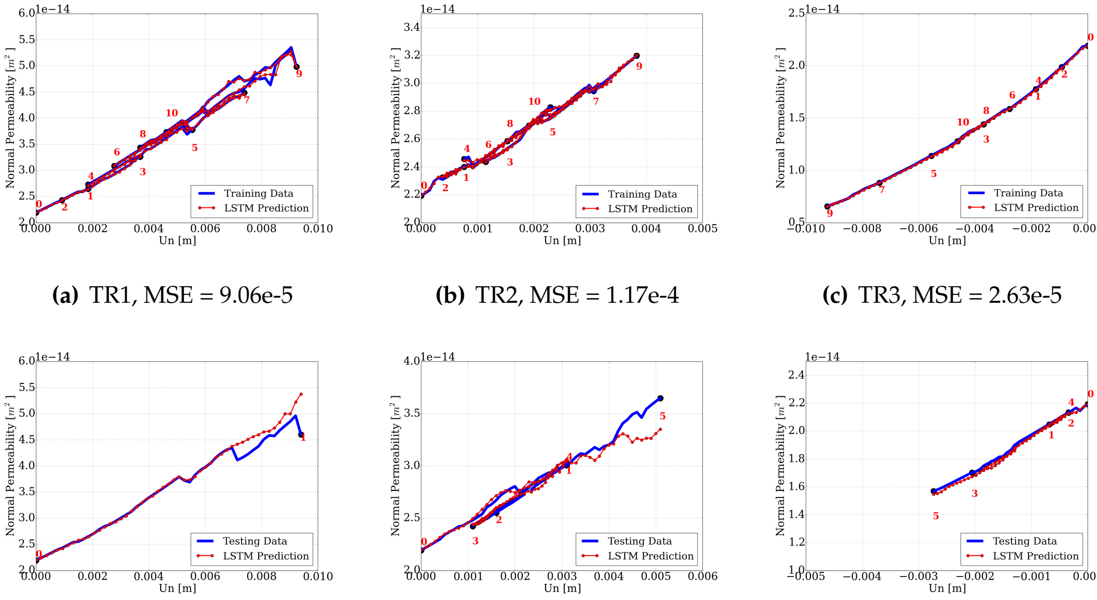
(d) TE1, MSE = 1.45e-3

(e) TE2, MSE = 7.73e-4

(f) TE3, MSE = 9.61e-3

Figure 4.15: Comparison of the micro-scale DEM simulation data and the trained meso-scale data-driven model. The minimum eigenvalue of fabric tensor against tangential displacement jump for the selected training and testing cases. The numbers mark the sequence of loading-unloading cycles. MSE refers to the scaled mean squared error defined in Eq. 4.27.

lar to the design in the previous section. In the meso-scale RVE, there is no definition of fabric tensor, thus the input data only consists of the displacement jumps and tractions in normal and tangential directions. Figure 4.20 shows a portion of the loading paths: TR1, TR2 and TR3 in the training sets and TE1, TE2 and TE3 in the testing sets. The physical parameters for the input of the LSTM neural network are the sequence of history values at time $[t_{n-1}, t_n, t_{n+1}]$ of the normal and tangential components of displacement jump, and the sequence of history values at time $[t_{n-2}, t_{n-1}, t_n]$ of the normal and tangential components of traction. The outputs of the LSTM neural network are the normal and tangential components of traction at time t_{n+1}, and the permeabilities normal and tangential to the



(a) TR1, MSE = 9.06e-5

(b) TR2, MSE = 1.17e-4

(c) TR3, MSE = 2.63e-5

(d) TE1, MSE = 2.33e-3

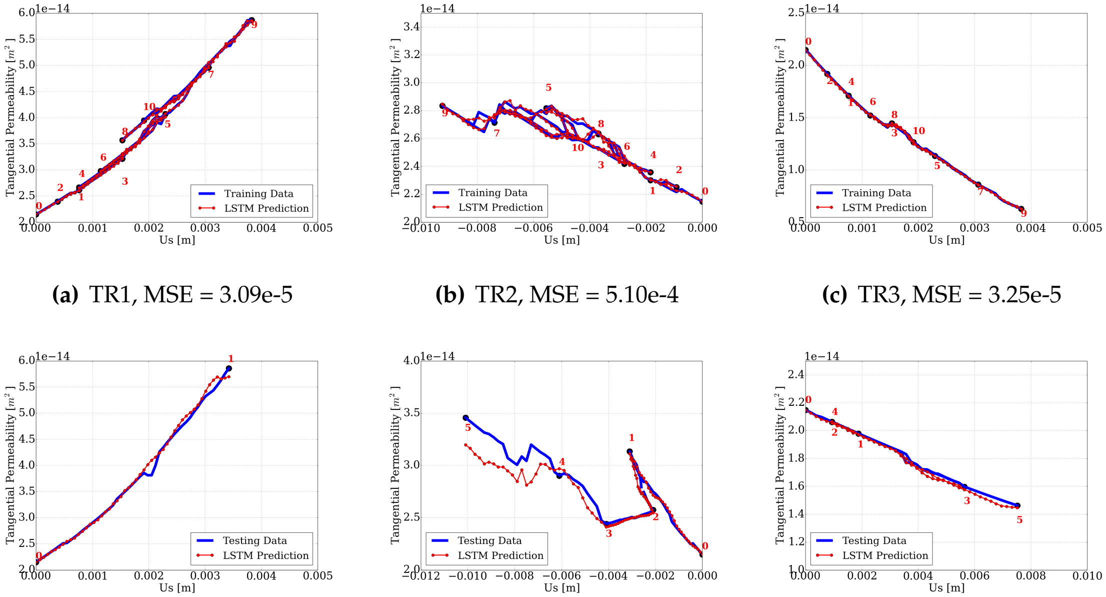
(e) TE2, MSE = 3.10e-3

(f) TE3, MSE = 9.07e-4

Figure 4.16: Comparison of the micro-scale DEM simulation data and the trained meso-scale data-driven model. Normal permeability against normal displacement jump for the selected training and testing cases. The numbers mark the sequence of loading-unloading cycles. MSE refers to the scaled mean squared error defined in Eq. 4.27.

strong discontinuity at time t_{n+1}.

Figures 4.21 and 4.22 shows the comparison between the FEM-LSTM simulations and the simulated mechanical responses generated from the recurrent neural network in the normal and tangential directions. The predicted responses are able to simulate both the damage-plastic flow and the elastic unloading in the cyclic responses. The predictions of normal and tangential permeabilities following the prescribed displacements are shown in Figures 4.23 and 4.24 respectively. The trained data-driven model is ready to be used as constitutive law for the strong discontinuity of the macro-scale problem.



(a) TR1, MSE = 3.09e-5

(b) TR2, MSE = 5.10e-4

(c) TR3, MSE = 3.25e-5

(d) TE1, MSE = 4.97e-4

(e) TE2, MSE = 6.57e-3

(f) TE3, MSE = 8.32e-4

Figure 4.17: Comparison of the micro-scale DEM simulation data and the trained meso-scale data-driven model. Tangential permeability against tangential displacement jump for the selected training and testing cases. The numbers mark the sequence of loading-unloading cycles. MSE refers to the scaled mean squared error defined in Eq. 4.27.

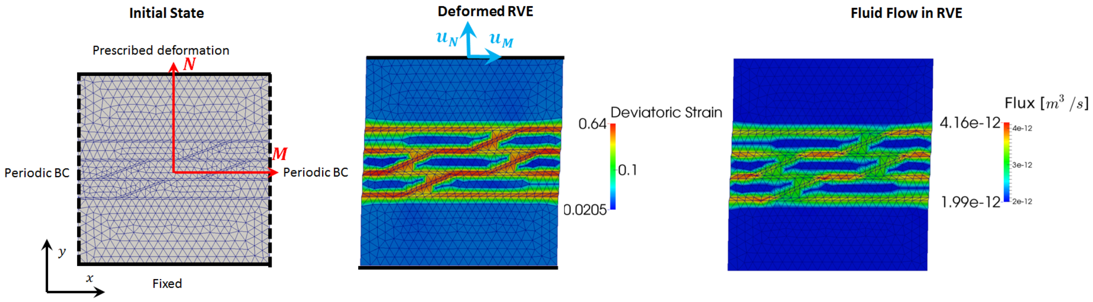
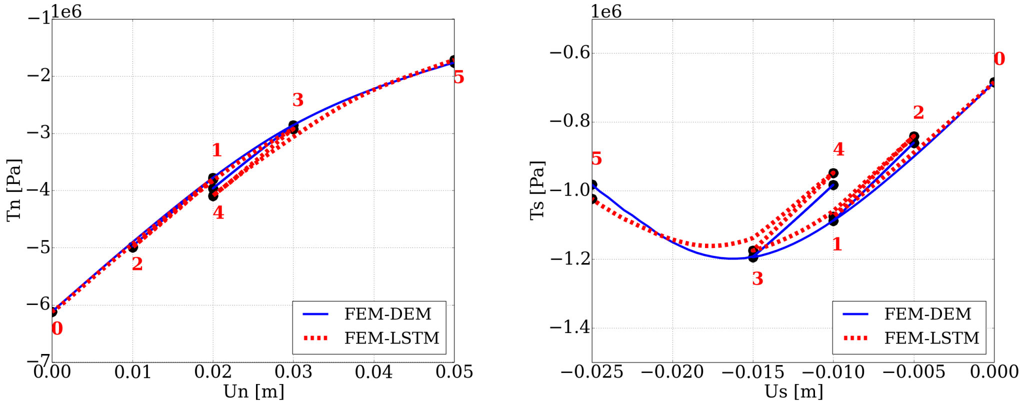


Figure 4.18: Meso-scale RVE. The initial configuration of the meso-scale RVE and its pre-embedded interfaces (LEFT), the deformed configuration and the deviatoric strain field of the meso-scale RVE (MIDDLE), and the fluid flux calculation used to predict the anisotropic effective permeability (RIGHT).



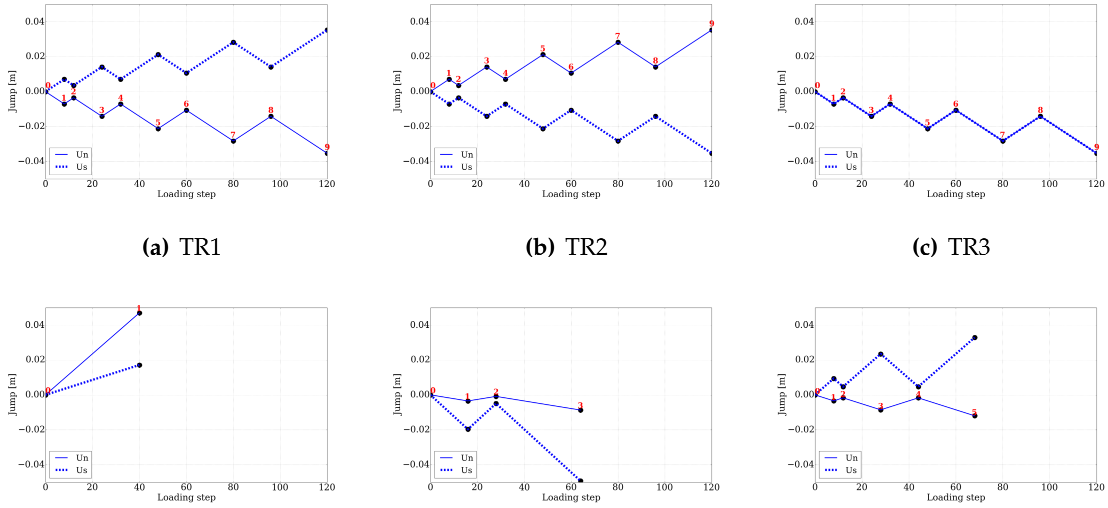
(a) Normal traction

(b) Tangential traction

Figure 4.19: Comparison of FEM-LSTM coupled model and FEM-DEM coupled model for meso-scale RVE. The numbers mark the sequence of loading-unloading cycles.

4.4.3 Simulation of macro-scale fault reactivation problem

Water is injected to the macro-scale field through the source S located to the right of the sealing fault. The distance between S and the fault is about 250 m (Fig. 4.26(a)). The prescribed Darcy velocity at the source is shown in Fig. 4.25(a). The injection profile is composed of three injection-pause cycles, where water supply is provided for 40 hours under constant rate of 50 m/s, followed by a pause for 10 hours before the next cycle of injection. We simulate the hydro-mechanical dual-porosity problem with the traction-separation law and macropore permeability tensors along the sealing fault given by the meso-scale data-driven model. The pore pressure in both scales at the source S is presented in Fig. 4.25(b). The fluid is injected to the macropore space. Upon injection or pause, the macropore injection pressure jumps up or plunge immediately, while the micropore pressure at the injection point has the opposite behavior. This is caused by the low mass transfer permeability between the macropores and micropores. Then in the transient regime, when fluid gradually diffuses into the micropores by mass transfer, micropore pressure slowly approaches



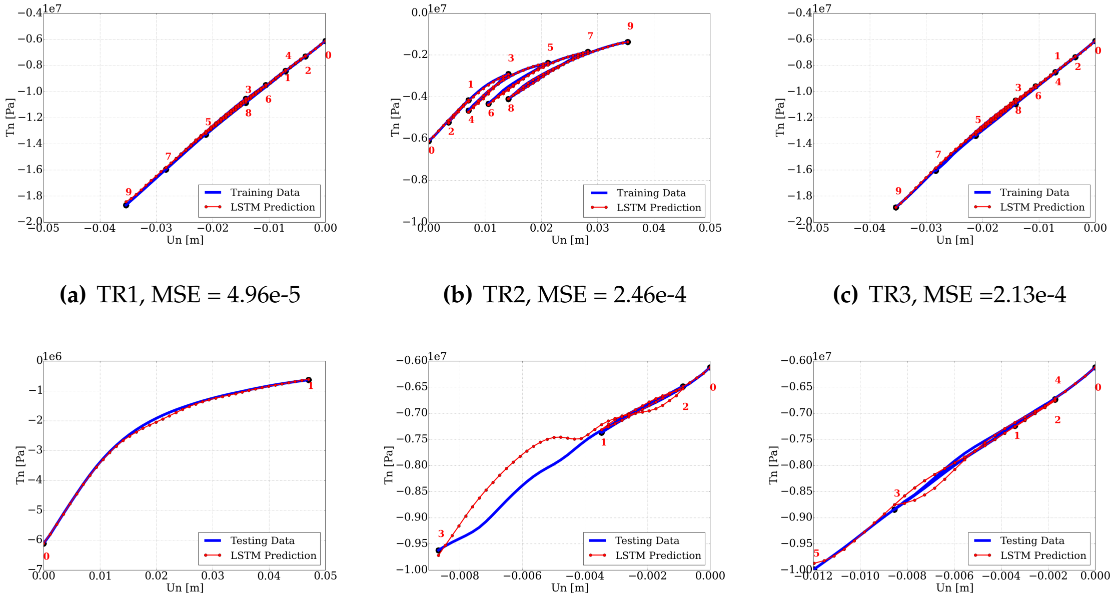
(a) TR1 **(b)** TR2 **(c)** TR3

(d) TE1 **(e)** TE2 **(f)** TE3

Figure 4.20: Loading path of three selected training cases TR1, TR2, TR3 and three selected testing cases TE1, TE2, TE3 on the meso-scale RVE. u_n and u_s are the normal and tangential displacement jumps. The coordinate system is $\{M, N\}$ (or $\{x, y\}$) depicted in Fig. 4.9. It can be seen that TR2 represents a tensile-shear loading case (as u_n is positive), TR1 and TR3 represent compressive-shear loading cases (u_n negative). The numbers mark the sequence of loading-unloading cycles.

the macropore pressure. The two pressure will eventually be identical when the diffusion between pores reaches equilibrium. To show the influence of the inter-pore transfer, we present an additional case where the transfer parameter $\bar{\alpha}$ is ten times higher (Fig. 4.25(c)). The discrepancy between the pressures is significantly reduced. The following results are from the low inter-pore transfer case.

To illustrate the hydraulic response, the macropore pressure field at time 40 h, 100 h and 140 h are presented in Fig. 4.26. The pressure plume is in the form of circle and expands with increasing amount of water injected through the source (t = 40 h). The pore pressure drops when the injection pauses, but the plume is still expanding (t = 100 h) driven by the

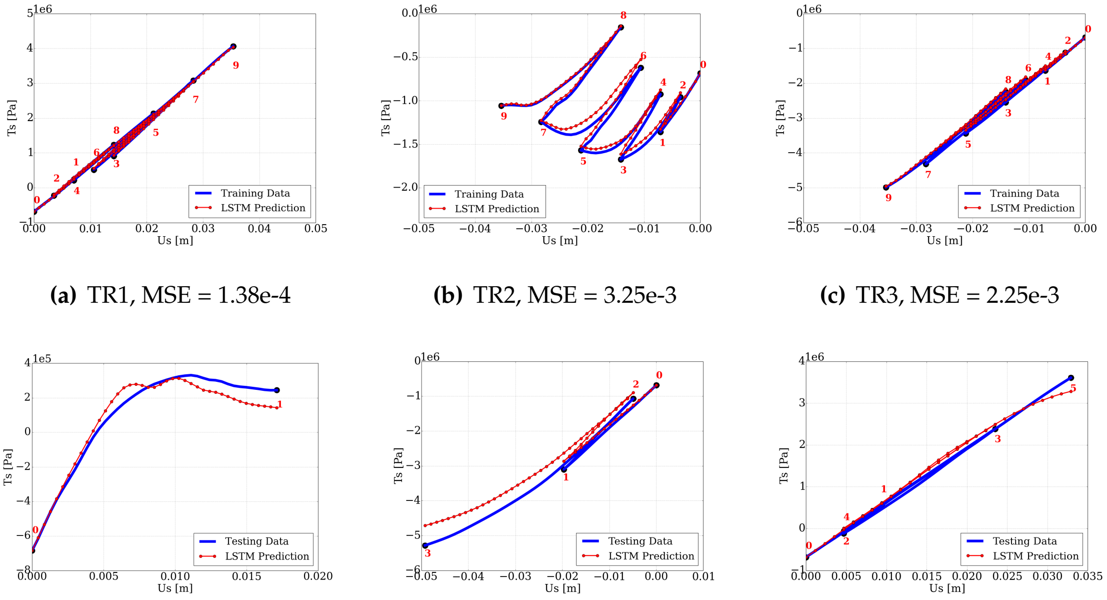


(a) TR1, MSE = 4.96e-5 **(b)** TR2, MSE = 2.46e-4 **(c)** TR3, MSE =2.13e-4

(d) TE1, MSE = 8.22e-5 **(e)** TE2, MSE = 6.04e-3 **(f)** TE3, MSE = 3.57e-4

Figure 4.21: Comparison of the meso-scale FEM-LSTM simulation data and the trained macro-scale data-driven model. Normal traction against normal displacement jump for the selected training and testing cases. The numbers mark the sequence of loading-unloading cycles. MSE refers to the scaled mean squared error defined in Eq. 4.27.

excess pore pressure that has not been entirely diffused. When the injection is resumed, the pore pressure also rises again. The form of the pressure plume is disturbed when it reaches the sealing fault, which has a two-order-lower macropore permeability (t = 140 h). As for the micropore pressure field, it has a similar but delayed evolution behavior, due to the time required for the fluid transfer between macropores and micropores. The local pressure responses are illustrated in Fig. 4.27 for three locations in the fault indicated in Fig. 4.26(a). The distances to the source point $d_{SB} > d_{SA} > d_{SC}$. It is seen that the closer a point is to the source, the faster the pressure increases upon injection. There is also significant pressure gradient across the less-permeable fault. The difference between macropore and micropore pressure is due to the different permeability in macropores and micropores for



(a) TR1, MSE = 1.38e-4

(b) TR2, MSE = 3.25e-3

(c) TR3, MSE = 2.25e-3

(d) TE1, MSE = 3.41e-3

(e) TE2, MSE = 4.76e-3

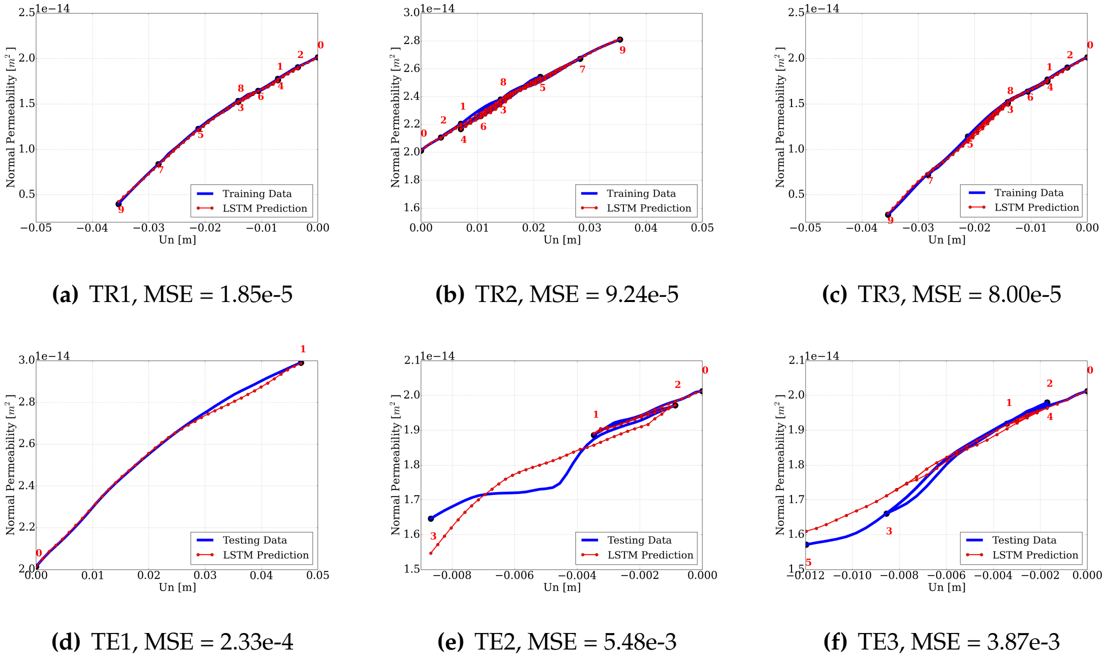
(f) TE3, MSE = 8.44e-4

Figure 4.22: Comparison of the meso-scale FEM-LSTM simulation data and the trained macro-scale data-driven model. Tangential traction against tangential displacement jump for the selected training and testing cases. The numbers mark the sequence of loading-unloading cycles. MSE refers to the scaled mean squared error defined in Eq. 4.27.

the fluid to diffuse in the macro-scale field, and also the low transfer permeability between pores.

Due to the fully coupled nature of the problem, the mechanical responses of the porous solid, especially the displacement jump and traction at the strong discontinuity, strongly depend on how pore fluid diffuses inside the pore space. The evolution of macro-scale mean effective stress field during the fluid injection cycles is shown in Fig. 4.28.

It is clear that this field has the same pattern as the pore pressure field. The increase in the mean effective stress is due to the increase in pore pressure, in agreement to the effective stress principle. The evolution of macro-scale differential stress field (Fig. 4.29) is a combined effect of the far field differential stress, fluid injection and presence of seal-



(a) TR1, MSE = 1.85e-5

(b) TR2, MSE = 9.24e-5

(c) TR3, MSE = 8.00e-5

(d) TE1, MSE = 2.33e-4

(e) TE2, MSE = 5.48e-3

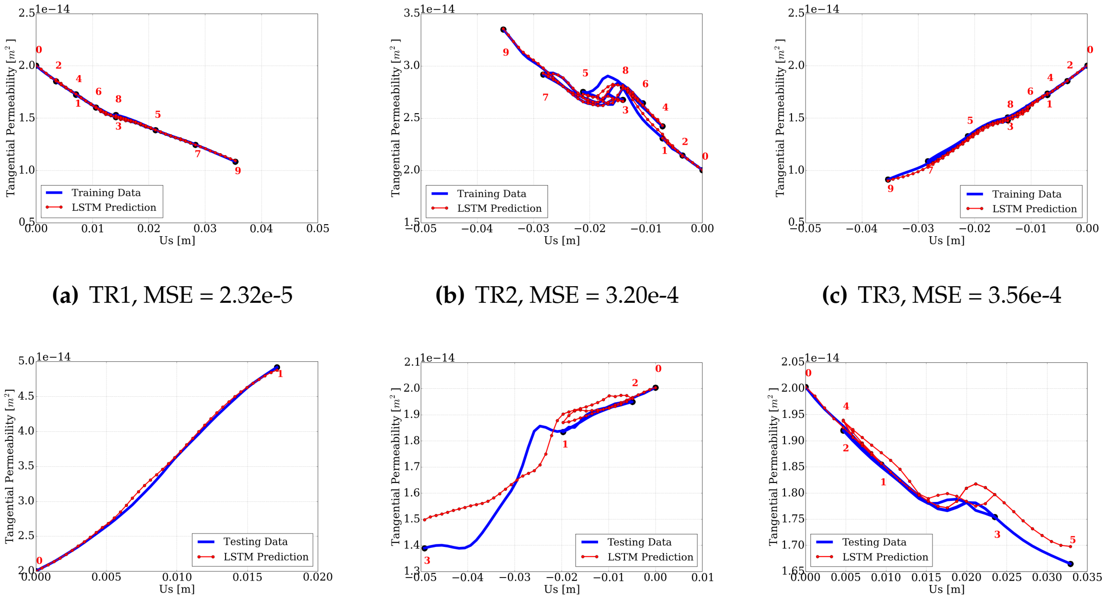
(f) TE3, MSE = 3.87e-3

Figure 4.23: Comparison of the meso-scale FEM-LSTM simulation data and the trained macro-scale data-driven model. Normal permeability against normal displacement jump for the selected training and testing cases. The numbers mark the sequence of loading-unloading cycles. MSE refers to the scaled mean squared error defined in Eq. 4.27.

ing fault. The decrease in normal compression traction makes the fault surface unable to sustain the shear traction, and the fault starts to mobilize.

The local displacement and traction responses are illustrated in Fig. 4.30 and Fig. 4.31 respectively for three locations in the fault indicated in Fig. 4.26(a). The traction states that the material at A, B, C experienced during the injection-pause cycles are depicted in Fig. 4.32.

These results clearly demonstrate the capacity of our data-driven model in capturing the complex and path-dependent interface behaviors. This is a significant improvement over the phenomenological traction-separation laws where idealized tensile and shear (linear or exponential) behavior is often adopted [165]. The data-driven model can preserve



(a) TR1, MSE = 2.32e-5

(b) TR2, MSE = 3.20e-4

(c) TR3, MSE = 3.56e-4

(d) TE1, MSE = 2.11e-4

(e) TE2, MSE = 1.27e-2

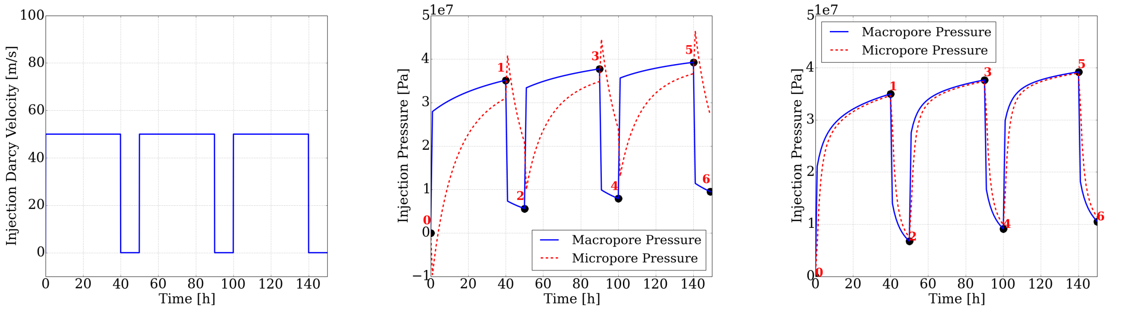
(f) TE3, MSE = 3.48e-3

Figure 4.24: Comparison of the meso-scale FEM-LSTM simulation data and the trained macro-scale data-driven model. Tangential permeability against tangential displacement jump for the selected training and testing cases. The numbers mark the sequence of loading-unloading cycles. MSE refers to the scaled mean squared error defined in Eq. 4.27.

important mechanical properties of the interface from sub-scale structures while reducing the computational costs compared to full micro-scale models such as DEM.

4.5 Conclusions

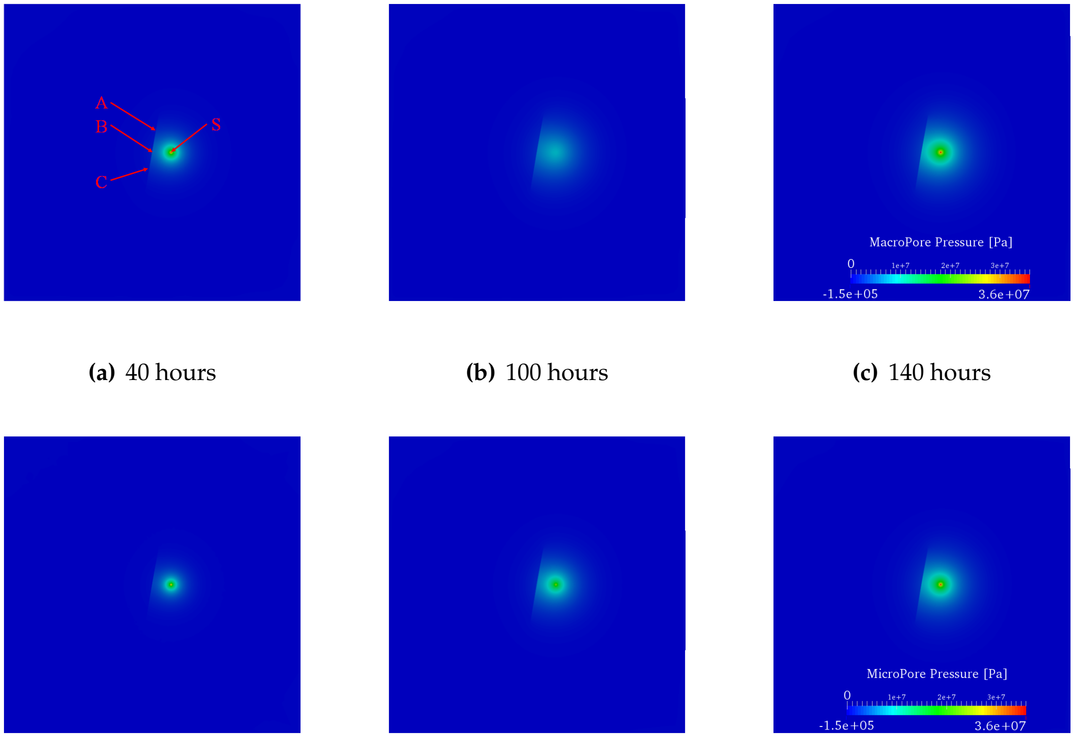
This chapter presents a semi-data-driven multiscale approach that obtains both the traction-separation law and the aperture-porosity-permeability relation from micro-mechanical simulations performed on representative elementary volumes in the finite deformation range. To speed up the multiscale simulations, the incremental constitutive updates of the mechanical responses are obtained from discrete element simulations at the representative



(a) **(b)** Low transfer between pores **(c)** High transfer between pores

Figure 4.25: Water supply in the macro-scale fault reactivation problem. (a) Time history of the prescribed injection velocity in macropores at the source point. (b) Computed responses of injection pressure in macropore and micropore at the source point (transfer parameter $\overline{\alpha} = \rho_f * k_m$). (c) Computed pressures in a comparison simulation where the transfer parameter $\overline{\alpha} = 10 * \rho_f * k_m$. The numbers mark the sequence of injection-pause cycles.

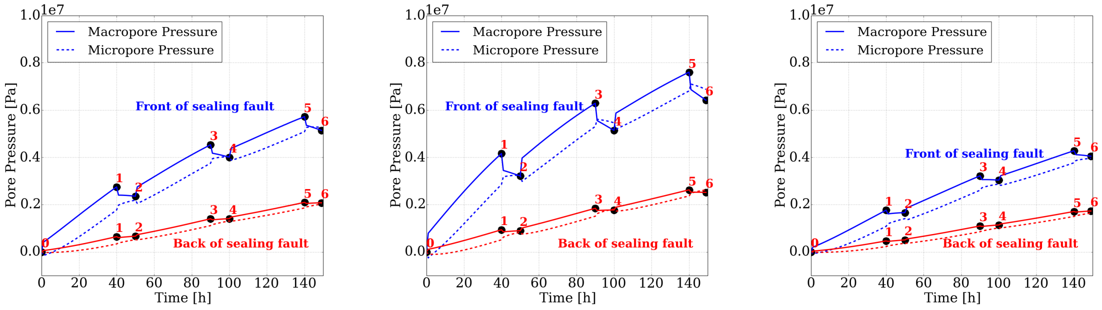
elementary volume whereas the hydraulic responses are generated from a neural network trained with data from lattice Boltzmann simulations. These responses are then linked to a macroscopic dual-permeability model. This approach allows one to bypass the need of deriving multi-physical phenomenological laws for complex loading paths. More importantly, it enables the capturing of the evolving anisotropy of the permeabilities of the macro- and micro-pores. A set of numerical experiments are used to demonstrate the robustness of the proposed model.



(a) 40 hours **(b)** 100 hours **(c)** 140 hours

(d) 40 hours **(e)** 100 hours **(f)** 140 hours

Figure 4.26: Evolution of macropore pressure (a-c) and micropore pressure (d-f) field. S denotes the fluid source. A, B, C are three locations on the sealing fault. 40h is the end of the first injection, 100h is the end of the second pause, and 140h is the end of the third injection.

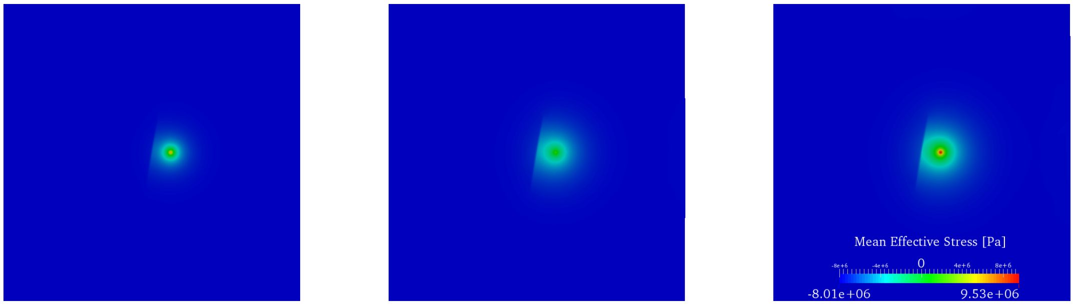


(a) A

(b) B

(c) C

Figure 4.27: Time history of local macropore and micropore pressure at locations A, B, C (Fig. 4.26(a)) of the sealing fault. ”Front” refers to the side of fault that is facing the source point. ”Back” is another side that is away from the source. The numbers mark the sequence of injection-pause cycles.

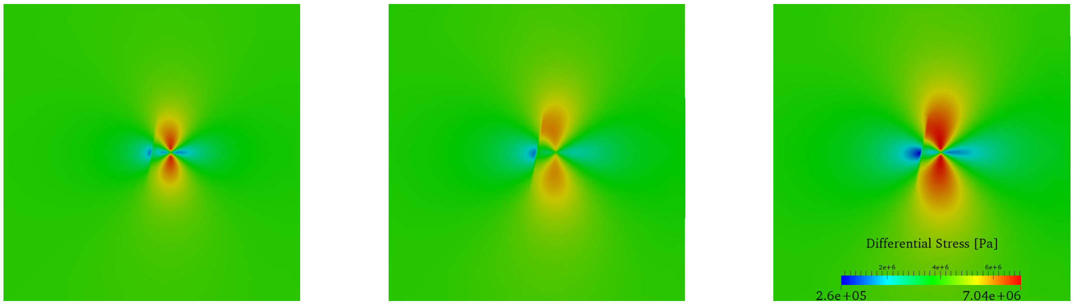


(a) 40 hours

(b) 100 hours

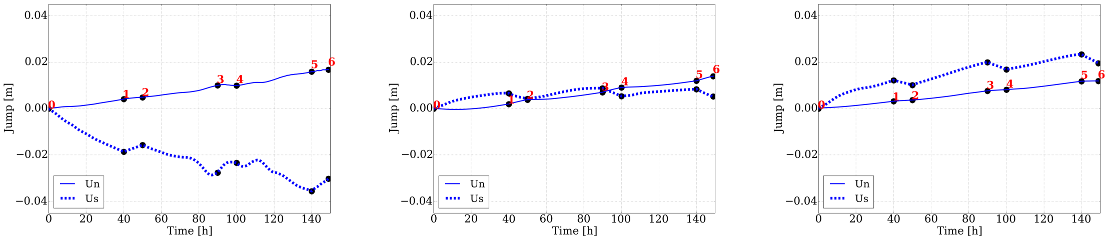
(c) 140 hours

Figure 4.28: Evolution of the mean effective stress field in the macro-scale simulation. 40h is the end of the first injection, 100h is the end of the second pause, and 140h is the end of the third injection.



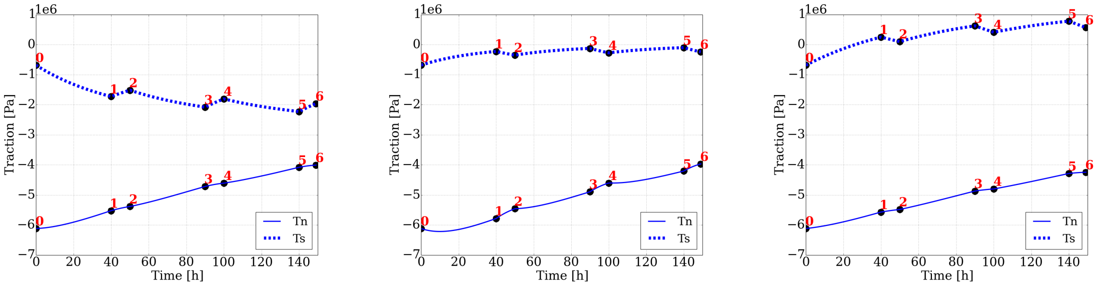
(a) 40 hours **(b)** 100 hours **(c)** 140 hours

Figure 4.29: Evolution of the differential stress field in the macro-scale simulation. 40h is the end of the first injection, 100h is the end of the second pause, and 140h is the end of the third injection.



(a) A **(b)** B **(c)** C

Figure 4.30: Time history of normal and tangential displacement jumps at locations A, B, C (Fig. 4.26(a)) of the sealing fault. The coordinate system is $\{M, N\}$ (or $\{x, y\}$) depicted in Fig. 4.9. The numbers mark the sequence of injection-pause cycles.

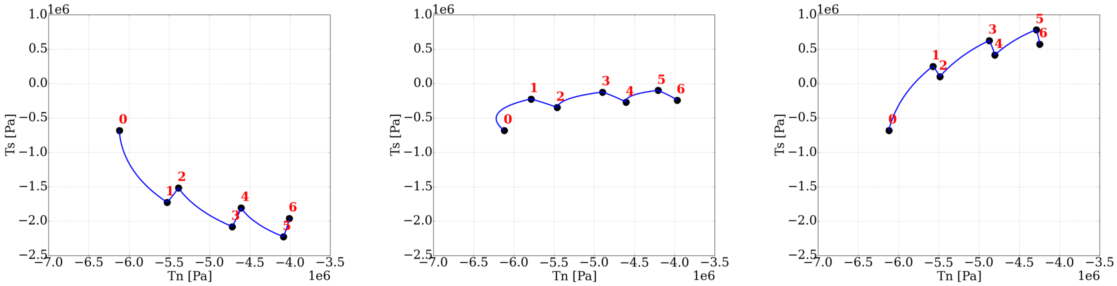


(a) A

(b) B

(c) C

Figure 4.31: Time history of normal and tangential displacement tractions at locations A, B, C (Fig. 4.26(a)) of the sealing fault. The coordinate system is $\{M, N\}$ (or $\{x, y\}$) depicted in Fig. 4.9. The numbers mark the sequence of injection-pause cycles.



(a) A

(b) B

(c) C

Figure 4.32: Traction Path at locations A, B, C (Fig. 4.26(a)) of the sealing fault. The coordinate system is $\{M, N\}$ (or $\{x, y\}$) depicted in Fig. 4.9. The numbers mark the sequence of injection-pause cycles.

Chapter 5

Metamodeling game for deriving micro-structure-based traction-separation models for geomaterials

This chapter is reproduced from the published paper: K. Wang, W.C. Sun, Meta-modeling game for deriving theory-consistent, micro-structure-based traction-separation laws via deep reinforcement learning, *Computer Methods in Applied Mechanics and Engineering*, 346:216-241, 2019.

5.1 Introduction

Constitutive responses of interfaces are important for a wide spectrum of problems that involve spatial domain with embedded strong discontinuity, such as fracture surfaces [182, 166, 246, 35], slip lines [180, 27], joints [65] and faults [157, 247, 216]. While earlier modeling efforts, in particular those involving the modeling of cohesive zones , often solely focus on mode I kinematics, the mixed mode predictions of traction-separation law relations are critical for numerous applications, ranging from predicting damage upon impacts [160], to predicting seismic events [186]. [166] provide a comprehensive account of the major characteristic of traction-separation laws and conclude that, while there are differences in details, most of the traction-separation laws obey a number of universal principles, such as the indifference of any superimposed rigid-body motion, the finite work required to create new surface, the existence of characteristic length scales, and the vanishing of cohesive traction with sufficient separations.

In the case where the loading history is not monotonic, constitutive responses of interfaces often become path-dependent. For instance, geomaterials, such as fault gauges, are known to exhibit rate- and state-dependent frictional responses [169, 206, 24, 251, 151]. While there are phenomenological models designed to capture the path-dependent responses of the interfaces, a recent trend that gains increasing popularity is to replace the phenomenological traction-separation laws with a computational homogenization procedure to capture the responses of materials with heterogeneous microstructures (cf. [147, 95, 94]. Nevertheless, as pointed out previously in [247], the major issue of applying hierarchical multiscale coupling on interfacial problems is the increasing computational demand due to the large number of required representative elementary simulations, a trade-off that is widely known in FEM^2 [67] and other homogenization-based multiscale methods, such as FEM-DEM [210, 209, 68, 214, 241, 133, 115, 244, 259].

To overcome this computational barrier, surrogate models are often derived to replicate the homogenized responses of sub-scale simulations [110, 238, 108, 163, 66, 131, 218]. Nevertheless, since surrogate models are often constitutive laws hand-crafted by modelers to incorporate morphology-dependent features [131], deriving, verifying and validating a surrogate model that can incorporate the essential information to yield macroscopic predictions with sufficient accuracy and robustness remain difficult and time-consuming. Data-driven models such as [119, 18, 239, 106] and [260] attempted to overcome this issue via supervised machine learning (e.g. neural network [121], symbolic regression model [239]) and unsupervised machine learning (e.g. dimensional reduction, feature extraction and clustering [18, 260]).

In particular, recent work by [247] attempted to resolve this issue by building a generic recurrent neural network that can easily incorporate different types of sub-scale information (e.g. porosity, fabric tensor, and relative displacement) to predict traction. This technique uses the concept of directed graph on the transfer learning approach (cf. [162]) in which multiple neural networks trained to make predictions on other physical quantities (e.g. relationship between porosity and fabric tensor) are re-used to generate additional inputs for predicting traction. However, the determination of the optimal input information (in addition to the displacement jump history) and configurations of information flow that enhances the prediction accuracy still requires a time-consuming trial-and-error task (cf. Section 4.3 [247]).

In this work, we introduce a general artificial intelligence approach to automate the creation and validation of traction-separation models. Unlike the previous approach in which neural networks are often used to either identify material parameters or create black-box constitutive laws, this work focuses on leveraging the capacity of a computer to improve via self-playing, a technique commonly referred as (deep) reinforcement learning in the computer science community [217, 194, 196]. In the past two years, the functionality of algorithms automatically generated from deep reinforcement learning have achieved remarkable success. In many cases, the demonstrated capacities were thought to be impossible in the past. For instance, the algorithm trained by deep reinforcement learning created by a company called DeepMind is able to outperform human experts in Go, Chess and Atari games. The most exciting part of this achievement is that, unlike previous AI such as the IBM Deep Blue, the deep reinforcement learning does not rely on hand-crafted policy evaluation functions and is therefore applicable to different kinds of games once they are defined and implemented.

This success motivates this research of proposing a meta-modeling approach where deep reinforcement learning may generate constitutive laws for (1) a given set of data, (2) a well-defined objective, and (3) a given set of universal principles. To achieve this goal, we recast the process of writing a constitutive model as a game with components suitable for deep reinforcement learning, involving a sequence of actions completely compatible with the stated rules (i.e., the law of physics). First, we define the model score, which could be any objective function suitable for a given task. For instance, this objective can be minimizing the discrepancy between calibrated experimental results and ***blind predictions*** measured by a norm, or a constrained optimization problem that gives considerations on other attributes such as consistency, speed, and robustness [251]. Once the score (i.e., the objective) is clearly defined, we then implement the rules, which are the universal principles of mechanics, such as material frame indifference, laws of thermodynamics. These rules are applied in an environment in which scores are sampled. In the case of traction-separation law, the environment is simply the validation process itself.

Following this, we then define the action space which consists of a number of actions available for the modelers to write constitutive models. Once the action space and the model score are defined, we leverage the directed graph modeling technique to generate a state. The state at the end of each game represents a constitutive model automatically generated from the computer algorithm. In reality, the action space could be of very high dimensions such that manually deriving, implementing, verifying and validating all possible configurations are not feasible. This situation is similar to playing the games of chess and Go where the number of possible combinations of decisions or moves (each can be represented by a decision tree) remains finite but is so enormous that it is not possible to seek the optimal moves by exhausting all possibilities [193].

With the state, action, rule and objective defined, the most critical part is to assign reward for each action. In principle, if the action space is of very low dimension, i.e., there are not many ways to model the physical processes, then the reward for each action can be determined by exhausting all the possible model configurations. However, in the case of writing a complex traction-separation model, we cannot evaluate the quality of the model until its predictions are compared with benchmark data. Therefore, the ability to approximate the reward for each action (in our case the modeling choices) without the need to evaluate all the available options becomes crucial for the success of the meta-modeling approach.

The deep reinforcement learning is therefore ideal for us to achieve this goal. We can approximate the rewards via neural networks and the Bellman expectation equation [15, 61]. By repeatedly generating new constitutive laws (i.e., playing the game of writing models), the agent will use the reward obtained from each played game, in analogy with the binary game result (win/loss) at the end of a Go game, to update the action probabilities and value functions to improve the agent's ability to write good constitutive laws. Through sufficient self-plays, the reinforcement learning algorithm then improves the modeling choices it made over time until it is ready for predictions.

There are a few major upshots for this approach. First, once the reinforcement learning

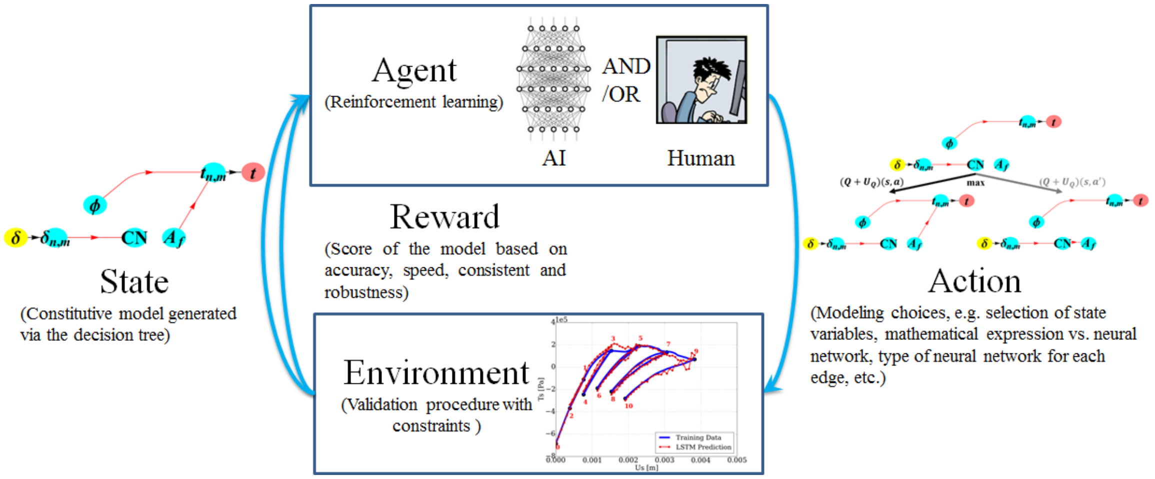

Agent
(Reinforcement learning)
AND /OR
AI
Human
Reward
(Score of the model based on accuracy, speed, consistent and robustness)
Environment
(Validation procedure with constraints)
State
(Constitutive model generated via the decision tree)
Action
(Modeling choices, e.g. selection of state variables, mathematical expression vs. neural network, type of neural network for each edge, etc.)
δ
$\delta_{n,m}$
CN
A_f
ϕ
$t_{n,m}$
t
$(Q + U_Q)(s,a)$
max
$(Q + U_Q)(s,a')$
1e5
Ts [Pa]
Us [m]
Training Data
LSTM Prediction
0
1
2
3
4
5
6
7
8
9
10
0.000
0.001
0.002
0.003
0.004
0.005
4
2
0
−2
−4
−6
−8

Figure 5.1: Scheme of the reinforcement learning algorithm in which an agent interacts with environment and receives rewards. Through exploration, the agent then determines better actions to achieve a particular goal defined by the reward. In our case, the reward is the score which represents the quality of the forward prediction, the action is any possible activities required to derive a constitutive law, and the environment is the procedure that compares the predictions with the benchmark data. Actual graphs of the Environment, State and Action will be detailed in the subsequent sections.

algorithm is established, it can serve as a model generator without any human intervention. Second, since we regard the validation process as the environment component of the reinforcement learning, the performance of a resultant model is simultaneously evaluated and therefore validations are always a part of the model writing process. Third, the metamodeling approach may easily embed any existing model generated by domain experts into the action space without re-implementing a new model. These unique capabilities enable us to have an unbiased tool to evaluate how well existing models fulfill a particular objective. Furthermore, since the model generation procedure is automated once an objective function is defined, this work may potentially eliminate the need of writing multiple incremental models for the same materials over time. Finally, this modeling approach is particularly powerful for discovering hidden physical coupling mechanisms that are otherwise too subtle to detect with human observation.

The rest of the chapter is organized as follows. We first review the directed graph approach that enables us to generate and utilize a decision tree to represent the modeling process (Section 5.2). The definition of model scores is then described in Section 5.3. We then provide a formal definition of a game invented to generate traction-separation laws for predictions (Section 7.4). This is followed by a description on how to use the reinforcement learning for the traction-separation law generation (Section 5.5). Two numerical experiments are then used to showcase the performance of the automated meta-modeling approach using synthetic data from microscale discrete element simulations (Section 5.6). The major findings are then summarized in the conclusions.

5.2 Representing traction-separation law in directed graph

In this section, we introduce a building block for a simplified and extensible game that generates traction-separation laws by considering the relationships among different types of data collected from sub-scale simulations. In this game, the goal is to find a specific way to link different types of data such that a score function is maximized. Before we introduce the formal definition of the game, one necessary step is to recast the algorithm that leads to predictions from constitutive laws as a network of unidirectional information flow, i.e., a directed graph (also referred to as digraph) [215, 212, 208, 188, 247]. Recall that a digraph $D = (V, E)$ is an ordered pair of non-empty finite sets which consists of a vertex set V and an edge set E [12]. Each edge connects a source vertex (tail) to a target vertex (head). Following the treatment in [208] and [247], the following **rules** are applied to generate the traction-separation law.

1. The traction $\boldsymbol{t}$ is placed as the only leaf of the digraph (i.e., the vertex that is not source to any other vertices).
2. The displacement jump $\boldsymbol{\delta}$ is placed as the only root of the digraph (i.e., the vertex that is not target of any other vertices).
3. There may exist isolated vertices in the digraph, i.e., some internal variables or microstructural features between $\boldsymbol{\delta}$ and $\boldsymbol{t}$ may not contribute to the final completed digraph and the corresponding constitutive model.
4. The digraph is acyclic, which means that there must be no cycle in the digraph.
5. If a vertex has sources or targets connected to it, it must be on at least one of the paths leading from $\boldsymbol{\delta}$ to $\boldsymbol{t}$. This ensures that an internal variable, once considered, is fully incorporated into the final constitutive model.

In the previous published work (cf. [208, 247]), we prescribed theoretical models or, in some cases, neural network models to create linkages and enforce the hierarchy among physical quantities (e.g. porosity-permeability relation). While this treatment is convenient for software engineering and code design [188], this approach only works if we have a prior knowledge about the relationships among the physical quantities. While one may presumably make ad hoc assumptions to complete the models, such a treatment is often at the expense of robustness. Another possible remedy is to gather all the measurement and data one may possibly obtain from observations and experiments, then find the key mechanisms that incorporate the most essential physics (e.g. the critical state plasticity for soil). This latter approach can be re-expressed as a problem in the directed graph in which we only know the elements of the vertex set but have no idea whether and how these vertices are connected, except that the traction is the leaf and the displacement jump is the root of the directed graph. Note that, in reality, the creation of a deterministic constitutive law does not only limit at determining connections among vertices (physical quantities), but also includes finding hidden vertices and appropriate edges. These actions are not modeled in this chapter, but will be considered in future studies. Furthermore, while our focus in this chapter is on deriving the traction-separation laws, in principle, the idea can be easily extended to other problems, such as the stress-strain relation for bulk materials, the porosity-temperature-fabric-tensor-permeability relations for porous media, among others.

For demonstration purposes, we consider a constitutive law $\boldsymbol{t}(\boldsymbol{\delta}, \boldsymbol{q})$ that predicts the traction vector $\boldsymbol{t}$ based on the history of the displacement jump $\boldsymbol{\delta}$ over a cohesive or cohesive-frictional surface with the normal direction vector being $\boldsymbol{n}$. $\boldsymbol{q}$ is a collection of state variables with n degrees of freedom, i.e., $q_1, q_2, q_3, ..., q_n$. We use sub-scale discrete element simulations to generate synthetic data and attempt to create a traction-separation model which can replicate the constitutive responses of complex loading histories.

Imposing restrictions of material frame indifference and assuming isotropic cohesive-frictional surface, the traction-separation model can be simplified to [160]

$$\boldsymbol{t}(\boldsymbol{\delta}, \boldsymbol{q}) = \boldsymbol{t}(\delta_n, \delta_m, \boldsymbol{q}), \tag{5.1}$$

where $\delta_n = \boldsymbol{\delta} \cdot \boldsymbol{n}$ and $\delta_m = |\boldsymbol{\delta_m}| = |\boldsymbol{\delta} - \delta_n \boldsymbol{n}|$. Hence, the traction $\boldsymbol{t}$ is related to its components t_n and t_m that

$$\boldsymbol{t}(\boldsymbol{\delta}, \boldsymbol{q}) = t_n(\delta_n, \delta_m, \boldsymbol{q})\boldsymbol{n} + t_m(\delta_n, \delta_m, \boldsymbol{q})\frac{\boldsymbol{\delta_m}}{\delta_m}. \tag{5.2}$$

The internal variables in $\boldsymbol{q}$, if the cohesive surface is composed of a thin layer of granular materials, can be chosen among a large set of geometrical measures on micro-structural attributes [214, 115]. In this work, we first manually select the following measures to be the intermediate vertices (the vertices that are neither the leaves nor the roots) to make forward predictions on the traction vector.

- Porosity ϕ, the ratio between the volume of the void and the total volume of a representative volume element (RVE) of the material layer.
- Coordination number $CN = N_{\text{contact}}/N_{\text{particle}}$ where N_{contact} is the number of particle contacts and N_{particle} is the number of particles in the RVE.
- Fabric tensor $\boldsymbol{A}_f = \frac{1}{N_{\text{contact}}} \sum_{c=1}^{N_{\text{contact}}} \boldsymbol{n}^c \otimes \boldsymbol{n}^c$, where $\boldsymbol{n}^c$ is the normal vector of a particle contact c, c = 1, 2, ...,N_{contact} in the RVE.
- Strong fabric tensor $\boldsymbol{A}_{sf} = \frac{1}{N_{\text{strongcontact}}} \sum_{c=1}^{N_{\text{strongcontact}}} \boldsymbol{n}^c \otimes \boldsymbol{n}^c$, where $\boldsymbol{n}^c$ is the normal vector of a strong particle contact (having a compressive normal force greater than mean contact force) c, c = 1, 2, ...,$N_{\text{strongcontact}}$ in the RVE.

All particle contacts inside the RVE can form a graph with particles as vertices and interactions as edges. Some quantitative measures of this graph of connectivity can be included in the internal variables $\boldsymbol{q}$ as additional microstructural characteristics. Here, we focus on four measures, which are computed using the software package NetworkX ([90]), and their detailed explanations can be found in the software documentation.

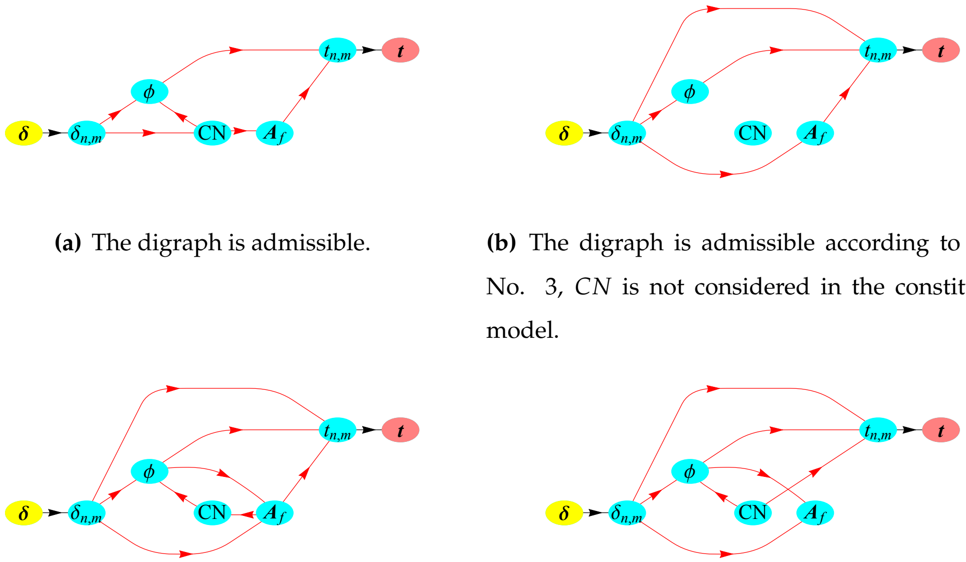
- d_a, degree assortativity, a scalar value between -1 and 1 measuring the similarity of connections in the graph with respect to the node degree.
- c_t, transitivity coefficient, $c_t = 3\frac{n_{triangles}}{n_{triads}}$, the fraction between the number of triangles and the number of triads present in contact graph.
- l_{sp}, average shortest path length in the contact graph.

- ρ_g, density of the graph, $\rho_g = \frac{2m}{n(n-1)}$, where n is the total number of nodes and m is the total number of edges in the graph.

To sum up, in the digraph representations of traction-separation models, $\boldsymbol{\delta}$ is the root and $\boldsymbol{t}$ is the leaf, and currently we consider $\boldsymbol{q}$ to be a subset of the following set of physical quantities $\{\delta_{n,m},\ t_{n,m},\ \phi,\ CN,\ \boldsymbol{A}_f,\ \boldsymbol{A}_{sf},\ d_a,\ c_t,\ l_{sp},\ \rho_g\}$. For the edges, we classify them as either "definitions" (such as $t_{n,m} \rightarrow \boldsymbol{t},\ \boldsymbol{\delta} \rightarrow \delta_{n,m}$) which are determined by universal principles in mechanics and should not be modified, or the "phenomenological relations" (such as $\delta_{n,m} \rightarrow \boldsymbol{A}_f$, $\phi \rightarrow CN$, $l_{sp} \rightarrow t_{n,m}$) which incorporate material parameters chosen to fit experimental data. The latter category of edges provide opportunities for researchers to propose hand-crafted constitutive relations of different degrees of complexities. For example, their forms can be linear, quadratic, exponential functions or be approximated by artificial neural networks (ANNs). For illustration purposes, we consider a simple digraph of traction-separation models involving only the nodes $\{\boldsymbol{\delta},\ \boldsymbol{t},\ \delta_{n,m},\ t_{n,m},\ \phi,\ CN,\ \boldsymbol{A}_f\}$. Figure 5.2 provides examples of two admissible and two illegal digraph configurations according to the Rules 1-5.

5.3 Score system for model evaluation and objective function

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



(a) The digraph is admissible.

(b) The digraph is admissible according to Rule No. 3, CN is not considered in the constitutive model.

(c) The digraph violates Rule No. 4, since there exists a cycle $\phi \rightarrow \boldsymbol{A}_f \rightarrow CN \rightarrow \phi$.

(d) The digraph violates Rule No. 5, since CN and $\boldsymbol{A}_f$ are not on any paths leading from δ to $\boldsymbol{t}$.

Figure 5.2: Examples of admissible directed graphs (a-b) and illegal directed graphs (c-d) representing information flow in traction-separation models involving internal physical quantities of porosity ϕ, coordination number CN and fabric tensor $\boldsymbol{A}_f$. The yellow node of separation δ refers to the root node, the pink node of traction $\boldsymbol{t}$ refers to the leaf node, and the cyan nodes refer to intermediate nodes. The black arrows refer to ”definition” or ”universal principles” edges. The red arrows refer to ”phenomenological relations” edges.

$(i = 1, 2, 3, ..., n)$ of n important features of a constitutive model,

$$\text{SCORE} = F(A_1, A_2, A_3, ..., A_n), \tag{5.3}$$

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,

$$\text{SCORE} = (\prod_{j=1}^{n_{\text{crit}}} A_j^{\text{crit}}) \cdot (\sum_{i=1}^{n_{\text{pfm}}} w_i A_i^{\text{pfm}}), \tag{5.4}$$

where $w_i \in [0,1]$ is the weight associated with the measure A_i^{pfm}, and $\sum_{i=1}^{n_{\text{pfm}}} w_i = 1$. In this section, two examples of measures of accuracy A_{accuracy} and prediction consistency $A_{\text{consistency}}$ are presented.

5.3.1 Accuracy of calibrations and forward predictions

In this work, the abilities of the models to replicate calibration data and make forward predictions are considered separately. Here we introduce a cross-validation procedure in which the dataset used for training the models (e.g. identifying material parameters (e.g. [251, 131]) or adjusting weights of neurons in recurrent neural networks (e.g. [121, 247]) is mutually exclusive to the testing dataset used to evaluate the quality of blind predictions. The details of the generation of these calibration and testing data sets using frictional discrete element simulations are presented in Appendix A. 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.,

$$\text{MSE}_i = \frac{1}{N_{\text{feature}}} \sum_{j=1}^{N_{\text{feature}}} [\mathcal{S}_j(Y_{i_j}^{\text{data}}) - \mathcal{S}_j(Y_{i_j}^{\text{model}})]^2, \tag{5.5}$$

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 [107],

$$F_N(\mathrm{MSE}) = \begin{cases} 0, & \mathrm{MSE} < \mathrm{MSE}_1, \\ \dfrac{r}{N}, & \mathrm{MSE}_r \leq \mathrm{MSE} < \mathrm{MSE}_{r+1},\ r = 1, ..., N-1, \\ 1, & \mathrm{MSE}_N \leq \mathrm{MSE}, \end{cases} \tag{5.6}$$

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,

$$A_{\text{accuracy}} = \max\left(\frac{\log[\max(\varepsilon_{P\%}, \varepsilon_{\text{crit}})]}{\log \varepsilon_{\text{crit}}}, 0\right), \tag{5.7}$$

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 $\varepsilon_{P\%} \leq \varepsilon_{\text{crit}}$.

5.3.2 Consistency of accuracy between calibrations and forward predictions

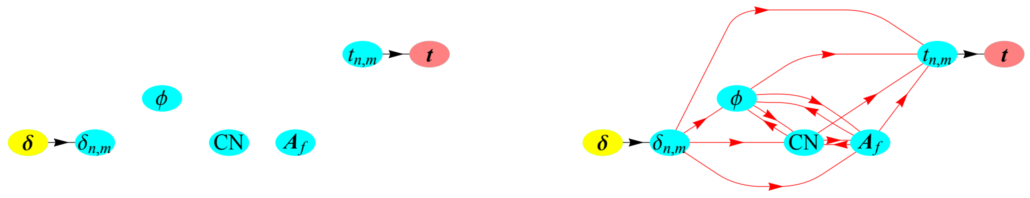
For the examination of the consistency in model predictions on training data and test data, the K-sample Anderson-Darling (AD) test of goodness-of-fit (gof) is conducted to check whether the eCDFs of training and test data come from the same probability distribution, while this distribution is unspecified [3, 191]. It is a non-parametric hypothesis test and determines whether the null hypothesis H_0 that the two eCDFs come from the same continuous distribution can be rejected or not, under a chosen significance level α_{gof}. The method consists of calculating a normalized AD test statistic, critical values of the AD statistic that depends on the sample sizes, and a p-value indicating the approximated significance level at which H_0 can be rejected. If the p-value is smaller than the significance level α_{gof}, the H_0 hypothesis is rejected. Otherwise there is insufficient evidence to reject H_0. In this work, we define the following binary measure for the consistency of the MSE distributions, with the significance level α_{gof},

$$A_{\text{consistency}} = H^{\alpha_{\text{gof}}} = \begin{cases} 0, & p\text{-value} < \alpha_{\text{gof}}, \\ 1, & p\text{-value} \geq \alpha_{\text{gof}}. \end{cases} \tag{5.8}$$

5.4 Game of the traction-separation law

Our focus in this chapter is primarily on the meta-modeling game invented for generating traction-separation models. Nevertheless, similar games can be defined for generating other types of constitutive models based on the ideas presented in this work. With the directed graph representations of traction-separation models as presented in Section 5.2, the process of developing a model can be recast as a game of making a sequence of decisions in generating edges between nodes in the digraphs. The player of the game can be a human or an AI agent. The game starts with an initial ”game board” of digraph with predefined nodes and predefined “definition” edges (e.g. displacement and deformation gradient, displacement jump vector and the scalar components in a given basis), while no ”phenomenological relation” edge is formed among them. Each step of the game consists of activating only one edge among all possible choices of edges in the predefined action space, following the predefined rules of the game. The game terminates when a complete and admissible digraph following the rules in Section 5.2 is established. The output models of the game are measured by a score system as presented in Section 5.3.

The game can be mathematically formalized as a Markov decision process. The human or AI agent observes the state of the game s_t at the current step t from the game environ-



(a) Initial configuration of the "game board"

(b) All possible actions on the "game board"

Figure 5.3: A game of traction-separation model for the digraph example in Figure 5.2. (a) the initial "board" on which the game is played. (b) All possible actions for picking the edges connecting the nodes are represented by the red arrows.

ment (the directed graph that represents a constitutive model) in the form of a vector of binaries indicating the on/off status of each valid edge choices in the action space. The agent takes an action a_t on the game environment in the form of an integer indicating the next edge to switch on in the action space. The action a_t is sampled from a vector of probabilities $\pi(s_t)$ of taking each valid action from the state s_t. Consequently, the state of the game becomes s_{t+1} at the next step $t+1$. The agent also receives a reward r_{t+1} for the action a_t of taking the game state from s_t to s_{t+1}. Each policy applied in a complete gameplay produces a particular trajectory $s_0, a_0, r_1, s_1, a_1, r_2, ..., a_{t-1}, r_t, s_t, a_t, ..., a_{T-1}, r_T, s_T$. Once a complete constitutive model is generated, the model score is evaluated. The final reward r_T 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.

Note that the exact values of the rewards $r_{t \leq T}$ are only known at the end of the game, similar to the game of Chess and Go. r_T is determined according to the final score of the generated model. If $r_T = 1$, then all previous intermediate rewards $r_{t<T} = 1$. If $r_T = -1$,

then $r_{t<T} = -1$. Before the game reaches an end, however, $r_{t<T}$ can only be estimated by the agent based on its knowledge of the game. For a human agent, both rewards $r_{t<T}$ and move probabilities $\pi(s)$ come from ”intuition” gained during many constitutive modeling practices. An experienced human modeler estimates the rewards and probabilities more accurately and hence more likely generates better constitutive models. For an AI agent, $r_{t<T}$ and $\pi(s)$ are approximated by hand-crafted mathematical functions or recently neural networks as in deep reinforcement Q-learning. They are estimated based on the expected game reward of taking action a from state s (Q-value) $Q(s, a)$ and the value of current state $v(s)$. The above-mentioned important quantities for mathematical descriptions of the gameplays are summarized in Table 5.1. Moreover, the constitutive modeling game is compared side-by-side with the game of Chess more familiar to the public in Table 5.2, in the aspects of the board to play on, the permitted actions to execute, the criteria for wining the game, etc.

Environment	Benchmark training and test data, idealized multigraph for constitutive models
Agent	Human or AI
State s	A list of binaries indicating the on/off status of each valid edge choice
Action a	An integer indicating the next edge to switch on from the current game state
Reward r	Win (1) / loss (-1) according to the score of the constitutive model in Section 5.3
$\pi(s, a)$	Probability of taking action a at state s
$v(s)$	Expected reward of state s
Q-value $Q(s, a)$	Expected reward from taking action a at state s

Table 5.1: Key ingredients of the game of constitutive models in directed graph.

	Game of Chess	Game of constitutive modeling in directed graph
Definition of game	Make a sequence of decisions to maximize the probability to win	Make a sequence of decisions to maximize the score of the constitutive model
Game board	8×8 grid	Directed graph with predefined nodes of physical quantities and edges of definition or universal principles
Game state	Configuration of chess pieces on the board	Configuration of directed graph representing the constitutive model
Game action	Move chess pieces	Select among modeling choices.
Game rule	Restrictions on chess piece movements	Universal principles Rules in Section 5.2 Specific restrictions on edge choices
Game reward	Win, draw or loss (discontinuous)	Win or loss (discontinuous) from comparison of model scores (continuous)
Reward evaluation	Only available at the end	Only available at the end

Table 5.2: Comparison of the essential definitions between the game of Chess and the game of constitutive modeling in directed graph.

For illustration purposes, we provide a simple game example for the digraph presented in Figure 5.2, which only involves the nodes $\{\boldsymbol{\delta},\ \boldsymbol{t},\ \delta_{n,m},\ \ t_{n,m},\ \phi,\ CN,\ \boldsymbol{A}_f\}$. Figure 5.3 presents the "initial game board" and all possible edges choices in the current game definition. The configuration of the digraph, or the state of the game, can be totally described by a list of binaries for 13 edges $[\delta_{n,m} \rightarrow \phi,\ \delta_{n,m} \rightarrow CN,\ \delta_{n,m} \rightarrow \boldsymbol{A}_f,\ \delta_{n,m} \rightarrow t_{n,m},\ \phi \rightarrow CN,\ \phi \rightarrow \boldsymbol{A}_f,\ \phi \rightarrow t_{n,m},\ CN \rightarrow \phi,\ CN \rightarrow \boldsymbol{A}_f,\ CN \rightarrow t_{n,m},\ \boldsymbol{A}_f \rightarrow \phi,\ \boldsymbol{A}_f \rightarrow CN,\ \boldsymbol{A}_f \rightarrow t_{n,m}]$ (The edges $\boldsymbol{\delta} \rightarrow \delta_{n,m}$ and $t_{n,m} \rightarrow \boldsymbol{t}$ are definitions and always active). The list also represents the entire action space. The action a is an integer $\in [0, 12]$ indicating the next edge ID to activate in the list. The legal moves at the current game state are represented by a list of 13 binaries indicating whether the corresponding edges are allowed to be activated for the next action step. The rule of the legal moves are as follows: (1) if one edge has already been selected, it is excluded from the selection of actions; (2) if an edge between two intermediate nodes has been selected, the other edge involving these two nodes but with opposite direction is also excluded (e.g., The edges $\phi \rightarrow CN$ and $CN \rightarrow \phi$ are mutually exclusive); (3) the final digraph of a complete traction-separation model must obey the rules in Section 5.2. This rule is explicitly checked for each action at each game state. All the actions resulting in a final digraph violating the rules in Section 5.2 will be forbidden. For example, if the edges $\phi \rightarrow CN$ and $CN \rightarrow \boldsymbol{A}_f$ have already been selected in the current incomplete digraph, then the action to select the edge $\boldsymbol{A}_f \rightarrow \phi$ is an illegal move, since it will lead to a final digraph that has a cycle.

Figure 5.4 provides a gameplay example of the constitutive modeling game in Figure 5.3, with mathematical representations of game states, actions and legal actions, as well as the Markov decision process.

The score evaluation (Section 5.3) requires model calibration on training data, and forward predictions on test data. The procedure for score evaluation is as follows. Once the final digraph configuration is determined, all paths (information flows) leading from $\boldsymbol{\delta}$ to $\boldsymbol{t}$ and all predecessors for each node in the paths are identified using the graph theory (software package NetworkX). Secondly, the predecessor nodes for the terminal node $\boldsymbol{t}$

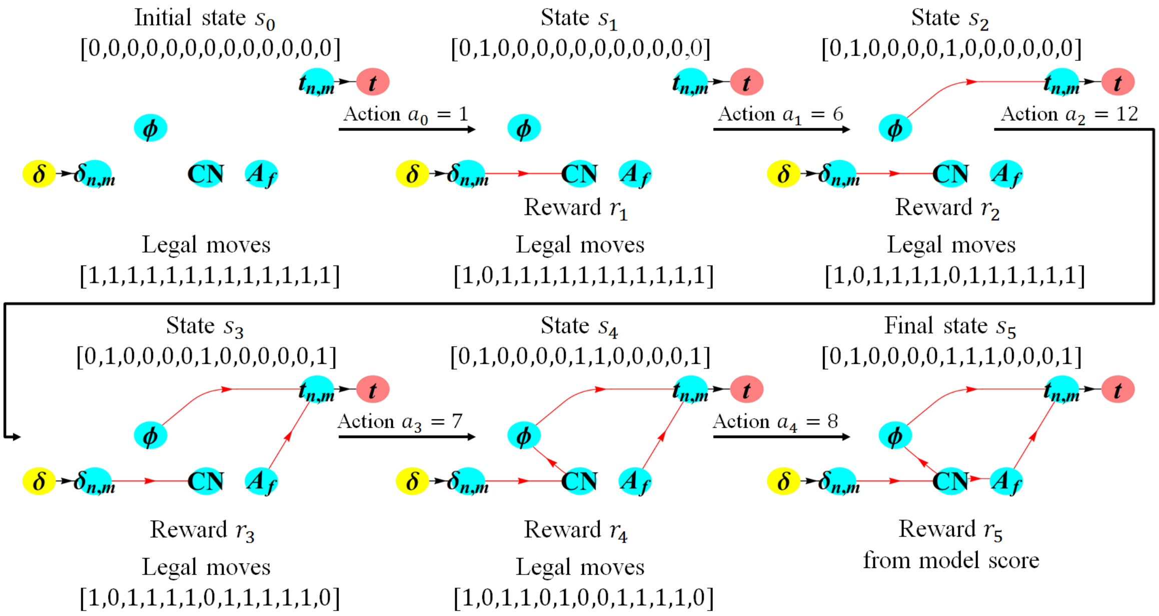


Figure 5.4: A gameplay example formalized as a Markov decision process (s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3, s_3, a_3, r_4, s_4, a_4, r_5, s_5) for the digraph game in Figure 5.3. The states are lists of binaries for 13 edges [$\delta_{n,m} \rightarrow \phi$, $\delta_{n,m} \rightarrow CN$, $\delta_{n,m} \rightarrow \boldsymbol{A}_f$, $\delta_{n,m} \rightarrow t_{n,m}$, $\phi \rightarrow CN$, $\phi \rightarrow \boldsymbol{A}_f$, $\phi \rightarrow t_{n,m}$, $CN \rightarrow \phi$, $CN \rightarrow \boldsymbol{A}_f$, $CN \rightarrow t_{n,m}$, $\boldsymbol{A}_f \rightarrow \phi$, $\boldsymbol{A}_f \rightarrow CN$, $\boldsymbol{A}_f \rightarrow t_{n,m}$]. The actions a are integers $\in [0, 12]$ (the list indices start from 0) indicating the next edge ID to activate. The legal moves are lists of binaries indicating whether the edges are allowed to be activated next. Final reward r_5 is determined by the model score evaluated at the end of the game. r_{1-4} are only estimated by ”intuitions” on whether the current policy can lead to a win or not, until r_5 is known. Note that the Markov decision process leading to the final digraph configuration s_5 is not unique.

within these paths are identified. Recursively going upstream along all the information flows, the predecessors for these nodes are identified, until the final predecessor node is the start node δ only. All the predecessor-successor node pairs can be connected by either mathematical equations frequently used in handcrafted constitutive models (linear, quadratic, exponential, power law, etc.) or artificial neural networks. In this work, we take the advantage of the flexibility of ANNs that they are universal function approximators to continuous functions of various complexity on compact subsets of R^n (Universal approximation theorem, [97]). Moreover, a special type of ANN, recurrent neural network (e.g., long short-term memory LSTM [96], gated recurrent units GRU [41, 42]), can capture the function of a time series of inputs, which is ideal for replicating the path-dependent material behavior. Hence, we only focus on ANN edges, without loss of generality of the meta-modeling games. The hybridized constitutive models with both mathematical equation edges and ANN edges will be studied in a separate research. The predecessor-successor node pairs are also inputs and outputs of all ANNs involved in the constitutive model. For example, there are two paths in the final digraph s_5 in Figure 5.4: $\{\boldsymbol{\delta} \rightarrow \delta_{n,m} \rightarrow CN \rightarrow \boldsymbol{A}_f \rightarrow t_{n,m} \rightarrow \boldsymbol{t}\}$ and $\{\boldsymbol{\delta} \rightarrow \delta_{n,m} \rightarrow CN \rightarrow \phi \rightarrow t_{n,m} \rightarrow \boldsymbol{t}\}$. Then the three required ANNs are, represented as input-output pairs, $[\delta_{n,m} \rightarrow CN]$, $[CN \rightarrow \phi, \boldsymbol{A}_f]$ and $[\phi, \boldsymbol{A}_f \rightarrow t_{n,m}]$. The parameters in each ANN are calibrated with training data of the input and output features using back propagations. The final output of $\boldsymbol{t}$ is predicted by executing consecutively the ANNs following the established paths from $\boldsymbol{\delta}$ to $\boldsymbol{t}$ in the directed graph. In the numerical examples of this chapter, the same neural network architecture is used for all ANNs for all edges in the directed graph: two hidden layers of 32 GRU neurons in each layer, and the output layer is a dense layer with linear activation function. All input and output data are pre-processed by standard scaling using mean values and standard deviations [172]. Each input feature considers its current value and 19 history values prior to the current loading step. Each ANN is trained for 1000 epochs using the Adam optimization algorithm [109], with a batch size of 256.

This particular form of ANN configurations is chosen after a number of trial-and-error numerical experiments similar to the one described in greater details in [247]. These experiments shown that the performance of the models is not particularly sensitive to small changes of the design parameters we considered (e.g. number of layers, number of neurons per layer), provided that a recurrent network is used and successfully trained. In fact, we found that the prediction accuracy does not exhibit significant different while we replace the long short-term memory neurons with the gated recurrent neurons. This robustness ensures that the predictions of each edges are sufficiently accurate and therefore potentially leads to more accurate predictions from the entire directed graph and ultimately higher game rewards.

In principle, other designs of neural networks can also be used as the universal function approximators for the digraph edges, provided that the true path-dependent relations are accurately captured. While it is possible that there exists better RNN setups, determining this optimal setup from all possible combinations of choices manually will nevertheless require a significant amount of trial-and-error effort. One way of overcoming this obstacle is to use the reinforcement learning to automate this trial-and-error procedure which determines the optimal setup for each edge in the directed graph. Exploring this fine-tuning option is out of the scope of this study, but we will nevertheless investigate this further in future studies.

5.5 Deep reinforcement learning for generating constitutive laws

With the game of constitutive modeling completely defined, a deep reinforcement learning (DRL) algorithm is employed as a guidance of taking actions in the game to maximize the final model score (Figure 7.6). This tactic is considered one of the key ideas leading to the major breakthrough in AI playing the game of Go (AlphaGo Zero) [196], Chess and shogi (Alpha Zero) [195] and many other games. The learning is completely free of human interventions. It does not need previous human knowledge in traction-separation model as a starter database. The AI agent simply learns to improve from a number of games it played and from the corresponding model scores and game rewards, even if the initially generated digraph configurations make very little sense for a traction-separation model. Moreover, during the self-plays and training, no human guidance is needed.

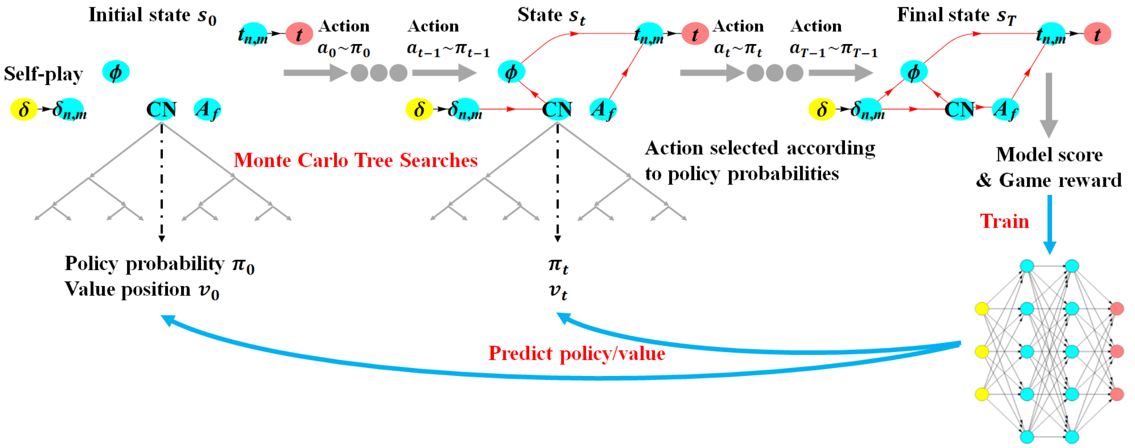


Figure 5.5: Self-play reinforcement learning of traction-separation law.

A (deep) neural network f_θ with parameters θ (weights, bias, ... of the artificial neurons) takes in the current configuration of the directed graph of the constitutive law s and outputs a policy vector $\boldsymbol{p}$ with each component $p_a = p(s, a)$ representing the probability of taking the action a from state s, as well as a scaler v estimating the expected score of the constitutive law game from state s, i.e.,

$$(\boldsymbol{p}, v) = f_\theta(s). \tag{5.9}$$

These outputs from the policy/value network guide the game play from the AI agent.

At each state s, the action to take is sampled from an action probability $\boldsymbol{\pi}(s)$. This probability is based on the policy $\boldsymbol{p}$ predicted from the neural network enhanced by a Monte

Carlo Tree Search (MCTS) ([34]). The search tree is composed of nodes representing states s of the game, and edges representing permitted actions a from s. Each edge (s, a) possesses a list of statistics $[N(s, a), W(s, a), Q(s, a)]$, where $N(s, a)$ is the number of visits to the edge during MCTS search, $W(s, a)$ is the total action value and $Q(s, a) = \frac{W(s,a)}{N(s,a)}$ is the mean action value. The search procedure consists of firstly a recursive selection of a sequence of optimal actions $a^0, a^1, a^2, ...$ leading to the corresponding child states $s^1, s^2, s^3, ...,$ starting from the root state s^0, until a leaf node of state s^l (that has never been encountered before in the search) is reached. The criteria for selection from a state s is that the action a maximizes the upper confidence bound $U(s, a)$ of the Q-value, among all valid actions. The upper bound is defined as

$$U(s,a) = Q(s,a) + U_Q(s,a) = Q(s,a) + c_{puct} p(s,a) \frac{\sqrt{\sum_b N(s,b)}}{1 + N(s,a)}. \tag{5.10}$$

where c_{puct} is a parameter controlling the level of exploration. If s^l is not a terminal state that ends the game, then its $\boldsymbol{p}(s^l)$ and $v(s^l)$ are predicted from the policy/value neural network $f_\theta(s^l)$. The search tree is expanded and the statistics for each edge (s^l, a) is initialized to $[N(s, a) = 0, W(s, a) = 0, Q(s, a) = 0]$. Otherwise, $v(s^l)$ is equal to the final reward of the constitutive modeling game. Finally, $v(s^l)$ is propagated back to the parent states $\{s^0, s^1, s^2, ...s^l\}$ and actions $\{a^0, a^1, a^2, ...a^{(l-1)}\}$ traversed during the seach. Their statistics are updated as

$$N(s^i, a^i) = N(s^i, a^i) + 1,\ W(s^i, a^i) = W(s^i, a^i) + v(s^l),\ Q(s^i, a^i) = \frac{W(s^i, a^i)}{N(s^i, a^i)}, \text{ for all } i < l. \tag{5.11}$$

The MCTS procedure is repeated a number of times. The searches in MCTS eventually yield a vector of search probabilities $\boldsymbol{\pi}(s^0)$ recommending actions to take from the root position s^0. $\boldsymbol{\pi}(s^0)$ is proportional to the exponentiated visit count for each edge, i.e.,

$$\boldsymbol{\pi}(s^0, a) = \frac{N(s^0, a)^{-\tau}}{\sum_b N(s^0, b)^{-\tau}}, \tag{5.12}$$

where τ is a positive temperature parameter that also controls the level of exploration. The MCTS algorithm for the game of constitutive models is illustrated in Figure 5.6.

During one episode of self-play by the AI agent, the above MCTS algorithm is executed for each state s_t in the sequence of encountered states $\{s_0, s_1, s_2, ..., s_{T-1}\}$. The root node s^0 of the search tree is set to s_t as the game progresses to the state s_t. All child nodes and their statistics constructed in the MCTS for the prior game states are preserved. The training data for the neural network consists of $(s_t, \boldsymbol{\pi}_t, z_t)$ obtained from a number of full plays of the constitutive law game guided by the aforementioned reinforcement learning algorithm. $\boldsymbol{\pi}_t$ is the estimation of policy after performing MCTS from state s_t and z_t is the reward of the generated constitutive model at the end of the game s_T. The loss function to be minimized by adjusting parameters θ using back propagation is,

$$1 = \sum_t (v(s_t) - z_t)^2 + \sum_t \boldsymbol{\pi}_t \log[\boldsymbol{p}(s_t)], \tag{5.13}$$

which is the combination of mean squared errors in game reward (1 or -1) and cross-entropy losses in policy probabilities. Hence, accordingly, the activation functions for the output layer are the hyperbolic tangent function $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ and the softmax function [152]. The procedure of DRL guided self-plays and the following training of the network f_θ is iterated until the score of the generated directed graph of the constitutive model does not improve further.

The pseudocode of the reinforcement learning algorithm designed to play the metamodeling game and improves via self-play is presented in Algorithm 1. As demonstrated

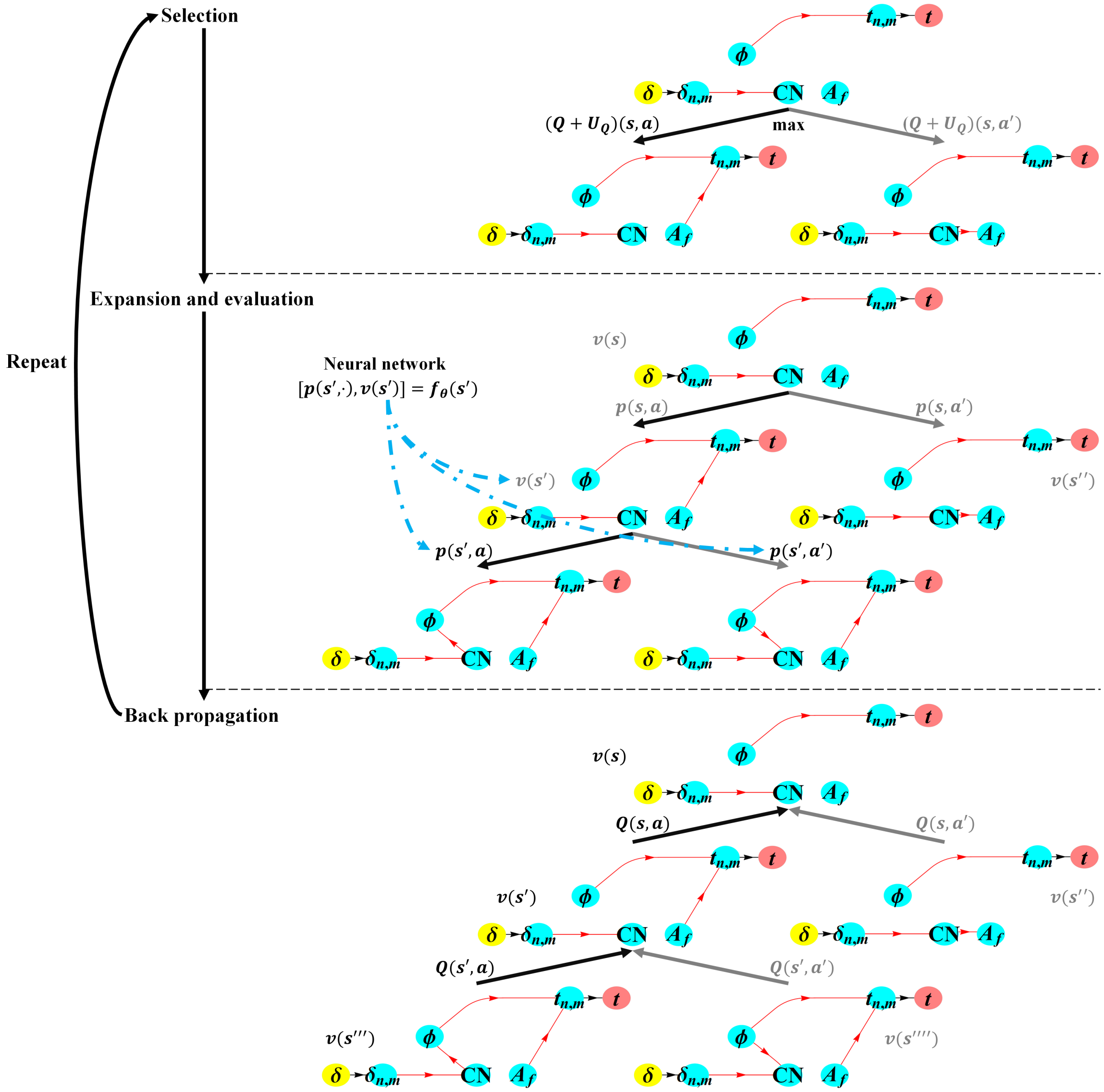


Figure 5.6: Monte Carlo Tree Search (MCTS) in a game of constitutive models (figure design adopted from [196]). A sequence of actions are selected from the root state s^0, each maximizing the upper confidence bound $Q(s,a)+U_Q(s,a)$. The leaf node s^l is expanded and its policy probabilities and position value are evaluated from the neural network $(\boldsymbol{p}(s^l), v(s^l)) = f_\theta(s^l)$. The action values Q in the tree are updated from the evaluation of the leaf node. Finally the search probability $\boldsymbol{\pi}(s^0)$ for the root state s^0 is returned to guide the next action in self-play.

in Algorithm 1, 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.

This design of algorithm is very similar to the one used in AlphaGo Zero to play and discover knowledge from the Game of Go and chess (cf. [196, 195]). The major difference is that we are now applying the reinforcement learning on a newly design game in which discoveries of physical relationships and quality of blind predictions are all based on making the right decisions in the step-by-step construction of directed graph rather than the movement of the game piece. This similarity of the algorithm design is encouraging, as it demonstrate that the DRL aglorithm is closer to reach the goal of becoming a form of general artificial intelligence and that it can can be easily applied to many other decision making ”games” designed for different research fields [146, 14].

5.6 Numerical Experiments

In this section, we present two traction-separation modeling games with different digraph complexities to demonstrate the intelligence, robustness and efficiency of the deep reinforcement learning algorithm on improving the accuracy and consistency of the generated traction-separation models through self-plays. For both examples, sub-scale discrete element simulations (DEM) are used to generate synthetic benchmark data for model calibrations and blind prediction evaluations. The accuracy scores of the DRL-generated models are evaluated by material point tests with deformation histories identical to those used to

Algorithm 1 Self-play reinforcement learning of the meta-modeling game

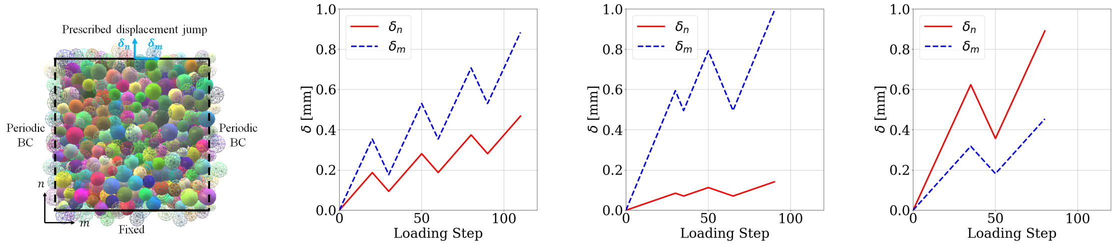
```
Require: The definition of meta-modeling game: game environment, state, action, model score, reward, game rules (Section 7.4).
Randomly initialize the policy/value network $f_\theta$.
Initialize empty set of the training examples $trainExamples \leftarrow []$.
for i in [0,..., $numIters$ (number 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 $trainExamples$.
    Train the policy/value network $f_\theta$ with $trainExamples$.
Use the final trained network $f_\theta$ and set $\tau = 0.01$ in MCTS for one more iteration of ”competitive gameplays” ($numEpisodes$ games) to generate the final converged digraph model.
Exit
```

generate the benchmark data.

5.6.1 Generation of synthetic data from discrete element modeling (DEM)

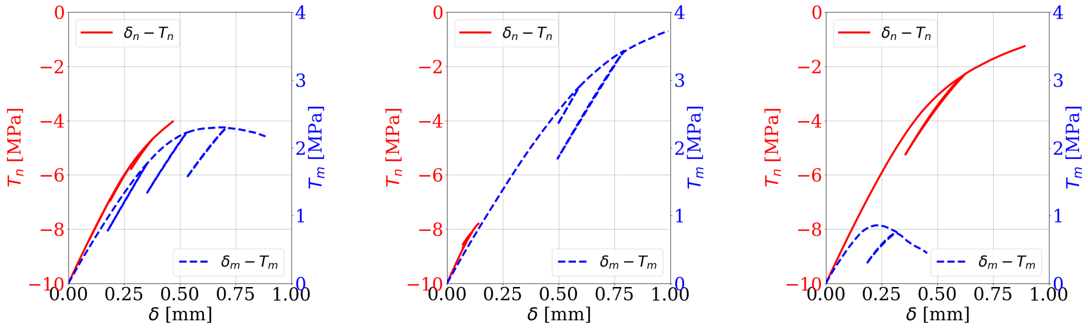
The data for calibration and evaluation of prediction accuracy of the deep-reinforcement-learned traction-separation models are generated by numerical simulations on a representative volume element (RVE) representing the granular materials on a frictional surface. The open-source software YADE for DEM is used [202]. The discrete element particles in the RVE have radii between 1 ± 0.3 mm with uniform distribution. The RVE has the height of 20 mm in the normal direction of the frictional surface and is initially consolidated to isotropic pressure of 10 MPa. The Cundall's elastic-frictional contact model ([49]) is used for the inter-particle constitutive law. The material parameters are: interparticle elastic modulus $E_{eq} = 1$ GPa, ratio between shear and normal stiffness $k_s / k_n = 0.3$, frictional angle $\varphi = 30^\circ$, density $\rho = 2600\ kg/m^3$, Cundall damping coefficient $\alpha_{damp} = 0.2$.

The DEM RVE is loaded in the normal $\boldsymbol{n}$ and tangential $\boldsymbol{m}$ directions of the frictional surface by displacement controls δ_n and δ_m (Figure 5.7(a)). The synthetic database consists of 200 numerical experiments under different loading paths. They differ from each other in the ratio of normal and tangential loading rate $\dot{\delta_n} / \dot{\delta_m}$, as well as the loading-unloading-reloading cycles, as illustrated in Figure 5.7(b), 5.7(c) and 5.7(d). The traction-separation curves of the experiments are recorded and three examples corresponding to the paths in Figure 5.7 are presented in Figure 5.8. The microstructural information required for the intermediate nodes in the directed graphs, such as porosity, coordination number and fabric tensor, are also recorded during the simulations. The open-source library NetworkX [90] is employed to analyze the graph of the particle interactions in the RVEs. Figure 5.9 presents examples of microstructural information and graph characteristics for the three example loading paths. The 200 numerical simulations in the database are shuffled. The first 50 simulations are used as ”training data” for the calibration of model parameters for the edges in the directed graphs. The other 150 simulations are ”test data” only for evaluating the blind prediction accuracy of the resultant constitutive model.



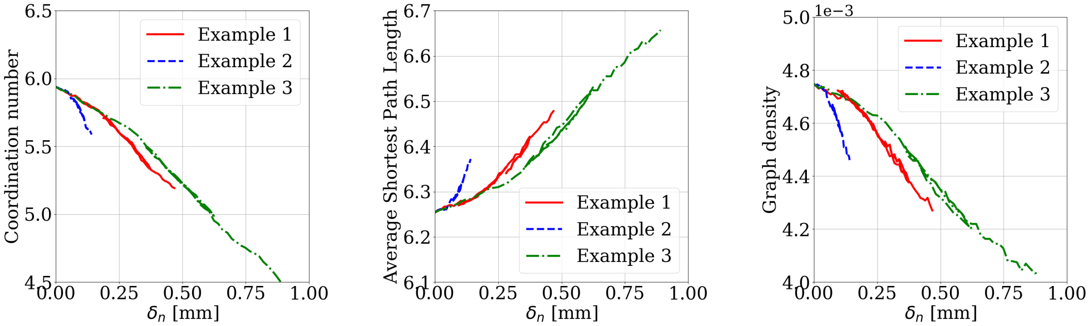
(a) RVE of frictional surface **(b)** Example loading path 1 **(c)** Example loading path 2 **(d)** Example loading path 3

Figure 5.7: Representative volume element of a frictional surface having normal $\boldsymbol{n}$ and tangential $\boldsymbol{m}$ directions. Three examples of different loading-unloading-reloading paths among 200 numerical experiments for the generation of database are shown.



(a) Example loading path 1 **(b)** Example loading path 2 **(c)** Example loading path 3

Figure 5.8: Examples of traction-separation curves corresponding to the three loading paths in Figure 5.7 among 200 numerical simulations. The normal traction T_n is plotted against the normal displacement jump δ_n. The tangential traction T_m is plotted against the tangential displacement jump δ_m.



(a) Coordination number **(b)** Average Shortest Path Length **(c)** Graph density

Figure 5.9: Examples of coordination number, average shortest path length and graph density for particle interactions corresponding to the three loading paths in Figure 5.7 among 200 numerical simulations. These quantities are plotted against the normal displacement jump δ_n.

5.6.2 Numerical Experiment 1: Determining optimal physical relationships for traction-separation laws

In the first example, our goal is to test the DRL algorithm and see whether it can determine the optimal topological relations among microstructural physical quantities of porosity ϕ, coordination number CN and fabric tensor $\boldsymbol{A}_f$. In [247], the authors use domain expertise, i.e., knowledge from previous literature on fabric tensor and critical state theory to deduce that the porosity and fabric tensor can be used as state variables to improve the forward prediction accuracy of the traction-separation law (cf. [72, 125, 206, 244]). In this work, we do not make any assumption or introduce any interpretation to the meta-modeling computer agent. Instead, we simply make a number of physical quantities measured from discrete element simulations available as vertices in the directed graph but do not introduce any relation (edge) manually. In other words, the edge set that represents the relations of the physical quantities is self-discovered by the computer agent from the reinforcement learning without any human intervention. We document our training procedure and analyze the performance of the models generated by the meta-modeling approach.

The directed graphs, states, actions, rewards and game rules of the modeling game have been defined in Sections 5.2 and 7.4, and illustrated in Figures 5.2, 5.3 and 5.4. The action space is of dimension 13. Through exhaustive plays of the game, the authors count 3200 possible game states, among which 591 states represent complete and admissible directed graph configurations according to the game rules. The model score is defined as:

$$\text{SCORE} = 0.45 * A^{\text{calibration}}_{\text{accuracy}} + 0.45 * A^{\text{prediction}}_{\text{accuracy}} + 0.1 * A_{\text{consistency}}, \tag{5.14}$$

where $P\% = 90\%$ and $\varepsilon_{\text{crit}} = 1e^{-6}$ for accuracy evaluations and $\alpha_{\text{gof}} = 1\%$ for consistency evaluations. The training data for model calibration contains 50 loading cases, and the test data for forward prediction evaluation contains 150 loading cases.

The DRL meta-modeling procedure (Algorithm 1) 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 $numEpisodes = 20$ self-play episodes of the game. Hence one execution of the entire DRL procedure contains $numIters * numEpisodes = 10 * 20 = 200$ 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 require $numMCTSSims = 20$ 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.

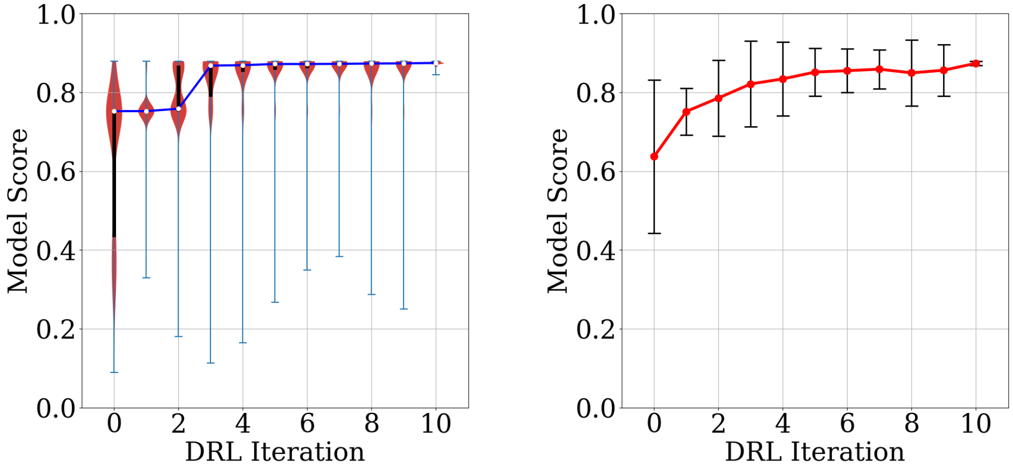
Because each run of the DRL procedure (Algorithm 1) is initialized randomly and the MCTS simulations also involve probability of the action possibilities, the gameplays in each individual DRL procedures could, in theory, lead to different resultant directed graph model for the same set of data, especially if the exploration is not sufficient. To analyze how much influence does the initial randomly directed graph affects the outcome of the gameplay, we conduct a numerical experiment in which 20 independent DRL procedures, each with a different initial configuration, are ran. We then compare the results obtained from these DRL procedures to assess the statistical performance of the DRL and also check whether they all converged to the same directed graph. The gameplay results are presented in Figure 5.10. The number of gameplays required for each DRL algorithm to successfully generate the optimal digraph model is 200, which is about 33% of total 591 possible digraphs the brutal-force approach required to evaluate and rank. such that the optimal graph can be determined. Yet, these 20 independent DRL procedures with different initial configurations all lead to the same optimal digraph. This optimal digraph is identical to the one determined from the brutal-force approach in our benchmark study.

At the first DRL iteration, the AI agent only knows the rule of the game without human knowledge on which physical quantities are essential in predicting the traction and how they should be connected. The AI just plays with trial-and-error following strategies guided by random initial neural network and MCTS. This lack of gameplay knowledge can be seen from the widely spread density distribution of model scores between maximum and minimum scores, large interquartile range between 25% and 75%, and the large standard deviation (Figure 5.10). In the subsequent iterations, the AI plays with increasing knowledge of game play reinforced by the ultimate game rewards, and it shows intelligence in keep playing games with better outcomes. This is shown by the increase in median and average of scores, the narrowing of interquartile range and the migration of the density distribution towards higher scores. The automatic learning is very efficient.

Statistically, after 5 iterations (100 games out of the total 591 possible game outcomes), the scores already concentrate around the maximum. Few bad games could be played, since the AI is still allowed to explore different game possibilities to avoid convergence to local maximum. The strength of the AI agent after 10 iterations is tested by suppressing the "exploration plays", and the outcome game scores show outstanding performances. Figure 5.11 illustrates the improvement of knowledge of traction-separation constitutive modeling by four representative digraph games played during the DRL iterations. The traction predictions from the resultant constitutive models are compared against both training data and unseen test data. In addition, four examples of blind predictions from the optimal digraph configuration (the 4th digraph in Figure 5.11) obtained in this game are shown in Figure 5.12.

5.6.3 Numerical Experiment 2: Data-driven discovery for enhancement of traction-separation laws

In the second example, we consider another common scenario in which we attempt to convert qualitative observations into quantitative predictions with the help of the reinforcement learning algorithm as a tool for augmented intelligence. The need to interpret observations of mechanisms into predictions is one of the oldest problems in constitutive modeling [167]. For instance, the observation that yielding depends on the amount of normal traction leads to the Mohr-Coulomb yield criterion [225]. The evolution of fabric tensor has been incorporated into the hardening law and the plastic flow rule to capture the induced anisotropy and critical state of sand [141, 52, 53]. However, recent advancements on the application of graph theory as well as the experimental techniques such as micro-CT imaging have revealed many geometric measures on the grain connectivity that



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

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

Figure 5.10: 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 20 games. (a) Violin Plot of model scores played in each DRL iteration. The shade 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.

help explain the onset of shear band [227, 226], coherent vortex structure [256] and post-bifurcation behaviors in granular materials [214, 241, 130]. While these discoveries of new knowledge are indeed encouraging, one cannot make use of them without investing significant efforts and time to derive, verify, and validate new constitutive laws that incorporate new information. Hence the graph-theoretical approaches, although have found great promises on analyzing the granular assemblies obtained from real or virtual exper-

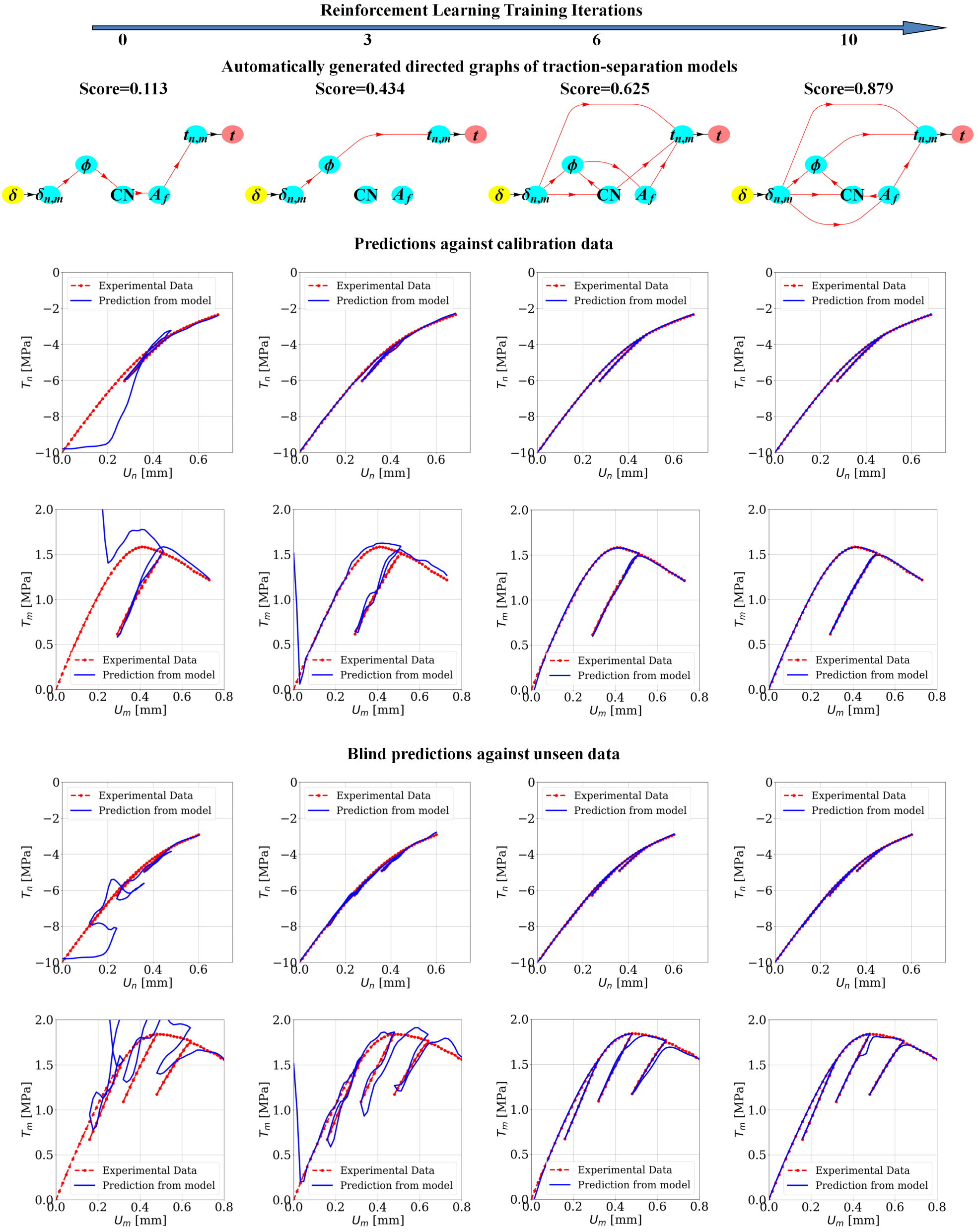


Figure 5.11: Knowledge of directed graphs of traction-separation models learned by deep reinforcement learning in Numerical Experiment 1. Four representative digraph games played during the DRL iterations and their prediction accuracy against training and test data are presented.

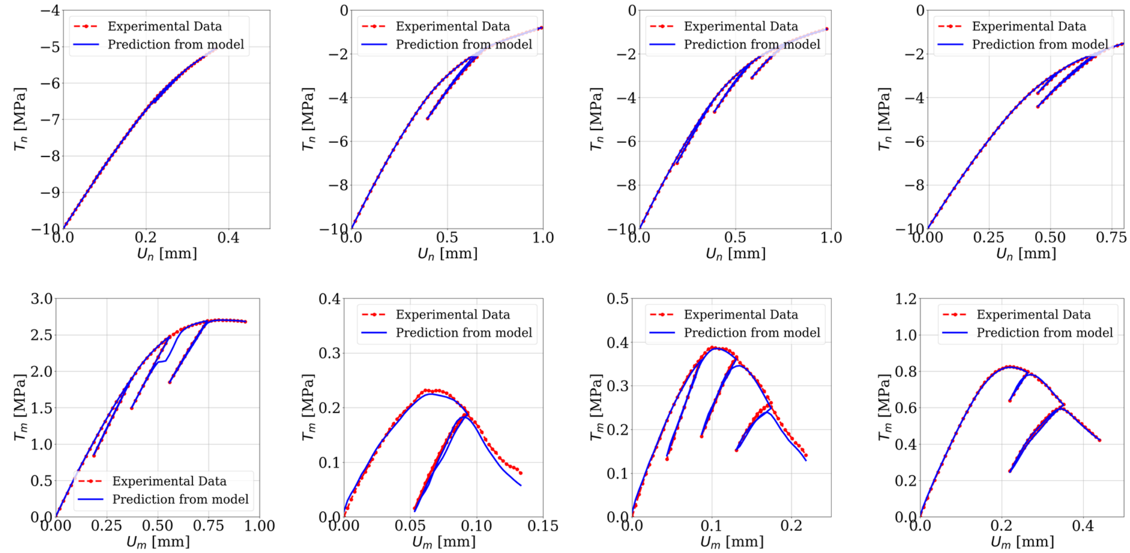
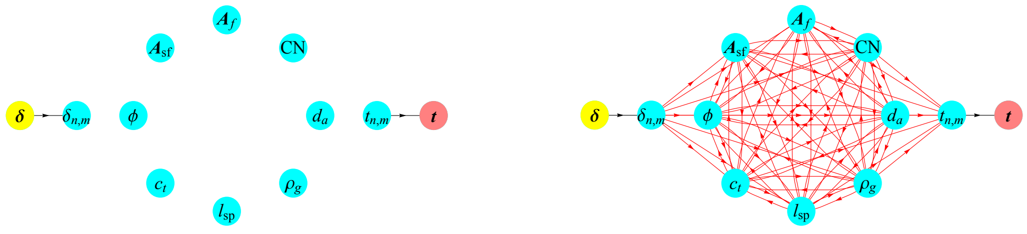


Figure 5.12: Four examples of blind predictions from the optimal digraph configuration (the 4th digraph in Figure 5.11) against unseen data among test database of 150 loading cases.

iments, have not yet made significant impacts on constitutive laws used for engineering applications. Our meta-modeling approach is capable of overcoming this bottleneck by efficiently automating some of these tasks currently undertaken by modelers. This second numerical experiment is used to demonstrate how the augmented intelligence can be used to incorporate the insights from observations into predictions without manually re-writing an existing constitutive law every time new information comes up.

This example is an extension of the first numerical experiment, in which more microstructual information are considered, including the fabric of strong interactions A_{sf} and four measures of grain connectivities d_a, c_t, l_{sp}, ρ_g. The task of identifying their roles in constitutive models for granular materials is now simply recast as defining a new game with augmented vertex set in digraph and extended action space. The ”game board” and all possible actions for this new game are shown in Figure 5.13. The dimension of the action space increases from 13 to 71. A particular game rule is added to test the flexibility of the DRL algorithm in handling different types of game constraints: the strong fabric tensor A_{sf} and the fabric tensor A_f, since both are geometric measures of inter-particles forces, are mutually exclusive in the final digraphs of constitutive models. The number of possible game states increases from 3200 to over 400000. The number of complete and admissible directed graph configurations increases from 591 to over 20000. The score definition is the same as Equation (5.14). The meta-modeling algorithm tries to learn the optimal ways to incorporate the microstructual information and make better predictions only from the training database of 50 loading cases, while the gained knowledge is validated on the test database of another 150 loading cases. The parameters for the DRL meta-modeling algorithm are set as: $numIters = 10$ iterations of ”exploration and exploitation”, 1 iteration of ”competitive gameplay”, $numEpisodes = 30$ self-plays in each iteration, and $numMCTSSims = 30$ MCTS simulations in each play step. Hence only a total number of 300 game episodes are needed to complete the DRL procedure, comparing to more than 20000 possible digraph models in this game setting.



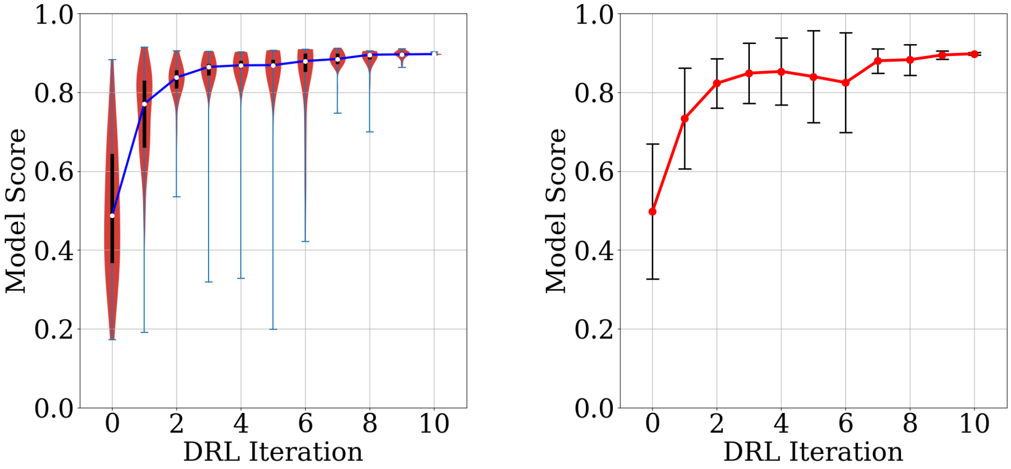
(a) Initial configuration of the ”game board” **(b)** All possible actions on the ”game board”

Figure 5.13: A game of traction-separation model for a digraph involving the nodes $\{\delta_{n,m}, t_{n,m}, \phi, CN, \boldsymbol{A}_f, \boldsymbol{A}_{sf}, d_a, c_t, l_{sp}, \rho_g\}$ (detailed in Section 5.2). (a) the initial ”board” on which the game is played. (b) All possible actions for picking the edges connecting the nodes are represented by the red arrows.

The statistics of the gameplay results from 5 separate runs of the DRL procedure are presented in Figure 6.4. We observe a very efficient improvement in generated traction-separation models, even though the number of legal game states in the new game has largely increased. Figure 5.15 exhibits four representative digraph configurations developed during DRL iterations, as well as their prediction quality on calibration data and unseen data. It can be seen that the information flows in a constitutive model are of crucial importance. Although the first and the fourth graphs both incorporate the same types of microstructual information, the difference in the ways these information are connected results in significant difference in model scores of 0.191 and 0.915, respectively. Moreover, the DRL algorithm develops the intelligence of selecting the strong fabric tensor A_{sf} over the fabric tensor A_f in order to further improve the prediction score of the model. Four blind prediction examples of the optimal digraph configuration (the 4th digraph in Figure 5.15) obtained in this game are presented in Figure 5.16. Comparing to the numerical example 1 (Figure 5.12), the augmented knowledge of additional microstructural information in constitutive models lead to more accurate representations of granular materials.

5.7 Conclusions

This chapter presents a new meta-modeling framework to employ deep reinforcement learning (DRL) to generate mechanical constitutive models for interfaces. The constitutive models are conceptualized as information flow in directed graphs. The process of writing constitutive models is simplified as a sequence of forming graph edges with the goal of maximizing the model score (a function of accuracy, robustness and forward prediction quality). Thus meta-modeling can be formulated as a Markov decision process with well-defined states, actions, rules, objective functions and rewards. By using



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

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

Figure 5.14: 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 games. (a) Violin Plot of model scores played in each DRL iteration. The shade 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.

neural networks to estimate policies and state values, the computer agent is able to efficiently self-improve the constitutive model it generated through self-playing, in the same way AlphaGo Zero (the algorithm that outplayed the world champion in the game of Go) improves its gameplay. Our numerical examples show that this automated metamodeling framework not only produces models which outperform existing cohesive models on benchmark traction-separation data, but is also capable of detecting hidden mech-

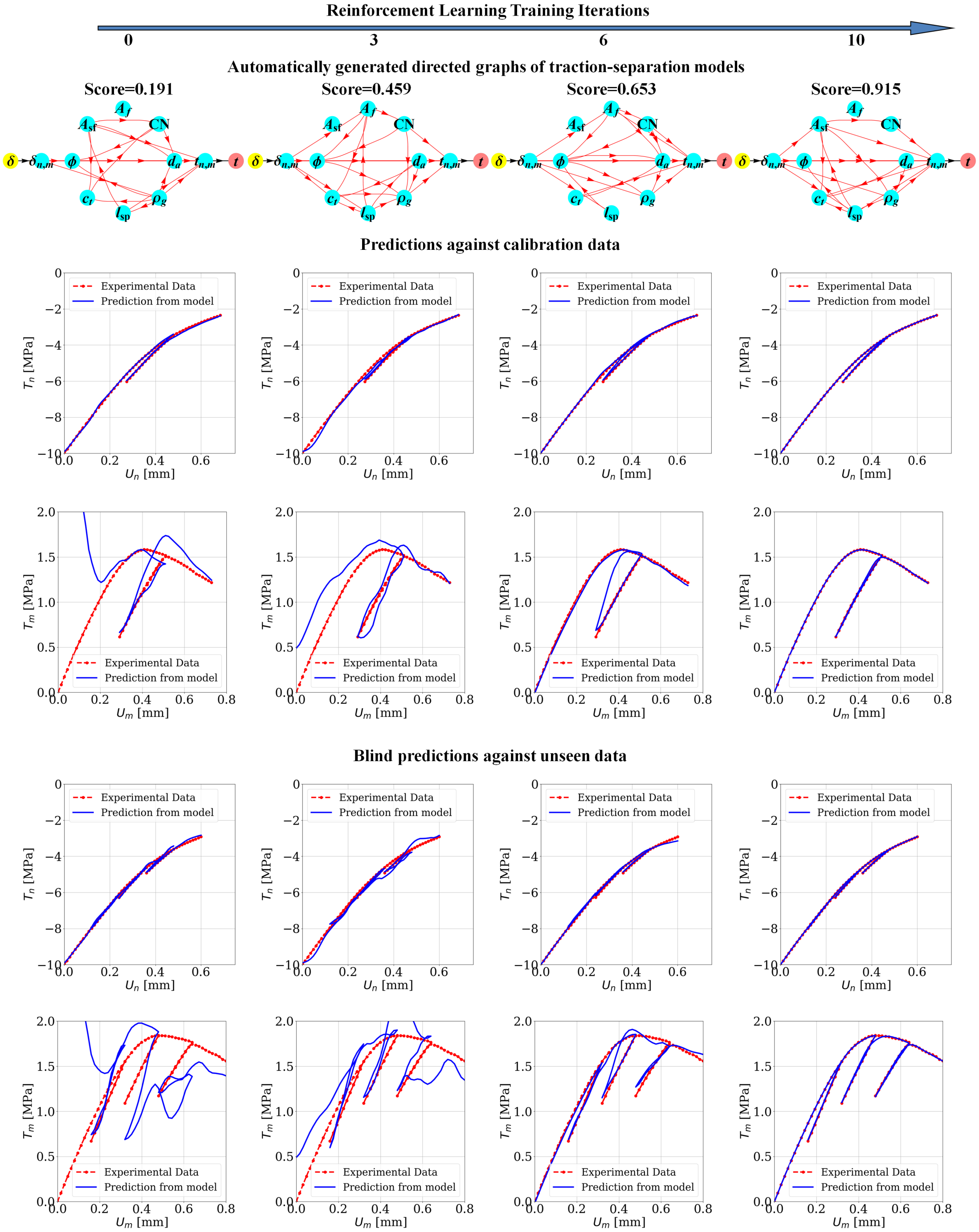


Figure 5.15: Knowledge of directed graphs of traction-separation models learned by deep reinforcement learning in Numerical Experiment 2.

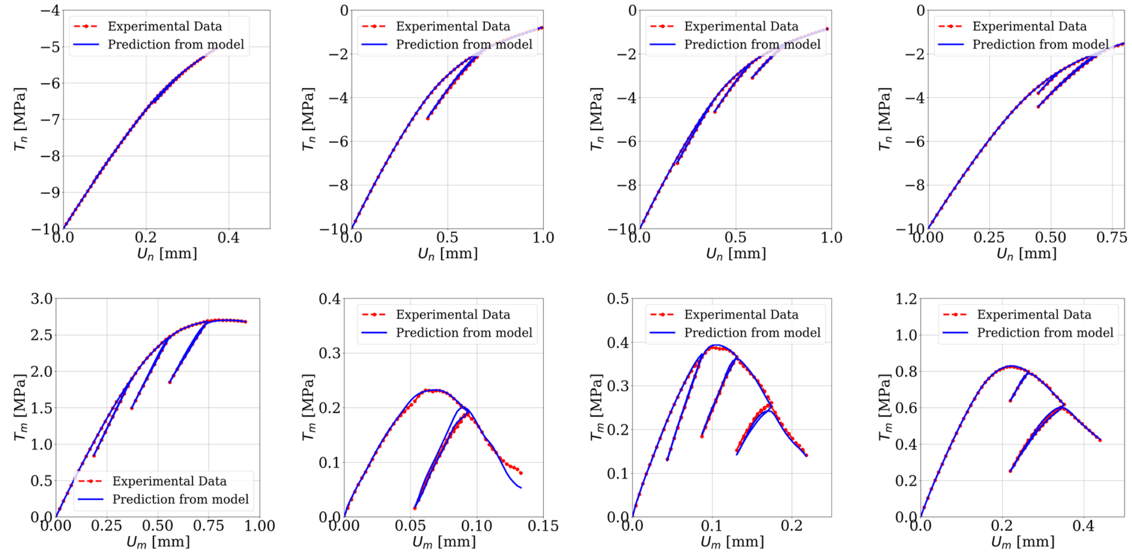


Figure 5.16: Four examples of blind predictions from the optimal digraph configuration (The 4th digraph in Figure 5.15) against unseen data among test database of 150 loading cases.

anisms among micro-structural features and incorporating them in constitutive models to improve the forward prediction accuracy, which are difficult tasks to do manually.

Chapter 6

Cooperative metamodeling game for automated learning of elastoplasticity models for geomaterials

This chapter is reproduced from the published paper: K. Wang, W.C. Sun, Q. Du, A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with AI-guided experimentation, *Comput Mech*, 64: 467, 2019.

6.1 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) [187, 188, 257, 44]. 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 [111, 101, 247].

In this chapter, 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 to 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. [77, 111, 247]). 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. [247]).

Within our framework, a black-box neural networks 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 [77, 122] and [247] 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 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 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 connection among events and mechanisms. As point out in [116], this model-building approach has an advantage over the model-free machine learning approaches in interperability. As considerable evidence has indicated that the model-based planning, such as the one introduced in this chapter, 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 [100, 14]. 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.

The rest of the chapter is organized as follows. We first introduce the meta-modeling cooperative game, including the method to recast a model as directed multigraph, and the generation of decision tree (Section 6.2). 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 (Section 7.4). In Section 6.4, 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 maximize 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.

For convenience, we provide a minimal review of the essential terminologies and concepts from graph theory that are used throughout this chapter. Their definitions can be found in, for instance, [253, 12].

Definition 6.1.1. 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 6.1.2. 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 6.1.3. 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 6.1.4. 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, $\boldsymbol{s} : \mathbb{E} \rightarrow \mathbb{V}$ is a mapping that maps each edge to its source node, and $\boldsymbol{t} : \mathbb{E} \rightarrow \mathbb{V}$ is a mapping that maps each edge to its target node.

Definition 6.1.5. An **underlying graph** $\mathbb{U}$ of a directed multigraph $\mathbb{G}$ is a multigraph whose edges are without directions.

Definition 6.1.6. 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 6.1.7. 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, $s : \mathbb{E} \rightarrow \mathbb{V}$ and $t : \mathbb{E} \rightarrow \mathbb{V}$ are the mappings that map the edges to the source and target vetrices, $n_V : \mathbb{V} \rightarrow \mathbb{L}_{\mathbb{V}}$ and $n_E : \mathbb{E} \rightarrow \mathbb{L}_{\mathbb{E}}$ are the mappings that give the vertices and edges the corresponding labels in $\mathbb{L}_{\mathbb{V}}$ and $\mathbb{L}_{\mathbb{E}}$ accordingly.

6.2 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. [78, 122]) 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 chapter, 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.

6.2.1 Material modeling algorithm as a directed multigraph

The architecture of an algorithm is often considered as a directed multigraph [51]. 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 governs 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 [247]. 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 $n_{\mathbb{V}} : \mathbb{V} \rightarrow \mathbb{L}_{\mathbb{E}}$ 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 $\boldsymbol{s} : \mathbb{E} \rightarrow \mathbb{V}$ and $\boldsymbol{t} : \mathbb{E} \rightarrow \mathbb{V}$ that provide the direction of the information flow, and the surjective mapping $\boldsymbol{n}_{\mathbb{E}} : \mathbb{E} \rightarrow \mathbb{L}_{\mathbb{E}}$ that assigns the edge labels (names) to the edges.

6.2.2 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) [111, 247]. 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 [196, 197]. 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 $f_i(l_1, l_2, \dots, l_n) = 0, i = 1, \dots, m$ where , i.e.,

$$\begin{aligned} \underset{l_i}{\text{maximize}} \quad & \text{SCORE}(l_1, l_2, \dots, l_n) \\ \text{subject to} \quad & f_i(i_i) = 0, \; i = 1, \dots, m. \end{aligned} \tag{6.1}$$

6.3 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 [272, 236] 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 Figure 6.1. This lead to a multi-agent multi-objective problem that can be solved by a deep reinforcement learning framework [220, 181]. In this work, the deep reinforcement learning algorithm is based on a model-free policy gradient algorithm that employs neural network to estimate the Q values of the policies (cf. [197, 248]). 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 [79] for a comprehensive review on these reinforcement learning algorithms.

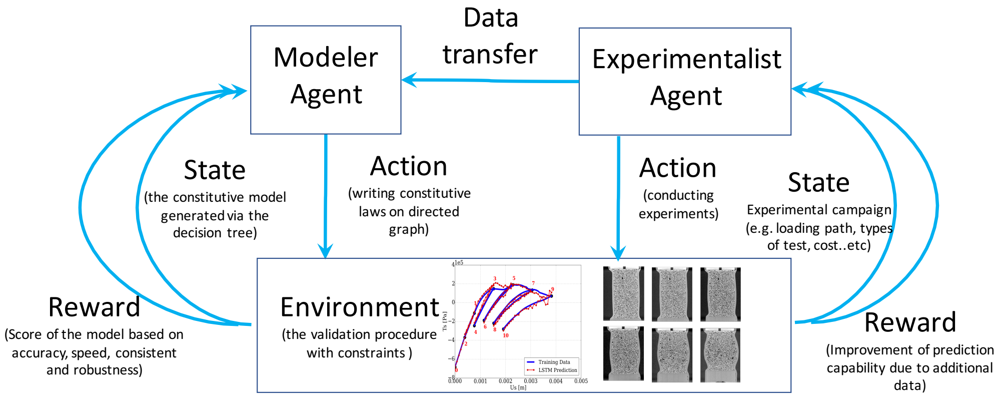

Data transfer
Modeler Agent
Experimentalist Agent
State
(the constitutive model generated via the decision tree)
Action
(writing constitutive laws on directed graph)
Action
(conducting experiments)
State
Experimental campaign (e.g. loading path, types of test, cost..etc)
Reward
(Score of the model based on accuracy, speed, consistent and robustness)
Environment
(the validation procedure with constraints)
Training Data
LSTM Prediction
Reward
(Improvement of prediction capability due to additional data)

Figure 6.1: 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).

6.3.1 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.

6.3.1.1 Game Board for Experimentalist

Consider a set of possible mechanical experiments on a material $\mathbf{T} = \{T_1, T_2, T_3, ..., T_n\}$. 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: $\mathbf{T}_c = \mathbf{T}_c^0 = \{T_{c1}, T_{c2}, T_{c3}, ..., T_{cn}\}$ and $\mathbf{T}_v = \mathbf{T}_v^0 = \{T_{v1}, T_{v2}, T_{v3}, ..., T_{vn}\}$. 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., $\mathbf{T}_v = \mathbf{T}_v^0 \cup (\mathbf{T}_c^0 \setminus \mathbf{T}_c)$. Hence the set $\mathbf{T}_c^0$ constitutes the ”game board” for the experimentalist agent to play the data collection game.

6.3.1.2 Game State for Experimentalist

The mathematical description of the current state of the game board is a list of binary indicators $s = [i_{c1}, i_{c2}, i_{c3}, ..., i_{cn}, i_{terminate}]$ 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.

6.3.1.3 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.

6.3.1.4 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 $LegalActions(s) = [ii_{c1}, ii_{c2}, ii_{c3}, ..., ii_{cn}, ii_{terminate}]$, 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 $LegalActions(s)$.

• 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 $ii_{terminate} = 1$.

6.3.1.5 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 $\mathbf{T}_v = \mathbf{T} \setminus \mathbf{T}_c$), the modeler could generate a number of different constitutive models with scores $[\text{SCORE}_{i,\ i=1,2,3,...}]_{\mathbf{T}_c}$. The reward should also consider the total cost of the calibration tests $\mathbf{T}_c$. This can be measured by a weighted sum $\mathrm{COST}(\mathbf{T}_c) = \sum^{\mathbf{T}_c^0} w_{ci}^{cost} * i_{ci}$, 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 (Figure 6.1), 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 ($[\mathrm{SCORE}_{i,\ i=1,2,3,...}]_{\mathbf{T}_c}$, $\forall \mathbf{T}_c \in \mathbb{T}_c^{\text{history}}$). Then, both agents have the knowledge of the highest model score for each $\mathbf{T}_c$: $\mathrm{SCORE}_{\mathbf{T}_c}^{\max} = \max([\mathrm{SCORE}_{i,\ i=1,2,3,...}]_{\mathbf{T}_c})$. Thus they know the highest model score in the history of self-played games: $\mathrm{SCORE}^{\max} = \max(\mathrm{SCORE}_{\mathbf{T}_c}^{\max})$, $\forall \mathbf{T}_c \in \mathbb{T}_c^{\text{history}}$. 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 $|\mathrm{SCORE}_{\mathbf{T}_c}^{\max} - \mathrm{SCORE}^{\max}| \leq \mathrm{TOL}$, 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 if its total cost is the lowest among all elements in $\mathbb{T}_c^{\max}$. Hence the reward is designed as

$$r(s) = \begin{cases} 1, & \text{if } \mathbf{T}_c \in \mathbb{T}_c^{\max} \text{and } \mathrm{COST}(\mathbf{T}_c) \leq \mathrm{COST}(\forall \mathbf{T}_c^i \in \mathbb{T}_c^{\max}) \\ 0, & \text{otherwise} \end{cases}, \tag{6.2}$$

6.3.1.6 Game Choices for Experimentalist

The elements in the set $\mathbf{T} = \{T_1, T_2, T_3, ..., T_n\}$ could be all possible mechanical experiments 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

$$\dot{\boldsymbol{\epsilon}} = \begin{bmatrix} \dot{\epsilon}_{11} & \dot{\epsilon}_{12} & \dot{\epsilon}_{13} \\ & \dot{\epsilon}_{22} & \dot{\epsilon}_{23} \\ \text{sym} & & \dot{\epsilon}_{33} \end{bmatrix}, \; \dot{\boldsymbol{\sigma}} = \begin{bmatrix} \dot{\sigma}_{11} & \dot{\sigma}_{12} & \dot{\sigma}_{13} \\ & \dot{\sigma}_{22} & \dot{\sigma}_{23} \\ \text{sym} & & \dot{\sigma}_{33} \end{bmatrix}. \tag{6.3}$$

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 [13]. $d\boldsymbol{\sigma}$ 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 [13] for their formulations for different loading constraints in geomechanics tests (e.g., drained and undrained triaxial tests).

6.3.2 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 (Section 6.2) 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 [21, 248] which are detailed as follows.

6.3.2.1 Game Board for Modeler

A constitutive model in the generalized elasto-plasticity framework [168, 271] 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 Figure 6.2. 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

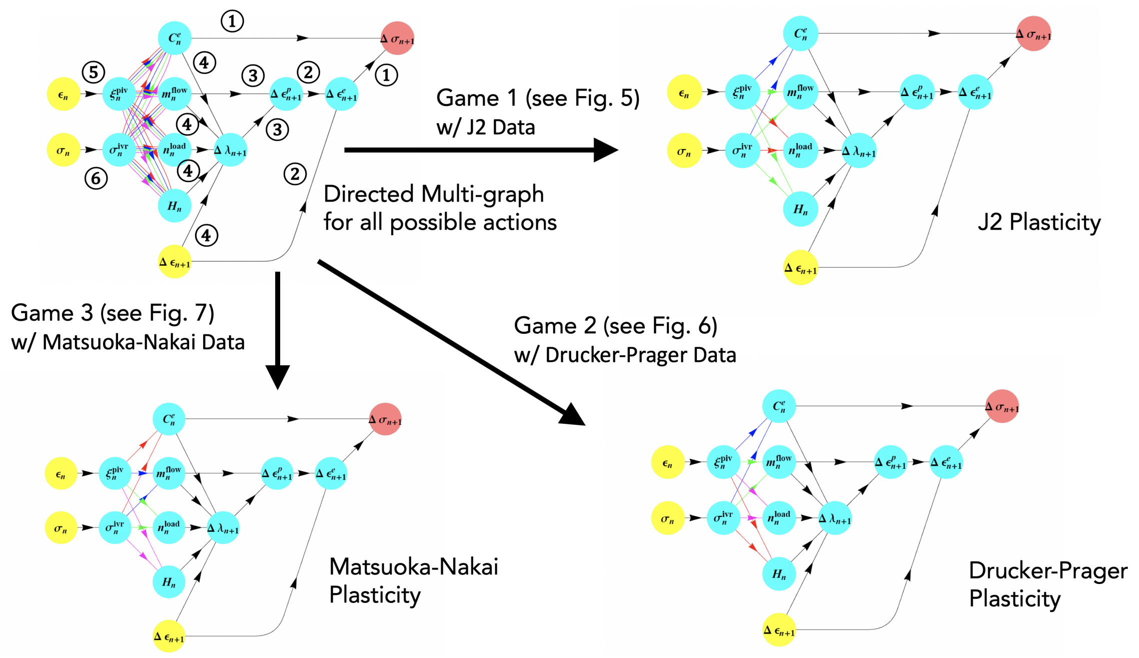


Figure 6.2: Directed multigraph of an elasto-plasticity model.The yellow nodes of the strain $\boldsymbol{\epsilon}_n$, stress $\boldsymbol{\sigma}_n$ and strain increment $\Delta\boldsymbol{\epsilon}_{n+1}$ refer to the root nodes, the pink node of the stress increment $\Delta\boldsymbol{\sigma}_{n+1}$ 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.

$$
\begin{aligned}
&\textcircled{1} \quad \Delta\boldsymbol{\sigma}_{n+1} = \boldsymbol{C}^e_n : \Delta\boldsymbol{\epsilon}^e_{n+1} \\
&\textcircled{2} \quad \Delta\boldsymbol{\epsilon}^e_{n+1} = \Delta\boldsymbol{\epsilon}_{n+1} - \Delta\boldsymbol{\epsilon}^p_{n+1} \\
&\textcircled{3} \quad \Delta\boldsymbol{\epsilon}^p_{n+1} = \Delta\lambda_{n+1}\boldsymbol{m}^{flow}_n \\
&\textcircled{4} \quad \Delta\lambda_{n+1} = \begin{cases} \dfrac{\boldsymbol{n}^{load}_n : \boldsymbol{C}^e_n : \Delta\boldsymbol{\epsilon}_{n+1}}{H_n + \boldsymbol{n}^{load}_n : \boldsymbol{C}^e_n : \boldsymbol{m}^{flow}_n} & \text{if plastic loading} \\ 0 & \text{if elastic loading} \end{cases},
\end{aligned}
\tag{6.4}
$$

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

$$\begin{cases} \Delta\boldsymbol{\sigma}^{e}_{n+1} : \boldsymbol{n}^{load}_{n} \neq 0 \rightarrow \text{plastic loading} \\ \Delta\boldsymbol{\sigma}^{e}_{n+1} : \boldsymbol{n}^{load}_{n} = 0 \rightarrow \text{elastic loading} \end{cases}, \tag{6.5}$$

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

$$\begin{cases} f(\boldsymbol{\sigma}_{n} + \Delta\boldsymbol{\sigma}^{e}_{n+1}, \boldsymbol{q}^{piv}_{n}(\boldsymbol{\xi}^{piv}_{n})) > 0 \rightarrow \text{plastic loading} \\ f(\boldsymbol{\sigma}_{n} + \Delta\boldsymbol{\sigma}^{e}_{n+1}, \boldsymbol{q}^{piv}_{n}(\boldsymbol{\xi}^{piv}_{n})) \leq 0 \rightarrow \text{elastic loading} \end{cases}, \tag{6.6}$$

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,

$$\textcircled{5} \begin{cases} \lambda_{n} = \displaystyle\int_{0}^{t_n} \dot{\lambda} dt \\ \bar{\epsilon}^{p}_{n} = \displaystyle\int_{0}^{t_n} ||\dot{\boldsymbol{\epsilon}^{p}}|| dt \\ \bar{\epsilon}^{p}_{v_n} = \displaystyle\int_{0}^{t_n} \operatorname{tr}(\dot{\boldsymbol{\epsilon}^{p}}) dt \\ \bar{\epsilon}^{p}_{s_n} = \displaystyle\int_{0}^{t_n} ||\dot{\boldsymbol{\epsilon}^{p}} - \frac{1}{3}\operatorname{tr}(\dot{\boldsymbol{\epsilon}^{p}})|| dt \\ e_{n} = e_{0} + \displaystyle\int_{0}^{t_n} \dot{e} dt = e_{0} + \int_{0}^{t_n} (1+e)\dot{\epsilon}_{v} dt \end{cases}, \tag{6.7}$$

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 [24]. As a result, the phenomenological relations can be represented as functions of a stress invariants $\boldsymbol{\sigma}_n^{ivr}$, which may include

$$\textcircled{6}\begin{cases} p_n = \dfrac{\operatorname{tr}(\boldsymbol{\sigma}_n)}{3} \\ q_n = \sqrt{3J_2} = \sqrt{\dfrac{3}{2}}||\boldsymbol{s}_n|| \\ \theta_n = \dfrac{1}{3}\sin^{-1}(-\dfrac{3\sqrt{3}}{2}\dfrac{J_3}{J_2^{3/2}}), \ -\dfrac{\pi}{6} \leq \theta \leq \dfrac{\pi}{6} \end{cases} \tag{6.8}$$

where $J_2 = \frac{1}{2}\text{trace}(\boldsymbol{s}_n^2)$, $J_3 = \frac{1}{3}\text{trace}(\boldsymbol{s}_n^3)$, $\boldsymbol{s}_n = \boldsymbol{\sigma}_n - p_n\boldsymbol{I}$ 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 [136]. 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,

$$\boldsymbol{n}^{load} = \frac{\partial f}{\partial \boldsymbol{\sigma}}||\frac{\partial f}{\partial \boldsymbol{\sigma}}||^{-1}, \tag{6.9}$$

or, in the case yield surface is absence, directly inferred from observations as those in the generalized plasticity framework (cf. [134, 168, 128]),

$$\boldsymbol{n}^{load} = n_v^{load}\boldsymbol{n}_v + n_s^{load}\boldsymbol{n}_s. \tag{6.10}$$

where

$$\begin{cases} \boldsymbol{n}_v = \dfrac{\partial p}{\partial \boldsymbol{\sigma}} = \dfrac{1}{3}\boldsymbol{I} \\ \boldsymbol{n}_s = \dfrac{\partial q}{\partial \boldsymbol{\sigma}} = \dfrac{\sqrt{3}}{2\sqrt{J_2}}\boldsymbol{S}. \end{cases} \tag{6.11}$$

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,

$$\boldsymbol{m}^{flow} = \frac{\partial g}{\partial \boldsymbol{\sigma}}||\frac{\partial g}{\partial \boldsymbol{\sigma}}||^{-1}. \tag{6.12}$$

or directly inferred from observations as those in the generalized plasticity framework (cf. [134, 168, 128])

$$\boldsymbol{m}^{flow} = m_v^{flow}\boldsymbol{n}_v + m_s^{flow}\boldsymbol{n}_s. \tag{6.13}$$

6.3.2.2 Game States for Modeler

The mathematical description of the current state of the game board is a list of binary indicators $s = [i_{e1}, i_{e2}, i_{e3}, ..., i_{en}]$ 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 generated 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 $i_{ei} = 0$, $\forall E_{ei} \in \mathbb{L}_{\mathbb{E}}$.

6.3.2.3 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.

6.3.2.4 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 $LegalActions(s) = [ii_1, ii_2, ii_3, ..., ii_n]$ 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 [198, 24], i.e.

$$\mathcal{D} = \sigma : \dot{\epsilon} - \frac{d\psi}{dt} \geq 0, \tag{6.14}$$

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,

$$\psi(\boldsymbol{\epsilon}^e, \boldsymbol{\xi}) = \psi^e(\boldsymbol{\epsilon}^e) + \psi^p(\boldsymbol{\xi}). \tag{6.15}$$

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. (6.14) could also be rewritten as [24],

$$\mathcal{D} = (\boldsymbol{\sigma} - \frac{\partial \psi^r}{\partial \boldsymbol{\epsilon}^e}) : \dot{\boldsymbol{\epsilon}} + \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}} - \frac{\partial \psi^p}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} \geq 0. \tag{6.16}$$

Readers interested to obtain further information on the constraints due to the thermodynamic laws are referred to [231, 24, 198]. In the coorperative 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 Figure 7.6. As the deep reinforcement learning progresses, the modeler agent will gradually learn to avoid generating model that violates 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 [121]. In this work, the frame indifference is enforced by representing tensors in spectral forms, then using Lie-algbera to establishing the mapping from one orthogonal basis to the other. The detailed operations have been documented in [247] and [248] and will not repeat in this chapter for brevity.

6.3.2.5 Game Choices for Modeler

This section specifies the candidate edges in the directed multi-graph of elasto-plasticity models (Fig. 6.2) 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 $\boldsymbol{\sigma}_n^{ivr} \rightarrow \boldsymbol{C}_n^e$ and $\boldsymbol{\xi}_n^{piv} \rightarrow \boldsymbol{C}_n^e$ represent the elasticity law. The edges $\boldsymbol{\sigma}_n^{ivr} \rightarrow \boldsymbol{n}_n^{load}$ and $\boldsymbol{\xi}_n^{piv} \rightarrow \boldsymbol{n}_n^{load}$ represent the definition of the loading direction. The edges $\boldsymbol{\sigma}_n^{ivr} \rightarrow \boldsymbol{m}_n^{flow}$ and $\boldsymbol{\xi}_n^{piv} \rightarrow \boldsymbol{m}_n^{flow}$ represent the definition of the plastic flow direction. The edges $\boldsymbol{\sigma}_n^{ivr} \rightarrow H_n$ and $\boldsymbol{\xi}_n^{piv} \rightarrow H_n$ represent the hardening law. Each edge allows multiple choices extracted from the phenomenological relations developed in the computational plasticity literature. In this chapter, 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 ($\boldsymbol{\sigma}_n^{ivr} \rightarrow \boldsymbol{C}_n^e$ and $\boldsymbol{\xi}_n^{piv} \rightarrow \boldsymbol{C}_n^e$) represent the definition and evolution of the elastic stiffness tensor

$$\boldsymbol{C}_n^e = K\boldsymbol{I} \otimes \boldsymbol{I} + 2G(\boldsymbol{I}_{sym}^4 - \frac{\boldsymbol{I} \otimes \boldsymbol{I}}{3}), \tag{6.17}$$

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

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

(E1) Linear elasticity

$$\begin{cases} K = K_0 \\ G = G_0 \end{cases}, \tag{6.18}$$

where K_0 and G_0 are constants.

(E2) Nonlinear elasticity with dependence on the mean pressure p [138]

$$\begin{cases} K = K_0(\dfrac{p}{p_{at}})^a \\ G = G_0(\dfrac{p}{p_{at}})^a \end{cases}, \tag{6.19}$$

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

(E3) Nonlinear elasticity with dependence on the mean pressure p and the void ratio e [52]

$$\begin{cases} K = \dfrac{2(1+\nu)}{3(1-2\nu)} G \\ G = G_0 p_{at} \dfrac{(2.97-e)^2}{1+e} (\dfrac{p}{p_{at}})^{1/2} \end{cases}, \tag{6.20}$$

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

$$\sigma_y = \sigma_{y0} + H_0 \bar{\epsilon}^p, \tag{6.21}$$

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

$$\frac{\sigma_y}{\sigma_{y0}} = (\frac{\sigma_y}{\sigma_{y0}} + \frac{3G}{\sigma_{y0}}\bar{\epsilon}^p)^n, \tag{6.22}$$

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

$$\sigma_y = \sigma_{y0} + H_0\bar{\epsilon}^p + H_\infty(1 - \exp(-b\bar{\epsilon}^p)), \tag{6.23}$$

where $\sigma_{y0}, H_0, H_\infty, b$ are material parameters.

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

$$\alpha = a_0 + a_1\bar{\epsilon}^p \exp(a_2 p - a_3\bar{\epsilon}^p), \tag{6.24}$$

where a_0, a_1, a_2, a_3 are material parameters [234].

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

$$\alpha = a_0 + 2a_1\frac{\sqrt{k\bar{\epsilon}^p}}{k + \bar{\epsilon}^p}, \tag{6.25}$$

where a_0, a_1, k are material parameters [24].

(L6) Yield surface of three-invariant Matsuoka–Nakai model [28]

$$\begin{cases} f = (k_1 I_3)^{1/3} - (I_1 I_2)^{1/3} \\ k_1 = c_0 + \kappa_1(\frac{p_{at}}{I_1})^m \\ \kappa_1 = a_1\bar{\epsilon}^p \exp(a_2 I_1)\exp(-a_3\bar{\epsilon}^p) \end{cases}, \tag{6.26}$$

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

(L7) Yield surface of Nor-Sand [103, 4]

$$\begin{cases} f = \zeta q + \eta p \\ \zeta = \dfrac{(1+\rho) + (1-\rho)\cos 3(\theta + \pi/6)}{2\rho} \\ \eta = \begin{cases} M[1 + \log(p_i/p)] & \text{if } N = 0 \\ (M/N)[1 - (1-N)(p/p_i)^{N/(1-N)}] & \text{if } N > 0 \end{cases} \\ \dot{p}_i = -\sqrt{\dfrac{2}{3}} h(p_i - p_i^*)||\dot{\boldsymbol{e}^p}||,\ \dot{\boldsymbol{e}^p} = \dot{\boldsymbol{\epsilon}^p} - \dfrac{1}{3} tr(\dot{\boldsymbol{\epsilon}^p})\boldsymbol{I} \\ \dfrac{p_i^*}{p} = \begin{cases} \exp(\bar{\alpha}\psi_i/M) & \text{if } \bar{N} = N = 0 \\ (1 - \bar{\alpha}\psi_i N/M)^{(N-1)/N}] & \text{if } 0 \leq \bar{N} \leq N \neq 0 \end{cases} \\ \bar{\alpha} = -3.5\dfrac{1-\bar{N}}{1-N} \\ \psi_i = e - e_{c0} + \tilde{\lambda}(p_i/p_{at})^a \end{cases}, \tag{6.27}$$

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 [52]

$$\begin{cases} f = ||\boldsymbol{S} - p\boldsymbol{\alpha}|| - \sqrt{2/3}pm \\ \dot{\boldsymbol{\alpha}} = \dot{\lambda}(2/3)h(\boldsymbol{\alpha}_\theta^b - \boldsymbol{\alpha}) \\ \boldsymbol{\alpha}_\theta^b = \sqrt{2/3}[\dfrac{1}{\zeta}M\exp(-n^b\psi) - m]\boldsymbol{n} \\ \zeta = \dfrac{(1+\rho) + (1-\rho)\cos 3(\theta + \pi/6)}{2\rho} \\ \boldsymbol{n} = \dfrac{\frac{\boldsymbol{S}}{p} - \boldsymbol{\alpha}}{\sqrt{2/3}m} \\ \psi = e - e_{c0} + \tilde{\lambda}(p/p_{at})^a \end{cases}, \tag{6.28}$$

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

(L9) Loading direction defined as [168, 271]

$$\begin{cases} n_v^{load} = \dfrac{d_f}{\sqrt{1+d_f^2}} \\ n_s^{load} = \dfrac{1}{\sqrt{1+d_f^2}} \\ \quad d_f = (1+\alpha)(M_f + q/p) \end{cases}, \tag{6.29}$$

where α, M_f are material parameters.

(L10) Loading direction defined as [128]

$$\begin{cases} n_v^{load} = \dfrac{d_f}{\sqrt{1+d_f^2}} \\ n_s^{load} = \dfrac{1}{\sqrt{1+d_f^2}} \\ \quad d_f = (1+\alpha)(M_f \exp(m_f(1-e)) + q/p) \end{cases}, \tag{6.30}$$

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. (6.24) or (6.25), and β_0 is an additional material parameter.

(P3) Plastic potential surface of three-invariant Matsuoka–Nakai model [28]

$$\begin{cases} g = (k_2 I_3)^{1/3} - (I_1 I_2)^{1/3} \\ k_2 = c_0 + \kappa_2 (\dfrac{p_{at}}{I_1})^m \\ \kappa_2 = \alpha \kappa_1 \end{cases}, \tag{6.31}$$

where κ_1 can be defined through Eq. 6.26 and β_0 is an additional material parameter.

(P4) Plastic potential surface of Nor-Sand [103, 4]

$$\begin{cases} g = \bar{\zeta} q + \bar{\eta} p \\ \bar{\zeta} = \dfrac{(1+\bar{\rho}) + (1-\bar{\rho})\cos 3(\theta + \pi/6)}{2\bar{\rho}} \\ \bar{\eta} = \begin{cases} M[1 + \log(\bar{p}_i/p)] & \text{if } \bar{N} = 0 \\ (M/\bar{N})[1 - (1-\bar{N})(p/\bar{p}_i)^{\bar{N}/(1-\bar{N})}] & \text{if } \bar{N} > 0 \end{cases} \end{cases}, \tag{6.32}$$

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 [52]

$$\begin{cases} \boldsymbol{m}^{flow} = B\boldsymbol{n} - C(\boldsymbol{n}^2 - \dfrac{1}{3}\boldsymbol{I}) + \dfrac{1}{3}D\boldsymbol{I} \\ B = 1 + \dfrac{3}{2}\dfrac{1-c}{c\xi}\cos 3(\theta + \pi/6) \\ C = 3\sqrt{3/2}\dfrac{1-c}{c\xi} \\ D = A_d(\boldsymbol{\alpha}_\theta^d - \boldsymbol{\alpha}) : \boldsymbol{n} \\ \boldsymbol{\alpha}_\theta^d = \sqrt{2/3}[\dfrac{1}{\zeta}M\exp(n^d\psi) - m]\boldsymbol{n} \end{cases}, \tag{6.33}$$

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

(P6) Plastic flow direction defined as [168, 271]

$$\begin{cases} m_v^{flow} = \dfrac{d_g}{\sqrt{1+d_g^2}} \\ m_s^{flow} = \dfrac{1}{\sqrt{1+d_g^2}} \\ \quad d_g = (1+\alpha)(M_g + q/p) \end{cases}, \tag{6.34}$$

where α, M_g are material parameters.

(P7) Plastic flow direction defined as [128]

$$\begin{cases} m_v^{flow} = \dfrac{d_g}{\sqrt{1+d_g^2}} \\ m_s^{flow} = \dfrac{1}{\sqrt{1+d_g^2}} \\ \quad d_g = (1+\alpha)(M_g \exp m_g \psi + q/p) \\ \quad \psi = e - e_{c0} + \tilde{\lambda}(p/p_{at})^a \end{cases}, \tag{6.35}$$

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 $\boldsymbol{m}^{flow}$.

$$H = -\frac{\partial f / \partial \boldsymbol{\epsilon}^p : \boldsymbol{m}^{flow}}{||\partial f / \partial \boldsymbol{\sigma}||}. \tag{6.36}$$

(H2) Hardening modulus defined as [168, 271]

$$\begin{cases} H = H_0(-p)H_f(H_v + H_s) \\ H_f = (1 + \dfrac{q}{pM_f}\dfrac{\alpha_f}{1+\alpha_f})^4 \\ H_v = 1 + \dfrac{q}{pM_g} \\ H_s = \beta_0\beta_1\exp(-\beta_0\bar{\epsilon}_s^p) \end{cases}, \tag{6.37}$$

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

(H3) Hardening modulus defined as [128]

$$\begin{cases} H = H_0\sqrt{p/p_{at}}H_f(1 + \dfrac{q}{pM_b}) \\ M_b = M_g\exp(-m_b\psi) \\ H_0 = H_{L0}\exp(m_0(1-e)) \\ H_f = (1 + \dfrac{q}{pM_f}\dfrac{\alpha_f}{1+\alpha_f})^4 \end{cases}, \tag{6.38}$$

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. (6.4))

$$\Delta\boldsymbol{\epsilon}_{n+1}^p = \Delta\boldsymbol{\epsilon}_{n+1} - (\boldsymbol{C}_n^e)^{-1} : \Delta\boldsymbol{\sigma}_{n+1}^{data}. \tag{6.39}$$

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

$$H_n = \frac{\boldsymbol{n}_n^{load} : \boldsymbol{C}_n^e : \Delta\boldsymbol{\epsilon}_{n+1}}{\Delta\lambda_{n+1}} - \boldsymbol{n}_n^{load} : \boldsymbol{C}_n^e : \boldsymbol{m}_n^{flow}. \tag{6.40}$$

6.3.2.6 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 [248]. 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.

6.3.2.7 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

$$\begin{aligned}
\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_{n+1}^{p}}, \\
&\quad \boldsymbol{\Delta\epsilon}_{n+1}^e, \boldsymbol{\Delta\sigma}_{n+1}\} && (6.41) \\
\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}} && (6.42) \\
\mathbb{E}_1 &= \{\boldsymbol{C}_n^e \rightarrow \boldsymbol{\Delta\sigma}_{n+1}, \boldsymbol{\Delta\epsilon}_{n+1}^e \rightarrow \boldsymbol{\Delta\sigma}_{n+1}\} && (6.43) \\
\mathbb{E}_2 &= \{\boldsymbol{\Delta\epsilon_{n+1}^{p}} \rightarrow \boldsymbol{\Delta\epsilon}_{n+1}^e, \boldsymbol{\Delta\epsilon}_{n+1} \rightarrow \boldsymbol{\Delta\epsilon}_{n+1}^e\} && (6.44) \\
\mathbb{E}_3 &= \{\boldsymbol{m}_n^{\text{flow}} \rightarrow \boldsymbol{\Delta\epsilon}_{n+1}^e, \Delta\lambda_{n+1} \rightarrow \boldsymbol{\Delta\epsilon}_{n+1}^e\} && (6.45) \\
\mathbb{E}_4 &= \{\boldsymbol{\Delta\epsilon}_{n+1} \rightarrow \Delta\lambda_{n+1}, \boldsymbol{C}_n^e \rightarrow \Delta\lambda_{n+1}, \boldsymbol{m}_n^{\text{flow}} \rightarrow \Delta\lambda_{n+1}, \boldsymbol{n}_n^{\text{load}} \rightarrow \Delta\lambda_{n+1}, \\
&\quad H_n \rightarrow \Delta\lambda_{n+1}\} && (6.46) \\
\mathbb{E}_5 &= \{\boldsymbol{\epsilon}_n \rightarrow \boldsymbol{\xi}_n^{piv}\} && (6.47) \\
\mathbb{E}_6 &= \{\boldsymbol{\sigma}_n \rightarrow \boldsymbol{\sigma}_n^{ivr}\} && (6.48) \\
\mathbb{E}_{\text{game}} &= \{\boldsymbol{\xi}_n^{piv} \rightarrow \boldsymbol{C}_n^e, \boldsymbol{\sigma}_n^{ivr} \rightarrow \boldsymbol{C}_n^e, \boldsymbol{\xi}_n^{piv} \rightarrow \boldsymbol{m}_n^{\text{flow}}, \boldsymbol{\sigma}_n^{ivr} \rightarrow \boldsymbol{m}_n^{\text{flow}}, && (6.49) \\
&\quad \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\}.
\end{aligned}$$

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. [247]). 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 Sections 6.3.2.5 and 6.3.2.6, i.e., $\mathbb{L}_{\mathbb{E}} = \{$ E1, E2, E3, L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, P1, P2, P3, P4, P5, P6, P7, P8, H1, H2, H3, H4 $\}$.

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 necessary trained individually. For each child vertex in the directed graph, all the edges that connects the parent vertices to it must be trained together.

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 [200, 201] (2) explicit integration of (potentially non-smooth) hardening laws [234] (3) integration of generalized plasticity models [58, 145] (4) linearized integration for loading constraints [13]. 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.

6.4 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 (Figure 7.6). 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) [196], Chess and shogi (Alpha Zero) [195] and many other games. In [248], the key ingredients (Policy/Value network, confidence bound for Q-value, Monte Carlo Tree Search) of the DRL technique are detailed and 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 target nodes (derive a directed graph from a directed multigraph) 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 [220, 70, 219]. The AI for experimentalist and modeler agents are separate, 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 (Figure 7.6). 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 [219]). In this work, we consider only the learning of fully cooperative strategies, as shown in the game reward system designed in Sections 6.3.1 and 6.3.2. In the reinforcement learning algorithm to play the two-player meta-modeling game, 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. **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.

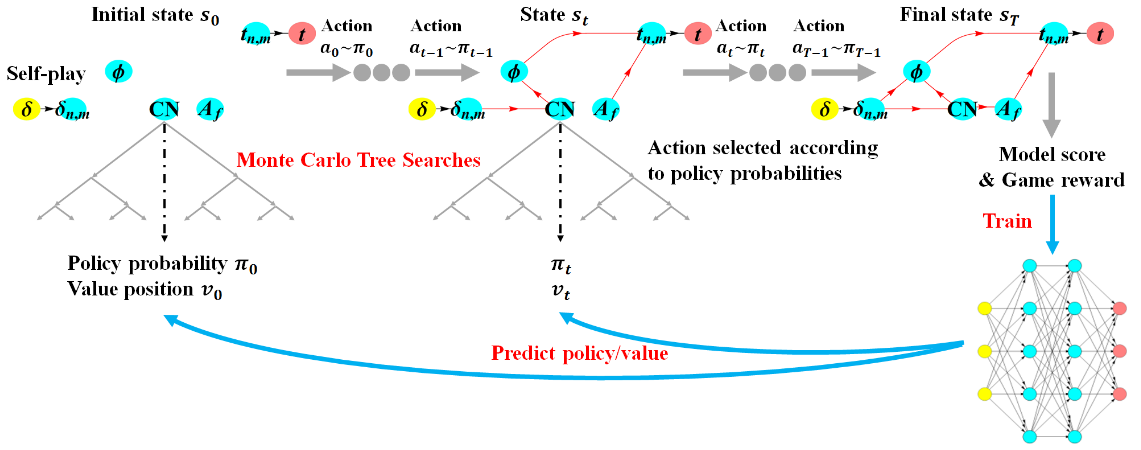


Figure 6.3: Multi-player interactive deep reinforcement learning for generating optimal strategy to automate the modeling, calibration and validation of an elasto-plastic model. In this framework, two deep neural networks are used to make decisions the mastermind model and data agents, while the model agents may also employ different strategies, including neural networks, mathematical expressions or other forms of mapping operators to compete a constitutive law.

6.5 Numerical Experiments

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.

6.5.1 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 [49, 115, 249]. 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 [247], 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 [202, 213]. 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 ([49]) 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: $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -300kPa, e_0 = 0.539$.

T2: $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -400kPa, e_0 = 0.536$.

T3: $\dot{\epsilon}_{11} < 0, b = 0, p_0 = -500kPa, e_0 = 0.534$.

T4: $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -300kPa, e_0 = 0.539$.

T5: $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -400kPa, e_0 = 0.536$.

T6: $\dot{\epsilon}_{11} > 0, b = 0, p_0 = -500kPa, e_0 = 0.534$.

T7: $\dot{\epsilon}_{11} < 0, b = 0.5, p_0 = -300kPa, e_0 = 0.539.$

T8: $\dot{\epsilon}_{11} < 0, b = 0.5, p_0 = -400kPa, e_0 = 0.536.$

T9: $\dot{\epsilon}_{11} < 0, b = 0.5, p_0 = -500kPa, e_0 = 0.534.$

T10: $\dot{\epsilon}_{11} > 0, b = 0.5, p_0 = -300kPa, e_0 = 0.539.$

T11: $\dot{\epsilon}_{11} > 0, b = 0.5, p_0 = -400kPa, e_0 = 0.536.$

T12: $\dot{\epsilon}_{11} > 0, b = 0.5, p_0 = -500kPa, e_0 = 0.534.$

T13: $\dot{\epsilon}_{11} < 0, b = 0.1, p_0 = -300kPa, e_0 = 0.539.$

T14: $\dot{\epsilon}_{11} < 0, b = 0.1, p_0 = -400kPa, e_0 = 0.536.$

T15: $\dot{\epsilon}_{11} < 0, b = 0.1, p_0 = -500kPa, e_0 = 0.534.$

T16: $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -300kPa, e_0 = 0.539.$

T17: $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -400kPa, e_0 = 0.536.$

T18: $\dot{\epsilon}_{11} > 0, b = 0.1, p_0 = -500kPa, e_0 = 0.534.$

T19: $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -300kPa, e_0 = 0.539.$

T20: $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -400kPa, e_0 = 0.536.$

T21: $\dot{\epsilon}_{11} < 0, b = 0.25, p_0 = -500kPa, e_0 = 0.534.$

T22: $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -300kPa, e_0 = 0.539.$

T23: $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -400kPa, e_0 = 0.536.$

T24: $\dot{\epsilon}_{11} > 0, b = 0.25, p_0 = -500kPa, e_0 = 0.534.$

T25: $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -300kPa, e_0 = 0.539.$

T26: $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -400kPa, e_0 = 0.536.$

T27: $\dot{\epsilon}_{11} < 0, b = 0.75, p_0 = -500kPa, e_0 = 0.534.$

T28: $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -300kPa, e_0 = 0.539.$

T29: $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -400kPa, e_0 = 0.536.$

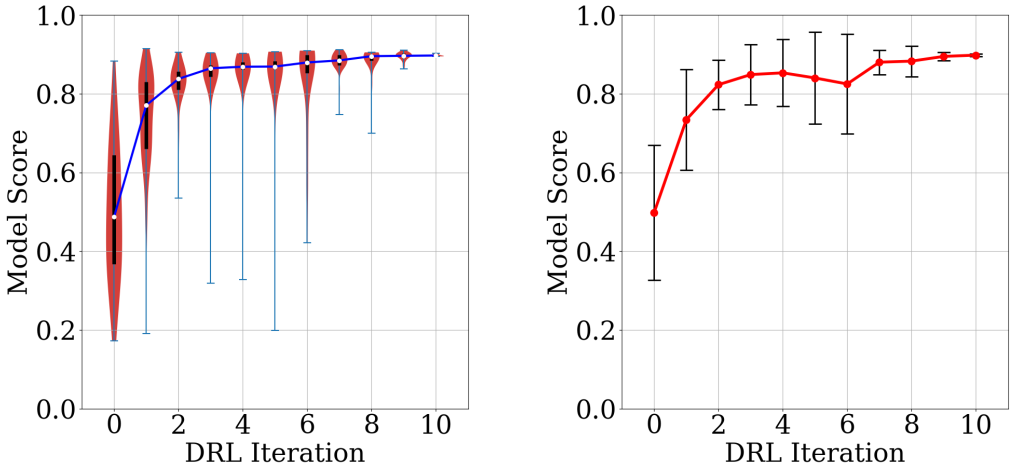
T30: $\dot{\epsilon}_{11} > 0, b = 0.75, p_0 = -500kPa, e_0 = 0.534.$

The candidate tests for the calibration data include $\mathbf{T}_c^0 = \{T1, T2, T3, ..., T11, T12\}$ and the validation tests are $\mathbf{T}_v^0 = \{T13, T14, T15, ..., T19, T30\}$. As explained in Section 6.3.1, 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 $numIters = 10$, $numEpisodes = 30$, $numMCTSSims = 300$.

6.5.2 Statistics of game scores via DRL iterations

The statistics of the gameplay results from 5 separate runs of the DRL procedure are presented in Figure 6.4. We observe efficient improvements in the generated elasto-plastic models over the DRL training iterations with the discrete element simulation data.

Figure 6.5 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. (6.19),



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

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

Figure 6.4: Statistics of the model scores in deep reinforcement learning iterations from 5 separate runs of the DRL procedure for Numerical Experiment 2. (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.

the loading direction defined as Eq. (6.30), the plastic flow direction defined as Eq. (6.35), and the hardening modulus defined as Eq. (6.37). 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 6.6 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].

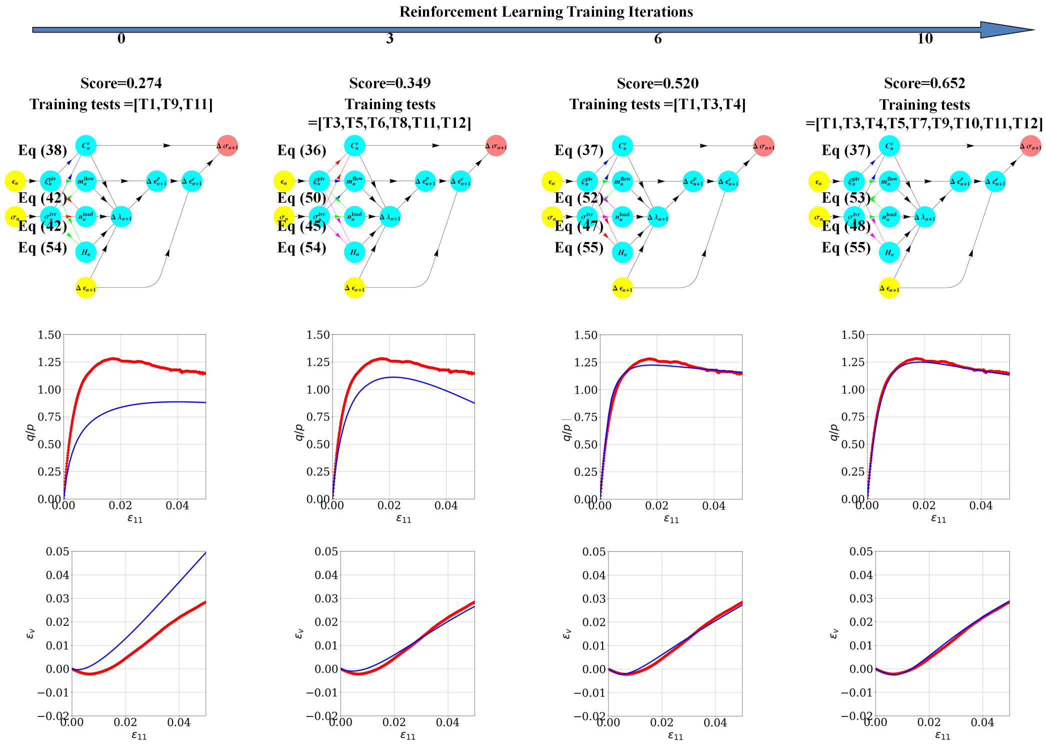


Figure 6.5: Knowledge of elasto-plastic models learned by deep reinforcement learning in Numerical Experiment 2 using data from drained triaxial tests. Four representative games played during the DRL iterations and their prediction accuracy against synthetic data are presented. The color edges illustrate different labeled edges selected in the constitutive models. The labels are represented by equation numbers and these equations are detailed in *Game Choices* in Section 6.3.2.

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. (6.39), (6.40) in Section 6.3.2) 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 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 a consistently good forward predictions [247, 248]. 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 [248]). 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 [205]. 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.

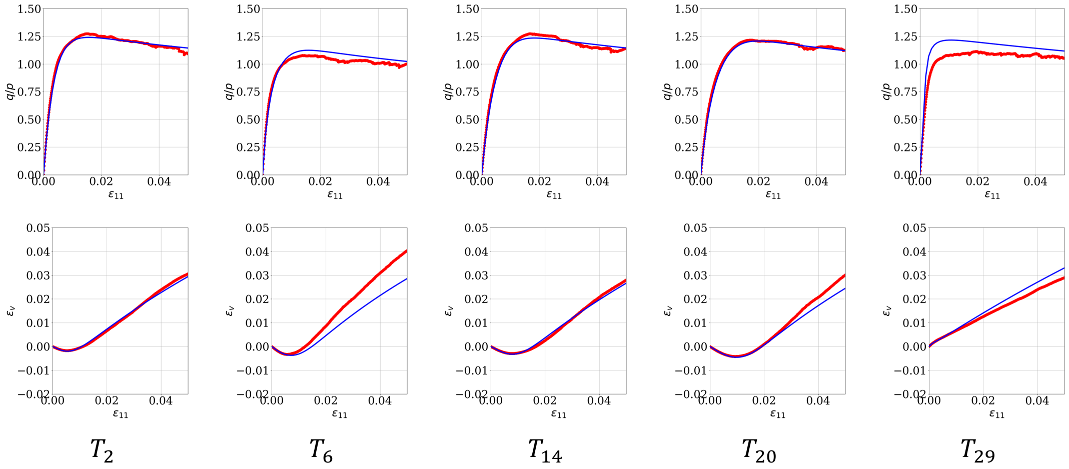


Figure 6.6: Five examples of **blind** predictions from the optimal digraph configuration (The 4th digraph in Figure 6.5) against data from the tests.

6.5.3 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 6.1 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

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				✓

Table 6.1: Five classes of the constitutive models generated during the deep reinforcement learning.

data in Table 6.1 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 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 Figure 6.7 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 [270, 168, 127, 190].

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. 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 [75, 138, 271, 173, 206]. 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 chapter.

6.5.4 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, 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 nu-

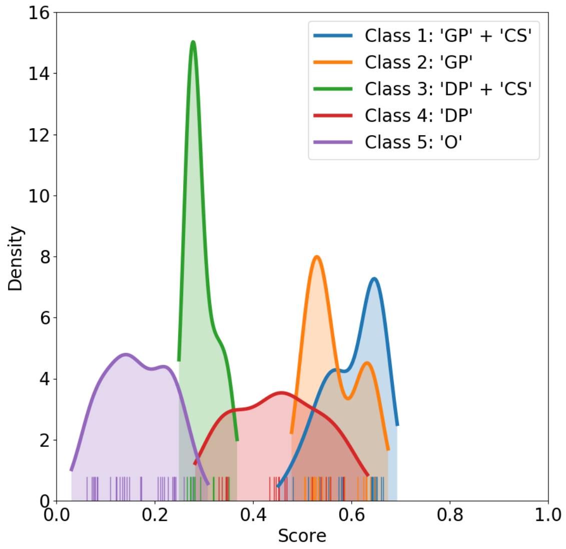


Figure 6.7: Distribution of the scores of the models generated during the deep reinforcement learning. The models are grouped into five families (see Table 6.1). The curves present the Gaussian kernel density estimation of the model score distributions (The estimated function $f_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x-x_i}{h})$ for score data $(x_1, x_2, ..., x_n)$, $K(x)$ 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).

merical 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 Figure 6.8. The specimen is initially consolidated to isotropic pressure of p_0 =-400kPa, 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 Figure 6.5).

The finite element implementation of the AI generated digraph-based model is simple and convenient. All modeling choices (Section 6.2) and the general purpose integration scheme (Algorithm 1) 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 Figure 6.9 and Figure 6.10, respectively. The global differential stress - axial strain and volumetric strain - axial strain curves are compared in Figure 6.11. The results demonstrate that all the local constitutive models, regardless of the quality, can all be implemented in the finite element solver. As mentioned previous, 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. [10, 206]), and the shear band mode converges in the 8th iteration.

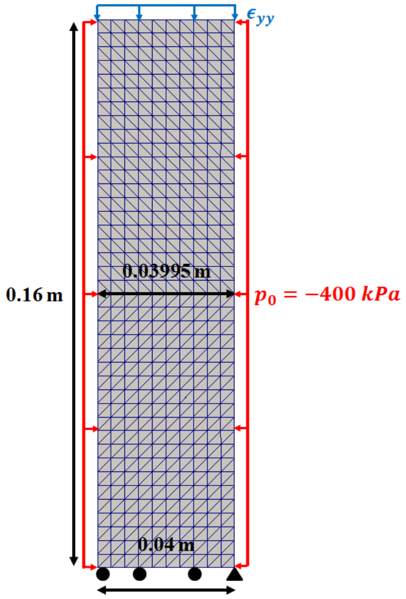
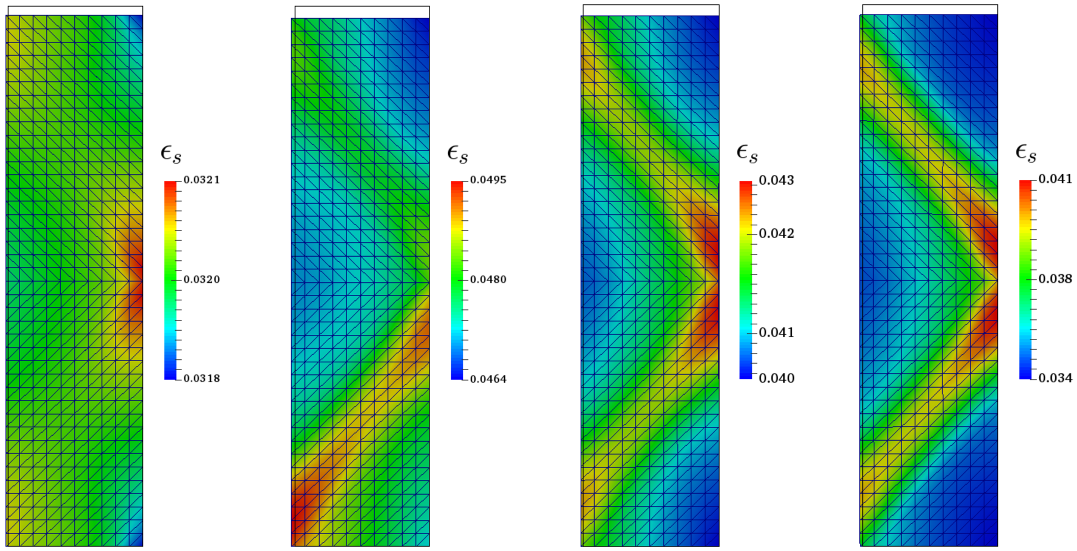


Figure 6.8: Description of the geometry, mesh, boundary, and loading conditions of the plane strain compression problem.

6.5.5 Efficiency compared to the brute force approach

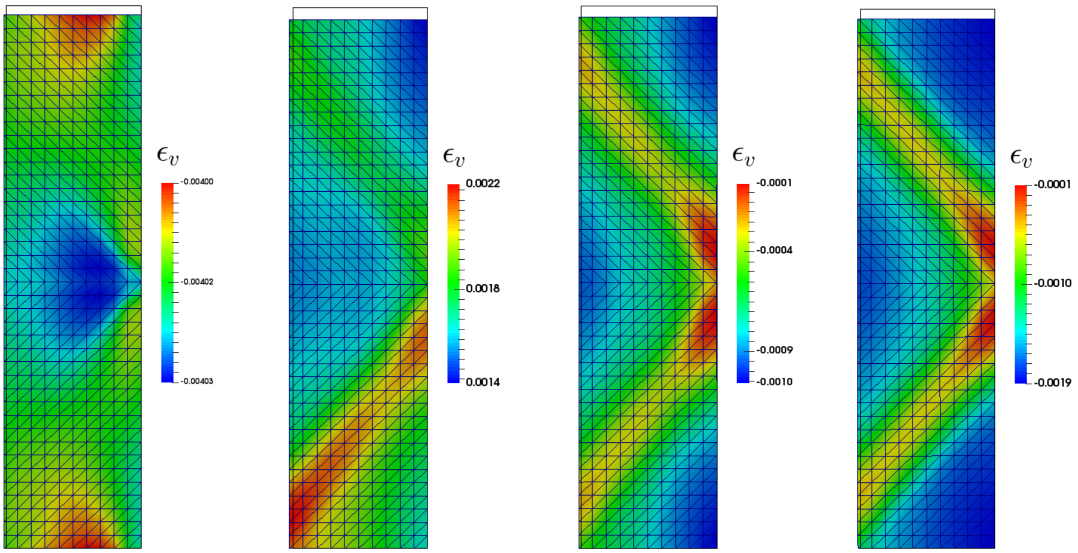
Consider the case in which reinforcement learning is not used. Instead, one simply generates all the possible directed graph from the labeled directed graph. In this case, the data agent must generate all possible sequences of calibration tests from the 12 candidates in the calibration test set (T1, 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 Section 6.3.2.5). 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



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

Figure 6.9: 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 Figure 6.5, Model 2 in the 5th iteration, Model 3 in the 8th iteration, Model 4 in the 10th iteration).

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 optimum 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.

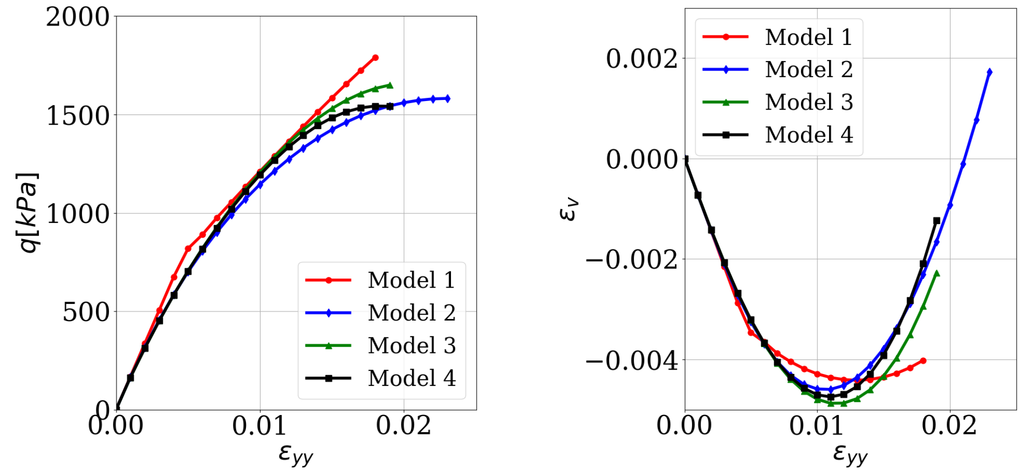


(a) Model 1 **(b)** Model 2 **(c)** Model 3 **(d)** Model 4

Figure 6.10: 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 Figure 6.5, Model 2 in the 5th iteration, Model 3 in the 8th iteration, Model 4 in the 10th iteration).

6.6 Conclusions

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



(a) Global diffrential stress q

(b) Global volumetric strain ϵ_v

Figure 6.11: Evolution of the global differential stress ($q = |\sigma_{yy} - p_0|$, $p_0 = -400kPa$, 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 Figure 6.5, 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.

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.

Chapter 7

Non-cooperative metamodeling game for automated validation and falsification of constitutive models

7.1 Introduction

As the constitutive models used to replicate material responses become increasingly sophisticated, the demand for accurately calibrating and validating those constitutive laws also increases. Engineering applications, particularly those involve high-risk and high-regret decision-making, require models that are not only accurate but also reliable such that the risk can be understood. Understanding the risk of a new constitutive law, however, is not an easy task. As many constitutive models are calibrated against limited amount and type of experimental data, identifying the reliable application range of these constitutive laws beyond the loading paths used in the calibration could be challenging. For example, the stress predictions on a loading path very different from the calibrating loading path could become questionable if over-fitting occurs. This chapter presents a novel designed non-cooperative metamodeling framework for experimentalist agents, aiming at helping third-party users to calibrate, validate, and falsify constitutive models in a very efficient manner. The idea is to first model the design procedures of experiments into decision trees. Then we introduce the recent emerging adversarial reinforcement learning techniques into the development of experimental strategies in collecting data for calibration and demonstration of the worst prediction scenarios of the models. Our non-cooperative game accelerates the convergence of the experimentalist agents' strategies to stable and reliable policies, similar to the impressive performance of the adversarial learning in image recognition, robotics and neural networks [83, 174, 263].

7.2 Recasting the design procedures of experiments as decision trees

The decision making of both AI experimentalists (protagonist and adversary) are modeled by "arborescences" in graph theory with labeled vertices and edges. An arborescence is a rooted polytree in which, for a single root vertex u and any other vertex v, there exists one unique directed path from u to v. A polytree (or directed tree) is a directed graph whose underlying graph is a singly connected acyclic graph. A brief review of the essential terminologies are given in [250], and their detailed definitions can be found in, for instance, [85, 253, 12]. Mathematically, the arborescence for decision making (referred to as "decision tree" hereafter) can be 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, $\boldsymbol{s} : \mathbb{E} \rightarrow \mathbb{V}$ and $\boldsymbol{t} : \mathbb{E} \rightarrow \mathbb{V}$ are the mappings that map each edge to its source vertex and its target vertex, $\boldsymbol{n_V} : \mathbb{V} \rightarrow \mathbb{L}_{\mathbb{V}}$ and $\boldsymbol{n_E} : \mathbb{E} \rightarrow \mathbb{L}_{\mathbb{E}}$ are the mappings that give the vertices and edges their corresponding labels (names) in $\mathbb{L}_{\mathbb{V}}$ and $\mathbb{L}_{\mathbb{E}}$.

The decision trees are constructed based on a hierarchical series of test conditions (type of test, pressure level, strain level, etc.) that an experimentalist needs to decide in order to design an experiment on a material. Assuming that an experiment can be completely and uniquely defined by an ordered list of selected test conditions $tc = [tc_1, tc_2, tc_3, ..., tc_n]$, where $N_{\mathbb{TC}}$ is the total number of test conditions. Each tc_i is selected from a finite set of choices $\mathbb{TC}_i = \{tc_i^1, tc_i^2, tc_i^3, ..., tc_i^{m_i}\}$, where m_i is the number of choices for the ith test condition. For conditions with continuous design variables, $\mathbb{TC}_i$ includes preset discrete values. For example, the target strain for a loading can only be chosen from discrete values of 1%, 2%, 3%, etc. Finally, an ordered list of sets $\mathbb{TC} = [\mathbb{TC}_1, \mathbb{TC}_2, \mathbb{TC}_3, ..., \mathbb{TC}_n]$ includes all design choices with a hierarchical relationship such that, if $i < j$, $tc_i \in \mathbb{TC}_i$ must be selected before $tc_j \in \mathbb{TC}_j$ is selected.

Example for hierarchical test conditions. Consider a simple design of mechanical experiments for geomaterials, for which all choices are listed in

$$\mathbb{TC} = [\text{'Sample', 'Type', 'Target'}]. \tag{7.1}$$

The first decision is to pick the initial geomaterial sample to test. Assuming that a sample is fully characterized by its initial pressure p_0, a simple set of discrete sample choices is given as

$$\mathbb{TC}_1 = \text{'Sample'} = \{\text{'300kPa', '400kPa'}\}. \tag{7.2}$$

The second test condition is the type of the experiment. The experiment can be either drained triaxial compression test ('DTC') or drained triaxial extension test ('DTE'). Then

$$\mathbb{TC}_2 = \text{'Type'} = \{\text{'DTC', 'DTE'}\}. \tag{7.3}$$

The third test condition to decide is the target strain magnitude for the loading. For example,

$$\mathbb{TC}_3 = \text{'Target'} = \{\text{'1\%'}, \text{'3\%'}\}. \tag{7.4}$$

After all three decisions are sequentially made, the experiment is completely determined by an ordered list, e.g., $tc = [\text{'300kPa'}, \text{'DTE'}, \text{'3\%'}]$. It indicates that the AI experimentalist will perform a monotonic drained triaxial extension test on a sample with $p_0 = 300kPa$ until the axial strain reaches 3%. □

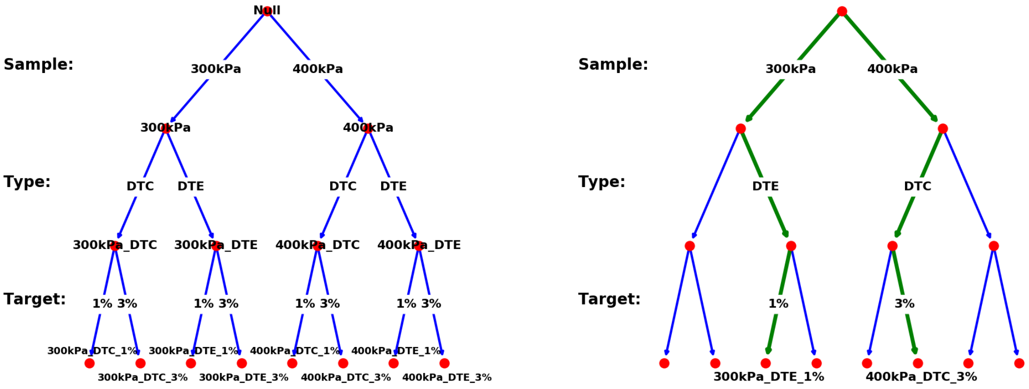
After the construction of $\mathbb{TC}$ for experimentalists, a decision tree is built top-down from a root node representing the 'Null' state that no test condition is decided. The root node is split into m_1 subnodes according to the first level of decisions $\mathbb{TC}_1$. Each subnode is further split into m_2 subnodes according to the second level of decisions $\mathbb{TC}_2$. The splitting process on the subnodes is carried out recursively for all the $N_{\mathbb{TC}}$ levels of decisions in $\mathbb{TC}$. Finally, the down-most leaf nodes represent all possible combinations of test conditions. The maximum number of possible configurations of experiments is $N_{test}^{max} = \prod_{i=1}^{N_{\mathbb{TC}}} m_i$, when all decisions across $\mathbb{TC}_i$ are independent. The number of possible experiments is reduced ($N_{test} < N_{test}^{max}$) when restrictions are specified for the selections of test conditions. E.g., the selection of $tc_i \in \mathbb{TC}_i$ will prohibit the selections of certain choices tc_j in subsequent test conditions $\mathbb{TC}_j, j > i$. The experimentalists can choose multiple experiments by taking multiple paths in the decision tree from the root node to the leaf nodes. The total number of possible combination of paths, if the maximum allowed number of simultaneously chosen paths is N_{path}^{max}, is $\sum_{k=1}^{N_{path}^{max}} C_{N_{test}}^{k}$, where $C_{N_{test}}^{k} = \frac{N_{test}!}{k!(N_{test}-k)!}$ is the combination number.

Example for the experimental decision tree. Consider the simple design of geomaterial experiments specified by Equations (7.1), (7.2), (7.3), (7.4). The decision tree $\mathbb{G}$ resulting from the hierarchical splitting process is shown in Fig. 7.1(a). The vertex sets and edge sets of the graph are

$$
\begin{aligned}
\mathbb{V} =& \{\text{'Null', '300kPa', '400kPa', '300kPa\_DTC', '300kPa\_DTE', '400kPa\_DTC', '400kPa\_DTE',}\\
& \text{'300kPa\_DTC\_1\%', '300kPa\_DTC\_3\%', '300kPa\_DTE\_1\%', '300kPa\_DTE\_3\%',}\\
& \text{'400kPa\_DTC\_1\%', '400kPa\_DTC\_3\%', '400kPa\_DTE\_1\%', '400kPa\_DTE\_3\%'}\},\\
\mathbb{E} =& \{\text{'Null'} \rightarrow \text{'300kPa', 'Null'} \rightarrow \text{'400kPa', '300kPa'} \rightarrow \text{'300kPa\_DTC',}\\
& \text{'300kPa'} \rightarrow \text{'300kPa\_DTE', '400kPa'} \rightarrow \text{'400kPa\_DTC', '400kPa'} \rightarrow \text{'400kPa\_DTE',}\\
& \text{'300kPa\_DTC'} \rightarrow \text{'300kPa\_DTC\_1\%', '300kPa\_DTC'} \rightarrow \text{'300kPa\_DTC\_3\%',}\\
& \text{'300kPa\_DTE'} \rightarrow \text{'300kPa\_DTE\_1\%', '300kPa\_DTE'} \rightarrow \text{'300kPa\_DTE\_3\%',}\\
& \text{'400kPa\_DTC'} \rightarrow \text{'400kPa\_DTC\_1\%', '400kPa\_DTC'} \rightarrow \text{'400kPa\_DTC\_3\%',}\\
& \text{'400kPa\_DTE'} \rightarrow \text{'400kPa\_DTE\_1\%', '400kPa\_DTE'} \rightarrow \text{'400kPa\_DTE\_3\%'}\},\\
\mathbb{L}_{\mathbb{V}} =& \mathbb{V},\\
\mathbb{L}_{\mathbb{E}} =& \{\text{'300kPa', '400kPa', 'DTC', 'DTE', '1\%', '3\%'}\}.
\end{aligned}
\tag{7.5}
$$

In this example, $N_{test} = N_{test}^{max} = 2 * 2 * 2 = 8$. If an experimentalist limits himself to only collect data from one or two experiments, i.e., $N_{path}^{max} = 2$, the total number of possible combinations is $C_8^1 + C_8^2 = 36$. Fig. 7.1(b) presents two example paths with edge labels illustrating the hierarchical decisions on the test conditions in order to arrive at the final experimental designs '300kPa_DTE_1%' and '400kPa_DTC_3%'.

□



(a) Decision tree with labels of vertices and edges

(b) Example of paths in the decision tree, the selected tests are ’300kPa_DTE_1%’ and ’400kPa_DTC_3%’

Figure 7.1: Decision tree for a simple experimental design for geomaterials (Eq. (7.1), (7.2), (7.3), (7.4)).

7.3 Zero-sum game for model calibration/validation with adversarial attack

This section presents a game design for the data acquisition involving the common decision-making activities of experimentalists in testing the mechanical properties of geomaterials. The goal of this game is for the protagonist agent to find the optimal set of experimental data for training/calibration of a material model. Meanwhile, with the zero-sum reward system, the adversary agent aims to find the extreme set of experiments for which forward predictions from the material model are the worst.

7.3.1 Decision tree for AI-guided experimentation

Firstly, we define the decision tree considered in this paper for the AI-guided experimentation on geomaterials, following the theory presented in Section 7.2. The hierarchical series of test conditions includes six elements, $\mathbb{TC} = [\mathbb{TC}_1, \mathbb{TC}_2, \mathbb{TC}_3, \mathbb{TC}_4, \mathbb{TC}_5, \mathbb{TC}_6]$, such that the experimentalists have the freedom to choose initially isotropic granular samples of different pressure p_0 and initial void ratio e_0, to perform different drained true triaxial tests, and to design different loading-unloading-reloading paths. The set of choices for each test conditions are shown in Table 7.1, with the labels representing their intended values.

$\mathbb{TC}$ Test Conditions	Choices
$\mathbb{TC}_1$ = 'Sample p_0'	{'300kPa', '400kPa', '500kPa'}
$\mathbb{TC}_2$ = 'Sample e_0'	{'0.60', '0.55'}
$\mathbb{TC}_3$ = 'Type'	{'DTC', 'DTE', 'TTC'}
$\mathbb{TC}_4$ = 'Load Target'	{'3%', '5%'}
$\mathbb{TC}_5$ = 'Unload Target'	{'NaN', '0%', '3%'}
$\mathbb{TC}_6$ = 'Reload Target'	{'NaN', '3%', '5%'}

Table 7.1: Choices of test conditions for AI-guided experimentation

The decision labels for the test type condition $\mathbb{TC}_3$ are defined as follows,

1. 'DTC': drained conventional triaxial compression test ($\dot{\epsilon}_{11} < 0$, $\dot{\sigma}_{22} = \dot{\sigma}_{33} = \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$),

2. 'DTE': drained conventional triaxial extension test ($\dot{\epsilon}_{11} > 0$, $\dot{\sigma}_{22} = \dot{\sigma}_{33} = \dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$),

3. 'TTC': drained true triaxial test with $b = 0.5$ ($\dot{\epsilon}_{11} > 0$, $b = \frac{\sigma_{22}-\sigma_{33}}{\sigma_{11}-\sigma_{33}} = const$, $\dot{\sigma}_{33} =$

$\dot{\sigma}_{12} = \dot{\sigma}_{23} = \dot{\sigma}_{13} = 0$),

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

$$\dot{\boldsymbol{\epsilon}} = \begin{bmatrix} \dot{\epsilon}_{11} & \dot{\epsilon}_{12} & \dot{\epsilon}_{13} \\ & \dot{\epsilon}_{22} & \dot{\epsilon}_{23} \\ \text{sym} & & \dot{\epsilon}_{33} \end{bmatrix}, \dot{\boldsymbol{\sigma}} = \begin{bmatrix} \dot{\sigma}_{11} & \dot{\sigma}_{12} & \dot{\sigma}_{13} \\ & \dot{\sigma}_{22} & \dot{\sigma}_{23} \\ \text{sym} & & \dot{\sigma}_{33} \end{bmatrix}. \quad (7.6)$$

Note that, since 'DTC' and 'DTE' are special cases of true triaxial tests, the choices {'DTC', 'DTE', 'TTC'} for $\mathbb{TC}_3$ is equivalent to choose the value of $b = \frac{\sigma_{22}-\sigma_{33}}{\sigma_{11}-\sigma_{33}}$ from {'0.0', '1.0', '0.5'}, respectively [185].

The decision labels 'NaN' in $\mathbb{TC}_5$ and $\mathbb{TC}_6$ are used to indicate that the unloading or reloading is not activated (stop the experiment). This design enables the freedom of generating monotonic loading paths (e.g., '5%_NaN_NaN'), loading-unloading paths (e.g., '5%_0%_NaN') and loading-unloading-reloading paths (e.g., '5%_0%_3%'). There are restrictions in choosing the strain targets. The experimentalist picks the loading target in $\mathbb{TC}_4$ first and the unloading target in $\mathbb{TC}_5$ must be, if not 'NaN', smaller than the loading strain. Then the reloading target in $\mathbb{TC}_6$ must be, if not 'NaN', larger than the unloading strain.

The decision tree following the splitting process in Section 7.2 is shown in Fig. 7.2. The subtree concerning the restricted decision-making in $\mathbb{TC}_4$, $\mathbb{TC}_5$ and $\mathbb{TC}_6$ is also detailed in the figure. The total number of experimental designs (which equals to the number of leaf nodes in the tree) is $N_{test} = 180$. The total number of experimental data combinations increases significantly when the maximum allowed simultaneous paths N_{path}^{max} increases. The combination number equals to $C_{180}^1 = 180$ when $N_{path}^{max} = 1$, equals to $C_{180}^1 + C_{180}^2 = 16290$ when $N_{path}^{max} = 2$, equals to $C_{180}^1 + C_{180}^2 + C_{180}^3 = 972150$ when $N_{path}^{max} = 3$, etc.

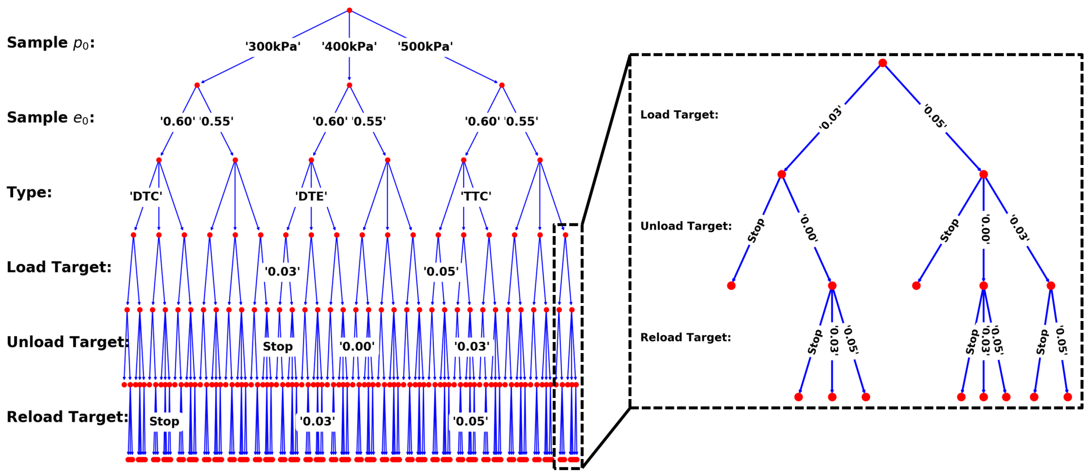


Figure 7.2: Decision tree for AI-guided experimentation. Due to the complexity of the graph, the vertex labels are omitted, and only a few edge labels are shown. See Fig. 7.1 for exhaustive vertex and edge labels in a simple decision tree example.

7.3.2 Design of the game for the experimentalist agents

The agent-environment interactive system (game) for the experimentalist agents consists of the game environment, game states, game actions, game rules, and game rewards [21, 248] (Fig. 7.3). These key ingredients are detailed as follows.

Game Environment consists of the geomaterial samples, the constitutive model for performance evaluation, and the decision tree in Fig. 7.2. The samples in this game are representative volume elements (RVEs) of virtual granular assemblies modeled by the discrete element method (DEM). The preparation of such DEM RVEs are detailed in the numerical examples. The constitutive model can be given by the modeler agent in a meta-modeling game [248, 250]. In this paper, we focus on the interactive learning of data acquisition

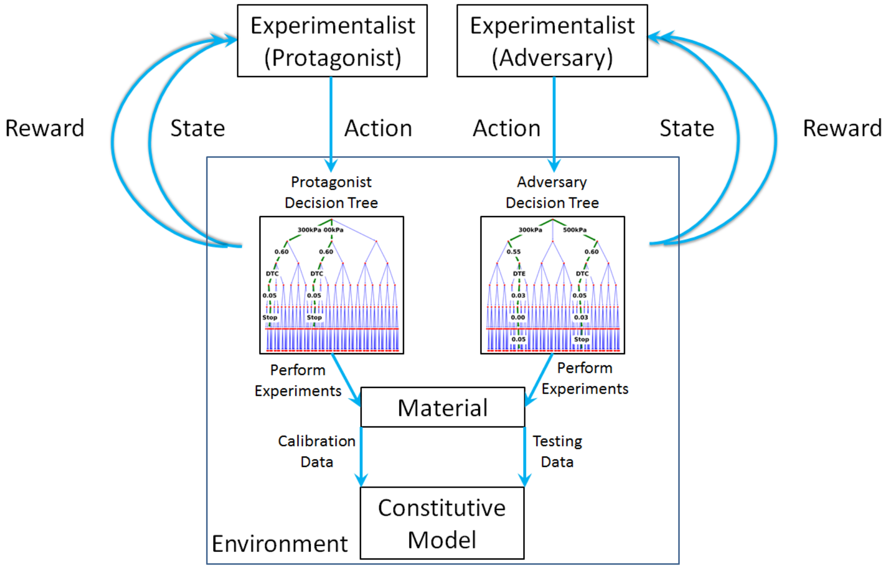


Figure 7.3: Key ingredients (environment, agents, states, actions, rules, and rewards) of the two-player non-cooperative agent-environment interactive system (game) for the experimentalist agents.

strategies for a particular constitutive model of interest. Three-agent protagonist-modeler-adversary reinforcement learning games is out of the scope of the current study. Hence the modeler agent's policy is held constant during the games. The protagonist and adversary agents determine the experiments on the RVEs in order to collect data for model parameter identification and testing the forward prediction accuracy of the constitutive model, respectively, via taking paths in their own experimental decision trees.

Game State For the convenience of deep reinforcement learning using policy/value neural networks, we use a 2D array to concisely represent the current paths that the protagonist or adversary agent has selected in the experimental decision tree. The mapping from the set of the 2D arrays to the set of path combinations in the decision tree is injective. The array has a row size of N_{path}^{max} and a column size of $N_{\mathbb{TC}} + 1$. Each row represents one path in the decision tree from the root node to a leaf node, i.e., a complete design of one experiment. The number of allowed experiments is restricted by the row size N_{path}^{max}, which is defined by the user. Each array entry in the first $N_{\mathbb{TC}}$ columns represents the selected decision label of each test condition in $\mathbb{TC}$. The entry a (integer) in the jth row and ith column indicates that the ath decision label in the set $\mathbb{TC}_i$ is selected for the jth experiment. The last column indicates whether the agent decides to take a new path in the decision tree (perform an another experiment), with 1 indicating continuation and 2 indicating stop. Initially, all entries in the array are 0, indicating no decisions has been made by the agents. The 2D state array needs to be flattened to a 1D array of size $N_{state} = N_{path}^{max} * (N_{\mathbb{TC}} + 1)$ and then input into the policy/value neural networks for policy evaluations.

Game Action The AI agents work on the game state array by changing the initial zero entries into integers representing the decision labels, in the left-to-right then top-to-bottom order. Suppose that the first zero element of the current state array s is in the jth row and ith column. If $i \leq N_{\mathbb{TC}}$, the agent will select an integer $1 \leq a \leq m_i$ (number of choices) to choose a decision label in $\mathbb{TC}_i$. If $i = N_{\mathbb{TC}} + 1$, the agent will select 1 for continuation or select 2 for stop. The size of the action space is $N_{action} = \max_{i \in [1, N_{\mathbb{TC}}]} m_i$. For the test conditions specified in Table 7.1 and the decision tree in Fig. 7.2, the action size is 3.

Game Rule The AI agents are restricted to follow existing edges in the constructed decision tree, which has already incorporated decision limitations such as the choices of loading/unloading/reloading strain targets. The game rules are reflected by a list of N_{action} binaries $LegalActions(s) = [ii_1, ii_2, ..., ii_{N_{action}}]$ at the current state s. If the ath decision is allowed, the ath entry is 1. Otherwise, the entry is 0. Figure 7.4 provides an example of the mathematical representations of the game states, actions and rules of the decision tree game.

Game Reward The reward from the game environment to the experimentalist agents

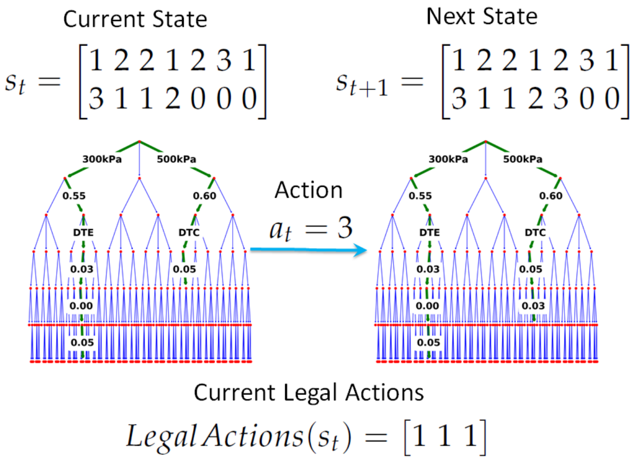

Current State
$s_t = \begin{bmatrix} 1\ 2\ 2\ 1\ 2\ 3\ 1 \\ 3\ 1\ 1\ 2\ 0\ 0\ 0 \end{bmatrix}$
Next State
$s_{t+1} = \begin{bmatrix} 1\ 2\ 2\ 1\ 2\ 3\ 1 \\ 3\ 1\ 1\ 2\ 3\ 0\ 0 \end{bmatrix}$
300kPa
500kPa
0.55
0.60
DTE
DTC
0.03
0.05
0.00
0.05
Action
$a_t = 3$
300kPa
500kPa
0.55
0.60
DTE
DTC
0.03
0.05
0.00
0.03
0.05
Current Legal Actions
$Legal\,Actions(s_t) = \begin{bmatrix} 1\ 1\ 1 \end{bmatrix}$

Figure 7.4: Example of the current s_t and next s_{t+1} game states, action by the agent a_t to ”advance” in the decision tree, and the legal actions at the current state, with $N_{path}^{max} = 2$.

should consider the performance of a given constitutive model on calibration data and testing data. After the decision of experiments by the protagonist, these experiments are performed on a given material to collect data. Then the constitutive model is calibrated with these data, and the accuracy is evaluated by a model score $\text{SCORE}_{\text{protagonist}}$. After the decision of experiments by the adversary, the calibrated constitutive model gives forward predictions on these testing data. The accuracy is evaluated by a model score $\text{SCORE}_{\text{adversary}}$. The total game score could be the average of both calibration and prediction accuracies $\text{SCORE} = 0.5 * \text{SCORE}_{\text{protagonist}} + 0.5 * \text{SCORE}_{\text{adversary}}$. +SCORE is returned to the protagonist as its game reward, while -SCORE is returned to the adversary. This zero-sum reward system is the key design to ensure that the protagonist generates the optimal calibration data, while the adversary tries to find out the testing data on which the predictions from the constitutive model are the worst.

7.3.3 Evaluation of model scores and game rewards

The accuracies of model calibrations and forward predictions are quantified by calculating the discrepancy between the vector of data points $[\overline{Y}_i^{\text{data}}]_{i=1}^{N_{data}}$ and the vector of predicted values $[\overline{Y}_i^{\text{model}}]_{i=1}^{N_{data}}$ under the same experimental conditions. For both data points and predictions, $\overline{Y}_i = \mathcal{S}_j(Y_i^j)$, where Y_i^j is the data that falls into the jth category of output features (quantities of interest, such as deviatoric stress q and void ratio e). $\mathcal{S}_j$ is the scaling operator (standardization, min-max scaling, ...) for the jth output feature.

The model scores are based on the modified Nash-Sutcliffe efficiency index [112]

$$E_{NS}^j = 1 - \frac{\sum_{i=1}^{N_{data}} |\overline{Y}_i^{\text{data}} - \overline{Y}_i^{\text{model}}|^j}{\sum_{i=1}^{N_{data}} |\overline{Y}_i^{\text{data}} - \text{mean}(\overline{Y}^{\text{data}})|^j},\ E_{NS}^j \in (-\infty, 1.0]. \tag{7.7}$$

When $j = 2$, it recovers the conventional Nash-Sutcliffe efficiency index. Here we adopt $j = 1$, and

$$\text{SCORE}_{\text{protagonist or adversary}} = -1.0 + 2.0 * \max(E_{NS}^1, 0.0),\ \text{SCORE} \in [-1.0, 1.0]. \tag{7.8}$$

The predictions $[\overline{Y}_i^{\text{model}}]_{i=1}^{N_{data}}$ come from a given constitutive model that is calibrated with data generated by the protagonist experimentalist. In this work, both gradient-based deterministic and Bayesian calibration method are used to find the optimal parameter values from the calibration data. The calibration procedures are accomplished by the Dakota software for optimization and uncertainty quantification [2].

For deterministic calibration of model parameters, the nonlinear least-squares solver ”NL2SOL” in Dakota is used to compute the model parameters that minimize the scaled mean squared error objective function

$$\text{scaled MSE} = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} (\overline{Y}_i^{\text{model}} - \overline{Y}_i^{\text{data}})^2 . \tag{7.9}$$

The initial guess, upper and lower bounds of each parameter are given by domain experts' knowledge and preliminary estimations. Dakota can also calculate 95% confidence intervals around the optimal value of the estimated parameters (mean $\pm 1.96 *$ standard deviation). These are not joint confidence intervals, but only on a per-parameter basis.

The uncertainty quantification of the model parameters are done by Bayesian calibration method. It is based on the Bayes' theorem of conditional probability, stating that the posterior joint probability density function (pdf) of the model parameters is given by

$$f_{\mathbf{\Theta}|D}(\boldsymbol{\theta}|\boldsymbol{d}) = \frac{f_{\mathbf{\Theta}}(\boldsymbol{\theta})\mathcal{L}(\boldsymbol{\theta};\boldsymbol{d})}{f_D(\boldsymbol{d})} \tag{7.10}$$

where $\mathbf{\Theta}$ is the space of model parameters and $\boldsymbol{\theta}$ is the vector of unknown model parameters. D is the space of experimental data and $\boldsymbol{d}$ is the vector of data with N_{data} measurements. $f_{\mathbf{\Theta}}(\boldsymbol{\theta})$ is the prior belief of the pdf of the parameters before the incorporation of data. It can be given by expert judgment or preliminary parameter studies. For example, the confidence interval given by "NL2SOL" could propose a prior bounded uniform distribution with upper and lower bounds of mean $\pm 5 *$ standard deviation. $\mathcal{L}(\theta;\boldsymbol{d})$ is the likelihood function that describes the probability of witnessing the data $\boldsymbol{d}$ based on predictions with parameters $\boldsymbol{\theta}$ from the model,

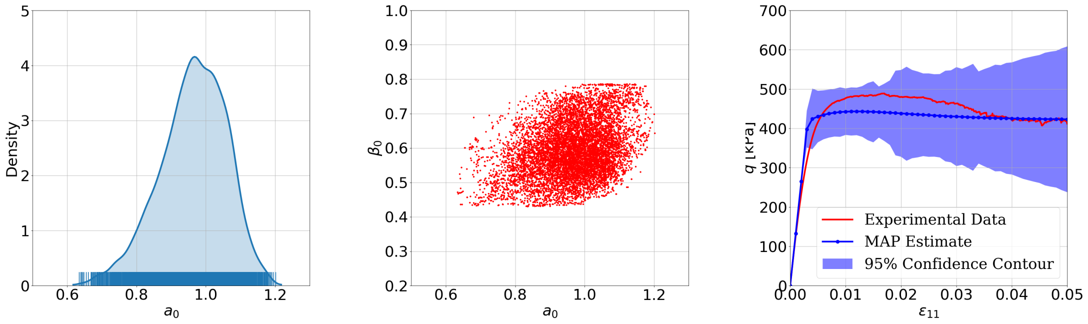
$$\mathcal{L}(\theta;\boldsymbol{d}) = \prod_{i=1}^{N_m} \frac{1}{\sigma_d\sqrt{2\pi}} \exp\Big[-\frac{(d_i - q_i(\boldsymbol{\theta}))^2}{2\sigma_d^2}\Big] \tag{7.11}$$

where $\boldsymbol{q}$ is the model predicted quantity of interest. σ_d^2 is the variance of the assumed zero-mean Gaussian distribution of the measurement error of the data. Finally, $f_D(\boldsymbol{d})$ is the total probability

$$f_D(\boldsymbol{d}) = \int_{\mathbf{\Theta}} f_{\mathbf{\Theta}}(\boldsymbol{\theta})\mathcal{L}(\boldsymbol{\theta};\boldsymbol{d})\boldsymbol{d\theta}, \tag{7.12}$$

which is a normalizing constant.

Using the ”DRAM” (delayed rejection and adaptive Metropolis) in the ”QUESO” toolbox of Dakota, the posterior joint pdfs are approximated numerically via Markov Chain Monte Carlo (MCMC) sampling in probabilistic spaces. MCMC generate samples by evolving the parameter vector $\boldsymbol{\theta}$ through a Markov process until the Markov chain reaches its stationary distribution. The starting point of the chain could be the optimal parameter vector given by the deterministic calibration. MCMC sampling could be very expensive for complicated constitutive models. In this case, simple surrogate models (emulators) are used to reduce the computation cost, e.g., polynomial chaos expansion, or stochastic collocation expansion. The optimal parameters after the Bayesian calibration procedure consists of the maximum a posteriori probability (MAP) point. Figure 7.5 presents an example of Bayesian calibration on the Drucker-Prager model with parameters specified by Eq. (7.13). The procedure yields joint posterior distributions of the parameters and confidence intervals of the predictions.



(a) Gaussian kernel density function of the distribution of a_0

(b) Accepted MCMC samples showing the joint posterior distribution of a_0 and β_0

(c) MAP model estimation with 95% confidence intervals (MAP predictions $\pm$ 1.96$*$standard deviation on error)

Figure 7.5: Example of Bayesian calibration on the Drucker-Prager model (cf. Eq.(7.13)).

Remarks on implementation of constitutive models. The elasto-plasticity models investigated in this work are implemented in the generalized elasto-plasticity framework [168, 271], which requires four essential components of ”phenomenological relations” : (1) elasticity law (2) loading direction (3) plastic flow direction (4) hardening modulus. They are numerically integrated using a general-purpose explicit integration algorithm, instead of using different implicit integration techniques necessary for different models. The algorithm is a combination of (1) the explicit integration with sub-stepping and automatic error control [200, 201] (2) explicit integration of (potentially non-smooth) hardening laws [234] (3) integration of generalized plasticity models [58, 145]. The loading constraints of geomechanics laboratory experiments are applied via a linearized integration technique $\boldsymbol{S}d\boldsymbol{\sigma} + \boldsymbol{E}d\boldsymbol{\epsilon} = d\boldsymbol{Y}$, combined with incremental constitutive equations, as proposed in [13]. $d\boldsymbol{\sigma}$ 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. The algorithm and implementations are detailed in [250]. □

7.4 Deep reinforcement learning algorithm for the zero-sum experimental game

With the two-player non-cooperative game completely defined in the previous section, a deep reinforcement learning (DRL) algorithm is employed as a guidance of taking actions of both protagonist experimentalist and adversary in the game to maximize the final game reward (Figure 7.6). 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) [196], Chess and shogi (Alpha Zero) [195] and many other games. In [248], the key ingredients (Policy/Value network, confidence bound for Q-value, Monte Carlo Tree Search) of the DRL technique are detailed and applied to a meta-modeling game for modeler agent only, focusing on finding the optimal topology of physical relations from fixed training/testing datasets. Since DRL needs to figure out the optimal strategies for two agents, the algorithm is extended to multi-agent multi-objective DRL [220, 70, 219]. The AI for protagonist and adversary agents are separate, each has its own Policy/Value network and decision tree search. But their intelligence are improved simultaneously during the self-plays of the entire Meta-modeling game, according to the individual rewards they receive from the game environment and the communications between themselves (Figure 7.6) [174]. 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.

7.5 Numerical Experiments

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 [202, 213]. 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 ([49]) is used for the

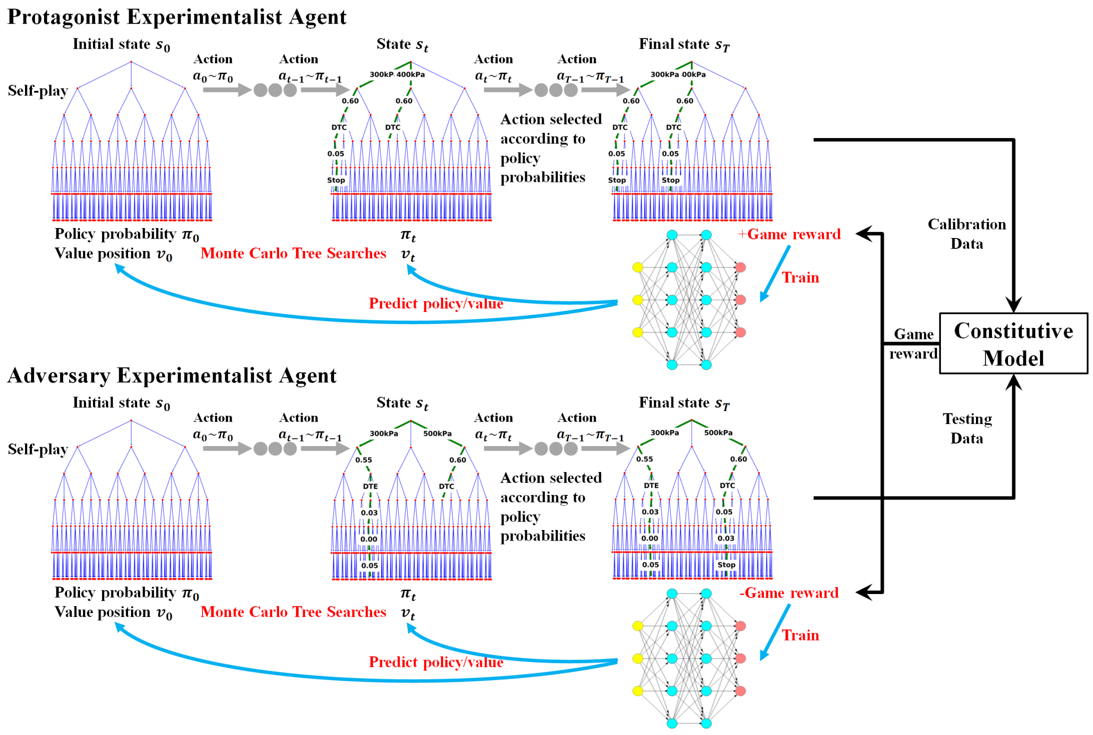


Figure 7.6: Two-player zero-sum adversarial reinforcement learning for generating optimal strategy to automate the calibration, validation and falsification of a constitutive model.

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 $\varphi = 30^\circ$, density $\rho = 2600\, kg/m^3$, Cundall damping coefficient $\alpha_{damp} = 0.4$. The generated RVE samples for the experimentalists to choose are presented in Table 7.2.

The constitutive model is a Drucker–Prager plasticity model [234]. It has a yield surface in the form $f = q + \alpha p$, and a potential surface in the form $g = q + \beta p - c_g$. α and β evolve according to

$$\begin{cases} \alpha = a_0 + a_1 \bar{\epsilon}^p \exp(a_2 p - a_3 \bar{\epsilon}^p) \\ \beta = \alpha - \beta_0 \end{cases}, \tag{7.13}$$

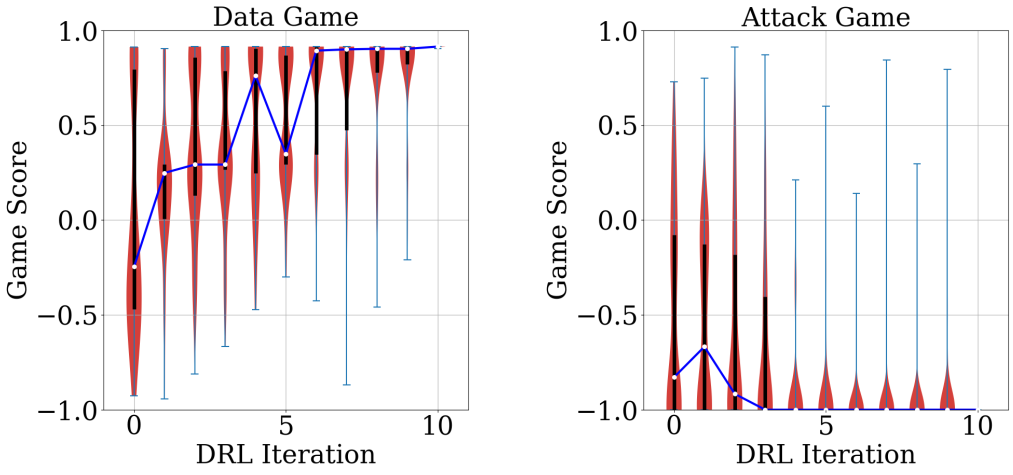
where $a_0, a_1, a_2, a_3, \beta_0$ are material parameters to calibrate. $\bar{\epsilon}^p$ is the accumulated plastic

Sample No.	′Sample p_0′	′Sample e_0′	p_0	e_0
1	′300kPa′	′0.60′	-300 kPa	0.5955
2	′300kPa′	′0.55′	-300 kPa	0.5554
3	′400kPa′	′0.60′	-400 kPa	0.5936
4	′400kPa′	′0.55′	-400 kPa	0.5538
5	′500kPa′	′0.60′	-500 kPa	0.5917
6	′500kPa′	′0.55′	-500 kPa	0.5521

Table 7.2: Initial DEM RVE Samples

strain.

The meta-modeling procedure with adversarial learning 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 $numEpisodes = 100$ self-play episodes of the game. Hence one execution of the entire DRL procedure contains $numIters * numEpisodes = 10 * 100 = 1000$ 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 require $numMCTSSims = 100$ 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. The success of the non-cooperative game is demonstrated by the statistical performance of the DRL in Figure 7.7. The distribution of games scores played in each training iteration clearly show an improvement of the protagonist agent in finding the optimal calibration data, while simultaneously the adversary agent converge to perform tests that the Drucker–Prager model behaves the worst.



(a) Protagonist

(b) Adversary

Figure 7.7: Violin plots of the density distributions of game scores in each DRL iteration in Numerical Experiment. 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.

Figure 7.8 and 7.9 illustrate the improvement of knowledge of which experiments to perform by the protagonist and the adversary during the training iterations. Initially, the AI agents just play with trial-and-error following strategies guided by random initial policy/value neural networks and MCTS. This lack of knowledge can be seen from the widely spread probability distribution of chosen experiments (leaf nodes in the decision trees). The evolution of pdfs in the subsequent iterations shows that the agents play with increasing knowledge of game strategies reinforced by their zero-sum game rewards. The pdfs of the protagonist tend to concentrate around monotonic drained triaxial compression tests. The examples of improved calibration accuracy of the Drucker–Prager model during the learning procedure, with calibration data provided by the protagonist, are shown in Figure

7.10. The pdfs of the adversary, however, tend to concentrate around cyclic drained triaxial extension tests. The examples of blind predictions from the Drucker–Prager model, with testing data provided by the adversary, are shown in Figure 7.11.

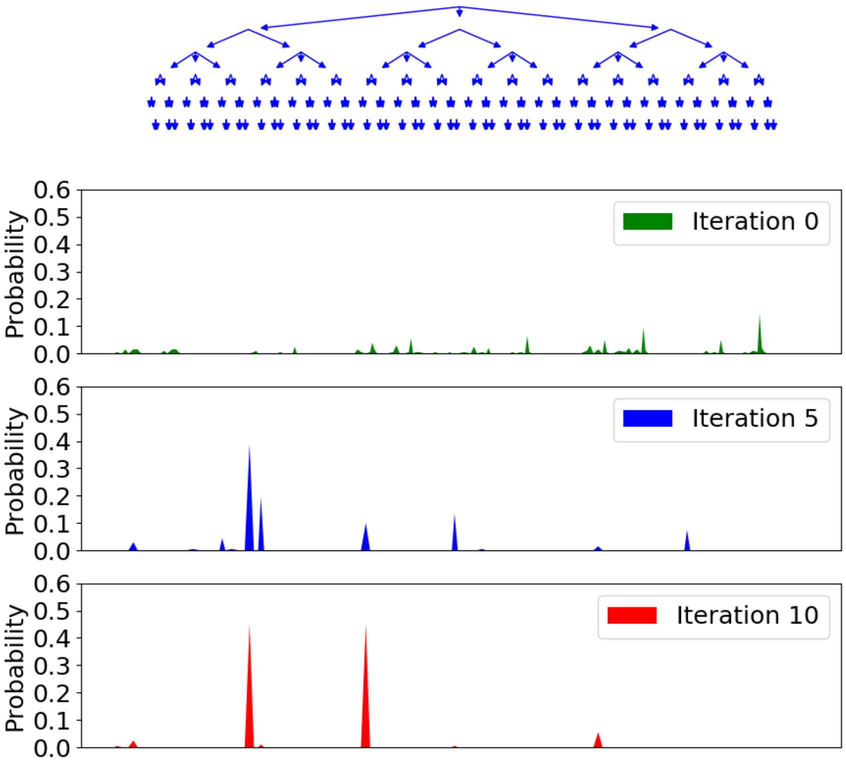


Figure 7.8: Distributions of the experiments determined by the protagonist during the DRL training iterations. The X axis corresponds to the leaf nodes of the decision tree for experimentalists.

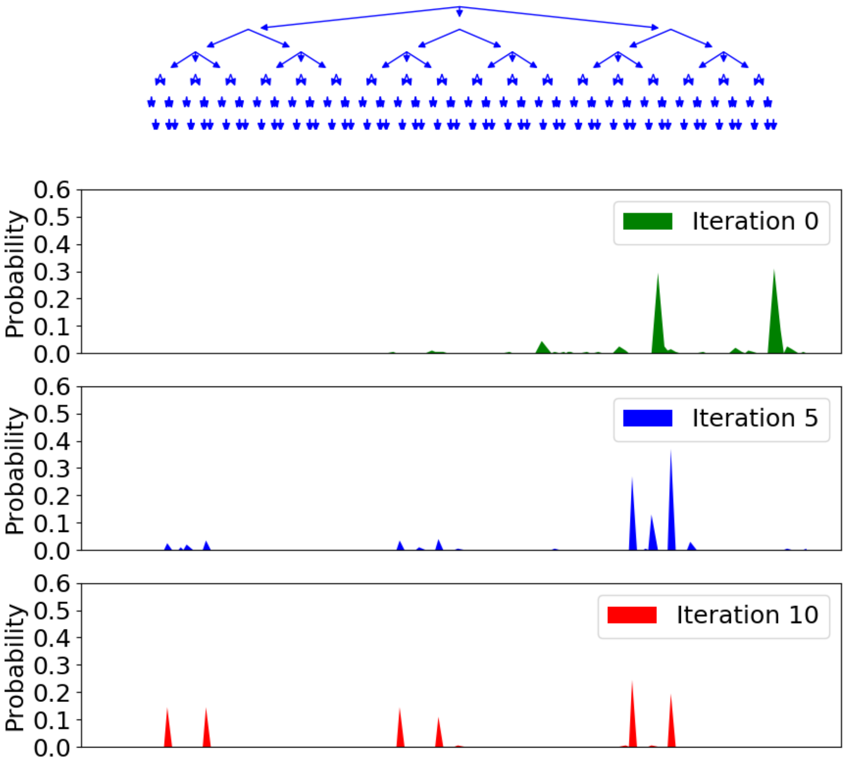
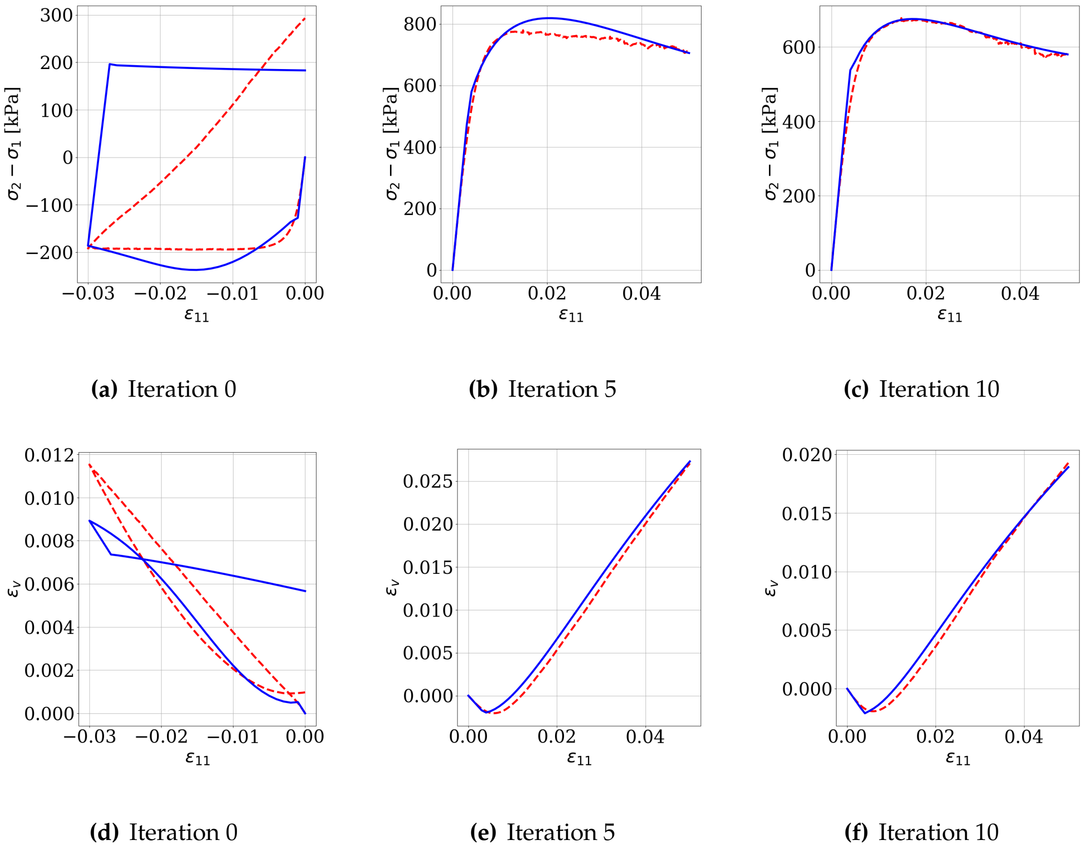


Figure 7.9: Distributions of the experiments determined by the adversary during the DRL training iterations. The X axis corresponds to the leaf nodes of the decision tree for experimentalists.

7.6 Conclusions

We introduce a zero-sum game where two AI experimentalists simultaneously generate experimental data to calibrate and explore weakness of a known constitutive law until the strengths and weaknesses of the constitutive laws on the application range can be identified through competition. In particular, we introduce a pair of AI agents in an adversarial reinforcement learning framework where the protagonist agent who seeks to find the opti-



(a) Iteration 0

(b) Iteration 5

(c) Iteration 10

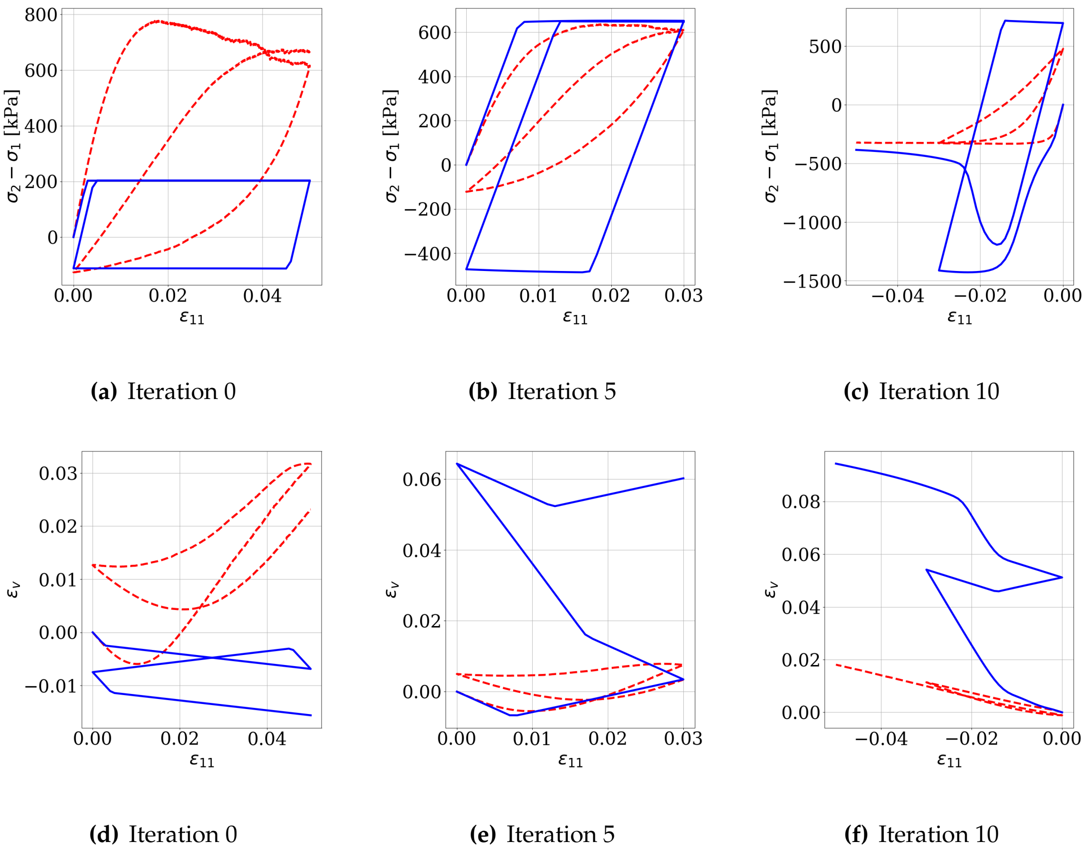
(d) Iteration 0

(e) Iteration 5

(f) Iteration 10

Figure 7.10: Examples of response curves of the games played by the protagonist during the DRL training iterations. Experimental data are plotted in red dashed curves, model predictions are plotted in blue solid curves.

mal way to generate data for model calibration is competing with an adversary who seeks to undermine the protagonist by finding the most devastating test scenarios that expose the weakness of the protagonist. By idealizing the intended range of applications and the experimental programs available as a polytree, we train both agents through deep reinforcement learning such that they can generate the optimal types of experiments that benefit their goals to win the competition. This competition is repeated multiple times, until the agents identify the types of predictions the models are fully capable of doing and the catastrophic failures can be sufficiently prevented. Through competition among



(a) Iteration 0

(b) Iteration 5

(c) Iteration 10

(d) Iteration 0

(e) Iteration 5

(f) Iteration 10

Figure 7.11: Examples of response curves of the games played by the adversary during the DRL training iterations. Experimental data are plotted in red dashed curves, model predictions are plotted in blue solid curves.

the AI agents, we essentially emulate an idealized scientific collaboration and competition among research teams such that we can better understand the valid application range of the learned material laws and prevent misinterpretations by spotting the weaknesses.

Chapter 8

Conclusion

8.1 Main contributions

In this dissertation, firstly, a finite strain multiscale hydro-mechanical model is established via an extended Hill-Mandel condition for two-phase porous media. By assuming that the effective stress principle holds at unit cell scale, we established a micro-to-macro transition that links the micromechanical responses at grain scale to the macroscopic effective stress responses, while modeling the fluid phase only at the macroscopic continuum level. We propose a dual-scale semi-implicit scheme, which treats macroscopic responses implicitly and microscopic responses explicitly. The dual-scale model is shown to have good convergence rate, and is stable and robust. By inferring effective stress measure and poroplasticity parameters, such as porosity, Biot's coefficient and Biot's modulus from microscale simulations, the multiscale model is able to predict effective poro-elasto-plastic responses without introducing additional phenomenological laws. The performance of the proposed framework is demonstrated via a collection of representative numerical examples. Fabric tensors of the representative elementary volumes are computed and analyzed via the anisotropic critical state theory when strain localization occurs.

For fissured porous media, we developed a semi-data-driven multiscale approach that obtains both the traction-separation law and the aperture-porosity-permeability relation from micro-mechanical simulations performed on representative elementary volumes in the finite deformation range. To speed up the multiscale simulations, the incremental constitutive updates of the mechanical responses are obtained from discrete element simulations at the representative elementary volume whereas the hydraulic responses are generated from a neural network trained with data from lattice Boltzmann simulations. These responses are then linked to a macroscopic dual-permeability model. This approach allows one to bypass the need of deriving multi-physical phenomenological laws for complex loading paths. More importantly, it enables the capturing of the evolving anisotropy of the permeabilities of the macro- and micro-pores. A set of numerical experiments are used to demonstrate the robustness of the proposed model.

For multiscale geomechanics problems across more than two scales, we introduced a recursive multiscale framework that captures the hydro-mechanical responses of multi-permeability porous media with embedded strong discontinuities across different length scales. Using the directed graph that represents the hierarchy of the numerical models as the starting point, we identify the knowledge gap and the weakest link of a multiscale multiphysics mutli-permeability model and replace this portion of the computational model with a data-driven counterpart. By creating, training and validating recurrent neural network that has the capacity to memorize and interpret history-dependent events, we established a new recursive data-driven approach where information from multiple sub-scales can be used sequentially to generate macroscopic prediction in a cost-efficient manner. The triple-scale coupling simulations are validated at each sub-scale level where data set for training and validation are mutually exclusive to each other. This hybrid data-driven modeling approach may play a critical role for analyzing problems where human-interpretable knowledge is sufficient to bring closure for forward predictions and for linking simulations across more than two scales in a cost-efficient manner.

Chapter 5 presents a meta-modeling approach in which we attempt to generate traction-separation laws not through explicitly writing a particular model but to provide the computer with modeling options such that it can explore on its own through self-practicing. Unlike previous deep-learning models that soley leverage supervised learning techniques to train neural networks that makes black-box predictions, this new approach focuses on reinforcement learning technique to discover hidden relationships among data and therefore make modeling decisions to emulate the process of writing constitutive models by human. To the best knowledge of the authors, this is the first work on using reinforcement learning and directed graph to form modeling ideas for writing path-dependent constitutive laws. Given the rules (frame indifference, thermodynamic laws, balance principles), we introduce an agenda-based approach where the DRL technique is used to find the optimal way to generate a forward prediction. As demonstrated in our numerical experiments, this approach can be regarded as a generalization of the previous models where neural network predictions may still embed in part of the predictions but are not necessarily completely replacing all components in the conventional models. This flexibility is the key for us to exploit the computer to make *repeated* trial-and-errors and improve from experiments over time to generate the best outcomes, instead of spending significant human time to explore through trial-and-errors.

In Chapter 6, 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 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. 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 metamodeling games are not limited to cooperative games only, they can be non-cooperative, as we introduced in Chapter 7 that a pair of AI agents compete against each other in an adversarial reinforcement learning framework. The agents simultaneously generate experimental data to calibrate and explore weakness of a known constitutive law until the strengths and weaknesses of the constitutive law on the application range can be identified through competition. Eventually, the Nash equilibrium of the game can be found automatically by the reinforcement learning algorithm, which can helps us understand the relationships among objectives between the experimentalist agents, the resultant actions taken by both players, and the outcomes, assuming each player is acting in a rational manner.

8.2 Perspectives

The idea of inventing the meta-modeling game for multiscale modeling of geomaterials could be significant in the sense that it frees modelers from focusing on curve-fitting a physical process. Instead, as future improvements on the models can be made by expanding the action space or simply leveraging the power of computer to improve the models over time, this allows us unprecedented luxury to place our focuses on finding best cause of actions that lead to the most predictive model.

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 resemble 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 amount 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.

In sum, this meta-modeling approach provides the following unique benefits against the conventional hand-crafting approach and black-box neural network models.

1. Since the machine learning procedure is automated, models intended for different purposes or designed to fulfill different demands (speed, accuracy, robustness) can be automatically generated and improved over time through self-plays in the model-creation game.

2. Since the validation procedure is introduced as the reward mechanism for the agent to find the optimal models available, the resultant models are always validated at the end of the games.

3. By recasting constitutive models as directed graphs, previous models established by domain experts can be easily embedded in the proposed framework to expand action spaces efficiently and shorten the training time.

4. The meta-modeling approach is generic and reusable, which means that it can handle different situations with different data, objective functions and rules set by human without going through additional derivation, implementation, material parameter identification and validation. Hence it does not require any debugging once the the game is implemented correctly.

There are also limitations of the current approaches. For instance, the demands for data could be higher than conventional modeling approach, particularly when more sophisticated and rigorous validation metric is used to assign model score and game reward. We also 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.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] B. M. Adams, W. Bohnhoff, K. Dalbey, J. Eddy, M. Eldred, D. Gay, K. Haskell, P. D. Hough, and L. P. Swiler. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.10 user's manual. *Sandia National Laboratories, Tech. Rep. SAND2014-4633*, 2014.

[3] T. W. Anderson and D. A. Darling. A test of goodness of fit. *Journal of the American statistical association*, 49(268):765–769, 1954.

[4] J. E. Andrade and R. I. Borja. Capturing strain localization in dense sands with random density. *International Journal for Numerical Methods in Engineering*, 67(11):1531–1564, 2006.

[5] T. Arbogast, J. Douglas, Jr, and U. Hornung. Derivation of the double porosity model of single phase flow via homogenization theory. *SIAM Journal on Mathematical Analysis*, 21(4):823–836, 1990.

[6] F. Armero. Formulation and finite element implementation of a multiplicative model of coupled poro-plasticity at finite strains under fully saturated conditions. *Computer methods in applied mechanics and engineering*, 171(3):205–241, 1999.

[7] F. Armero. Large-scale modeling of localized dissipative mechanisms in a local continuum: applications to the numerical simulation of strain localization in rate-dependent inelastic solids. *Mechanics of Cohesive-frictional Materials*, 4(2):101–131, 1999.

[8] F. Armero and K. Garikipati. An analysis of strong discontinuities in multiplicative finite strain plasticity and their relation with the numerical simulation of strain localization in solids. *International Journal of Solids and Structures*, 33(20-22):2863–2885, 1996.

[9] F. Armero and C. Linder. New finite elements with embedded strong discontinuities in the finite deformation range. *Computer Methods in Applied Mechanics and Engineering*, 197(33-40):3138–3170, 2008.

[10] A. Aydin, R. I. Borja, and P. Eichhubl. Geological and mathematical framework for failure modes in granular rock. *Journal of Structural Geology*, 28(1):83–98, 2006.

[11] K. Bagi. Stress and strain in granular assemblies. *Mechanics of materials*, 22(3):165–177, 1996.

[12] J. Bang-Jensen and G. Z. Gutin. *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.

[13] J. Bardet and W. Choucair. A linearized integration technique for incremental con-

stitutive equations. *International Journal for Numerical and Analytical Methods in Geomechanics*, 15(1):1–19, 1991.

[14] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[15] R. Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684, 1957.

[16] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary. *Nonlinear finite elements for continua and structures*. John wiley &amp; sons, 2013.

[17] R. Berger, C. Kloss, A. Kohlmeyer, and S. Pirker. Hybrid parallelization of the liggghts open-source dem code. *Powder Technology*, 278:234–247, 2015.

[18] M. Bessa, R. Bostanabad, Z. Liu, A. Hu, D. W. Apley, C. Brinson, W. Chen, and W. K. Liu. A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality. *Computer Methods in Applied Mechanics and Engineering*, 320:633–667, 2017.

[19] M. A. Biot. General theory of three-dimensional consolidation. *Journal of applied physics*, 12(2):155–164, 1941.

[20] J. E. Bishop. Simulating the pervasive fracture of materials and structures using randomly close packed voronoi tessellations. *Computational Mechanics*, 44(4):455–471, 2009.

[21] E. Bonabeau. Agent-based modeling: Methods and techniques for simulating hu-

man systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287, 2002.

[22] R. I. Borja. Finite element simulation of strain localization with large deformation: capturing strong discontinuity using a petrov–galerkin multiscale formulation. *Computer Methods in Applied Mechanics and Engineering*, 191(27):2949–2978, 2002.

[23] R. I. Borja. Assumed enhanced strain and the extended finite element methods: A unification of concepts. *Computer Methods in Applied Mechanics and Engineering*, 197(33):2789–2803, 2008.

[24] R. I. Borja. *Plasticity: modeling & computation*. Springer Science & Business Media, 2013.

[25] R. I. Borja and E. Alarcón. A mathematical framework for finite strain elastoplastic consolidation part 1: Balance laws, variational formulation, and linearization. *Computer Methods in Applied Mechanics and Engineering*, 122(1):145–171, 1995.

[26] R. I. Borja and J. Choo. Cam-clay plasticity, part viii: A constitutive framework for porous materials with evolving internal structure. *Computer Methods in Applied Mechanics and Engineering*, 2016.

[27] R. I. Borja and C. D. Foster. Continuum mathematical modeling of slip weakening in geological systems. *Journal of Geophysical Research: Solid Earth*, 112(B4), 2007.

[28] R. I. Borja, K. M. Sama, and P. F. Sanz. On the numerical integration of three-invariant elastoplastic constitutive models. *Computer methods in applied mechanics and engineering*, 192(9-10):1227–1258, 2003.

[29] R. I. Borja, C. Tamagnini, and E. Alarcón. Elastoplastic consolidation at finite strain part 2: finite element implementation and numerical examples. *Computer Methods in Applied Mechanics and Engineering*, 159(1):103–122, 1998.

[30] E. Bosco, V. Kouznetsova, E. Coenen, M. Geers, and A. Salvadori. A multiscale framework for localizing microstructures towards the onset of macroscopic discontinuity. *Computational Mechanics*, 54(2):299–319, 2014.

[31] R. M. Bowen. Compressible porous media models by use of the theory of mixtures. *International Journal of Engineering Science*, 20(6):697–735, 1982.

[32] S. C. Brenner and R. Scott. *The mathematical theory of finite element methods*, volume 15. Springer Science & Business Media, 2008.

[33] M. D. Brothers, J. T. Foster, and H. R. Millwater. A comparison of different methods for calculating tangent-stiffness matrices in a massively parallel computational peridynamics code. *Computer Methods in Applied Mechanics and Engineering*, 279:247–267, 2014.

[34] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[35] E. C. Bryant and W. Sun. A mixed-mode phase field fracture model in anisotropic rocks with consistent kinematics. *Computer Methods in Applied Mechanics and Engineering*, 2018.

[36] S. L. Bryant, P. R. King, and D. W. Mellor. Network model evaluation of permeability and spatial correlation in a real random sphere packing. *Transport in Porous Media*, 11(1):53–70, 1993.

[37] C. Callari and F. Armero. Finite element methods for the analysis of strong discontinuities in coupled poro-plastic media. *Computer Methods in Applied Mechanics and Engineering*, 191(39):4371–4400, 2002.

[38] C. Callari and F. Armero. Analysis and numerical simulation of strong discontinuities in finite strain poroplasticity. *Computer Methods in Applied Mechanics and Engineering*, 193(27-29):2941–2986, 2004.

[39] C. Callari and F. Federico. Fem validation of a double porosity elastic model for consolidation of structurally complex clayey soils. *International journal for numerical and analytical methods in geomechanics*, 24(4):367–402, 2000.

[40] B. Chareyre, A. Cortis, E. Catalano, and E. Barthélemy. Pore-scale modeling of viscous flow and induced forces in dense sphere packings. *Transport in porous media*, 94(2):595–615, 2012.

[41] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[42] F. Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[43] J. Choo and R. I. Borja. Stabilized mixed finite elements for deformable porous media with double porosity. *Computer Methods in Applied Mechanics and Engineering*, 293:131–154, 2015.

[44] J. Choo and W. Sun. Coupled phase-field and plasticity modeling of geological materials: From brittle fracture to ductile flow. *Computer Methods in Applied Mechanics and Engineering*, 330:1–32, 2018.

[45] J. Choo, J. A. White, and R. I. Borja. Hydromechanical modeling of unsaturated flow in double porosity media. *International Journal of Geomechanics*, page D4016002, 2016.

[46] J. Christoffersen, M. Mehrabadi, and S. Nemat-Nasser. A micromechanical description of granular material behavior. *Journal of Applied Mechanics*, 48(2):339–344, 1981.

[47] E. W. Coenen, V. Kouznetsova, E. Bosco, and M. G. Geers. A multi-scale approach to bridge microscale damage and macroscale failure: a nested computational homogenization-localization framework. *International Journal of Fracture*, pages 1–22, 2012.

[48] O. Coussy. *Poromechanics*. John Wiley & Sons, 2004.

[49] P. A. Cundall and O. D. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29(1):47–65, 1979.

[50] J. S. Curtis and B. Van Wachem. Modeling particle-laden flows: A research outlook. *AIChE journal*, 50(11):2638–2645, 2004.

[51] R. Dabrowski, K. Stencel, and G. Timoszuk. Software is a directed multigraph. In *European Conference on Software Architecture*, pages 360–369. Springer, 2011.

[52] Y. F. Dafalias and M. T. Manzari. Simple plasticity sand model accounting for fabric change effects. *Journal of Engineering mechanics*, 130(6):622–634, 2004.

[53] Y. F. Dafalias, A. G. Papadimitriou, and X. S. Li. Sand plasticity model accounting for inherent fabric anisotropy. *Journal of Engineering Mechanics*, 130(11):1319–1333, 2004.

[54] R. de Borst. Fluid flow in fractured and fracturing porous media: A unified view. *Mechanics Research Communications*, 2016.

[55] R. de Borst. A classification of poromechanical interface elements. *Journal of Modeling in Mechanics and Materials*, 1(1), 2017.

[56] R. de Borst. Fluid flow in fractured and fracturing porous media: A unified view. *Mechanics Research Communications*, 80:47–57, 2017.

[57] R. de Borst, M. A. Crisfield, J. J. Remmers, and C. V. Verhoosel. *Nonlinear finite element analysis of solids and structures*. John Wiley & Sons, 2012.

[58] R. de Borst and O. M. Heeres. A unified approach to the implicit integration of standard, non-standard and viscous plasticity models. *International journal for numerical and analytical methods in geomechanics*, 26(11):1059–1070, 2002.

[59] W. Degruyter, A. Burgisser, O. Bachmann, and O. Malaspinas. Synchrotron x-ray microtomography and lattice boltzmann simulations of gas flow through volcanic pumices. *Geosphere*, 6(5):470–481, 2010.

[60] J. Dolbow and T. Belytschko. A finite element method for crack growth without remeshing. *International journal for numerical methods in engineering*, 46(1):131–150, 1999.

[61] I. C. Dolcetta and H. Ishii. Approximate solutions of the bellman equation of deterministic control theory. *Applied Mathematics and Optimization*, 11(1):161–181, 1984.

[62] X. Du and M. Ostoja-Starzewski. On the size of representative volume element for darcy law in random media. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 462, pages 2949–2963. The Royal Society, 2006.

[63] H. Edelsbrunner and N. R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15(3):223–241, 1996.

[64] U. El Shamy and Y. Abdelhamid. Modeling granular soils liquefaction using coupled lattice boltzmann method and discrete element method. *Soil Dynamics and Earthquake Engineering*, 67:119–132, 2014.

[65] M. Elices, G. Guinea, J. Gomez, and J. Planas. The cohesive zone model: advantages, limitations and challenges. *Engineering fracture mechanics*, 69(2):137–163, 2002.

[66] T. R. Faisal, N. Hristozov, T. L. Western, A. D. Rey, and D. Pasini. Computational study of the elastic properties of rheum rhabarbarum tissues via surrogate models of tissue geometry. *Journal of structural biology*, 185(3):285–294, 2014.

[67] F. Feyel. A multilevel finite element method (fe2) to describe the response of highly non-linear structures using generalized continua. *Computer Methods in applied Mechanics and engineering*, 192(28-30):3233–3244, 2003.

[68] J. Fish. *Practical multiscaling*. John Wiley & Sons, 2013.

[69] J. Fish and W. Wu. A nonintrusive stochastic multiscale solver. *International Journal for Numerical Methods in Engineering*, 88(9):862–879, 2011.

[70] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson. Learning to communicate

with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.

[71] I. Frankenreiter, D. Rosato, and C. Miehe. Hybrid micro-macro-modeling of evolving anisotropies and length scales in finite plasticity of polycrystals. *PAMM*, 11(1):515–518, 2011.

[72] P. Fu and Y. F. Dafalias. Fabric evolution within shear bands of granular materials and its relation to critical state theory. *International Journal for numerical and analytical methods in geomechanics*, 35(18):1918–1948, 2011.

[73] S. Galindo-Torres, A. Scheuermann, H. Mühlhaus, and D. Williams. A micro-mechanical approach for the study of contact erosion. *Acta Geotechnica*, 10(3):357–368, 2013.

[74] S. Galindo-Torres, A. Scheuermann, H. Mühlhaus, and D. Williams. A micro-mechanical approach for the study of contact erosion. *Acta Geotechnica*, 10(3):357–368, 2015.

[75] A. Gens and D. Potts. Critical state models in computational geomechanics. *Engineering Computations*, 5(3):178–197, 1988.

[76] H. Gerke and M. V. Genuchten. A dual-porosity model for simulating the preferential movement of water and solutes in structured porous media. *Water resources research*, 29(2):305–319, 1993.

[77] J. Ghaboussi, J. Garrett Jr, and X. Wu. Knowledge-based modeling of material behavior with neural networks. *Journal of engineering mechanics*, 117(1):132–153, 1991.

[78] J. Ghaboussi, D. A. Pecknold, M. Zhang, and R. M. Haj-Ali. Autoprogressive training of neural network constitutive models. *International Journal for Numerical Methods in Engineering*, 42(1):105–126, 1998.

[79] M. Ghavamzadeh, S. Mannor, J. Pineau, A. Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

[80] I. Gitman, H. Askes, and L. Sluys. Representative volume: existence and size determination. *Engineering fracture mechanics*, 74(16):2518–2534, 2007.

[81] B. Gong, M. Karimi-Fard, L. J. Durlofsky, et al. Upscaling discrete fracture characterizations to dual-porosity, dual-permeability models for efficient simulation of flow with strong gravitational effects. *SPE Journal*, 13(01):58–67, 2008.

[82] C. Goniva, C. Kloss, A. Hager, and S. Pirker. An open source cfd-dem perspective. In *Proceedings of OpenFOAM workshop Gothenburg, Sweden*, 2010.

[83] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[84] W. Graf, S. Freitag, M. Kaliske, and J.-U. Sickert. Recurrent neural networks for uncertain time-dependent structural behavior. *Computer-Aided Civil and Infrastructure Engineering*, 25(5):322–323, 2010.

[85] R. L. Graham, D. E. Knuth, O. Patashnik, and S. Liu. Concrete mathematics: a foundation for computer science. *Computers in Physics*, 3(5):106–107, 1989.

[86] W. G. Gray, C. T. Miller, and B. A. Schrefler. Averaging theory for description of environmental problems: What have we learned? *Advances in water resources*, 51:123–138, 2013.

[87] W. G. Gray, B. A. Schrefler, and F. Pesavento. The solid phase stress tensor in porous media mechanics and the hill–mandel condition. *Journal of the Mechanics and Physics of Solids*, 57(3):539–554, 2009.

[88] N. Guo and J. Zhao. A coupled fem/dem approach for hierarchical multiscale modelling of granular media. *International Journal for Numerical Methods in Engineering*, 99(11):789–818, 2014.

[89] N. Guo, J. Zhao, and W. Sun. Multiscale analysis of shear failure of thick-walled hollow cylinder in dry sand. *Géotechnique Letters*, 6(1), 2016.

[90] A. Hagberg, P. Swart, and D. S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[91] K. Han, Y. Feng, and D. Owen. Numerical simulations of irregular particle transport in turbulent flows using coupled lbm-dem. *Computer Modeling in Engineering and Sciences*, 18(2):87, 2007.

[92] Y. Han and P. A. Cundall. Lbm–dem modeling of fluid–solid interaction in porous media. *International Journal for Numerical and Analytical Methods in Geomechanics*, 37(10):1391–1407, 2013.

[93] M. Hilpert, R. Glantz, and C. T. Miller. Calibration of a pore-network model by a pore-morphological analysis. *Transport in Porous Media*, 51(3):267–285, 2003.

[94] C. Hirschberger, S. Ricker, P. Steinmann, and N. Sukumar. Computational multiscale modelling of heterogeneous material layers. *Engineering Fracture Mechanics*, 76(6):793–812, 2009.

[95] C. B. Hirschberger, N. Sukumar, and P. Steinmann. Computational homogenization of material layers with micromorphic mesostructure. *Philosophical Magazine*, 88(30-32):3603–3631, 2008.

[96] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[97] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[98] T. Hughes. The finite element method: linear static and dynamic finite element analysis. 1987.

[99] T. J. Hughes, K. S. Pister, and R. L. Taylor. Implicit-explicit finite elements in nonlinear transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 17:159–182, 1979.

[100] W. Humboldt. On language: On the diversity of human language construction and its influence on the mental development of the human species. 1999.

[101] R. Ibañez, E. Abisset-Chavanne, J. V. Aguado, D. Gonzalez, E. Cueto, and F. Chinesta. A manifold learning approach to data-driven computational elasticity and inelasticity. *Archives of Computational Methods in Engineering*, 25(1):47–57, 2018.

[102] J. R. Jain and S. Ghosh. Damage evolution in composites with a homogenization-

based continuum damage mechanics model. *International Journal of Damage Mechanics*, 18(6):533–568, 2009.

[103] M. Jefferies. Nor-sand: a simle critical state model for sand. *Géotechnique*, 43(1):91–103, 1993.

[104] A. Johari, G. Habibagahi, and A. Ghahramani. Prediction of soil–water characteristic curve using genetic programming. *Journal of Geotechnical and Geoenvironmental Engineering*, 132(5):661–665, 2006.

[105] S. Jung and J. Ghaboussi. Neural network constitutive model for rate-dependent materials. *Computers &amp; Structures*, 84(15):955–963, 2006.

[106] O. L. Kafka, C. Yu, M. Shakoor, Z. Liu, G. J. Wagner, and W. K. Liu. Data-driven mechanistic modeling of influence of microstructure on high-cycle fatigue life of nickel titanium. *JOM*, pages 1–5, 2018.

[107] M. G. Kendall et al. The advanced theory of statistics. *The advanced theory of statistics.*, (2nd Ed), 1946.

[108] S. Keshavarz and S. Ghosh. Multi-scale crystal plasticity finite element model approach to modeling nickel-based superalloys. *Acta Materialia*, 61(17):6549–6561, 2013.

[109] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[110] K. Kirane and S. Ghosh. A cold dwell fatigue crack nucleation criterion for polycrystalline ti-6242 using grain-level crystal plasticity fe model. *International Journal of Fatigue*, 30(12):2127–2139, 2008.

[111] T. Kirchdoerfer and M. Ortiz. Data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 304:81–101, 2016.

[112] P. Krause, D. Boyle, and F. Bäse. Comparison of different efficiency criteria for hydrological model assessment. *Advances in geosciences*, 5:89–97, 2005.

[113] K. L. Kuhlman, B. Malama, and J. E. Heath. Multiporosity flow in fractured low-permeability rocks. *Water Resources Research*, 51(2):848–860, 2015.

[114] M. R. Kuhn, H. E. Renken, A. D. Mixsell, and S. L. Kramer. Investigation of cyclic liquefaction with discrete element simulations. *Journal of Geotechnical and Geoenvironmental Engineering*, 140(12):04014075, 2014.

[115] M. R. Kuhn, W. Sun, and Q. Wang. Stress-induced anisotropy in granular materials: fabric, stiffness, and permeability. *Acta Geotechnica*, 10(4):399–419, 2015.

[116] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[117] K. Lamorski, Y. Pachepsky, C. Sławiński, and R. Walczak. Using support vector machines to develop pedotransfer functions for water retention of soils in poland. *Soil Science Society of America Journal*, 72(5):1243–1247, 2008.

[118] J. Larsson and R. Larsson. Finite-element analysis of localization of deformation and fluid pressure in an elastoplastic porous medium. *International journal of solids and structures*, 37(48):7231–7257, 2000.

[119] B. Le, J. Yvonnet, and Q.-C. He. Computational homogenization of nonlinear elastic materials using neural networks. *International Journal for Numerical Methods in Engineering*, 104(12):1061–1084, 2015.

[120] E. Le Garzic, T. de L'Hamaide, M. Diraison, Y. Géraud, J. Sausse, M. De Urreiztieta, B. Hauville, and J.-M. Champanhet. Scaling and geometric properties of extensional fracture systems in the proterozoic basement of yemen. tectonic interpretation and fluid flow implications. *Journal of Structural Geology*, 33(4):519–536, 2011.

[121] M. Lefik and B. Schrefler. Artificial neural network for parameter identifications for an elasto-plastic model of superconducting cable under cyclic loading. *Computers & structures*, 80(22):1699–1713, 2002.

[122] M. Lefik and B. Schrefler. Artificial neural network as an incremental non-linear constitutive model for a finite element code. *Computer methods in applied mechanics and engineering*, 192(28):3265–3283, 2003.

[123] J. Lewandowska, A. Szymkiewicz, K. Burzyński, and M. Vauclin. Modeling of unsaturated water flow in double-porosity soils by the homogenization approach. *Advances in Water Resources*, 27(3):283–296, 2004.

[124] X. Li and Y. Dafalias. Dissipation consistent fabric tensor definition from dem to continuum for granular media. *Journal of the Mechanics and Physics of Solids*, 78:141–153, 2015.

[125] X. S. Li and Y. F. Dafalias. Anisotropic critical state theory: role of fabric. *Journal of Engineering Mechanics*, 138(3):263–275, 2011.

[126] X. S. Li and Y. F. Dafalias. Anisotropic critical state theory: role of fabric. *Journal of Engineering Mechanics*, 138(3):263–275, 2012.

[127] H. I. Ling and H. Liu. Pressure-level dependency and densification behavior of sand

through generalized plasticity model. *Journal of Engineering Mechanics*, 129(8):851–860, 2003.

[128] H. I. Ling and S. Yang. Unified sand model based on the critical state and generalized plasticity. *Journal of Engineering Mechanics*, 132(12):1380–1391, 2006.

[129] F. Liu and R. I. Borja. A contact algorithm for frictional crack propagation with the extended finite element method. *International Journal for Numerical methods in engineering*, 76(10):1489–1512, 2008.

[130] G. Liu, W. Sun, S. M. Lowinger, Z. Zhang, M. Huang, and J. Peng. Coupled flow network and discrete element modeling of injection-induced crack propagation and coalescence in brittle rock. *Acta Geotechnica*, pages 1–26, 2018.

[131] Y. Liu, W. Sun, and J. Fish. Determining material parameters for critical state plasticity models based on multilevel extended digital database. *Journal of Applied Mechanics*, 83(1):011003, 2016.

[132] Y. Liu, W. Sun, Z. Yuan, and J. Fish. A nonlocal multiscale discrete-continuum model for predicting mechanical behavior of granular materials. *International Journal for Numerical Methods in Engineering*, 2015.

[133] Y. Liu, W. Sun, Z. Yuan, and J. Fish. A nonlocal multiscale discrete-continuum model for predicting mechanical behavior of granular materials. *International Journal for Numerical Methods in Engineering*, 106(2):129–160, 2016.

[134] J. Lubliner and F. Auricchio. Generalized plasticity and shape-memory alloys. *International Journal of Solids and Structures*, 33(7):991–1003, 1996.

[135] M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[136] L. Malcher, F. A. Pires, J. C. de Sá, and F. Andrade. Numerical integration algorithm of a new model for metal plasticity and fracture including pressure and lode angle dependence. *International Journal of Material Forming*, 2(1):443–446, 2009.

[137] M. Mansouri, J.-Y. Delenne, M. S. El Youssoufi, and A. Séridi. A 3d dem-lbm approach for the assessment of the quick condition for sands. *Comptes Rendus Mécanique*, 337(9):675–681, 2009.

[138] M. T. Manzari and Y. F. Dafalias. A critical state two-surface plasticity model for sands. *Geotechnique*, 47(2):255–272, 1997.

[139] M. J. Martinez, P. Newell, J. E. Bishop, and D. Turner. Coupled multiphase flow and geomechanics model for analysis of joint reactivation during co 2 sequestration operations. *International Journal of Greenhouse Gas Control*, 17:148–160, 2013.

[140] W. McKinney et al. Data structures for statistical computing in python.

[141] M. M. Mehrabadi, S. Nemat-Nasser, and M. Oda. On statistical description of stress and fabric in granular materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 6(1):95–108, 1982.

[142] C. Miehe and C. Bayreuther. On multiscale fe analyses of heterogeneous structures: from homogenization to multigrid solvers. *International Journal for Numerical Methods in Engineering*, 71(10):1135–1180, 2007.

[143] C. Miehe and J. Dettmar. A framework for micro–macro transitions in periodic par-

ticle aggregates of granular materials. *Computer Methods in Applied Mechanics and Engineering*, 193(3):225–256, 2004.

[144] C. Miehe, J. Dettmar, and D. Zäh. Homogenization and two-scale simulations of granular materials for different microstructural constraints. *International Journal for Numerical Methods in Engineering*, 83(8-9):1206–1236, 2010.

[145] P. Mira, L. Tonni, M. Pastor, and J. Fernandez Merodo. A generalized midpoint algorithm for the integration of a generalized plasticity model for sands. *International journal for numerical methods in engineering*, 77(9):1201–1223, 2009.

[146] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[147] N. Moës, M. Cloirec, P. Cartraud, and J.-F. Remacle. A computational approach to handle complex microstructure geometries. *Computer methods in applied mechanics and engineering*, 192(28-30):3163–3177, 2003.

[148] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *Int. J. Numer. Meth. Engng*, 46:131–150, 1999.

[149] T. Mohammadnejad and A. Khoei. An extended finite element method for hydraulic fracture propagation in deformable porous media with the cohesive crack model. *Finite Elements in Analysis and Design*, 73:77–95, 2013.

[150] J. Mosler. Modeling strong discontinuities at finite strains—a novel numerical implementation. *Computer Methods in Applied Mechanics and Engineering*, 195(33):4396–4419, 2006.

[151] S. Na and W. Sun. Computational thermo-hydro-mechanics for multiphase freezing and thawing porous media in the finite deformation range. *Computer Methods in Applied Mechanics and Engineering*, 318:667–700, 2017.

[152] N. M. Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.

[153] T.-T. Ng. Input parameters of discrete element methods. *Journal of Engineering Mechanics*, 132(7):723–729, 2006.

[154] M. Nikolic, A. Ibrahimbegovic, and P. Miscevic. Modelling of internal fluid flow in cracks with embedded strong discontinuities. In *Computational Methods for Solids and Fluids*, pages 315–341. Springer, 2016.

[155] M. Nitka, G. Combe, C. Dascalu, and J. Desrues. Two-scale modeling of granular materials: a dem-fem approach. *Granular Matter*, 13(3):277–281, 2011.

[156] A. Nur and J. Byerlee. An exact effective stress law for elastic deformation of rock with fluids. *Journal of Geophysical Research*, 76(26):6414–6419, 1971.

[157] M. Ohnaka and T. Yamashita. A cohesive zone model for dynamic shear faulting based on experimentally inferred constitutive relation and strong motion source parameters. *Journal of Geophysical Research: Solid Earth*, 94(B4):4089–4104, 1989.

[158] J. Oliver, M. Caicedo, E. Roubin, A. E. Huespe, and J. Hernández. Continuum approach to computational multiscale modeling of propagating fracture. *Computer Methods in Applied Mechanics and Engineering*, 294:384–427, 2015.

[159] E. Onate and J. Rojek. Combination of discrete element and finite element meth-

ods for dynamic analysis of geomechanics problems. *Computer methods in applied mechanics and engineering*, 193(27):3087–3128, 2004.

[160] M. Ortiz and A. Pandolfi. Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. *International journal for numerical methods in engineering*, 44(9):1267–1282, 1999.

[161] M. Ostoja-Starzewski, X. Du, Z. Khisaeva, and W. Li. Comparisons of the size of the representative volume element in elastic, plastic, thermoelastic, and permeable random microstructures. *International Journal for Multiscale Computational Engineering*, 5(2), 2007.

[162] S. J. Pan, Q. Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[163] J. H. Panchal, S. R. Kalidindi, and D. L. McDowell. Key computational modeling issues in integrated computational materials engineering. *Computer-Aided Design*, 45(1):4–25, 2013.

[164] A. Pandolfi, P. Guduru, M. Ortiz, and A. Rosakis. Three dimensional cohesive-element analysis and experiments of dynamic fracture in c300 steel. *International Journal of Solids and Structures*, 37(27):3733–3760, 2000.

[165] K. Park and G. H. Paulino. Cohesive zone models: a critical review of traction-separation relationships across fracture surfaces. *Applied Mechanics Reviews*, 64(6):060802, 2011.

[166] K. Park, G. H. Paulino, and J. R. Roesler. A unified potential-based cohesive model

of mixed-mode fracture. *Journal of the Mechanics and Physics of Solids*, 57(6):891–908, 2009.

[167] M. Pastor, A. Chan, P. Mira, D. Manzanal, J. Fernández Merodo, and T. Blanc. Computational geomechanics: the heritage of olek zienkiewicz. *International Journal for Numerical Methods in Engineering*, 87(1-5):457–489, 2011.

[168] M. Pastor, O. Zienkiewicz, and A. Chan. Generalized plasticity and the modelling of soil behaviour. *International Journal for Numerical and Analytical Methods in Geomechanics*, 14(3):151–190, 1990.

[169] M. S. Paterson and T.-f. Wong. *Experimental rock deformation-the brittle field*. Springer Science & Business Media, 2005.

[170] R. P. Pawlowski, E. T. Phipps, and A. G. Salinger. Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, part i: Template-based generic programming. *Scientific Programming*, 20(2):197–219, 2012.

[171] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[172] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[173] J. M. Pestana, A. J. Whittle, and L. A. Salvati. Evaluation of a constitutive model

for clays and sands: Part i–sand behaviour. *International journal for numerical and analytical methods in geomechanics*, 26(11):1097–1121, 2002.

[174] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2817–2826. JMLR. org, 2017.

[175] M. Piri and M. J. Blunt. Three-dimensional mixed-wet random pore-scale network modeling of two-and three-phase flow in porous media. i. model description. *Physical Review E*, 71(2):026301, 2005.

[176] J. H. Prevost. Nonlinear transient phenomena in saturated porous media. *Computer Methods in Applied Mechanics and Engineering*, 30(1):3 – 18, 1982.

[177] J. H. Prevost. Implicit-explicit schemes for nonlinear consolidation. *Computer Methods in Applied Mechanics and Engineering*, 39(2):225–239, 1983.

[178] J. H. Prevost and N. Sukumar. Faults simulations for three-dimensional reservoir-geomechanical models with the extended finite element method. *Journal of the Mechanics and Physics of Solids*, 86:1–18, 2016.

[179] S. R. Pride and J. G. Berryman. Linear dynamics of double-porosity dual-permeability materials. i. governing equations and acoustic attenuation. *Physical Review E*, 68(3):036603, 2003.

[180] T. Rabczuk and P. Areias. A new approach for modelling slip lines in geological materials with cohesive models. *International Journal for Numerical and Analytical Methods in Geomechanics*, 30(11):1159–1172, 2006.

[181] R. Raileanu, E. Denton, A. Szlam, and R. Fergus. Modeling others using oneself in multi-agent reinforcement learning. *arXiv preprint arXiv:1802.09640*, 2018.

[182] J. R. Rice. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of applied mechanics*, 35(2):379–386, 1968.

[183] J. R. Rice and M. P. Cleary. Some basic stress diffusion solutions for fluid-saturated elastic porous media with compressible constituents. *Reviews of Geophysics*, 14(2):227–241, 1976.

[184] M. Robinson, M. Ramaioli, and S. Luding. Fluid–particle flow simulations using two-way-coupled mesoscale sph–dem and validation. *International journal of multiphase flow*, 59:121–134, 2014.

[185] N. M. Rodriguez and P. V. Lade. True triaxial tests on cross-anisotropic deposits of fine nevada sand. *International Journal of Geomechanics*, 13(6):779–793, 2013.

[186] J. W. Rudnicki. Fracture mechanics applied to the earth's crust. *Annual Review of Earth and Planetary Sciences*, 8(1):489–525, 1980.

[187] J. Rutqvist, Y. Ijiri, and H. Yamamoto. Implementation of the barcelona basic model into tough–flac for simulations of the geomechanical behavior of unsaturated soils. *Computers & Geosciences*, 37(6):751–762, 2011.

[188] A. Salinger, R. Bartlett, A. Bradley, Q. Chen, I. Demeshko, X. Gao, G. Hanson, A. Mota, R. Muller, E. Nielsen, et al. Albany: Using component-based design to develop a flexible, generic multiphysics analysis code. *International Journal for Multiscale Computational Engineering*, 2016.

[189] L. Sanavia, B. Schrefler, and P. Steinmann. A formulation for an unsaturated porous medium undergoing large inelastic strains. *Computational Mechanics*, 28(2):137–151, 2002.

[190] M. Sánchez, A. Gens, L. do Nascimento Guimarães, and S. Olivella. A double structure generalized plasticity model for expansive materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 29(8):751–787, 2005.

[191] F. W. Scholz and M. A. Stephens. K-sample anderson–darling tests. *Journal of the American Statistical Association*, 82(399):918–924, 1987.

[192] H. Shahir, A. Pak, M. Taiebat, and B. Jeremić. Evaluation of variation of permeability in liquefiable soil under earthquake loading. *Computers and Geotechnics*, 40:74–88, 2012.

[193] C. E. Shannon. Xxii. programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(314):256–275, 1950.

[194] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[195] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[196] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[197] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[198] J. C. Simo and T. J. Hughes. *Computational inelasticity*, volume 7. Springer Science & Business Media, 2006.

[199] B. Simon, J.-S. Wu, O. Zienkiewicz, and D. Paul. Evaluation of u – w and u – π finite element methods for the dynamic response of saturated porous media using one-dimensional models. *International Journal for Numerical and Analytical Methods in Geomechanics*, 10(5):461–482, 1986.

[200] S. W. Sloan. Substepping schemes for the numerical integration of elastoplastic stress–strain relations. *International journal for numerical methods in engineering*, 24(5):893–911, 1987.

[201] S. W. Sloan, A. J. Abbo, and D. Sheng. Refined explicit integration of elastoplastic models with automatic error control. *Engineering Computations*, 18(1/2):121–194, 2001.

[202] V. Šmilauer, E. Catalano, B. Chareyre, S. Dorofeenko, J. Duriez, A. Gladky, J. Kozicki, C. Modenese, L. Scholtès, L. Sibille, et al. Yade documentation. *The Yade Project.(http://yade-dem. org/doc/)*, 2010.

[203] X. Song and R. I. Borja. Mathematical framework for unsaturated flow in the finite deformation range. *International Journal for Numerical Methods in Engineering*, 97(9):658–682, 2014.

[204] P. Steinmann. A finite element formulation for strong discontinuities in fluid-saturated porous media. *Mechanics of Cohesive-frictional Materials*, 4(2):133–152, 1999.

[205] Q. Sun, Y. Tao, and Q. Du. Stochastic training of residual networks: a differential equation viewpoint. *arXiv preprint arXiv:1812.00174*, 2018.

[206] W. Sun. A unified method to predict diffuse and localized instabilities in sands. *Geomechanics and Geoengineering*, 8(2):65–75, 2013.

[207] W. Sun. A stabilized finite element formulation for monolithic thermo-hydro-mechanical simulations at finite strain. *International Journal for Numerical Methods in Engineering*, 2015.

[208] W. Sun. A stabilized finite element formulation for monolithic thermo-hydro-mechanical simulations at finite strain. *International Journal for Numerical Methods in Engineering*, 103(11):798–839, 2015.

[209] W. Sun, J. E. Andrade, and J. W. Rudnicki. Multiscale method for characterization of porous microstructures and their impact on macroscopic effective permeability. *International Journal for Numerical Methods in Engineering*, 88(12):1260–1279, 2011.

[210] W. Sun, J. E. Andrade, J. W. Rudnicki, and P. Eichhubl. Connecting microstructural attributes and permeability from 3d tomographic images of in situ shear-enhanced compaction bands using multiscale computations. *Geophysical Research Letters*, 38(10), 2011.

[211] W. Sun, Z. Cai, and J. Choo. Mixed arlequin method for multiscale poromechanics problems. *International Journal for Numerical Methods in Engineering*, 2016.

[212] W. Sun, Q. Chen, and J. T. Ostien. Modeling the hydro-mechanical responses of strip and circular punch loadings on water-saturated collapsible geomaterials. *Acta Geotechnica*, 9(5):903–934, 2014.

[213] W. Sun, M. Kuhn, and J. Rudnicki. A micromechanical analysis on permeability evolutions of a dilatant shear band. In *48th US Rock Mechanics/Geomechanics Symposium*. American Rock Mechanics Association, 2014.

[214] W. Sun, M. R. Kuhn, and J. W. Rudnicki. A multiscale dem-lbm analysis on permeability evolutions inside a dilatant shear band. *Acta Geotechnica*, 8(5):465–480, 2013.

[215] W. Sun, J. T. Ostien, and A. G. Salinger. A stabilized assumed deformation gradient finite element formulation for strongly coupled poromechanical simulations at finite strain. *International Journal for Numerical and Analytical Methods in Geomechanics*, 37(16):2755–2788, 2013.

[216] W. Sun and T.-f. Wong. Prediction of permeability and formation factor of sandstone with hybrid lattice boltzmann/finite element simulation on microtomographic images. *International Journal of Rock Mechanics and Mining Sciences*, 106:269–277, 2018.

[217] R. S. Sutton. Introduction: The challenge of reinforcement learning. In *Reinforcement Learning*, pages 1–3. Springer, 1992.

[218] A. E. Tallman, L. P. Swiler, Y. Wang, and D. L. McDowell. Reconciled top-down and bottom-up hierarchical multiscale calibration of bcc fe crystal plasticity. *International Journal for Multiscale Computational Engineering*, 15(6), 2017.

[219] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vi-

cente. Multiagent cooperation and competition with deep reinforcement learning. *PloS one*, 12(4):e0172395, 2017.

[220] M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.

[221] M. Tawhai, J. Bischoff, D. Einstein, A. Erdemir, T. Guess, and J. Reinbolt. Multiscale modeling in computational biomechanics. *IEEE Engineering in medicine and biology magazine*, 28(3), 2009.

[222] K. Terzaghi. *Theory of consolidation*. Wiley Online Library, 1943.

[223] K. Terzaghi, K. Terzaghi, C. Engineer, A. Czechoslowakia, K. Terzaghi, I. Civil, A. Tchécoslovaquie, and E. Unis. *Theoretical soil mechanics*, volume 18. Wiley New York, 1943.

[224] v. K. Terzaghi. The shearing resistance of saturated soils and the angle between the planes of shear. In *Proceedings of the 1st International Conference on Soil Mechanics and Foundation Engineering*, volume 1, pages 54–56. Harvard University Press Cambridge, MA, 1936.

[225] S. Timoshenko. *History of strength of materials: with a brief account of the history of theory of elasticity and theory of structures*. Courier Corporation, 1953.

[226] A. Tordesillas, D. M. Walker, and Q. Lin. Force cycles and force chains. *Physical Review E*, 81(1):011302, 2010.

[227] A. Tordesillas, D. M. Walker, A. L. Rechenmacher, and S. Abedi. Discovering community structures and dynamical networks from grain-scale kinematics of shear

bands in sand. In *Advances in Bifurcation and Degradation in Geomaterials*, pages 67–73. Springer, 2011.

[228] S. Toro, P. Sánchez, J. Podestá, P. Blanco, A. E. Huespe, and R. Feijóo. Cohesive surface model for fracture based on a two-scale formulation: computational implementation aspects. *Computational mechanics*, 58(4):549–585, 2016.

[229] S. Toro, P. J. Sánchez, A. E. Huespe, S. M. Giusti, P. J. Blanco, and R. Feijóo. A two-scale failure model for heterogeneous materials: numerical implementation based on the finite element method. *International Journal for Numerical Methods in Engineering*, 97(5):313–351, 2014.

[230] D. Tran, N. Prime, F. Froiio, C. Callari, and E. Vincens. Numerical modelling of backward front propagation in piping erosion by dem-lbm coupling. *European Journal of Environmental and Civil Engineering*, 21(7-8):960–987, 2017.

[231] C. Truesdell and W. Noll. The non-linear field theories of mechanics. In *The non-linear field theories of mechanics*, pages 1–579. Springer, 2004.

[232] C. Truesdell and R. Toupin. *The classical field theories*. Springer, 1960.

[233] A. Truty and T. Zimmermann. Stabilized mixed finite element formulations for materially nonlinear partially saturated two-phase media. *Computer methods in applied mechanics and engineering*, 195(13):1517–1546, 2006.

[234] X. Tu, J. E. Andrade, and Q. Chen. Return mapping for nonsmooth and multiscale elastoplasticity. *Computer Methods in Applied Mechanics and Engineering*, 198(30-32):2286–2296, 2009.

[235] V. Tvergaard and J. W. Hutchinson. The relation between crack growth resistance and fracture process parameters in elastic-plastic solids. *Journal of the Mechanics and Physics of Solids*, 40(6):1377–1397, 1992.

[236] O. I. Ulven and W. Sun. Capturing the two-way hydromechanical coupling effect on fluid-driven fracture in a dual-graph lattice beam model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 42(5):736–767, 2018.

[237] I. Vardoulakis. Deformation of water-saturated sand: I. uniform undrained deformation and shear banding. *Géotechnique*, 46(3):441–456, 1996.

[238] C. V. Verhoosel, J. J. Remmers, M. A. Gutiérrez, and R. De Borst. Computational homogenization for adhesive and cohesive failure in quasi-brittle solids. *International Journal for Numerical Methods in Engineering*, 83(8-9):1155–1179, 2010.

[239] D. Versino, A. Tonda, and C. A. Bronkhorst. Data driven modeling of plastic deformation. *Computer Methods in Applied Mechanics and Engineering*, 318:981–1004, 2017.

[240] V. Šmilauer and B. Chareyre. Dem formulation. In *Yade Documentation 2nd ed*. The Yade Project, 2015. http://yade-dem.org/doc/.

[241] K. Wang and W. Sun. Anisotropy of a tensorial bishop coefficient for wetted granular materials. *Journal of Engineering Mechanics*, page B4015004, 2015.

[242] K. Wang and W. Sun. A semi-implicit micropolar discrete-to-continuum method for granular materials. In M. Papadrakakis, V. Papadopoulos, G. Stefanou, and V. Plevris, editors, *Proceedings of European Congress on Computational Methods in Applied Science and Engineering*, number June, pages 5–10, Crete Island, 2016.

[243] K. Wang and W. Sun. A semi-implicit discrete-continuum coupling method for porous media based on the effective stress principle at finite strain. *Computer Methods in Applied Mechanics and Engineering*, 304:546 – 583, 2016.

[244] K. Wang and W. Sun. A semi-implicit discrete-continuum coupling method for porous media based on the effective stress principle at finite strain. *Computer Methods in Applied Mechanics and Engineering*, 304:546–583, 2016.

[245] K. Wang and W. Sun. Data-driven discrete-continuum method for partially saturated micro-polar porous media. In *Poromechanics VI*, pages 571–578. 2017.

[246] K. Wang and W. Sun. A unified variational eigen-erosion framework for interacting brittle fractures and compaction bands in fluid-infiltrating porous media. *Computer Methods in Applied Mechanics and Engineering*, 318:1–32, 2017.

[247] K. Wang and W. Sun. A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. *Computer Methods in Applied Mechanics and Engineering*, 334:337–380, 2018.

[248] K. Wang and W. Sun. Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning. *Computer Methods in Applied Mechanics and Engineering*, 346:216–241, 2019.

[249] K. Wang and W. Sun. An updated lagrangian lbm–dem–fem coupling model for dual-permeability fissured porous media with embedded discontinuities. *Computer Methods in Applied Mechanics and Engineering*, 344:276–305, 2019.

[250] K. Wang, W. Sun, and Q. Du. A cooperative game for automated learning of elasto-

plasticity knowledge graphs and models with ai-guided experimentation. *Computational Mechanics*, pages 1–33, 2019.

[251] K. Wang, W. Sun, S. Salager, S. Na, and G. Khaddour. Identifying material parameters for a micro-polar plasticity model via x-ray micro-computed tomographic (ct) images: Lessons learned from the curve-fitting exercises. *International Journal for Multiscale Computational Engineering*, 14(4), 2016.

[252] C. Wellmann, C. Lillie, and P. Wriggers. Homogenization of granular material modeled by a three-dimensional discrete element method. *Computers and Geotechnics*, 35(3):394–405, 2008.

[253] D. B. West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[254] J. A. White and R. I. Borja. Stabilized low-order finite elements for coupled solid-deformation/fluid-diffusion and their application to fault zone transients. *Computer Methods in Applied Mechanics and Engineering*, 197(49):4353–4366, 2008.

[255] C. A. Wibberley, J. Gonzalez-Dunia, and O. Billon. Faults as barriers or channels to production-related flow: insights from case studies. *Petroleum Geoscience*, pages petgeo2016–057, 2016.

[256] J. R. Williams and N. Rege. Coherent vortex structures in deforming granular materials. *Mechanics of Cohesive-frictional Materials: An International Journal on Experiments, Modelling and Computation of Materials and Structures*, 2(3):223–236, 1997.

[257] I. Wollny, W. Sun, and M. Kaliske. A hierarchical sequential ale poromechanics

model for tire-soil-water interaction on fluid-infiltrated roads. *International Journal for Numerical Methods in Engineering*, 112(8):909–938, 2017.

[258] D. M. Wood. *Soil behaviour and critical state soil mechanics*. Cambridge university press, 1990.

[259] H. Wu, N. Guo, and J. Zhao. Multiscale modeling and analysis of compaction bands in high-porosity sandstones. *Acta Geotechnica*, 13(3):575–599, 2018.

[260] S. Wulfinghoff, F. Cavaliere, and S. Reese. Model order reduction of nonlinear homogenization problems using a hashin–shtrikman type finite element method. *Computer Methods in Applied Mechanics and Engineering*, 330:149–179, 2018.

[261] X.-P. Xu and A. Needleman. Numerical simulations of fast crack growth in brittle solids. *Journal of the Mechanics and Physics of Solids*, 42(9):1397–1434, 1994.

[262] N. N. Yanenko. *The method of fractional steps*. Springer, 1971.

[263] Y. Yang and P. Perdikaris. Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics*, 394:136–152, 2019.

[264] M. Yoshimine, K. Ishihara, and W. Vargas. Effects of principal stress direction on intermediate principal stress on undrained shear behavior of sand. *Soils and Foncations*, 38(3):179–188, 1998.

[265] J. Yvonnet and Q.-C. He. The reduced model multiscale method (r3m) for the non-linear homogenization of hyperelastic media at finite strains. *Journal of Computational Physics*, 223(1):341–368, 2007.

[266] M. J. Zahr, P. Avery, and C. Farhat. A multilevel projection-based model order reduction framework for nonlinear dynamic multiscale problems in structural and solid mechanics. *International Journal for Numerical Methods in Engineering*, 2017.

[267] J. Zhao and N. Guo. Unique critical state characteristics in granular media considering fabric anisotropy. *Géotechnique*, 63(8):695–704, 2013.

[268] H. Zhu, Z. Zhou, R. Yang, and A. Yu. Discrete particle simulation of particulate systems: theoretical developments. *Chemical Engineering Science*, 62(13):3378–3396, 2007.

[269] J.-H. Zhu, M. M. Zaman, and S. A. Anderson. Modeling of soil behavior with a recurrent neural network. *Canadian Geotechnical Journal*, 35(5):858–872, 1998.

[270] O. Zienkiewicz and Z. Mroz. Generalized plasticity formulation and applications to geomechanics. *Mechanics of engineering materials*, 44(3):655–680, 1984.

[271] O. C. Zienkiewicz, A. Chan, M. Pastor, B. Schrefler, and T. Shiomi. *Computational geomechanics*. Wiley Chichester, 1999.

[272] T. I. Zohdi. Rapid simulation of laser processing of discrete particulate materials. *Archives of Computational Methods in Engineering*, 20(4):309–325, 2013.