

# Machine Learning-Based Constitutive Modelling for Granular Materials



Prifysgol Abertawe  
Swansea University

**Shaoheng Guan**

School of Aerospace, Civil, Electrical, General and Mechanical Engineering  
Swansea University

Submitted in fulfilment of the requirements for the degree of  
*PhD*

Sep. 22, 2023





## Dedication

Dedicated to my dearest wife, Ma Yukun,

Our journey began in Swansea, UK, where we crossed paths. From those early days to the final stages of my doctoral thesis and our shared moments in Graz, Austria, each step has been illuminated by your presence. Together, we strolled along Swansea's coastline, savoured the taste of fish and chips in Tenby, and listened to the enchanting tunes of Scottish bagpipes.


In times of doubt, you've remained my constant and steadfast source of support. As I dedicate the accomplishment of my doctoral thesis, I'm also dedicating it to you. It symbolises not only my achievement but also the love and companionship we've shared on this incredible journey.

With boundless love,

Shaoheng Guan


## Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ...  .....


Date ... 14 Aug. 2024 .....

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed ...  .....


Date ... 14 Aug. 2024 .....

I hereby give consent for my thesis, if accepted, to be available for electronic sharing.

Signed ...  .....

Date ... 14 Aug. 2024 .....

The University's ethical procedures have been followed and, where appropriate, ethical approval has been granted.

Signed ...  .....

Date ... 14 Aug. 2024 .....

Shaoheng Guan  
Aug. 2023

## Acknowledgements

I want to convey my deepest appreciation to Prof. Feng Y.T., my supervisor, for his unwavering assistance, and endless patience throughout my doctoral voyage. He not only granted me considerable freedom in my research but also consistently provided me with useful and timely help whenever I encountered obstacles. Being a student of such a brilliant scholar and wise man was a true honour.

I would also like to express my gratitude to Prof. Zhou Wei and Prof. Ma Gang from Wuhan University, Dr Xiao Dunhui at Swansea Univeristy, and Dr Zhang Xue from Liverpool University for their valuable guidance and support.

My time in Swansea allowed me to forge meaningful friendships. I am deeply appreciative of individuals like Qu Tongming, Fu Jinlong, Wang Mengqi, Fu Rui, Li Yang, Liu Biao, Chai Yanjiang, Sun Guangshuai, Peng Linzhi, Li Zhanfeng, Zou Xi, Chen Bingbing, Nathan Ellmer, Yash and Agustina Felipe.

Heartfelt thanks go to my parents and my wife for their consistent support. Their love acts as the driving force that propels me forward.

I also want to acknowledge the beauty of Swansea's grassy landscapes, expansive sea, and captivating blue skies – elements that contributed to an enriching experience.

## Abstract

As a material second only to liquids in nature, granular materials are widely used in hydraulic structures, roads, bridges etc. Dam-building granular materials are complex systems of pore structures and continuously graded rock particles. An accurate description of their mechanical properties is essential for the safety analysis of ultra-high rockfill dams. At the microscopic scale, granular materials are discrete elementary systems aggregated by complex internal interactions, and their microscopic mechanical structure and statistical characteristics influence the macroscopic mechanical properties; at the macroscopic scale, especially in engineering-scale computational analysis, granular materials are often regarded as continuous media and their constitutive relationship are described using non-linear or elastic-plastic theories. Yet, there is no unified theory to characterise all their constitutive properties.

Constitutive modelling stands as a pivotal topic within mechanical calculations. Establishing an accurate description of the relationship between deformation and constitutive response serves as the foundation for Boundary Value Problem analysis. With the growing prominence of machine learning techniques in the data-driven realm, they are expected to enhance constitutive modelling and potentially surpass classical models based on simplifying assumptions. More and more endeavours have been dedicated to integrating machine learning into mechanical calculations and assessing its efficacy.

This PhD thesis focuses on the use of machine learning techniques to investigate the feasibility of developing a constitutive model for granular materials and applying it in boundary value problem calculations. The main areas of research include the following aspects:

1. In Chapter 2, we introduce a deep learning model designed to reproduce the macroscopic mechanical response of granular materials across various particle size distributions (PSDs) and initial states, considering different loading conditions. We start by extracting stress-strain data from massive DEM simulations and then proceed to capture the mechanical behaviour of these granular materials through the Long Short-Term Memory networks. The work contains three central issues: LSTM cell customisation,

granular materials stress-strain sampling, and loading history pasteurisation. The validation results demonstrate that this deep learning model achieves good generalisation and a high level of prediction accuracy when tested on the true triaxial loading dataset.

2. For the different loading and unloading paths in the conventional triaxial simulation of the DEM, an Active Learning approach is introduced to guide the sampling (Chapter 3). Based on the positive correlation between the prediction error and the uncertainty given by activate learning method, the strain paths are evaluated without DEM simulations, from which the worst predicted paths are selected for sampling. To prevent data redundancy, points in the vicinity of one selected point will not be selected for the current resampling round. The model was trained on single-cyclic loading datasets and performed quite well under multiple-cyclic loading paths.
3. In order to circumvent the reliance on phenomenological assumptions in boundary value problem analysis, a computational framework coupled with FEM and neural network (FEM-NN) is proposed (Chapter 4). Building on the work in Chapter 2 and 3, we further introduce FEM-DEM multiscale simulations by employing the Random Gaussian Process to generate macroscopic random loading paths to be applied to the macro-scale model. A large amount of stress-strain data on the integration points is collected. Part of them are subsequently, used to train the neural network. Material loading histories represented by encoded variables. Active learning is employed here again to assess the informativeness of the data points, according to which the points are resampled from the massive database. Two examples are provided to demonstrate the effectiveness of the implemented framework which provides considerable improvements in computational efficiency and the ability to reproduce the mechanical response of granular materials at the macroscopic scale.
4. In Chapter 5 the trained network-based constitutive model is embedded into the explicit FEM solver. In implicit FEM solvers for non-linear static problems, a global equilibrium solution is typically obtained via Newton-Raphson iteration. However, the non-linear iterations may not converge when the predicted tangential matrix is not accurate enough. Therefore, the explicit FEM solver is employed to circumvent non-linear iteration. The network is trained and investigated on data generated from two constitutive models (IME model and CSUH model) separately. The trained network is able to reproduce almost exactly the ground truth results at the macroscopic level. However, the error accumulation problem resulting from a large number of steps is an-

other challenge to the prediction accuracy and robustness of the data-driven model. A check-and-revision method is proposed to iteratively optimise the model by expanding the training range and improving the network generalisation.

5. Chapter 6 focuses on evaluating the capacity and performance of a network-based material cell with physics extension against boundary-value problems. The proposed material cell aims to reproduce constitutive relationships learned from datasets generated by random loading paths following random Gaussian Process. The material cell demonstrates its effectiveness across three progressively complex constitutive models by incorporating physics-based basis functions as prior/assumptions. An adaptive linear transformation is introduced to mitigate the error caused by magnitude gaps between strain increments in training sets and finite element simulations. The material cell successfully reproduces constitutive relationships in FEM simulations, and its performance is comprehensively evaluated by comparing two different material cells: the sequentially trained gated recurrent unit (GRU)-based material cell and the one-to-one trained deep network-based material cell. The GRU-based material cell can be trained without prior knowledge about the internal variables. Consequently, this enables us to directly derive the constitutive model using stress-strain data obtained from experiments.
6. A universal constitutive model has been introduced, combining the recurrent machine learning structure with traditional constitutive models in Chapter 7. A dramatic drop in prediction accuracy emerges when the input strain exceeds the training space because of the poor generalisation ability of the purely data-driven method. Therefore, we introduce the widely accepted elasticity theory, yielding, hardening and plastic flow as physical constraints to build a machine learning-based universal constitutive model. These constraints serve as priors/assumptions for the machine learning model. During the sample preparation stage, they alleviate the stringent demands for the completeness of data sampling. In the model calculations, they guide the model to make predictions, even for unseen loading paths. The proposed model has been calibrated and tested with FEM-DEM datasets.

# Contents

<b>List of Figures</b>	<b>VI</b>
<b>List of Tables</b>	<b>XIV</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Constitutive models of granular materials . . . . .	2
1.2 The phenomenological constitutive model . . . . .	2
1.3 Hierarchical Multi-scale modelling, HMM . . . . .	4
1.4 ML in mechanical computation . . . . .	5
1.4.1 A brief introduction of the machine learning method . . . . .	6
1.4.2 ML in computational mechanics . . . . .	7
1.5 Challenges . . . . .	10
<b>2 A predictive framework for the path-dependent mechanical response of multi-graded granular assemblies</b>	<b>11</b>
2.1 Particle size distribution (PSD) of the granular assembly . . . . .	12
2.1.1 Loading history dependency . . . . .	13
2.1.2 Training data set preparation: DEM simulations . . . . .	14
2.1.3 Strain and stress of particle assembly . . . . .	17
2.1.4 Quasi-static loading rate . . . . .	18
2.1.5 Critical step size in time integration . . . . .	19
2.2 Methodology for DL-based mechanical response prediction . . . . .	20
2.2.1 Challenge of the recurrent network unit: gradient explosion or fading	20
2.2.2 Modified LSTM considering the initial state . . . . .	23
2.2.3 Extracting sequences of training sets via sliding window . . . . .	25
2.2.4 Network model training . . . . .	27

2.3	Model validation: stress and void ratio prediction . . . . .	29
2.3.1	On different PSDs . . . . .	29
2.3.2	On different initial void ratios . . . . .	32
2.3.3	On different loading paths . . . . .	34
2.4	Concluding remarks . . . . .	37
<b>3</b>	<b>Deep active learning for constitutive modelling of granular materials</b>	<b>39</b>
3.1	Introduction . . . . .	39
3.2	A deep active learning strategy for constitutive modelling . . . . .	41
3.3	A comprehensive examination of active learning-assisted data-driven constitutive modelling based on a data pool . . . . .	44
3.3.1	The adopted GRU neural network and accuracy evaluation . . . . .	45
3.3.2	Examination and verification of the role of active learning in constitutive modelling . . . . .	48
3.3.3	Batch-mode active learning scheme . . . . .	49
3.3.4	Prediction performance of three training cases . . . . .	50
3.3.5	Active learning-informed data preparation . . . . .	51
3.4	Interactive constitutive training and data labelling through active learning . . . . .	52
3.4.1	Strain path falsification . . . . .	52
3.4.2	A surrogate error indicator for data-driven constitutive modelling . . . . .	54
3.4.3	Examination of the whole domain and adaptive resampling: an example with varied unloading-reloading cycles . . . . .	55
3.4.4	Verification of the interactive learning strategy . . . . .	57
3.5	Discussion . . . . .	60
3.5.1	Significance of active learning . . . . .	60
3.5.2	Limitation of active learning and future work . . . . .	62
3.6	Concluding remarks . . . . .	63
<b>4</b>	<b>An FEM-NN framework for accelerating the multi-scale computation</b>	<b>65</b>
4.1	FEM-DEM . . . . .	65
4.1.1	Marco solver: FEM . . . . .	66
4.1.2	Lower-scale solver: DEM . . . . .	69
4.2	FEM-NN: neural network-based multi-scale method . . . . .	71
4.2.1	Neural network modelling of micro-RVE response . . . . .	73
4.2.2	Training samples preparation . . . . .	77



4.2.3	Active learning-based resampling . . . . .	80
4.2.4	FEM-NN coupling . . . . .	83
4.3	Numerical examples . . . . .	85
4.3.1	Bixial compression . . . . .	86
4.3.2	Retaining wall . . . . .	93
4.4	Dicussion . . . . .	95
4.4.1	Limitations of Active Learning . . . . .	95
4.5	Concluding remarks . . . . .	95
<b>5</b>	<b>An explicit FEM-NN framework and analysis of error caused by NN-</b>	
	<b>predicted stress</b>	<b>98</b>
5.1	Methodology of the explicit FEM . . . . .	98
5.1.1	Governing equation and solving process . . . . .	98
5.1.2	Stable time increment . . . . .	99
5.1.3	Damping . . . . .	100
5.2	Constitutive model and the network-constitutive model . . . . .	101
5.2.1	Classical constitutive model . . . . .	102
5.2.2	Neural network-based constitutive model . . . . .	103
5.2.3	A novel MLP network enhanced by Fourier feature mask and the multiplied residual block . . . . .	103
5.2.4	Check-and-revision method . . . . .	105
5.3	Numerical simulations . . . . .	107
5.3.1	Biaxial compression . . . . .	107
5.3.2	Retaining wall . . . . .	113
5.3.3	Rigid strip footing . . . . .	115
5.4	Stability of the netowrk-based exFEM computation . . . . .	119
5.4.1	The emergence and propagation process of errors in the FEM calculation	120
5.4.2	Stability analysis of NN-based predictions after adding noise . . . . .	121
5.5	Discussion . . . . .	122
5.5.1	Challenges in network-based constitutive model development . . . . .	122
5.5.2	Possible development . . . . .	123
5.5.3	Improvement in calculation efficiency . . . . .	124
5.6	Concluding remarks . . . . .	125
<b>6</b>	<b>Recurrent network-based constitutive model</b>	<b>126</b>

6.1	Introduction . . . . .	126
6.2	Methodology . . . . .	130
6.2.1	Governing equations and explicit FEM solver . . . . .	130
6.2.2	Material cell . . . . .	131
6.2.3	Loading path generating via random Gaussian Process . . . . .	134
6.3	Material cell without physics: training and testing . . . . .	136
6.4	Material cell with physics extensions: training, test and FE analysis . . . . .	138
6.4.1	Adaptive step size adjustment . . . . .	139
6.4.2	Utilisation of the symmetry . . . . .	141
6.4.3	Techniques for physical extensions . . . . .	141
6.4.4	Pressure independent material behaviour: $J_2$ model . . . . .	144
6.4.5	Pressure dependent material behaviour: Drucker-Prager model . . . . .	146
6.4.6	Plastic hardening behaviour: $J_2$ -harden model . . . . .	149
6.5	Concluding remarks . . . . .	157
6.6	Appendix . . . . .	159
6.6.1	Regression via Gaussian Process . . . . .	159
6.6.2	Classic constitutive model . . . . .	160
6.6.3	Tensor transformation . . . . .	162
<b>7</b>	<b>A universal machine learning-based material cell</b>	<b>163</b>
7.1	Introduction . . . . .	163
7.2	Components of the elastoplastic constitutive model . . . . .	163
7.2.1	Yield function . . . . .	164
7.2.2	Hardening function . . . . .	165
7.2.3	Plastic flow rule . . . . .	165
7.3	Enhanced IME model . . . . .	167
7.3.1	Original IME model . . . . .	167
7.3.2	Enhance the original model . . . . .	169
7.4	CSUH model . . . . .	170
7.4.1	Yield surface and hardening function . . . . .	170
7.4.2	Current state representation . . . . .	171
7.4.3	Influence of the medium principal stress ratio . . . . .	174
7.5	Specificity of mechanical properties of granular materials . . . . .	175
7.6	Optimisation of constitutive models based on the datasets collected from exFEM-DEM simulations . . . . .	176

7.6.1	Baseline: $J_2$ model . . . . .	178
7.6.2	Enhanced IME model . . . . .	179
7.6.3	CSUH model . . . . .	181
7.7	Concluding remarks . . . . .	189
<b>8</b>	<b>Conclusion</b>	<b>191</b>
	<b>Bibliography</b>	<b>196</b>

# List of Figures

2.1	(a) Five hypothetical PSDs/GSD (grain size distributions) [1], (b) Different metrics of the five mixtures. . . . .	13
2.2	Loading paths included in training sample preparation: (a) constant- $\sigma_3$ -constant- $b$ loading and constant- $p$ -constant- $b$ loading, (b) cyclic loading, and (c) strain-controlled random loading. Note: $\sigma_3$ corresponds to the principal stress along the third axis (or $z$ -direction), while the subscripts 1-2-3 and x-y-z are used interchangeably to denote principal axes and coordinate directions, respectively. . . . .	18
2.3	Influence of inertial number $I$ on (a) stress ratio $\eta$ : as $I$ decreases to $2.5e-3$ , it converges to around 0.7; (b) volumetric strain: converges to 0.8% after $I$ decreases to $2.5e-3$ ; and (c) deviatoric fabric. [2] . . . . .	19
2.4	Architecture of the recurrent neural network . . . . .	20
2.5	The modified LSTM unit . . . . .	23
2.6	Comparison of prediction errors (a) classical LSTM cell; (b) mLSTM cell . . . . .	24
2.7	The sliding window to extract data sequences . . . . .	25
2.8	Validation error with different sliding window sizes . . . . .	26
2.9	(a) Input and output of the network model. (b) Network structure. . . . .	27
2.10	Training and validation loss during the training process . . . . .	28
2.11	Predicted macroscopic mechanical responses with different PSDs under constant- $\sigma_3$ -constant- $b$ loading paths: (a) (top) fractal function control graded granular material aggregate ( $\beta = -5.0$ , $I_G = 1.016$ , initial void ratio $e_0 = 0.508$ ), (bottom) binary mixed granular aggregate ( $I_G = 1.117$ , initial void ratio $e_0 = 0.508$ ). (b) and (c) Comparison of depth-learning predictions (solid lines) and DEM simulations (hollow points) for the stress and pore ratio curves, respectively. . . . .	30

2.12	Predicted macroscopic mechanical responses with unseen PSD ( $\beta = 1.5$ and $I_G = 1.290$ ) under constant- $p$ -constant- $b$ loading paths. . . . .	31
2.13	Predicted macroscopic mechanical responses with different initial void ratio $e_0$ under conventional triaxial compression . . . . .	32
2.14	Predicted macroscopic mechanical responses with different initial void ratio $e_0$ under conventional triaxial compression . . . . .	33
2.15	(a) Relative prediction error of shear stress $q$ for specimens with different initial void ratios $e_0$ : the relative error increases as the initial pore ratio increases. (b) Poorer prediction results for the specimen with a large void ratio under constant- $\sigma_3$ -constant- $b$ loading . . . . .	34
2.16	Macromechanical response predictions for true triaxial conditional constant- $p$ -constant- $b$ loading path (mean stress $p = 2.00\text{MPa}$ ): (a) particle specimen with the linear distribution of particle size ( $I_G = 1.064$ ) (b) and (c) comparison of depth learning predictions (solid line) and DEM simulations (hollow points) for stress and pore ratio curves, respectively . . . . .	35
2.17	Macromechanical response predictions for the true triaxial conditional constant- $\sigma_3$ -constant- $b$ loading path (small principal stress $\sigma_3 = 2.00\text{MPa}$ ): (a) particle specimen with linear particle size distribution ( $I_G = 1.064$ ) (b) and (c) comparison of depth-learning predictions (solid lines) and DEM simulations (hollow points) for the stress and pore ratio curves, respectively . . . . .	35
2.18	Network for predicting the distribution of peak and critical state stresses in the $\pi$ plane under constant- $p$ -constant- $b$ loading paths. . . . .	36
2.19	Macroscopic mechanical response prediction of granular material under random strain loading: (a) the particle assembly with fractal function controlled grading ( $\beta = 2.9$ , $I_G = 1.478$ , $e_0 = 0.187$ ); (b) and (c) network prediction (solid line) and DEM simulation (hollow point) comparison of stress curve and void ratio curve, respectively . . . . .	37
2.20	Comparison of the predicted macroscopic mechanical response of granular materials under cyclic loading and unloading with an initial enclosing pressure $\sigma_3 = 0.50\text{MPa}$ . (a) Fractal function control grading of granular material specimens ( $\beta = 1.0$ , $I_G = 1.209$ , $e_0 = 0.375$ ) (b) and (c) Comparison of depth-learning predictions (solid lines) and DEM simulations (hollow points) of stress and porosity ratio curves, respectively . . . . .	38
3.1	Procedures of deep active learning <sup>1</sup> . . . . .	43

3.2	Specimen distribution used for preliminary training . . . . .	45
3.3	Basic active learning procedures for Case 1 and Case 2 . . . . .	46
3.4	Learning curves during training . . . . .	47
3.5	Illustration of data resampling in a batch-mode active learning scheme . . . . .	50
3.6	Learning curves during training . . . . .	51
3.7	The worst predictions given by a DNN model in Case 1 with 60 groups of training data . . . . .	52
3.8	The worst predictions given by a DNN model in Case 3 with 60 groups of training data . . . . .	53
3.9	The added data specimens via active learning in each round . . . . .	54
3.10	Inputs and outputs for data-driven stress-strain modelling in a conventional triaxial testing condition . . . . .	55
3.11	The workflow of global domain examination and resampling strategy . . . . .	57
3.12	Sampling space for training and examination . . . . .	58
3.13	The worst forecasts discovered in the sixth round of active learning . . . . .	61
3.14	The newly added data specimens in each active learning round . . . . .	62
4.1	Flowchart of FEM-DEM multi-scale coupling calculation . . . . .	67
4.2	The shear stress with different numbers of particles in the assembly [3] . . . . .	70
4.3	RVEs with different numbers of particles [4] . . . . .	72
4.4	Comparison of network prediction under normal and Sobolev training [5] . . . . .	75
4.5	Comparison between tangent (green) and secant (red) matrices on the stress-strain curve . . . . .	75
4.6	Network architecture . . . . .	76
4.7	Gaussian processes corresponding to different covariance matrices are shown. Figures (a) and (b) display the covariance matrices. Figures (c) and (d) illustrate a series of random values sampled from Gaussian processes based on the covariance matrices in (a) and (b), respectively. In (c) and (d), the x-axis represents the points index (normalized to 0-1), while the y-axis represents their values. . . . .	79
4.8	Random paths generated by the Gaussian process for $v_c$ . The left column is the kernel and the right column is the generated random paths. . . . .	81
4.9	Comparison of neural network prediction results with DEM simulation . . . . .	84
4.10	Biaxial compression simulation with different mesh densities . . . . .	86
4.11	Mixed data set generation programme . . . . .	87

4.12	Curves of top forces and global volumetric strain corresponding to different Cases. . . . .	88
4.13	Comparison of displacement results corresponding to different Cases . . . . .	89
4.14	Flowchart depicting the process of network training and FEM-NN simulation using active learning for resampling. . . . .	90
4.15	The shear strain-based active learning resampling at seven load steps: Top row – 30% of the data points with highest uncertainty levels (the number in red is the average uncertainty); Bottom row – equivalent shear strain distribution. . . . .	91
4.16	Comparison of the local strain and stress responses from different solution schemes . . . . .	92
4.17	The number of iterations required for the FEM-DEM and FEM-NN simulations with the three levels of mesh . . . . .	94
4.18	Schematic of the retaining wall simulation . . . . .	94
4.19	Displacement distributions of the soil at some load steps when compressed by the retaining wall and obtained by (a) FEM-DEM, (b) FEM-NN 1, and (c) FEM-NN 2 . . . . .	96
4.20	Comparisons of integrated wall forces . . . . .	96
5.1	Contrasting various $\gamma$ values to optimise damping coefficient selection . . . . .	101
5.2	Architecture of the neural network used to reproduce the stress tensor . . . . .	104
5.3	Fourier feature mask . . . . .	104
5.4	Multiplied residual block in the network architecture . . . . .	105
5.5	(a) Hyperparameters (layer type, layer number and the node number) sensitivity analysis. The dot-dashed curves represent the results with multiplied residual blocks; (b) Comparison of results with and without the Fourier mask (c) The normalised distribution of the network prediction on the 1:1 line . . . . .	106
5.6	Schematic of the on-the-fly check-and-revision method . . . . .	107
5.7	Biaxial compression simulation: (a) The model discretisation and boundary conditions; (b) Comparison of the shear-strain field; (c) Comparison of the top force, global volume strain and maximum nodal acceleration . . . . .	108
5.8	Curves of stresses belonging to Gauss points with datasets collected from the IME model simulations . . . . .	109

5.9	Comparison between the CSUH-based and NN-based exFEM simulations: (a) The top force, global volume strain and the maximum acceleration; (b) Shear strain field . . . . .	110
5.10	Stress–strain responses at representative Gauss points with datasets collected from the CSUH-based simulations . . . . .	111
5.11	The exFEM-NN computation results with the check-and-revision iterations: (a) Strain field and (b) top force, total volume strain and maximum acceleration curves after three and six iterations of check-and-revision; (c) evolution of the number of Gauss points whose error of stress prediction greater than 50% with loading step . . . . .	112
5.12	Stress curves on the Gauss points after six iterations of check-and-revision .	113
5.13	Retaining wall simulations with CSUH and NN-based model: (a) Discretisa- tion and boundary condition; (b) Curves of the reaction force, macroscopic volumetric strain and the maximum acceleration; (c) Displacement field . . .	114
5.14	Rigid strip footing simulation results with CSUH and NN-based model (a) Discretisation and boundary condition; (b) Curves of the footing reaction force, global volumetric strain and maximum acceleration; (c) Evolution of the number of error points with loading steps . . . . .	115
5.15	Error points selected at the last load step in each check-and-revision iteration	116
5.16	Shear strain field with different number of check-and-revision iterations . . .	117
5.17	The strain-stress curves of four Gauss points of the rigid strip footing simulation	118
5.18	Evolutaion of the stress error during loading . . . . .	120
5.19	Network prediction results after introducing different levels of $\xi$ (0, 0.2, 0.5, 1.0) . . . . .	121
6.1	(a) Schematic of the material cell $\mathcal{M}$ . (b) Tensor flow in the recurrent neural network cell. . . . .	132
6.2	The sequential and one-to-one mapping method . . . . .	133
6.3	Finite training paths and infinite possible loading paths . . . . .	134
6.4	Three examples of the random loading paths. The curves represent each of the three components of the strain tensor in the two-dimensional case. . . . .	136
6.5	$\mathcal{M}_{NN}$ consists of two layers of GRU and a fully connected (FC) layer and (b) the loss evaluation with different numbers of training sets, where the solid line indicates the training loss and the dots represent the validation loss. . .	137



6.6	Results of the two-layer GRU model trained on data generated based on $J_2$ model (a) Prediction error distribution. The vertical axis indicates the sample distribution's density, with the curve representing an area integral equal to 1. . . . .	139
6.7	Distribution of the norm of the strain increment . . . . .	140
6.8	Results of the material cell without physical extensions trained on datasets generated with $J_2$ model: only $\sigma$ as the internal $\mathcal{I}$ (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions. . . . .	143
6.9	The material cell with physics extensions. . . . .	145
6.10	Training loss evaluation: material cell with physics extension trained on datasets generated via $J_2$ model. . . . .	145
6.11	Predictions of material cell with physical extensions under $J_2$ model. (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions. . . . .	145
6.12	The meshes and boundary conditions of the biaxial compression simulation .	146
6.13	Biaxial monotonic compression: $J_2$ model. (a) and (b) are the equivalent von-Mises stress of $J_2$ model and material cell (with adaptive step size adjustment) simulations, respectively. (c) The curves of top force and the global volumetric strain for different cases, involving the ground truth $J_2$ model, the material cell (Fig. 6.9) with $s = 1$ , $s = 60$ , and adaptive step size adjustment. MC in the legend denotes material cell. The top force compresses the specimen in a downward direction. A volumetric strain less than zero indicates that the volume is undergoing compression. . . . .	147
6.14	Training loss of material cell without/with $p$ under Drucker-Prager model . .	147
6.15	Predictions of material cell after training with $p$ included. (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions. . . . .	148
6.16	Biaxial simulation: Drucker-Prager model and the material cell shown in Fig. 6.9. (a) and (b) are the mean stress results of Drucker-Prager model and material cell (with adaptive step size adjustment) simulations, respectively. (c) The curves of top force and the global volumetric strain for different cases. MC in the legend denotes material cell. . . . .	148

6.17	The material cell with a single layer of GRU: $J_2$ -harden model (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions. . . . .	149
6.18	Material cell with implicit internal variables. . . . .	151
6.19	The material cell with implicit internal variables: $J_2$ -harden (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions. . . . .	152
6.20	Results of biaxial calculations in FEM with the material cell based on the implicit internal variable (Fig. 6.18). MC in the legend denotes material cell.	153
6.21	Dog bone stretching simulation: $J_2$ -harden model with the material cell shown in Fig. 6.18. MC in the legend denotes material cell. The top force pulls the dog bone upward. In the second subplot, the global volumetric strain $\epsilon_v$ is summarised over the simulated domain. In the subplot of the third row, $a$ denotes the acceleration of nodes in the FEM simulation. . . . .	154
6.22	Stretching simulation with the quarter perforated plate: $J_2$ -harden model with the material cell shown in Fig. 6.18. MC in the legend denotes material cell. Tension is considered positive, while compression is considered negative.	155
6.23	Comparison of the computational time required for biaxial simulations using the material cell and three conventional constitutive models . . . . .	155
6.24	The DNN-based material cell . . . . .	156
6.25	DNN-based material cell: $J_2$ -harden model (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions. . . . .	156
6.26	Results (top force and global volumetric strain) of biaxial simulations with the DNN-based material cell (Fig. 6.24) in multiple cyclic loading. (a): $J_2$ yield surface with ideal plasticity; (b) Drucker-Prager yield surface with ideal plasticity and (c) $J_2$ yield surface with hardening plasticity. MC in the legend denotes material cell. . . . .	157
7.1	Yield surfaces in stress space. (a) Tresca yield surface; (b) von Mises yield surface (c) Drucker-Prager Yield surface; (d) Mohr-Coulomb yield surface; (e) Spatial mobilised plane; (f) modified Cam-Clay surface. . . . .	164
7.2	The reference and current yield surface of the CSUH model [6] . . . . .	170
7.3	Normal consolidation line (NCL), anisotropic compression line (ACL), and the critical state line (CSL) on the $e - \ln p$ plane . . . . .	172

7.4	Shear stress ratio according to different kinds of Hvorslev envelope [7] . . . . .	173
7.5	The logarithmic coordinate ICL (Isotropic compression line) of Cambria sand and the asymptote on the logarithmic coordinate [8]; (b) The two forms of the isotropic compression line on the logarithmic coordinate with $N = 1.9$ , $Z = 0.9$ .	173
7.6	Mechanical responses on Gauss points in exFEM-DEM biaxial simulations (a) Shear stress and shear strain relationship; (b) Difference in angle between strain tensor and stress tensor $\theta_\epsilon - \theta_\sigma(^{\circ})$ . . . . .	176
7.7	Mechanical responses on Gauss points in CSUH model-based biaxial simulations (a) Shear stress and shear strain relationship; (b) Difference in angle between the principal direction of strain and stress tensor $\theta_\epsilon - \theta_\sigma(^{\circ})$ . . . . .	177
7.8	Constitutive model calculation process, from the strain sequence to the stress sequence . . . . .	178
7.9	Optimising process of the $J_2$ model. (a) The calculation error. (b) The trainable model parameters. . . . .	179
7.10	Evolution of the loss during the optimisation process: enhanced IME model .	180
7.11	Predicted $\sigma_{yy}$ curves after optimisation of Enhanced IME model. . . . .	181
7.12	Predicted stress component $\sigma_{yy}$ of the CSUH model after optimisation on the data sets collected from exFEM-DEM footing simulation. . . . .	184
7.13	Test on data sets collected from the biaxial simulation: predicted stress component $\sigma_{yy}$ after the CSUH model optimised on the footing simulation. The test data sets are from the exFEM-DEM biaxial simulation. . . . .	185
7.14	Macroscopic results of biaxial compression based on the optimised CSUH model	186
7.15	Comparison of the constitutive response at integration points of the biaxial compression simulation. The $x$ coordinate is the strain in the vertical direction.	188
7.16	Macroscopic results of the optimised CSUH model in the retaining simulation	189

# List of Tables

2.1	Details of the particle assemblies . . . . .	17
2.2	Summary of the loading paths . . . . .	17
2.3	Summary of the critical time steps for different particle assembles . . . . .	20
2.4	Summary of network architecture and optimiser configuration . . . . .	29
3.1	The adopted network architecture and some key hyperparameters . . . . .	47
3.2	The first-round active learning-assisted forecasts and verifications for Case 1	48
3.3	The prediction performance of the selected ten samples with the greatest standard deviations but extracting neighbouring points . . . . .	59
3.4	The prediction performance on the ten specimens with the smallest standard deviations . . . . .	60
4.1	Summary of datasets from FEM-DEM simulations . . . . .	82
4.2	Parameters of the lower-scale DEM simulations . . . . .	86
4.3	Summary for the different training and testing cases . . . . .	87
4.4	Efficiency improvement . . . . .	93
5.1	Parameters of the CSUH model . . . . .	102
5.2	Summary of time consumed in different simulations with different constitutive models . . . . .	124
7.1	Material constants of IME model . . . . .	168
7.2	The optimised material constants of the enhanced IME model . . . . .	180
7.3	The CSUH parameters optimised on the exFEM-DEM footing simulation . .	183

# Chapter 1

## Introduction

Since Hooke's Law, there have been many attempts to develop constitutive models to accurately describe the relationship between material deformation and mechanical stress  $\epsilon_{ij} \rightarrow \sigma_{ij}$ . From the last century to the present, the emergence of a wide range of constitutive models has seen remarkable development in this field. As a mathematical approximation of material behaviour, constitutive relationships are the basis not only for understanding the mechanical properties but also for performing macroscopic numerical calculations (e.g. by FEM).

Granular materials, such as sand, powder, and foam, are ubiquitous in nature as well as in industrial production and geotechnical applications, and they are considered to be the second most abundant material on earth after liquids [9]. Typical granular materials, such as rock piles and sand, are collections of different microscopic particles, and the outstanding feature of granular materials deforming under external loads is the complex evolutionary process of particle contact. The diversity and dissipation of particle contact have evolved in loading, giving rise to a rich and complex mechanical behaviour of granular materials, which makes them very different from continuous solids, liquids, and gases. For researchers and engineers, the development of accurate ontological models of granular materials that take into account the various properties that granular materials have in macroscopic mechanics, including state/history dependence [10], dilatancy [11], non-coaxially, strain localisation, loading liquefaction characteristics, critical states, strength, and structural anisotropy [12, 13], is a prerequisite for accurate numerical simulations.

## 1.1 Constitutive models of granular materials

Unlike continuous media, the strength and stiffness of granular materials are largely influenced by the ground of inter-particle action: the load on the granular material is supported by the contact network between the particles or rather the distribution of the load to each particle through the contact network. With each slip of the inter-particle contact, the particle aggregate undergoes (either large or small) irrecoverable plastic deformation, which leads to changes in the strength and stiffness characteristics of the particle aggregate [6]. Due to the complexity of the particle contact network, these simple inter-particle slips will result in a particle aggregate with a rather complex mechanical behaviour at the macroscopic level.

Within the framework of continuum mechanics, granular materials are usually considered homogenised continua, without taking into account their discrete nature and micro structure. For granular materials, the macroscopic mechanical properties should be describable by particle-scale parameters such as size and shape distribution, contact fabric coefficients and void ratios [14]. After the advent of DEM [15] simulation methods, researchers were able to get their hands on the relationships of complex particle contact configurations and the evolution during loading for the first time, and a series of research works on this were initiated [15–17]. The hope was to include the contact micro-structure in the constitutive model and to build a model that could take into account the macroscopic mechanical properties. However, building a highly accurate predictive constitutive model from the particle-level analysis is challenging due to the complex properties of the particle assemblies. A large amount of work has been devoted to the study of the relationship between macroscopic mechanics and microscopic properties and structure using experimental or DEM simulation type approaches [4, 18–31]. However, most of the work can only provide a qualitative description of the mechanical properties of the material. The study of the statistics of particle contact networks and their evolutionary processes [32–35] has inspired modelling approaches related to the contact fabric tensor of granular materials [36–38].

## 1.2 The phenomenological constitutive model

Constitutive modelling is one of the most important and challenging studies in studying granular materials. As early as the 1780s the French scholar Charles-Augustin de Coulomb published a paper on the frictional interaction between two particles leading to their static stability. In 1882, the German civil engineer Christian Otto Mohr introduced Mohr's Circle

in his study of stresses, which describes the strength of a material based on shear stress. The research was later referred to as the famous Mohr-Coulomb criterion,  $\tau = \sigma \tan \phi + c$ . The theory has been used to describe the strength of frictional materials to this day. Among the granular materials, rockfills and sands are typical of frictional granular materials.

Common constitutive models for granular materials include the non-linear elastic model and the phenomenological elastoplastic model. In parallel with the development of the elastoplastic model, Kondner [39] used the hyperbolic function to describe the stress-strain nonlinear relationship. Based on this, the Duncan-Chang non-linear elastic model was developed [40] in the 1970s, which has been widely accepted and used in engineering simulations until today.

The success of the elastoplastic constitutive model is based on three main hypotheses: (1) the yield surface, which is used to distinguish between elastic and plastic deformation; (2) the hardening function (including isotropic, mobilising and mixed hardening), which is used to describe the shifting in the size and position of the yield surface after the plastic deformation of the material has occurred; and (3) the plastic potential function (or associated and non-associated flow rule), which is used to decide the direction of the plastic deformation after yielding the material into after yielding, in geomechanical materials, is mainly used to determine the dilation angle. Elastoplastic models have evolved considerably from the last century to the present.

The Critical state solid mechanics (CSSM) framework was proposed by the Cambridge soil mechanics research group represented by Roscoe in the 1950s [41–44]. The framework is able to approximate a frictional, volumetrically deformable material such as clay or sand. The critical state theory facilitates the model gradually forgetting its historical state and reaching a critical state after a sufficient shear loading. The normal compression line (NCL/ICL, Isotropic Compression Line) and critical state line (CSL) in this framework are the basis for subsequent critical state geomechanics.

To model overconsolidated clays, a series of models have been proposed based on the modified Cam-Clay model. By introducing a path-independent unified hardening parameter into the modified Cam-Clay model, Yao et al [45–47] proposed a Unified hardening (UH) model for overconsolidated clays. The UH model uses the same material parameters as the modified Cam-Clay model. For normally consolidated clays, the UH model degenerates to the modified Cam-Clay model. Poorooshasb et al [48] found that the plastic potential function does not coincide with the yield surface and therefore proposed the non-associated

flow rule to describe the elastoplastic behaviour of sand.

Been and Jefferies [49, 50] proposed a Nor-sand model for sandy soils based on the state variables related to the critical state concept. The highlight of the model is the state parameter describing the distance between the current state and the critical state. Li, Dafalias and Yang [37, 38, 51, 52] introduced the state variables into the dilatation equation by which the Anisotropic Critical State Theory (ACST) was developed.

Many works are proposed to establish a unified constitutive model for clay and sand via their similarities [47, 53–57]. Among them, the work of Yao et al. 2019 [57] analysed the mechanical properties of granular materials towards isotropic compression, shear yielding and dilatation, approximating them via mathematical formulas, and extended their proposed UH model for clay to sands with different initial void ratios (loose/dense sand), over-consolidation ratio to establish the CSUH (Critical state unified hardening) model. Due to the consideration of the transformed stress space [58], the loading in the CSUH model with different Lode angles has different critical stress ratio ( $M = q/p$ ), on the  $\pi$  plane at different distances from the hydrostatic pressure axis (considering the mid principal stress coefficient).

Meanwhile, Pastor, Zienkiewicz and Chan *et al.* proposed a generalised plasticity model [59–61], which is distinguished by the fact that the yield surface and hardening conditions do not need to be explicitly defined, and that the magnitude and direction of plastic deformation are determined by additional material parameters.

However, in some physical experiments [62–64] or low-scale numerical simulations [65], the simplifying phenomenological assumptions do not fit well enough with the material properties. Firstly, our limited knowledge of the material constitutive relationship highlights the need for the assumptions' further improvements. Likewise, in order to improve their generalisation ability or accuracy, the constitutive model is constantly adding model parameters. The more advanced the constitutive model, the more material parameters, the more difficult the calibration of the parameters becomes, and the more difficult to generalise in engineering.

### 1.3 Hierarchical Multi-scale modelling, HMM

Almost simultaneously, due to the development of computer hardware, a class of bottom-up modelling methods has emerged, namely Hierarchical Multi-scale Modelling (HMM), which uses a coupled FEM-DEM computational approach to directly relate the micro-structure and macroscopic boundary conditions of granular materials [4, 66–72]. In addition to the



macroscopic calculations using FEM, other researchers have used the Material point method (MPM) for macroscopic scale calculations and proposed a coupled MPM-DEM approach [73–75]. Compared to the FEM-DEM method, macro-scale calculations using MPM can mitigate the grid distortion by large deformations, such as slopes and snow piles [76–79], because the Lagrangian-Eulerian method in mesh and material points mapping.

In these FEM/MPM-DEM studies, macroscopic computations derive mechanical responses at Gaussian points directly from DEM RVE simulations. It is important to note that the homogenisation assumption is employed to compute stresses via the inter-particle contact fabric, averaging within the RVE where particle contact networks which may be unevenly distributed or experience strain localisation. Within the FEM/MPM-DEM framework, it is generally assumed that each Gaussian point corresponds to a particle-scale RVE, with the macro-scale strain serving as the boundary condition for its associated RVE. In MPM, due to the explicit solver (time integration), extensive RVE calculations are required, making it time-consuming and challenging to perform large-scale engineering simulations.

Liang et al. [75] combined the MPM with low-scale DEM simulation. They employed the Message message-passing interface for computation distribution across numerous nodes and CPUs. The DEM computation shows accelerated enhancement with an increase in computational nodes. Nevertheless, as the node count rises, the speedup gradually diminishes, eventually stabilising beyond 64 nodes. In their paper’s solid object intrusion example, involving 287,496 RVEs, the computation was distributed across 40 nodes (each equipped with 24 CPUs), consuming 39.6 hours. Multi-scale computation is notably time-intensive, and even with multiple hardware accelerations, the efficiency gains are limited by the overhead of node communication costs.

## 1.4 ML in mechanical computation

In addition to continuum mechanics modelling and HMM, with the development of machine learning methods, especially neural networks, some scholars have started to adopt neural networks to assist mechanics calculations, which not only include reproducing constitutive relationships but also other data-driven mechanics calculation methods were inspired. These studies provided new ideas for reproducing granular materials’ constitutive responses.

### 1.4.1 A brief introduction of the machine learning method

ML is a branch of artificial intelligence, a way to achieve artificial intelligence. In the last 30 years, machine learning has developed into a multi-disciplinary subject. ML research focuses on the design and analysis of algorithms that allow programs to "learn" automatically. ML algorithms are a class of algorithms that automatically analyse data to obtain patterns and use the patterns to make predictions about unknown data. ML can be divided into supervised learning, common supervised learnings are regression analysis (Regression) and classification problems; unsupervised learning, including cluster analysis, and semi-supervised learning; and reinforcement learning (RL), generally consisting of a strategic network and evaluation network. Combined with the Monte Carlo search, RL can be achieved by adapting the strategy to the changing environment to constantly improve its strategies, for example in the applications of Alpha-Go and Alpha-Zero, including Alpha-Fold [80–83].

Supervised regression, i.e. neural network, random forest, support vector method (SVM), and Gaussian process regression (GPR), are the most popular ML methods in mechanical analysis.

For **neural networks**, the most prevalent open-source library for neural network methods include TensorFlow [84] and PyTorch [85]. Neural network methods have been widely used in various regression and classification analyses, and their applications have been involved in a wide range of industries.

The **random forest** is a kind of supervised ML algorithm. The term "random decision forests" was first introduced by Ho in 1995 [86]. Due to its accuracy, simplicity and flexibility, it has become one of the most commonly used supervised ML algorithms. It can be used for both classification and regression tasks, which, together with its non-linear nature, makes it highly adaptable to training on a variety of data. Extreme Gradient Boosting (XGBoost), is a distributed gradient boosting library designed for efficiency, flexibility and portability. It is commonly used to train gradient-boosting decision trees and other gradient-boosting models. Random forests use the same model as gradient-boosting decision trees but use a different training algorithm. It is possible to use XGBoost to train a standalone random forest or to use a random forest as the base model for gradient boosting.

The Gaussian Process Regression (GPR) is a regression method based on Bayesian inference [87]. Compared with other kinds of ML methods, it highlights uncertainty quantification and continuity. Generally, GPR is described from the perspective of weight space and basis

function space, respectively. In contrast to regression analysis that determines the number of basis functions, GPR uses the transformation of the “kernel trick” to obtain the covariance matrix, which can have infinite basis functions, equivalent to a regression method on a continuous infinite dimensional space. Gaussian process regression Python open source libraries commonly used are Sci-kit learn and GPyTorch. GPR is becoming an increasingly popular regression model in mechanical computation because of its excellent generalisation, continuity and non-parametric features.

However, when encountering the rising pieces of training data, the GPR’s nature of infinite dimensionality makes it excessively difficult to optimise their hyper-parameters. For example, in neural network training, there may be tens of thousands or even millions of training samples, while if GPR handles such a large data set without special tricks, it will result in an unmanageable amount of computation. As a result, there are several methods proposed to deal with large training samples:

- Sparse Gaussian process regression is proposed to deal with huge datasets [88], and there is an integrated method in the Python library GPflow;
- the local approximate Gaussian process regression training [89], which selects a dataset closer to the input data points to retrain the model each time the Gaussian process is invoked, and then invokes it;
- and the matrix calculation accelerations that use GPU for Gaussian process training, as in the previously mentioned GPyTorch Python library.

In this work, the stochastic nature and smooth continuity of the Gaussian process are harnessed for creating random loading paths. These paths are then utilised for generating training samples for neural networks.

### 1.4.2 ML in computational mechanics

In the 1990s, Ghaboussi et al. started using neural networks for stress reproduction in FEMs to describe the material responses at integration points in BVP calculations [90–96], demonstrating that multilayer perceptron neural networks can indeed be used to reconstruct intrinsic structure models from stress-strain data.

Historical dependence, which is the fundamental nature of granular materials [97], is another challenge for the data-driven constitutive modelling for granular materials. In traditional elastic-plastic constitutive models, internal variables based on plasticity mechanics

methods are often used to describe the historical state of the material. In recent years, recurrent neural networks (RNN) have been introduced into this field. Mozaffar, Ma, Qu, Zhang and Bonatti et al. attempted to employ recurrent neural networks to reproduce the mechanical response to historical state-dependent/memory effects of materials [98–103]. Qu et al. [102] developed a stress-strain constitutive modelling approach based on micromechanical information by integrating the elastic stiffness matrix and invariants. Ma et al. [101] proposed a modified long short-term memory model considering the initial state of the granular material. Wang et al. [104] used the temporal convolutional networks to reproduce the history-dependent macroscopic stresses of the granular material. The minimum state cell was proposed in [103] to reduce the number of state variables (memory cells) and the linear minimum state cell was developed to mitigate the dependence of RNN on the incremental length size, and it was demonstrated to be useful in FEM simulations of aluminium alloys. Simone et al. [105] and Logarzo [106] embedded RNN into FEM to perform FEM<sup>2</sup> multi-scale calculations, which significantly improved the computational efficiency. The memory effect of RNN matches well with the material memory effect, however, RNN training is usually quite time-consuming as this model is not suitable for GPU parallel training due to its inherent iterative nature.

Xu et al. [107] used a symmetric positive definite neural network based on Cholesky decomposition of the material matrix to present the stress increment as  $d\boldsymbol{\sigma} = \mathbf{L}_\theta \mathbf{L}_\theta^T d\boldsymbol{\epsilon}$ , the loss function is constructed through the explicit equation of motion  $\mathcal{L} = (\mathbf{M}\mathbf{a} + \mathbf{P}(\mathbf{u}, \boldsymbol{\sigma}) - \mathbf{f})^2$ , where  $\mathbf{P}$  is the internal force,  $\mathbf{f}$  is the external force,  $\mathbf{M}$  and  $\mathbf{a}$  are the mass and acceleration respectively, and training is done in a similar way to RNN training. The highlight of the method is that the stress increment is calculated via the Cholesky multiplier as  $\mathbf{L}_\theta \mathbf{L}_\theta^T$ , which ensures that the stress increment is zero when the loading of the material is zero. Without this operation, it cannot be strictly guaranteed in purely data-driven training, which will introduce quite an instability encountering tiny strain increments.

In Huang et al.’s work [108] an explicit historical variable ( $\phi = \sum_i |\Delta\epsilon^{(i)}|$ ) is introduced to describe the historical influence, and the proper orthogonal decomposition (POD) is employed to reduce the six components of the stress/strain vector to three principal directions. Then they used the MLP map relationship between  $(\boldsymbol{\epsilon}, \phi)$  and  $(\boldsymbol{\sigma})$ .

Tang et al. introduced the coaxiality assumption to reduce the six components of the stress/strain vector in the 3D issues into volume strain-hypostatic stress ( $\epsilon_v - p$ ) and shear strain-effective/shear stress ( $\epsilon_s - \sigma_s$ ). Their works [109–111] attempted to reproduce the

nonlinear hyperelastic model, the J2 plasticity, and the associative Drucker–Prager model via searching in the datasets, respectively, and the multi-scale calculation of porous materials using the network as regression tool [112]. The highlight of their works is the coaxiality assumption and its analogous usage in the case of the plastic constitutive model. By assuming the relationship  $(\epsilon_v - p)$  and  $(\epsilon_s - \sigma_s)$ , the relationship can be converted to 3D via rotation operator  $\sigma_{ij} = \sum_I \gamma \epsilon^{(I)} v_i^{(I)} v_j^{(I)}$ , where  $\gamma$  is calculated from the relationship between  $\epsilon_s$  and  $\sigma_s$ ,  $I$  is the number of the principal,  $v$  is the eigen value of the strain tensor  $\epsilon_{ij}$ . Note that, in Tang’s ”Map123” works, the output is obtained through searching in the dataset, instead of training an agent regression model.

The searching method for mechanical calculations first appeared in a series of works by Otiz et al. [73, 113–115]. The most representative of work for multi-scale calculations of granular materials is [73], where a low-scale LS-DEM (Level-Set DEM) was used to construct datasets and macroscopic calculations are implemented in MPM. The data-driven algorithms were invoked to return the constitutive response on the material points. Fairly perfect results are displayed in this paper. Yang et al. 2019 proposed the Structure-Genome-Driven method for composite material calculations based on Otiz’s search method.

Fuhg et al. [116] compared the performance of Gaussian process, ANN and Sobolev training-based ANN for stress prediction on local data points for the multi-scale properties of isotropic hyperelastic materials and accelerated multiscale calculations using a LaGPR (local approximation GPR) agent model. They also addressed the convexity of the yield surface in support vector, Gaussian process regression and neural network respectively, and proposed a hybrid metallic material yielding surface combining the phenomenological model (model part) and the data-driven model (data part) [117]. The data-driven model only uses the uniaxial and biaxial experimental data describing the shape of the initial yield surface.

Sun et al. have done a lot of work in data-driven mechanics calculations. Vlassis [118] used training a deep learning network in Sobolev space to reproduce a smooth elastoplastic model with the elastic part controlled by the energy equation, the yield surface and plastic hardening and flow controlled by a set of Level-sets; Wang [119, 120] introduced the idea of Reinforce learning by introducing directed graphs for finding the best path (i.e. the microscopic parameters and order of the granular material contained in the model) for modelling the elastoplastic model, while training and scoring this choice, and automatically selecting from a decision tree by reinforcement learning search to select the best matching intrinsic model.

## 1.5 Challenges

Since the field is still in its early stages and lacks reliable methods, constructing a constitutive model for granular materials via machine learning presents several challenges. These include the limited scope of training data, difficulties in capturing path-dependent behaviour, issues with generalisation across different scenarios, maintaining high prediction accuracy, and preventing error accumulation during multi-step load calculations.

## Chapter 2

# A predictive framework for the path-dependent mechanical response of multi-graded granular assemblies

With the advancements in numerical simulation techniques and measurement instruments, we now have access to a growing volume of high-quality data. However, the traditional granular material constitutive model is gradually insufficient to fully utilise this increasingly rich data set. To address this, we propose a deep learning-based granular material model that incorporates modified long short-termed method (mLSTM) cells. These cells establish a relationship between macroscopic mechanical properties and structural characteristics for granular materials.

By initialising the hidden states of the mLSTM cell based on the initial state of the granular material, we can effectively combine the historical dependence of the granular material with the memory properties of the LSTM cell. To validate the predictive capability of our model, we employ DEM simulation results encompassing various particle size distributions, initial void ratios, different loading paths, and different initial surrounding pressures.

## 2.1 Particle size distribution (PSD) of the granular assembly

At the fine scale, dam-building granular material consists of a combination of continuously graded rock particles and complex pore structures. Therefore, this deep learning model is required to consider the PSD. Based on geotechnical testing, the most widely used parameters describing the PSD are as follows:

- $D_{50}$  represents the diameter where the cumulative mass fraction of particles with sizes smaller than  $D_{50}$  is 50%.
- The coefficient of curvature  $C_c = \frac{D_{30}^2}{D_{10}D_{60}}$ . A curvature coefficient within  $[1, 3]$  indicates a soil grading with a complete and continuous PSD. A high curvature coefficient suggests the presence of abrupt changes between the particle size ranges of  $D_{10}$  and  $D_{30}$ , indicating a lack of smaller particles. Conversely, a low curvature coefficient suggests the presence of abrupt changes between the particle size ranges of  $D_{30}$  and  $D_{60}$ , indicating a lack of larger particles.
- The coefficient of uniformity,  $C_u = \frac{D_{60}}{D_{10}}$  is generally greater than 5 when the material contains enough fine particles to fill the voids between coarse particles, aiding in easy compaction.

The curvature coefficient  $C_c$  and uniformity coefficient  $C_u$  are widely used in engineering practice due to their simplicity. However, they fall short of capturing the complete picture of the particle gradation curve. Extracting features solely from two points on the curve, namely  $D_{10}$  and  $D_{60}$ , provides insufficient information. This limitation becomes evident when examining Fig. 2.1a, which represents hypothetical PSDs of the 5 mixtures. In Figure 2.1a, it can be observed that Mixture 3 exhibits a uniform distribution of particle sizes, while Mixture 5 shows a step-like pattern in the PSD. Surprisingly, the uniformity coefficient  $C_u$  of them are the same.

The main issue stems from the fact that calculating the material parameter  $C_u$  only considers two points on the curve, overlooking the comprehensive characteristics of the entire gradation curve. The crucial aspect to describe is the dispersion of the particle size. It is desirable to encode the particle material gradation dispersion using a minimal set of physical metrics. Building on the work of Guida et al. [121], the PSD was condensed into a single



physical metric:

$$I_G = \exp \sqrt{\sum_i^N \frac{w_i}{w_{tot}} \ln^2 \left( \frac{d_i}{\bar{d}} \right)} \quad (2.1)$$

where  $w_i$  represents the weight of particles whose diameter lies between  $d_{i-1}$  and  $d_i$ ,  $w_{tot}$  is the total weight of all particles,  $N$  is the total number of ranges, and  $\bar{d} = \exp \sum_i (w_i/w_{tot} \ln(d_i))$ . The effect of  $I_G$  is shown in Fig. 2.1 where it is easy to see that the greater the dispersion the greater the  $I_G$  of the particle mixture, and it is quite sensitive to the absence of particles with certain size. Mixture 5, which lacks particles of medium size, despite having the largest  $d_{max}/d_{min}$ , has a smaller dispersion index  $I_G$  than Mixture 3 and 4.

The PSD is encoded into  $I_G$  in this Chapter. Note that the DEM simulations in this paper do not take into account particle fragmentation. As a result, the PSD remains unchanged, making  $I_G$  a constant for a specific particle assembly.

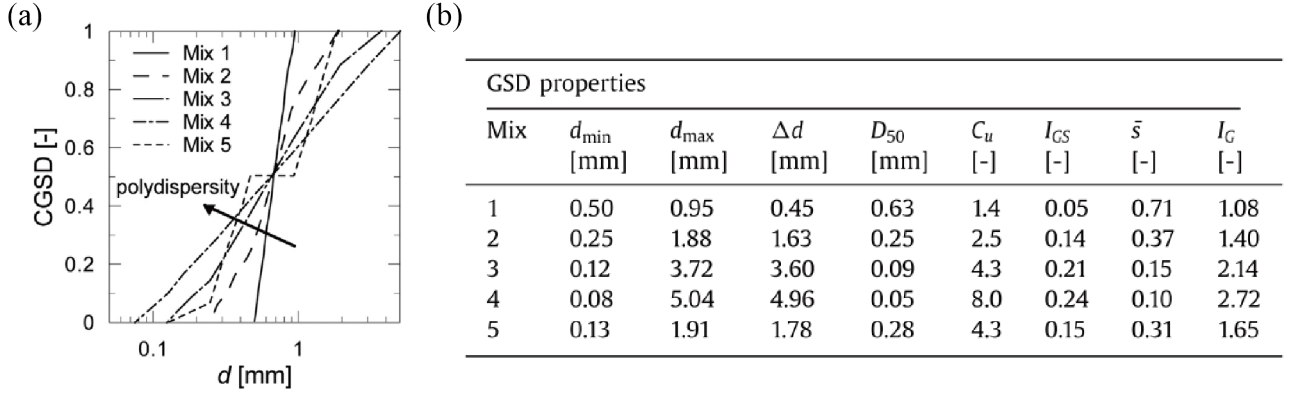


Figure 2.1: (a) Five hypothetical PSDs/GSD (grain size distributions) [1], (b) Different metrics of the five mixtures.

### 2.1.1 Loading history dependency

A large sequence of macroscopic data such as stress-strain and void ratios  $\{(\boldsymbol{\sigma}, \boldsymbol{\epsilon}, e)^{(n)} | n = 1, 2, \dots, N\}$ , where  $e$  represents the void ratio and  $N$  is the total number of loading steps, can be obtained from the DEM simulations. The hidden state of the LSTM cell can be used to implicitly represent the loading history. An explicit vector  $\boldsymbol{\chi}$  is proposed and appended to the input to enhance the network to reproduce the **long-term** historical influence. The parameter can be expressed as the accumulation of absolute strain increment:

$$\boldsymbol{\chi}^{(t)} = \sum_{i=0}^t |\Delta \boldsymbol{\epsilon}^{(i)}| \quad (2.2)$$

Thus,  $\chi$  has a clear physical meaning and increases monotonically, enabling the current material state to be uniquely calibrated.

### 2.1.2 Training data set preparation: DEM simulations

Developing accurate constitutive models that describe the mechanical properties of granular materials is a complex task, particularly when considering history-dependent material properties. As the acquisition of high-quality material data becomes easier, data-driven modelling methods can be employed to extract constitutive properties from the data. However, data generation through laboratory experiments is costly and challenging. Consequently, simulation-based approaches are commonly used for training sample generation, such as Discrete Element Method (DEM), Molecular Dynamics (MD), or low-scale Finite Element Method (FEM) simulations. Once the model demonstrates effective training on simulation data, it can then be applied to experimental datasets.

In this section, DEM simulations are utilised to conduct triaxial tests on granular materials, generating a comprehensive database for the following training. The particles in the simulations are modelled as incompressible spheres, employing the Hertz-Mindlin contact model [122]. Sliding between particles occurs when the tangential force exceeds the Coulomb frictional resistance, and the contact model accounts for rolling resistance as well. The following material parameters are used: particle density  $\rho = 2600\text{kg/m}^3$ , Young's modulus  $E = 0.8\text{GPa}$ , Poisson's ratio  $\nu = 0.12$ , friction coefficient  $\mu = 0.4$ , recovery coefficient  $e=0.95$ , and rolling friction coefficient  $\mu_r = 0.05$ .

A phenomenon similar to an 'avalanche' takes place within the particle assembly, resulting in a sharp decline in the stress-strain curve. However, this steep drop introduces challenges for the network to accurately discern constitutive patterns from the noise caused by curve oscillations. To obtain a smoother curve, a small Young's modulus and a low recovery factor of  $C_r = 0.5$  are employed.

A total of 4590 sets of DEM simulations were conducted. The simulations were executed using the LIGGGHTS software [123], which leverages the OpenMPI library to implement Message Passing Interface parallelism. The calculations were executed on the Wuhan University supercomputing server.

To account for the diverse microstructures and initial states of granular materials, we prepared granular aggregates with varying PSDs and different initial states, including initial

void ratio  $e$  and initial consolidation pressure. The PSDs encompassed Bell-shaped, Mono-sized, Binary, Linear-distributed, and Fractal-distributed distributions. In Figure 2.2, a diagram showcasing the particle size distribution and the ensemble is presented. The particle distribution function based on fractal function distribution [124] is defined as follows:

$$F(d) = \frac{d^{3-\beta} - d_{\min}^{3-\beta}}{d_{\max}^{3-\beta} - d_{\min}^{3-\beta}} \quad (2.3)$$

where  $\beta$  is the fractal coefficient deciding the shape of PSD.  $d_{\min}$  and  $d_{\max}$  represent the maximum and minimum particle diameters, respectively.

In order to ensure that the particle assemblages with different PSDs have the same number of particles and that the particle solid volumes  $v_s$  are the same,  $d_{\max}$  and  $d_{\min}$  need to be adjusted. In this calculation, the number of particles is set to  $N = 10,000$  in order to control the simulation consumption, while  $r = d_{\max}/d_{\min} = 10$  in order to prevent the minimum particle size from being too small and leading to too short a critical time step (see Eq. 2.8), so that, given  $F(d)$ , the solid volume  $v_s$  is determined by the smallest particle size  $d_{\min}$ , i.e. the function  $v_s = \hat{v}_s(d_{\min})$ . The minimum particle size  $d_{\min}$  is determined by dichotomous iteration with a fixed particle number  $N = 10,000$  as is shown in Alg. 1.

Once the minimum particle size is obtained,  $N = 10,000$  particles can be generated within a cube without contact with each other according to  $F(d)$ . The cube is then compressed isotropically and servo-loaded to achieve the target envelope pressure. Periodic boundaries are used in the simulations to avoid boundary effects.

Different friction coefficients  $\mu_0 = [0.0, 0.005, 0.01, 0.04, 0.1, 0.2, 0.3]$  were used in the initial sample preparation to generate granular samples with different initial void ratios  $e_0$ . The larger  $\mu_0$ , the larger  $e_0$  at the same surrounding pressure. Details of the granular assemblage, including the types of PSDs, mean particle size, particle number, initial void ratio range and initial envelope pressure, are summarised in Tab. 2.1.

The macroscopic mechanical behaviour of the granular material under various loading paths was evaluated using DEM. Four different loading paths were used in this simulation, including constant- $p$ -constant- $b$  loading, constant- $p$ -constant- $b$  loading, conventional triaxial cyclic loading, and random strain loading, as shown in Fig. 2.2, with details of the loading paths in Table 2.2 is shown. Note the medium principal stress coefficient is calculated as  $b = \frac{\sigma_2 - \sigma_3}{\sigma_1 - \sigma_3}$ . In the constant- $\sigma_3$ -constant- $b$  loading path, the principal stress coefficients  $b = 0$  and  $b = 1$  represent conventional triaxial compression and tension simulations respectively. The assembly of granular materials was pre-consolidated to ensure uniform initial conditions

---

**Algorithm 1** Dichotomous method for minimum particle size  $d_{\min}$ 

---

**Require:** PSD  $F(d)$ , fix solid volume  $v_s$ , particle number  $N$ **Ensure:**  $v_s^{cal} = v_s$ 

- 1:  $N^{cal} = 0$ , tolerance  $Tol = 5e - 4$
  - 2: Set the initial range  $[d_{\min}^l, d_{\min}^r]$ ,  $error = 1$
  - 3: **while**  $error > Tol$  **do**
  - 4:      $d_{\min}^m = \frac{d_{\min}^l + d_{\min}^r}{2}$ ,  $d_{\max} = 10d_{\min}^m$
  - 5:     Split range  $[d_{\min}^m, d_{\max}]$  into  $n + 1$  equidistant subintervals  
       $\{[d_{\min}^m, d_1), [d_1, d_2), \dots, [d_n, d_{\max}]\}$
  - 6:     **for**  $i$  in  $[1, 2, \dots, n + 1]$  **do**
  - 7:         Linearly interpolate  $M + 1$  particles with size  $\{d_i^0, d_i^1, \dots, d_i^M\}$
  - 8:         The average particle volume of this interval  $\bar{v}_i = \frac{1}{M+1} \sum_j^{M+1} \frac{4}{3}\pi(0.5d_i^j)^3$
  - 9:         Number of particles in this interval  $N_i = v_s (F(d_i) - F(d_{i-1})) / \bar{v}_i$
  - 10:     **end for**
  - 11:     Calculated total particle number  $N^{cal} = \sum N_i$
  - 12:     **if**  $N^{cal} \leq N$  **then**
  - 13:          $d_{\min}^l = d_{\min}^m$
  - 14:     **else**
  - 15:          $d_{\min}^r = d_{\min}^m$
  - 16:     **end if**
  - 17:      $error = |N^{cal} - N|/N$
  - 18: **end while**
-

Table 2.1: Details of the particle assemblies

Types of PSDs	Range of diameters	Average diameters	Particle number	Initial void ratio	Confining pressure (MPa)
Bell-shaped	0.006 ~ 0.0175	0.0125	10000	0.265 ~ 0.722	0.1 ~ 4.0
Mono-sized	0.0118 ~ 0.0118	0.0118		0.275 ~ 0.744	
Binary	0.009 ~ 0.018	0.0180		0.245 ~ 0.673	
Linear	0.0022 ~ 0.018	0.0151		0.237 ~ 0.623	
Fractal	0.0014 ~ 0.0139	0.0127		0.267 ~ 0.727	
	0.0031 ~ 0.0308	0.0217		0.198 ~ 0.565	
	0.0044 ~ 0.0433	0.0235		0.175 ~ 0.488	
	0.0051 ~ 0.0502	0.0216		0.180 ~ 0.469	
	0.0057 ~ 0.0532	0.0188		0.191 ~ 0.496	

Table 2.2: Summary of the loading paths

Types of loading paths	Parameter 1	Parameter 2
Constant- $p$ -constant- $b$	$p = 0.1, 0.5, 1.0, 2.0, 4.0$ (MPa)	$b = 0.00, 0.25, 0.50, 0.75, 1.00$
Constant- $\sigma_3$ -constant- $b$	$\sigma_3 = 0.1, 0.5, 1.0, 2.0, 4.0$ (MPa)	$b = 0.00, 0.25, 0.50, 0.75, 1.00$
Cyclic loading	$\sigma_2 = \sigma_3 = 0.1, 0.5, 1.0, 2.0, 4.0$ (MPa)	
Random loading	/	

across the samples. The tension observed in Fig. 2.2c refers to tensile strain relative to the pre-consolidated state, but the stress remains compressive.

### 2.1.3 Strain and stress of particle assembly

The strain is calculated as:

$$\epsilon_j = \ln \left( 1 + \frac{\Delta L_j}{L_j} \right) \quad (2.4)$$

where  $L_j$  represents the initial length in  $j$ -direction and  $\Delta L_j$  is the change.

The homogenised stress is calculated as [125]:

$$\sigma_{ij} = \frac{1}{V} \sum_{\alpha}^N f_i^{\alpha} d_j^{\alpha} \quad (2.5)$$

where  $N$  is the number of contacts within the volume  $V$ ,  $f_i$  and  $d_j$  represent the contact force and branch vector (the line connecting the centres of the 2 spheres), respectively.

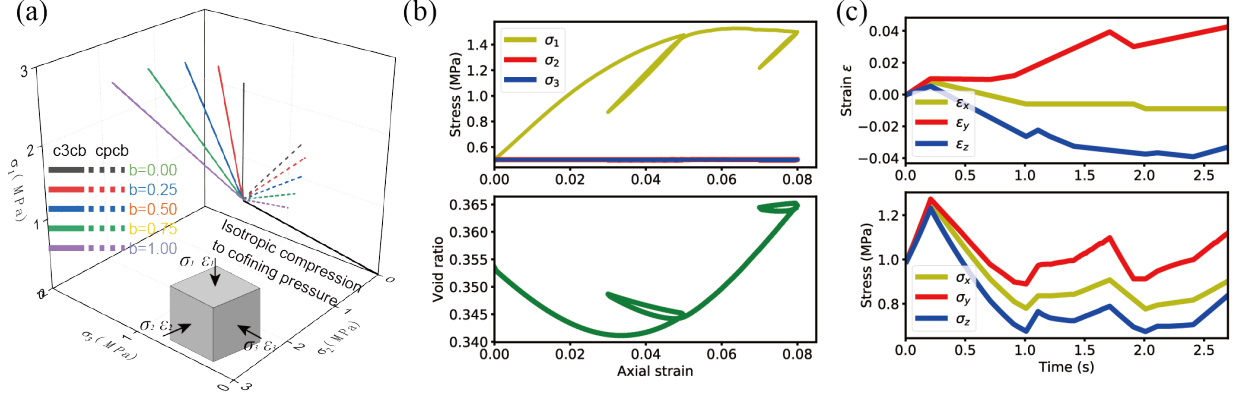


Figure 2.2: Loading paths included in training sample preparation: (a) constant- $\sigma_3$ -constant- $b$  loading and constant- $p$ -constant- $b$  loading, (b) cyclic loading, and (c) strain-controlled random loading. Note:  $\sigma_3$  corresponds to the principal stress along the third axis (or  $z$ -direction), while the subscripts 1-2-3 and  $x$ - $y$ - $z$  are used interchangeably to denote principal axes and coordinate directions, respectively.

### 2.1.4 Quasi-static loading rate

In geotechnical testing, loading is typically performed under quasi-static conditions. The inertial number serves as a key criterion for defining these quasi-static conditions:

$$I = \dot{\epsilon} d \sqrt{\frac{\rho}{p}} \quad (2.6)$$

where  $\dot{\epsilon}$  represents the strain rate,  $d$  is the characteristic diameter,  $\rho$  denotes the density, and  $p$  is the effective mean stress. A small inertial number  $I$  indicates quasi-static loading conditions. As  $I$  increases, the simulation transitions to a dense flow regime and eventually to a collisional dynamic regime [121].

Therefore, it is important to control the inertia  $I$  to ensure quasi-static loading. The literature [2] investigates the effect of inertia  $I$  on loading results, in particular on the critical state stress ratio  $\eta = q/p$  ( $q$  is the shear stress), the volume strain and the particle contact configuration tensor  $\Phi = \left( \sum_{c=1}^{N_c} \mathbf{n}^c \otimes \mathbf{n}^c \right) / N_c$ , where  $\mathbf{n}^c$  is the normal direction of contact  $c$ . As shown in Fig. 2.3 below, in (a) and (c), when the inertia  $I \leq 5e - 2$  then the simulation results are essentially the same, and examination of (b) reveals that the critical state volume strain  $\epsilon_v$  gradually converges when the inertia  $I \leq 2.5e - 3$ . Therefore, inertial  $I \leq 2.5e - 3$  should be maintained in the simulations to ensure the quasi-static loading.

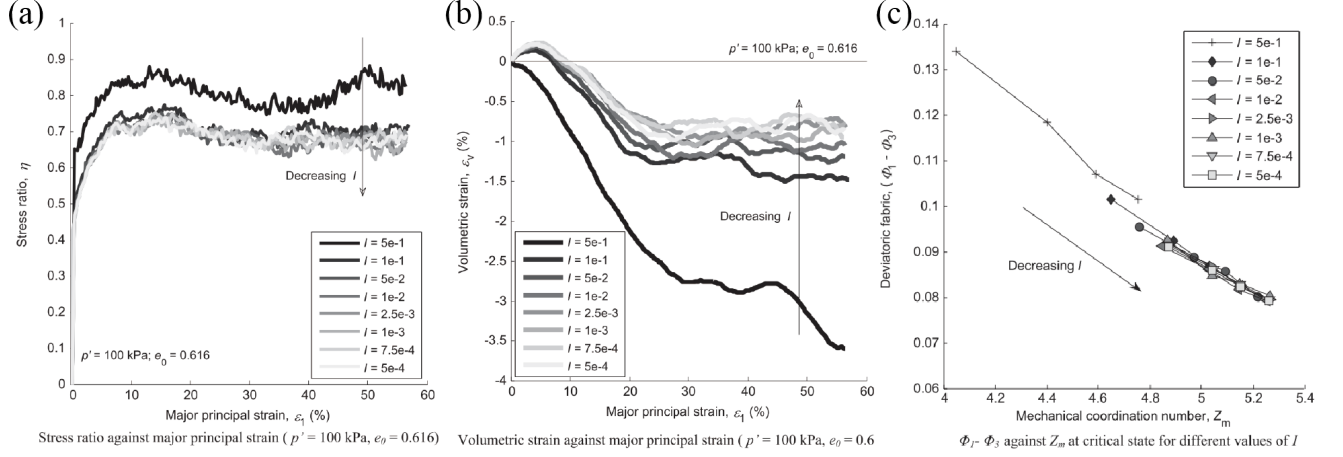


Figure 2.3: Influence of inertial number  $I$  on (a) stress ratio  $\eta$ : as  $I$  decreases to  $2.5e-3$ , it converges to around 0.7; (b) volumetric strain: converges to  $0.8\%$  after  $I$  decreases to  $2.5e-3$ ; and (c) deviatoric fabric. [2]

## 2.1.5 Critical step size in time integration

In DEM calculations, time-integration-based particle position generally updates like:

$$x = x_0 + v\Delta t + \frac{1}{2}a\Delta t^2 \quad (2.7)$$

where  $x$  is particle's position,  $v$  is the particle's velocity and  $a$  is the acceleration. The time-integration method needs  $\Delta t < \Delta t_{crit}$  to ensure computational stability.

Systems with different types of freedom require different methods to evaluate the critical load step, as is listed in [126]. Here we use the method in the LS-DYNA DEM to calculate the critical load step:

$$\Delta t_{crit} = 0.2\pi \sqrt{\frac{m_{min}}{3(1+2\nu)E}\beta} \quad (2.8)$$

where  $\beta$  is the stiffness penalty, generally ranging from 0.001 to 0.1, and  $m_{min}$  is the minimum mass of the particles, which explains the significant increase in computational effort for too small  $m_{min}$ .

The critical step sizes for different particle assemblies are shown in Tab. 2.3, where  $\beta = 0.01$  is relatively conservative to ensure correctness. The critical time steps for all assemblies are greater than  $1e-6$ s, except for the smallest particle size  $d_{min} = 0.0014$ , where the critical time step is slightly less than  $1e-6$ s. Considering that it is better to adopt a uniform standard for a large number of simulations,  $\Delta t = 1e-6$ s can be accepted as the time step for all assemblies in the simulation.

Table 2.3: Summary of the critical time steps for different particle assemblies

Types of PSDs	$d_{\min}$	$E$ (GPa)	$\nu$	$\beta$	$\Delta t_{crit}$
Bell-shaped	0.006	0.8	0.12	2600	7.34e-6
Mono-sized	0.0118				2.02e-5
Binary	0.009				1.34e-5
Linear	0.0022				1.63e-6
Fractal	0.0014				8.28e-7
	0.0031				2.72e-6
	0.0044				4.61e-6
	0.0051				5.75e-6
	0.0057				6.80e-6

## 2.2 Methodology for DL-based mechanical response prediction

In this section, we present deep learning for predicting the macroscopic mechanical response of granular materials: deep network construction, and data set preparation.

### 2.2.1 Challenge of the recurrent network unit: gradient explosion or fading

The gradient of recurrent neural networks (RNN) has the problem of vanishing and exploding, as shown in Fig. 2.4 for an RNN with the input sequence  $\{x_1, x_2, \dots, x_r\}$  and the output sequence  $\{h_1, h_2, \dots, h_r\}$ , with the cell state information stored in  $\{c_0, c_1, c_2, \dots, c_{r-1}\}$ . To illustrate the error explosion or hour problem, here we consider 1D inputs and outputs without considering the biases.

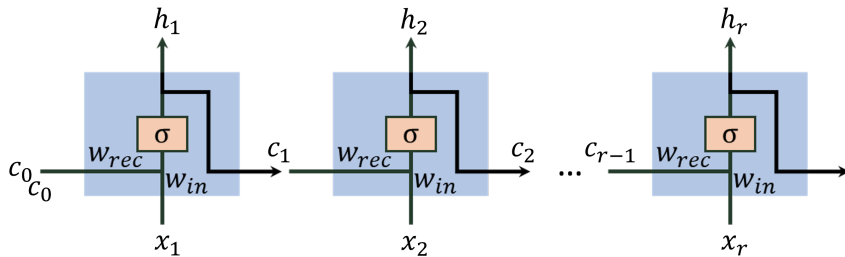


Figure 2.4: Architecture of the recurrent neural network



As in the classical RNN unit, the hidden state  $h_t$  equals the cell state  $c_t$ . The calculation can be expressed as:

$$h_t = \sigma(w_{rec} \cdot h_{t-1} + w_{in} \cdot x_{t-1}) \quad (2.9)$$

where  $\sigma$  is the Sigmoid activation function,  $w_{rec}$  and  $w_{in}$  are the weights for the recurrent input  $h_{t-1}$  and the input  $x_{t-1}$ , respectively.

Define the overall prediction error as  $\mathcal{L} = \sum_{t=1}^T \mathcal{L}_t$ , so the gradient of the serial prediction error against the weight matrix  $w = [w_{rec}, w_{in}]$  is calculated as:

$$\frac{\partial \mathcal{L}}{\partial w} = \sum_{t=1}^T \frac{\partial \mathcal{L}_t}{\partial w} = \left[ \sum_{t=1}^T \frac{\partial \mathcal{L}_t}{\partial h_t} \frac{\partial h_t}{\partial w_{rec}}, \sum_{t=1}^T \frac{\partial \mathcal{L}_t}{\partial h_t} \frac{\partial h_t}{\partial w_{in}} \right] \quad (2.10)$$

where  $T$  is the length of the training sequence, and for simplicity, the derivatives  $\frac{\partial \mathcal{L}}{\partial w}$  indicates  $[\frac{\partial \mathcal{L}}{\partial w_{rec}}, \frac{\partial \mathcal{L}}{\partial w_{in}}]$ .

The derivatives to  $w_{in}$  is:

$$\frac{\partial h_t}{\partial w_{in}} = x_{t-1} \sigma'(w_{rec} \cdot h_{t-1} + w_{in} \cdot x_{t-1}) \quad (2.11)$$

The global loss derivative to  $w_{in}$  can be shown as:

$$\frac{\partial \mathcal{L}}{\partial w_{in}} = \sum_{t=1}^T x_{t-1} \sigma'(w_{rec} \cdot h_{t-1} + w_{in} \cdot x_{t-1}) \frac{\partial \mathcal{L}_t}{\partial h_t} \quad (2.12)$$

Let's pay attention to the derivatives of  $h_t$  to  $w_{rec}$ :

$$\frac{\partial h_t}{\partial w_{rec}} = \sigma'(w_{rec} \cdot h_{t-1} + w_{in} \cdot x_{t-1}) \cdot (h_{t-1} + w_{rec} \frac{\partial h_{t-1}}{\partial w_{rec}}) \quad (2.13)$$

This is an iteration with the final term  $\frac{\partial h_0}{\partial w_{rec}} = 0$  and  $\sigma'_0 = \sigma'(w_{rec} h_0 + w_{in} x_1)$ . Finally, the derivatives can be shown as:

$$\frac{\partial h_t}{\partial w_{rec}} = \sum_{i=1}^t (h_i w_{rec}^{t-i} \prod_{j=i}^t \sigma'(w_{rec} \cdot h_{j-1} + w_{in} \cdot x_{j-1})) \quad (2.14)$$

Substitute Eq. 2.14 into Eq. 2.10, the derivatives of global loss to  $w_{rec}$  is:

$$\frac{\partial \mathcal{L}}{\partial w_{rec}} = \sum_{t=1}^T \frac{\partial \mathcal{L}_t}{\partial h_t} \sum_{i=1}^t (h_i w_{rec}^{t-i} \prod_{j=i}^t \sigma'(w_{rec} \cdot h_{j-1} + w_{in} \cdot x_{j-1})) \quad (2.15)$$

The exponential term  $w_{rec}^{t-i}$  in the above equation can either tend towards infinity or approach zero as the exponential number increases, potentially leading to gradient explosion

or gradient vanishing. This issue is particularly pronounced during the training of long sequences.

The LSTM (Long Short-Term Memory) unit [127] was introduced to alleviate these problems through the specially designed gates. The operations in the LSTM unit are represented as the following equations:

$$\begin{cases} f_t = \sigma_g(x_t \cdot W_f + h_{t-1} \cdot U_f + b_f) \\ i_t = \sigma_g(x_t \cdot W_i + h_{t-1} \cdot U_i + b_i) \\ o_t = \sigma_g(x_t \cdot W_o + h_{t-1} \cdot U_o + b_o) \\ \tilde{C}_t = \sigma_g(x_t \cdot W_c + h_{t-1} \cdot U_c + b_c) \\ C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\ h_t = o_t \odot \sigma_h(C_t) \end{cases} \quad (2.16)$$

where  $W$  and  $U$  are weights for the input  $x_t$  and hidden state  $h_{t-1}$ , respectively, and  $C$  is the cell state.

$C_t$  depends on the values of  $x_t, h_{t-1}, C_{t-1}$ . As  $h_{t-1}$  recurrently depends on of  $C_{t-1}, h_{t-2}, x_{t-1}$ , the partial differential can be written as:

$$\frac{\partial C_t}{\partial C_{t-1}} = \frac{\partial f_t}{\partial C_{t-1}} \odot C_{t-1} + f_t + \frac{\partial i_t}{\partial C_{t-1}} \odot \tilde{C}_{t-1} + i_t \odot \frac{\partial \tilde{C}_t}{\partial C_{t-1}} \quad (2.17)$$

where the derivatives of the forget gate, input gate and cell gate can be expressed as:

$$\begin{cases} \frac{\partial f_t}{\partial C_{t-1}} = \sigma'_{g,f,t} U_f \frac{\partial h_{t-1}}{\partial C_{t-1}} \\ \frac{\partial i_t}{\partial C_{t-1}} = \sigma'_{g,i,t} U_i \frac{\partial h_{t-1}}{\partial C_{t-1}} \\ \frac{\partial \tilde{C}_t}{\partial C_{t-1}} = \sigma'_{g,c,t} U_c \frac{\partial h_{t-1}}{\partial C_{t-1}} \\ \frac{\partial h_{t-1}}{\partial C_{t-1}} = o_{t-1} \odot \sigma'_h(C_{t-1}) \end{cases} \quad (2.18)$$

Then Eq. 2.17 can be written as:

$$\frac{\partial C_t}{\partial C_{t-1}} = \left( \sigma'_{g,f,t} U_f \odot C_{t-1} + \sigma'_{g,i,t} U_i \odot \tilde{C}_t + i_t \odot \sigma'_{g,c,t} U_c \right) o_{t-1} \odot \sigma'_h(C_{t-1}) + f_t \quad (2.19)$$

Compared with the hidden state of classical RNN unit  $\frac{h_t}{h_{t-1}} = \sigma'_t w_{rec}$ ,  $\frac{\partial C_t}{\partial C_{t-1}}$  is not a term related only to one of  $U_f, U_i$  or  $U_c$ , but rather a sum of them. This will result in this term being almost around 1, which will alleviate the gradient explosion and fading. But note, even with LSTM units that prevent the gradient problem, too long a training sequence can still pose a problem.

## 2.2.2 Modified LSTM considering the initial state

LSTM units can learn sequences to predict long-term dependencies in various scenarios, and are used in natural language processing and also in engineering and mechanical calculations for predicting historical dependencies [101, 105, 128]. In addition to the loading history, PSD represented by  $I_G$ , initial void ratio  $e$  and initial confining pressure significantly influence the mechanical behaviour of granular materials. Therefore, it is necessary to feed PSD information, initial void ratio  $e$  and initial stress state into the network at the head of the sequence.

In natural language processing, no special attention is generally paid to the influence of the initial implicit state on prediction. But the initial state matters a lot in granular materials' mechanical responses.

As shown in Fig. 2.5, the classical LSTM cell was modified to enable the implicit state initialisation. In practice, we add a fully-connected layer (FC) structure before the first implied state, through which the initial implied state vector  $h_{t-1} \in \mathbb{R}^d$  is expanded into a new vector  $h'_{t-1} \in \mathbb{R}^h$  before it is input to the cell for computation, as follows:

$$h'_t = \sigma_g(w_h \cdot h_{t-1} + b_h) \quad (2.20)$$

where the weight  $w_h \in \mathbb{R}^{h \times d}$  and bias  $b_h \in \mathbb{R}^h$ . Then the tensor  $h'_t$  participates in Eq. 2.16 as the hidden state.

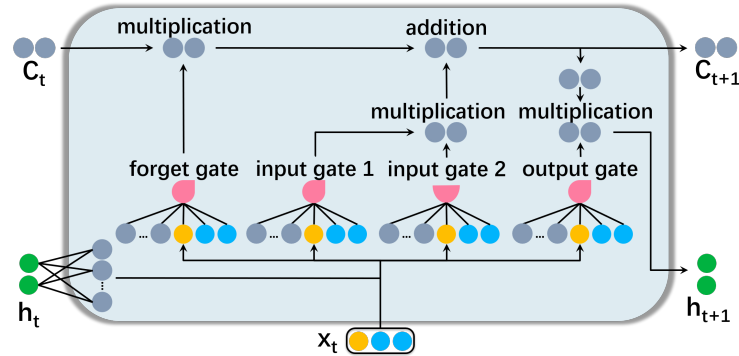


Figure 2.5: The modified LSTM unit

To demonstrate the effectiveness of the mLSTM cell, a comparison was made between its predictive ability and that of the classical LSTM cell. In order to eliminate the influence of chance on the evaluation of network prediction errors, the training process was repeated five times using different random seeds for each scenario with varying numbers of hidden layer

cells. Additionally, one-fifth of the data was randomly selected from the database for each training session.

The validation error, which was obtained after 2000 training generations for the same set of scenarios, exhibited fluctuations within a certain range due to the chance nature of the network training effect. Fig. 2.6 displays the validation error for each set of solutions after five random training sessions.

Based on the distribution of the validation errors, it can be observed that the classical LSTM cell-built model continues to improve as the width of the hidden state exceeds 100. The validation error for this model lies in the order of  $1e-3$ . On the other hand, the error of the mLSTM cell-built network gradually stabilises at around  $2.5e-5$  after the width of the hidden state exceeds 50 or so.

The fact that the error of the mLSTM cell is smaller than that of the classical LSTM cell indicates that the modified network aligns well with the initial state-dependent properties of the granular material. Leveraging this property can significantly enhance the accuracy of the network in predicting macroscopic stresses in granular materials.

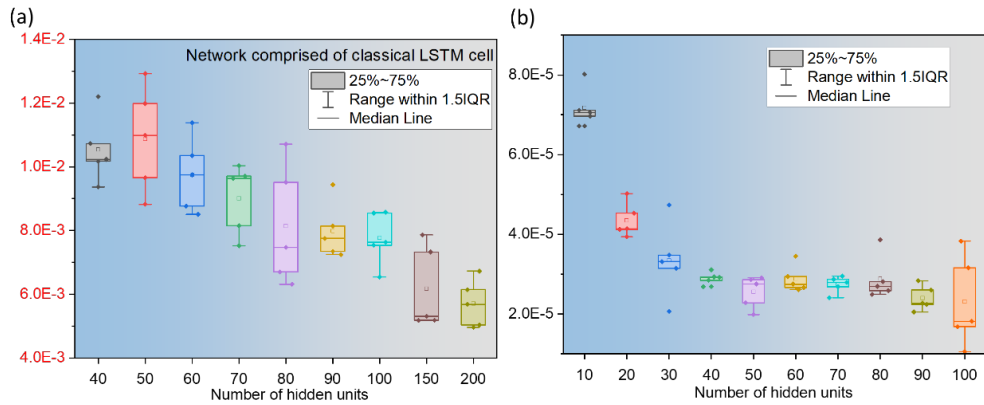


Figure 2.6: Comparison of prediction errors (a) classical LSTM cell; (b) mLSTM cell

The validation errors were compared for different numbers of LSTM cells with varying numbers of hidden cells, as depicted in Fig. 2.6b. For the mLSTM cells, it was observed that once the width of the hidden state exceeds 50, further increasing the width did not significantly enhance the predictive power of the model. This suggests that the network is already sufficiently capable of extracting the patterns present in the data set. Additionally, increasing the number of implied units makes the training process more challenging and time-consuming. Moreover, overly complex networks are more prone to over-fitting, resulting in

the network capturing more noise rather than useful constitutive relationships. Therefore, for the subsequent work in this chapter, a total of **50** hidden nodes are employed in the mLSTM unit.

### 2.2.3 Extracting sequences of training sets via sliding window

In the simulations conducted, the macroscopic state of the granular material is characterised by the principal stresses  $(\sigma_1, \sigma_2, \sigma_3)$  and strains  $(\epsilon_1, \epsilon_2, \epsilon_3)$  in three directions, and void ratio  $e$ . Each DEM simulation generates a sequence of data comprising strain, stress, and porosity ratio  $\{(\sigma, \epsilon, e)^{(n)}\}_{n=1}^N$ , where  $N$  represents the number of the total steps. With DEM loading involving massive load steps, using all this data for training is impractical. It's also crucial to set appropriate intervals between training data points, as large gaps can reduce training accuracy.

To ensure consistency, the DEM simulations conducted in this study employed the same loading velocity. To extract training data sequences from the dataset, equal load step intervals were utilized. As depicted in Fig. 2.7, a sliding time window was implemented to divide the entire long series of DEM simulation results into multiple batches of sequences.

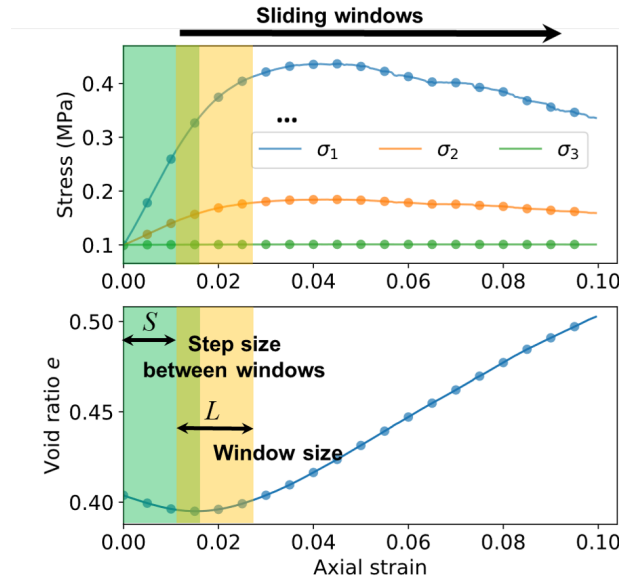


Figure 2.7: The sliding window to extract data sequences

The length of the sliding time window determines the length of the training data sequence, which in turn affects both the training difficulty and accuracy. Empirically, longer sequences result in more complex computations for derivatives during error backpropagation, making

optimization more challenging as shown in Sec. 2.2.1. However, longer sequences also retain more historical information, allowing the mLSTM units to capture the historical dependence of the granular material effectively.

To investigate the impact of the sliding time window length  $L$ , which also represents the length of the training data sequence, on training accuracy, a comparison was made among different window lengths. Similar to the sensitivity analysis of the width of hidden states, five trials were repeated for each case, with one-fifth of the original data used for training in each trial. The average training error was then calculated for analysis, and the mean error comparison is presented in Fig. 2.8.

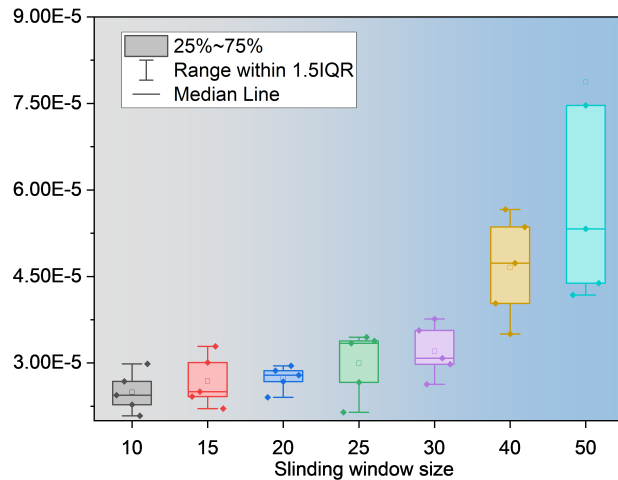


Figure 2.8: Validation error with different sliding window sizes

It was observed that once the training data sequence length exceeds 25, the validation error significantly increases with longer sequence lengths. Moreover, the variance also increases with  $L$ , indicating the training becomes unstable.

In some respects, shorter training data sequences require fitting fewer data points in a single training session, making error back-propagation and network parameter optimisation easier. However, shorter training lengths result in a reduced inclusion of historical information in the training data sequence, thereby under-utilising the memory properties of the network.

Considering these factors, for this particular study, the length of the training data sequence was set to 20 as a balance between training complexity and historical information incorporation.

## 2.2.4 Network model training

The Adam optimizer, which stands for Adaptive Moment Estimation, is a highly efficient stochastic gradient descent method that adjusts the learning rate based on adaptive momentum. It combines the benefits of the AdaGrad method [129] for handling sparse gradients and the RMSProp method [130] for addressing non-stationary problems.

By doing so, it offers several advantages, such as requiring only the calculation of first-order derivatives and consuming minimal memory, adjusting the gradient does not impact the magnitude of the optimised parameter change, and it resembles an annealing process in which the model progresses towards the state of lowest energy.

As shown in Fig. 2.9, the input features consist of PSD information  $I_G$ , material history information  $\chi$ , and current strain increments  $\Delta\epsilon$ . The outputs are the current stress  $\sigma$  and void ratio  $e$ . The network consists of a single mLSTM layer and a fully-connected layer. The red dashed line represents where to feed the first two snaps of data. The data at step 0 is fed to the initial hidden state. The output at step 1 is utilised to evaluate the prediction error.

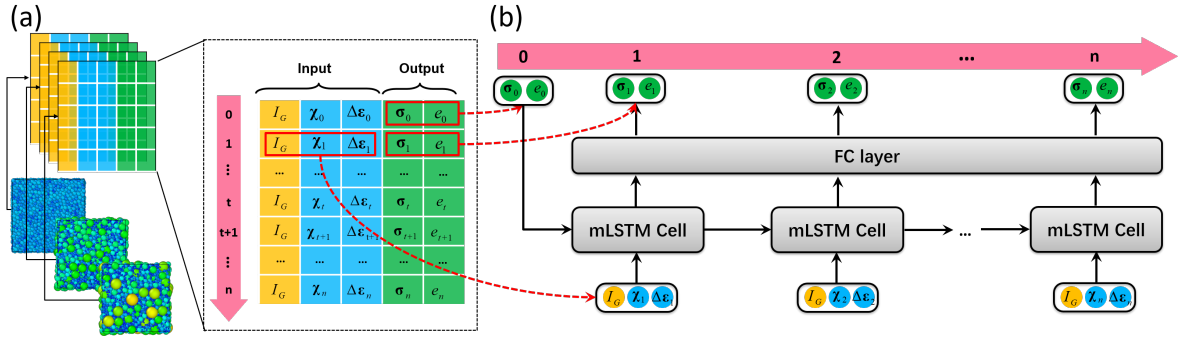


Figure 2.9: (a) Input and output of the network model. (b) Network structure.

Before commencing the training process, the data were randomly divided into three subsets: training, validation, and testing, which accounted for 70%, 15%, and 15% of the data, respectively. This division ensured that the network was trained on a diverse set of data and that its performance could be evaluated on unseen data.

To prevent over-fitting and enhance the generalisation of the network model, an Early Stopping strategy [131] was implemented in the training process. This strategy involves periodically assessing the network's performance on a random subset of data from the validation set. The network's error on this subset is evaluated, and if the error does not improve, a counter is incremented. Conversely, if the error improves, the counter is reset. If the counter

reaches a predefined threshold after each evaluation, indicating that the network's performance has plateaued, it is considered to be in the best condition possible and the training process is terminated.

By employing the Early Stopping strategy, the network training is monitored and halted when the model reaches its optimal performance, preventing it from being excessively trained and overfitting the data.

When the network converged, the mean squared error on the training and test sets was measured to be  $1.03e-5$  and  $2.16e-5$ , respectively, as shown in Fig. 2.10. The training error steadily decreased as the number of training generations increased. However, the error on the validation set reached a point of stabilization and did not show further improvement even after an additional 1000 epochs (patience number). At this stage, it is crucial to stop the network training to prevent overfitting, where the network starts capturing the errors specific to the training set, such as the noises.

The Early Stopping strategy involved the validation data set, which accounted for 15%, to evaluate the model's performance during training. The test dataset, which also accounted for 15%, was not involved in the training process at all. Consequently, the test dataset served as a means to assess the prediction accuracy and generalisation ability of the trained model.

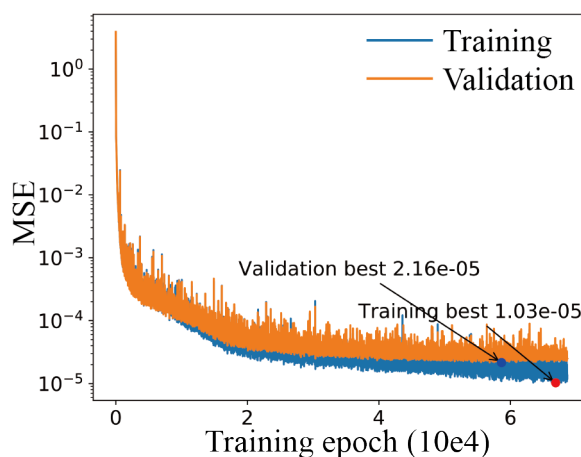


Figure 2.10: Training and validation loss during the training process

Details of the network structure and hyperparameters of the optimiser are shown in Tab. 2.4.



Table 2.4: Summary of network architecture and optimiser configuration

Layer type		Hidden layer Node number	Activation	Note
modified LSTM	hidden state preprocess	40	ReLU	<b>optimizer</b> =Adam (with learning rate=0.001,beta1=0.9, beta2=0.999 epsilon=1e-8) <b>loss function</b> = MSE <b>maximum number of attempts</b> =1000 <b>sliding window size</b> =20 <b>batch size</b> =256 <b>initializer</b> = random uniform
	LSTM Regular steps	50	Sigmoid function & tanh function	
FC layers		20	ReLU	
FC layers		4	/	

## 2.3 Model validation: stress and void ratio prediction

To validate the effectiveness of the network in capturing the macroscopic mechanical properties of granular materials, various tests are conducted on granular specimens with different gradations, initial pore ratios, and loading paths. Additionally, the effectiveness of the model in predicting historical path-related behaviour is explored which involves the internal variable  $\chi$  to predict macro responses for cyclic loading paths. It is important to note that all of these test data were not included in the model training process to ensure validity.

### 2.3.1 On different PSDs

Granular materials, both in engineering and in nature, consist of particles with varying shapes and sizes. The PSD plays a significant role in its microstructure, and different gradations result in variations in natural stacking compactness, average coordination number, and *etc.*

By considering the PSD characteristics in both DEM simulations and network model training, the model can effectively capture the influence of gradation on the macroscopic mechanical responses. Fig. 2.11 presents predictions of the macroscopic responses for two materials under the same loading path (constant- $\sigma_3$ -constant- $b$ ) and the same initial pore

ratio ( $e_0 = 0.508$ ): one with a fractal coefficient  $\beta = -0.5$  controlled PSD and the other with binary PSD.

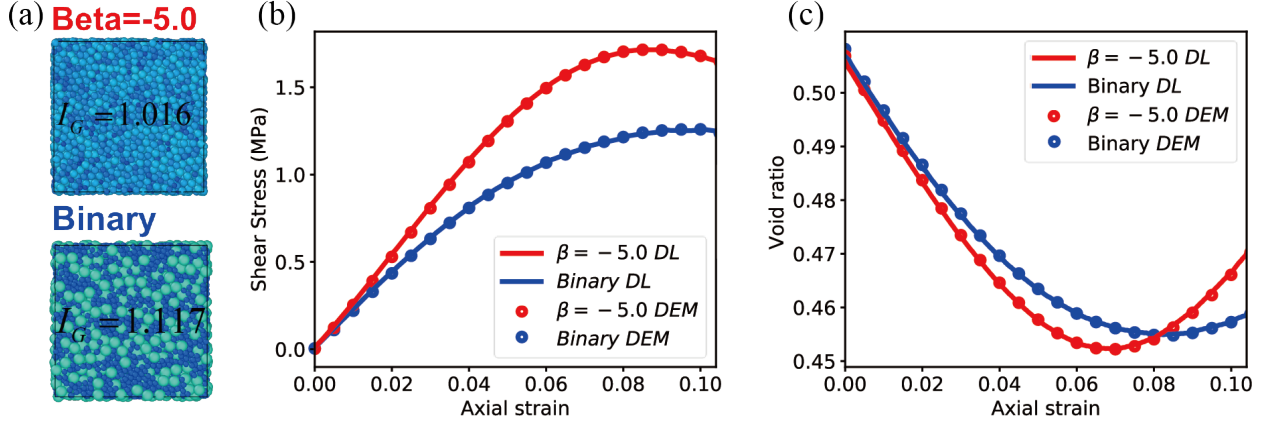


Figure 2.11: Predicted macroscopic mechanical responses with different PSDs under constant- $\sigma_3$ -constant- $b$  loading paths: (a) (top) fractal function control graded granular material aggregate ( $\beta = -5.0$ ,  $I_G = 1.016$ , initial void ratio  $e_0 = 0.508$ ), (bottom) binary mixed granular aggregate ( $I_G = 1.117$ , initial void ratio  $e_0 = 0.508$ ). (b) and (c) Comparison of depth-learning predictions (solid lines) and DEM simulations (hollow points) for the stress and pore ratio curves, respectively.

Despite their similar initial pore ratios and initial confining pressures  $\sigma_c = 1.0\text{MPa}$ , the two materials exhibit significant differences in macro-mechanical responses. The deep learning model can distinguish between materials with different PSD with  $I_G$ , and serves as a reliable predictor of the mechanical response for distinct particle assemblies under identical initial conditions.

A higher  $I_G$  value indicates a better continuity, a wider dispersion and thus a greater optimum density of the granular assembly. As depicted in Fig. 2.11, the assembly with a fractal coefficient  $\beta = -5.0$  ( $I_G = 1.016$ ) exhibits higher peak stress compared to the specimen from the binary mixture ( $I_G = 1.117$ ). Despite their equal initial void ratios  $e_0 = 0.508$ , the latter material, characterised by a higher index  $I_G$ , suggests it can be compressed more easily, resulting in a smaller void ratio. Therefore, even with equal initial void ratios, the former material (with a smaller  $I_G$ ) exhibits a higher relative density considering the optimum initial void ratio. This explains the higher peak stress and more pronounced shear expansion observed in the former material ( $I_G = 1.016$ ).

Deep learning networks trained on datasets featuring various gradations and initial pore

ratios effectively capture and reproduce the impact of particle gradation on the macro-mechanical properties of the material.

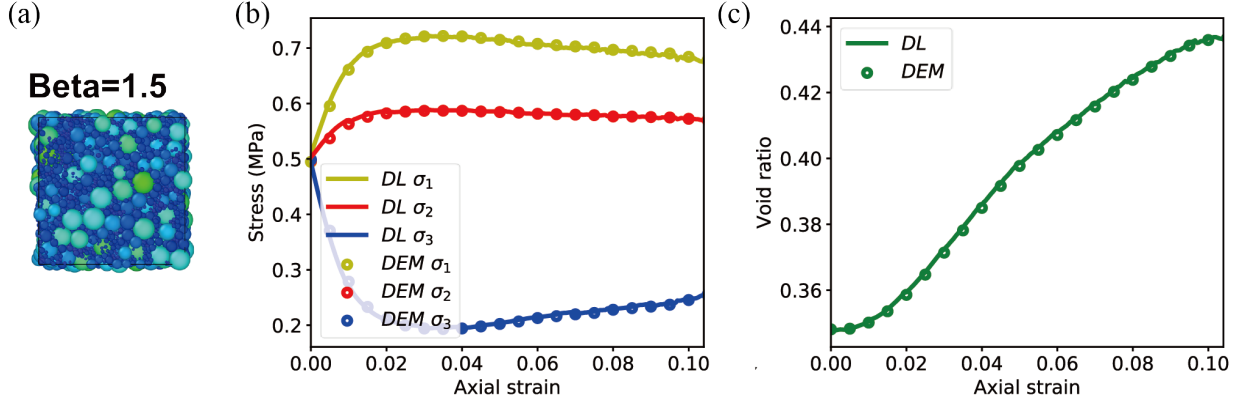


Figure 2.12: Predicted macroscopic mechanical responses with unseen PSD ( $\beta = 1.5$  and  $I_G = 1.290$ ) under constant- $p$ -constant- $b$  loading paths.

To assess the generalization ability of the deep learning network to particle gradations, an ensemble of particles with fractal parameters  $\beta = 1.5$  and  $I_G = 1.290$  was utilized. These particles were not included in Tab. 2.1 and were not involved in training the deep network model. As depicted in Fig. 2.12, when the strain sequences were fed into the trained deep learning network model, it successfully predicted the macroscopic mechanical response of the material under constant- $p$ -constant- $b$  loading paths. The predicted results exhibited good agreement with the DEM simulation results. Thus, the deep learning network can effectively predict not only the trained particle gradations but also untrained ones.

However, it's essential to note that the neural network fundamentally functions as an interpolation method, performing well within the interpolation range. When extrapolation is required, the prediction accuracy of the network drops significantly. The network excels within the training range of  $I_G \in [1.0, 1.479]$ , highlighting the importance of expanding the dataset to enhance the network's generalisation ability. However, expanding the dataset alone is not a definitive solution to all problems. While it can address some practical issues temporarily, it falls short when encountering new problems that were not represented in the training set. In such cases, dataset expansion alone is insufficient, and the network's generalization ability may be limited.

### 2.3.2 On different initial void ratios

The initial void ratio  $e_0$  of a granular material plays a crucial role in determining its mechanical properties. The network predictions are compared with those obtained from DEM simulations conducted on particle assemblies with different initial void ratios, as depicted in Figure 2.13.

During the initial stages of small strains, the material experiences shear contraction. Dense particle assemblies exhibit a higher modulus of elasticity and peak stress compared to less dense ones. As the material undergoes further deformation, it experiences shear expansion, leading to a drop in stress and exhibiting a more "brittle" behaviour compared to other less dense specimens.

As the shear progresses ( $\epsilon_{axial} > 0.28$ ), the material enters a critical state after the peak stress reduction [41]. In this state, specimens with different initial densities exhibit similar macroscopic mechanical responses. Based on the critical state theory, the behaviour of the material solely depends on its PSD and the mechanical parameters of the particle material, independent of the initial state. The influence of initial conditions and loading history gradually diminishes, displaying a fading property [132, 133]. The deep learning network model's predictions align well with the DEM simulations, indicating that the proposed model effectively captures the behaviour of the granular material concerning the initial pore ratio. Additionally, it can reproduce the material's behaviour as it enters the critical state, free from dependence on the initial state and loading history.

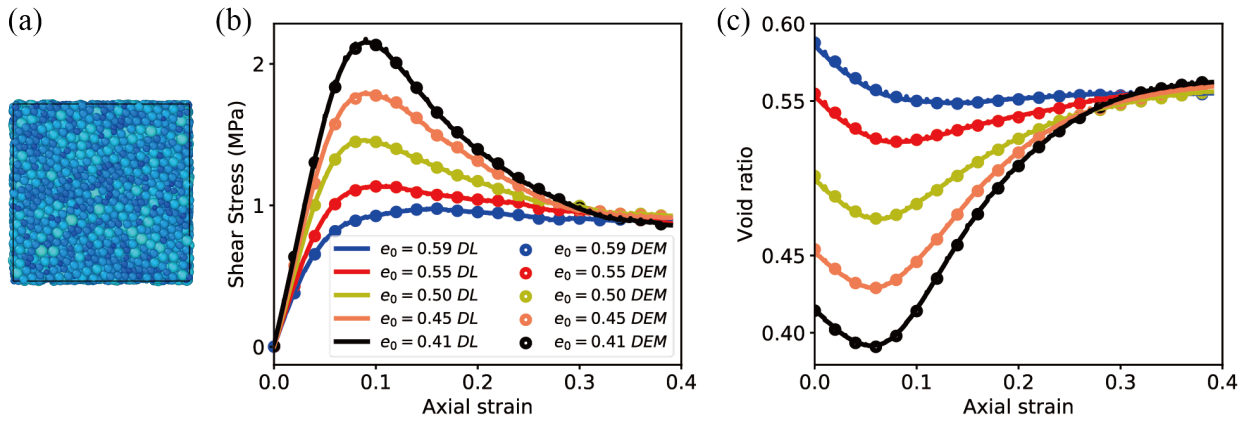


Figure 2.13: Predicted macroscopic mechanical responses with different initial void ratio  $e_0$  under conventional triaxial compression

The network successfully predicted the stress  $\sigma$  and void ratio  $e$  for both the critical and

peak states. The data points representing the peak and critical states are displayed on the  $p - q$  and  $p - e$  spaces, as depicted in Fig. 2.14. Each data point represents the location of the predicted results in that particular space. According to the critical state theory, the critical states of all materials are uniquely distributed in the  $p - q$  stress space, which can be fitted by the linear function  $q = Mp$ , and in the  $p - e$  space described by the exponential function  $e = e_{\Gamma} - \lambda (p'_c/p_a)^\xi$ . The parameters  $M, e_{\Gamma}, \lambda$  and  $\xi$  are material parameters that can be obtained by fitting the data.

Through the fitting process, we observed that predictions of the network model were not precisely on the curve, but rather very close to it. Additionally, our predictions aligned well with the DEM results. It is important to note that the curves obtained by fitting the material parameters are not sufficient to fully describe the critical and peak states. The macroscopic mechanical properties exhibited are also influenced by other factors. Therefore, a data-driven network, which relies on the input of a large amount of relevant data and historical information, is necessary to accurately and comprehensively describe the macroscopic mechanical properties of materials.

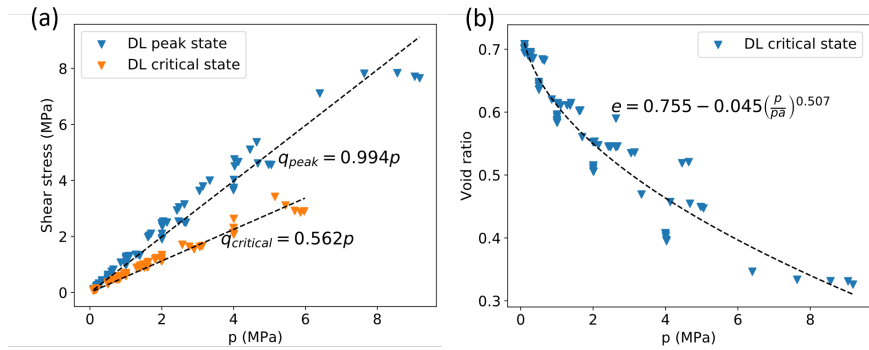


Figure 2.14: Predicted macroscopic mechanical responses with different initial void ratio  $e_0$  under conventional triaxial compression

During the analysis of network training and prediction, it was observed that the model's performance was relatively poor when dealing with specimens having a high initial void ratio  $e_0$ . Upon further investigation, it was discovered that loose specimens tended to experience an "avalanche" phenomenon during DEM loading, resulting in a sharp decline in the stress-strain curve, as depicted at the right side of Fig. 2.15. Consequently, the network model faced challenges in distinguishing between valid constitutive patterns and noises caused by curve oscillations.

Although the study of sloshing processes in granular materials has been extensively ex-

plored and linked to crustal activity in some research works, our study primarily focuses on examining the quasi-static mechanical behaviour. The dynamic process of gradual energy accumulation and abrupt release of particle contact is considered noise. When the relative density of the granular material is low, the influence of kinetic energy becomes significant, leading to an increasing amount of noise in the datasets. Consequently, this resulted in curve fluctuations and increased difficulty in training the network. Fig. 2.15 illustrates how the prediction error of the network gradually rises as the initial void ratio increases.

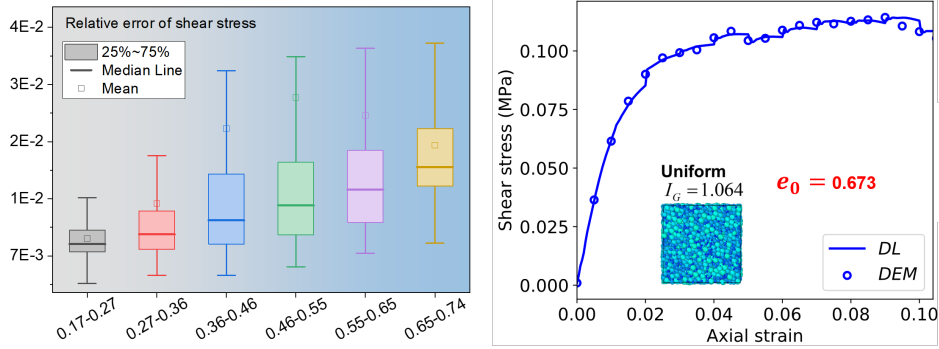


Figure 2.15: (a) Relative prediction error of shear stress  $q$  for specimens with different initial void ratios  $e_0$ : the relative error increases as the initial pore ratio increases. (b) Poorer prediction results for the specimen with a large void ratio under constant- $\sigma_3$ -constant- $b$  loading

### 2.3.3 On different loading paths

The mechanical characteristics of granular materials exhibit some variation when subjected to different macroscopic true triaxial loading conditions. Zhou et al. [27] examined the influence of the medium principal stress coefficient, denoted as  $b = (\sigma_2 - \sigma_3) / (\sigma_1 - \sigma_3)$ , on the stress response at the macroscopic level.

Fig. 2.16 and Fig. 2.17 demonstrate the significant impact of  $b$  on the macroscopic stress state during constant- $p$ -constant- $b$  and constant- $\sigma_3$ -constant- $b$  loading. In the case of  $b = 1.0$  (conventional triaxial tension), the material exhibits greater shear expansion and reaches the peak state earlier. As  $b$  decreases, the peak stress occurs at a later stage, and the shear expansion effect gradually diminishes. Remarkably, the predictions generated by the deep learning network align well with the results obtained from DEM simulations conducted under various loading paths. These results indicate that the network can effectively replicate the macroscopic mechanical response observed under different loading conditions.

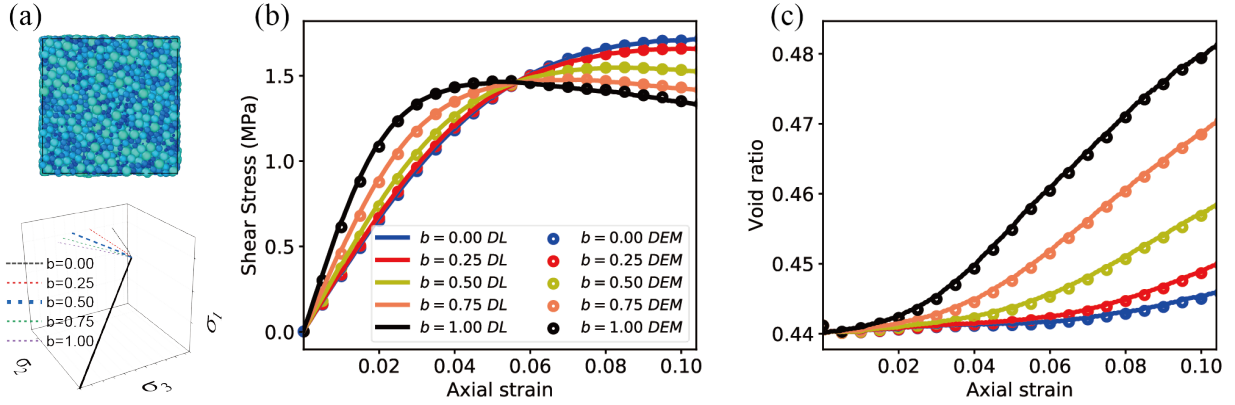


Figure 2.16: Macromechanical response predictions for true triaxial conditional constant- $p$ -constant- $b$  loading path (mean stress  $p = 2.00\text{MPa}$ ): (a) particle specimen with the linear distribution of particle size ( $I_G = 1.064$ ) (b) and (c) comparison of depth learning predictions (solid line) and DEM simulations (hollow points) for stress and pore ratio curves, respectively

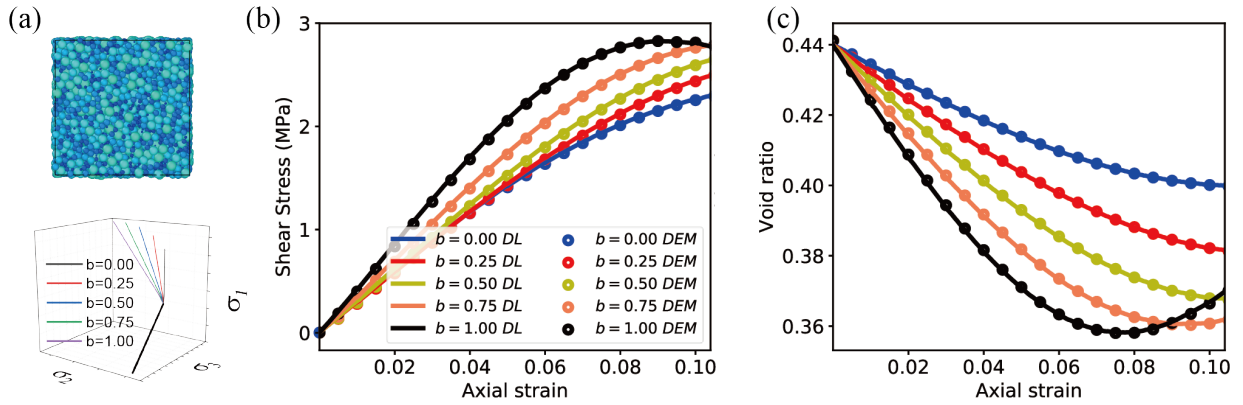


Figure 2.17: Macromechanical response predictions for the true triaxial conditional constant- $\sigma_3$ -constant- $b$  loading path (small principal stress  $\sigma_3 = 2.00\text{MPa}$ ): (a) particle specimen with linear particle size distribution ( $I_G = 1.064$ ) (b) and (c) comparison of depth-learning predictions (solid lines) and DEM simulations (hollow points) for the stress and pore ratio curves, respectively

Fig. 2.18 illustrates that the peak and critical state stress distributions exhibit strong alignment across various  $b$ . These stress distributions are found to be situated on the Spatial Mobilized Plane, which corresponds to the yielding surface proposed by Matsuoka [58].

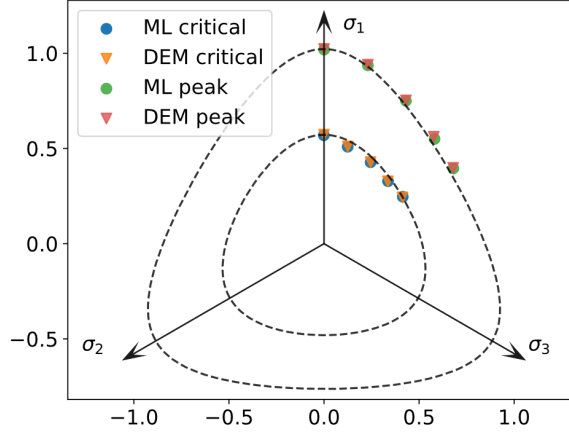


Figure 2.18: Network for predicting the distribution of peak and critical state stresses in the  $\pi$  plane under constant- $p$ -constant- $b$  loading paths.

To further verify the predictive capacity of the network model, random strain-controlled DEM simulations were conducted, and the results are presented in Fig. 2.19. The network demonstrated its ability to accurately predict the stress and void ratio values throughout the random loading path. It is important to acknowledge that the network's accurate predictions are limited to the specific random paths generated based on the current assumptions.

Comparing the mechanical response of the granular material under cyclic loading with a mono-loading, as shown in Fig. 2.20, the network was able to capture that the void ratio  $e$  was slightly less than that of the mono-loaded specimen when the unloaded material was loaded again. If the LSTM network and the history variable  $\chi$  were not involved, and only the mapping relationship between strain  $\epsilon$  and stress  $\sigma$  and void ratio  $e$  were established, the mechanical response produced would be the same for the same strain and would not reflect the effect of the loading history. However, on this DEM simulation and prediction, we were able to clearly identify a small difference in the macroscopic mechanical response of specimens that had undergone unloading and those that had not, when loaded to the same axial strain. This difference is influenced by the size of the hysteresis loop and the number of loading and unloading cycles [102].

When comparing the mechanical response of the granular material under cyclic loading with that of mono-loading, as depicted in Fig. 2.20, the network successfully captured that



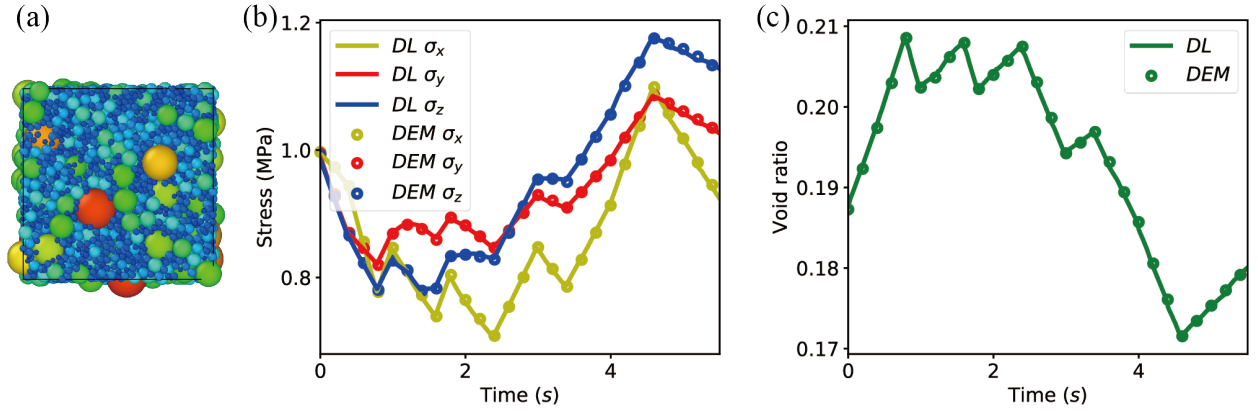


Figure 2.19: Macroscopic mechanical response prediction of granular material under random strain loading: (a) the particle assembly with fractal function controlled grading ( $\beta = 2.9$ ,  $I_G = 1.478$ ,  $e_0 = 0.187$ ); (b) and (c) network prediction (solid line) and DEM simulation (hollow point) comparison of stress curve and void ratio curve, respectively

the void ratio  $e$  was slightly lower when the material was reloaded after unloading compared to the mono-loaded specimen. This observation highlights the significance of incorporating the LSTM network and the history variable  $\chi$  in the prediction process. If only the mapping relationship between strain  $\epsilon$ , stress  $\sigma$ , and void ratio  $e$  were established without considering the loading history, the mechanical response would be identical for the same strain, failing to reflect the influence of the loading history. The size of the distinction is influenced by factors such as the size of the loading-unloading loop and the number of loading-unloading cycles.

## 2.4 Concluding remarks

The presented network model utilises a modified LSTM unit to extract the constitutive patterns directly from the DEM simulation dataset. The mechanical behaviour of granular materials is influenced by various factors such as PSD, initial void ratio, confining pressure, loading histories and *etc.* To generate the required dataset, a large number of DEM simulations are performed.

The datasets were sampled using a sliding time window method to create training sequences for network training. The trained network is capable to capture the complex constitutive relationships, including shear shrinkage, shear expansion, history dependence, strain hardening/softening, and critical state behaviour.

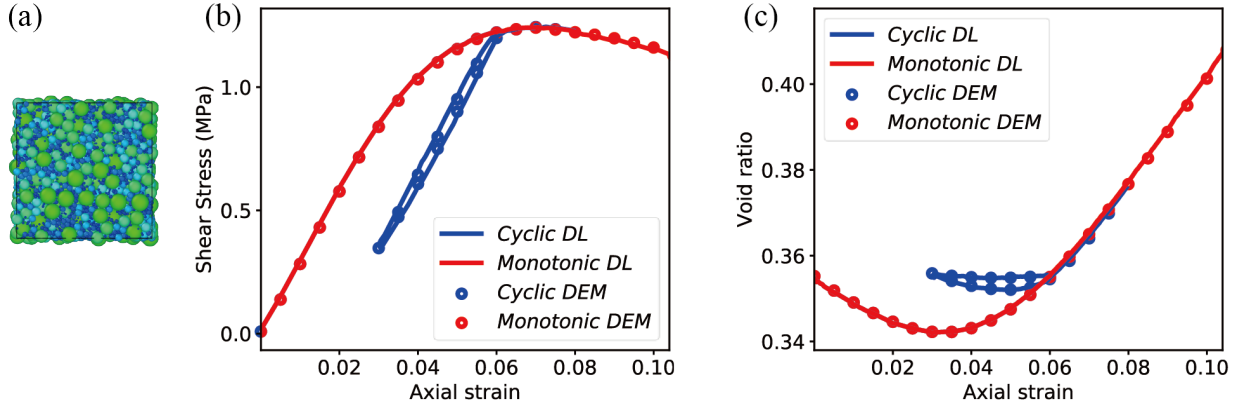


Figure 2.20: Comparison of the predicted macroscopic mechanical response of granular materials under cyclic loading and unloading with an initial enclosing pressure  $\sigma_3 = 0.50\text{MPa}$ . (a) Fractal function control grading of granular material specimens ( $\beta = 1.0$ ,  $I_G = 1.209$ ,  $e_0 = 0.375$ ) (b) and (c) Comparison of depth-learning predictions (solid lines) and DEM simulations (hollow points) of stress and porosity ratio curves, respectively

The network model offers a powerful means of predicting macroscopic stresses in granular materials, leveraging the rich information obtained from DEM simulations and enabling the potential to be used as a data-driven constitutive model.

# Chapter 3

## Deep active learning for constitutive modelling of granular materials

### 3.1 Introduction

The description of the deformation and instability of materials under external loads has been an open scientific challenge for human beings since the origin of modern science. As a mathematical approximation of material behaviour, the constitutive relation serves as the cornerstone not only for understanding the mechanical performance of materials but also for performing macroscale numerical computations (e.g. by FEM).

The most prevalent approach to formulating constitutive relations of granular materials until now has been the phenomenological constitutive theory of plasticity. Its success is underpinned by four primary assumptions: (1) an explicit partition of elastic and plastic zones in a stress-strain sequence; (2) the yield surface distinguishing the boundary between elasticity and plasticity; (3) the associative or non-associative flow rule describing the direction of plastic deformation; and (4) the hardening rule characterising the evolution of the yield surface. This classical plasticity theory has received extensive applications, but it also confronts dilemmas owing to the prior assumptions and increasing model and state parameters to be calibrated.

Machine learning offers a promising alternative to tackle the challenge of constitutive modelling. Unlike the phenomenological constitutive theory and multiscale modelling, data-driven models do not require parameter calibration and phenomenological assumptions, nei-

ther do they request unaffordable computational resources to infer stress responses from strain paths. Deep learning-based constitutive modelling includes two indispensable components: model and data. In contrast to the reviewed progress from the perspective of model in previous chapters, rather few efforts have been taken to guarantee that the dataset used to train neural networks is sufficiently representative. Yet data is crucial to developing a successful machine learning model and a high-quality dataset should cover all possible scenarios that the surrogate model is intended for. Data-driven constitutive modelling tends to suffer from two challenges pertaining to data: (1) the training process is data-demanding while it is costly to obtain high-fidelity data, via either laboratory experiments or microscale numerical simulations; (2) as data-driven models are not underpinned by physical principles, the reliability of a data-driven model is always a concern.

Active learning can provide a useful perspective to tackle the above challenges. For conventional supervised learning with a “passive learning” framework, where the training dataset is generated before the training process is performed, the model has to be prepared blindly and inefficiently. The idea behind active learning is that not all data is created equally, because different data may carry a different information intensity for the model. Provided that the most instructive data is identified, labelled, and employed to fit a model, it can be expected that such data may enable a trained model to reach the desired generalisability. However, incorporating active learning in the constitutive modelling of materials remains a relatively unexplored territory to date.

This chapter aims to fill the above gap by developing a deep learning committee-based active constitutive learning strategy. Through three different scenarios, the unique advantages of deep active learning are revealed. On the one hand, active learning can prioritise selecting the most informative data when training a surrogate model. On the other hand, active learning can serve as an effective tool to detect potential predictive blind points and improve the model continuously. We demonstrate that active learning works for both MLP and RNNs, but its application can go beyond any type of data-driven surrogate model. We also confirm that deep active learning are not only suitable for the data from common conventional or true triaxial tests but also complex strain-stress paths experienced in a BVP problem. Furthermore, the trained DNNs are embedded into FEM computations to bypass phenomenological constitutive models or particle-scale DEM simulations.

The remainder of the chapter is structured as follows. Section 3.2 introduces the basic idea of active learning in general and the committee-based active learning strategy in particular.

In Section 3.3, a comprehensive examination is conducted to understand different active learning strategies in data-driven constitutive modelling. An interactive constitutive training scheme is demonstrated in Section 3.4, where data generation and the training of DNNs are integrated seamlessly. Such a procedure allows models to identify the most informative data to make DNNs learn faster in a cost-effective manner. The significance and limitations of active learning in constitutive modelling are discussed in Section 3.5. Some concluding remarks are made in Section 3.6.

## 3.2 A deep active learning strategy for constitutive modelling

Active learning is originally motivated by many machine learning scenarios where input-output data pairs are difficult or expensive to collect, whereas most supervised learning models are data-greedy. An important issue arises: can we use as small datasets as possible to train a reliable predictive model? By identifying the most instructive data for a surrogate model, active learning can maximise the performance of models while minimising the amount of data to be labelled. Note that active learning is a strategy, rather than a specific model. When combining deep learning models with the active learning strategy, the model is called deep active learning.

The basic procedure of deep active learning is shown in Fig. 3.1. For a mapping relation connecting inputs  $\mathbf{X}$  and outputs  $\mathbf{Y}$ , we have a labelled data pool  $\mathbf{D}_l = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_l}$  and an unlabelled data pool  $\mathbf{D}_u = \{\mathbf{x}_i\}_{i=1}^{N_u}$  before training. The first step is to build deep neural networks  $F^{(0)}$  with prescribed  $\mathbf{X}$  and  $\mathbf{Y}$ . These models can be either trained or untrained with labelled datasets  $\mathbf{D}_l$ . Second, these deep neural networks are utilised to infer outputs  $\hat{\mathbf{Y}}$  based on the unlabelled data pool  $\mathbf{D}_u$ . Third, an active querying strategy is adopted to identify some unlabelled data which is most likely to be mispredicted in  $\mathbf{D}_u$ . Then the fourth step is to label/resample these recognised datasets via performing laboratory experiments or numerical simulations. These data are moved from  $\mathbf{D}_u$  to  $\mathbf{D}_l$  and new deep neural networks  $F^{(1)}$  are fitted based on the updated labelled data  $\mathbf{D}_l$ . The workflow is an iterative procedure including “exploration—labelling—training”. The procedure is repeated until the trained DNNs reach the desired accuracy over a targeted stress-strain space. Over these active learning rounds  $t = \{0, 1, 2, \dots, T\}$ , a series of DNNs  $F^{(0)}, F^{(1)}, F^{(2)}, \dots, F^{(T)}$  are trained until the mapping relation from  $\mathbf{X}$  to  $\mathbf{Y}$  has been well captured by the latest

DNN.

The core of active learning lies in the querying strategy. Many active query strategies are reported in the existing literature, such as uncertainty sampling [134], expected error reduction [135], density-weighted methods [135] and committee-based query (CBQ) strategy [136,137]. However, most active learning strategies are domain-dependent and cannot be adapted to other fields. Nevertheless, the CBQ approach can be a universal option which also works for sequence-based regression problems, such as the constitutive modelling of granular materials. In addition, the CBQ method is conceptually simple and easy to implement, and therefore this strategy is adopted in this study.

The CBQ strategy requires a committee  $C = \{M^{(1)}, M^{(2)}, \dots, M^{(N)}\}$  of  $N$  surrogate models ( $M$ ) which are fitted based on an available labelled dataset  $\mathbf{D}_l$ . Each committee member provides forecasts on the data points extracted from the unlabelled data pool  $\mathbf{D}_u$ . Then the predictive disagreement among different committee members, i.e. surrogate models can serve as an indicator of the degree of uncertainty. Specifically, the sample with the largest degree of disagreement among all the committee members is selected as the most informative one for the current surrogate model.

This approach is established based on the idea that the model should make a better prediction in the domain that is sufficiently covered by the training dataset, compared to the region where the training data is sparsely distributed. It is expected that the forecast in the region with sufficient data coverage will converge to the ground truth and thus the inferences made by the committee surrogate models are relatively close to each other. By contrast, the predictions around the region where the training data is insufficient will scatter with a measurable variance. The discrepancy among predictions can be quantified by the standard deviation  $S_N$  for a given specimen:

$$S_N = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2} \quad (3.1)$$

where  $y_i$  is the forecast of the  $i^{th}$  committee member with  $i \in [1, N]$ , and  $\bar{y}$  is the mean value of these forecasts. By ranking the standard deviations of the predictions from the unlabelled pool in their magnitude, the highest variances in the output forecasts indicate

---

<sup>1</sup>The cartoon is accessed from <https://kidsread.wordpress.com/2022/06/03/3-steps-for-teaching-kids-to-read/>)

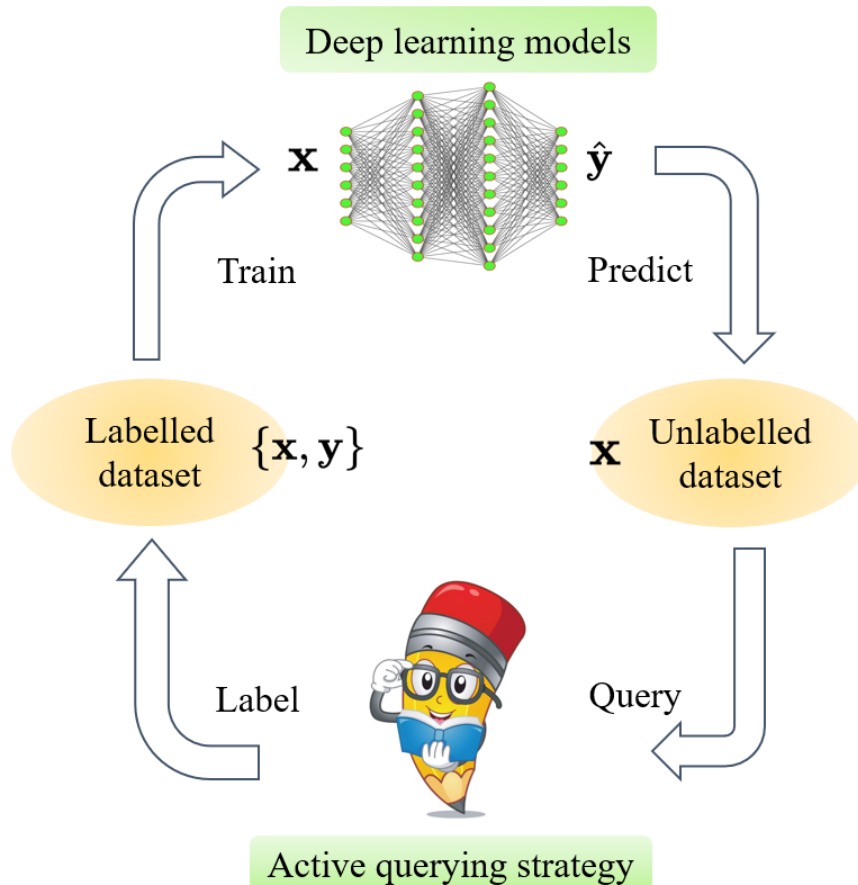


Figure 3.1: Procedures of deep active learning <sup>1</sup>

the most instructive data.

One more question is how to choose committee members. This study adopts a group of DNN models as committee members with the only difference consisting in the random initialisation of weights and biases before training. The predictive capability of DNNs is empowered by weight and bias parameters. As a high-dimensional mapping of the objective function, different (random) weight initialisations give rise to marginally different DNNs after training, even though the network architecture and all other hyperparameters are the same. Thus, one can leverage the active learning strategy regardless of what types of DNNs are in use.

For the constitutive modelling of path-dependent materials, currently, there are two typical DNNs available: One is a multilayer perceptron (MLP) and the other one is a time-series model, such as RNNs and temporal convolutional neural networks (TCN). MLP can capture the point-to-point mapping efficiently but must introduce some internal variables to encode

the loading history. In contrast, time-sequence models can capture extremely complex stress-strain relations with multiple unloading-reloading cycles but must rely on a large number of parameters and complex network structures to achieve good performance.

In the following sections, we will consider different constitutive training scenarios which may be encountered in path-dependent materials. Also, both MLPs and time-series-based DNNs will be investigated in the subsequent sections. The detailed road map is as follows:

*The role of active learning.* Section 3.3 aims to clarify the applicability of active learning in the constitutive modelling of granular materials by performing parametric and comparative examinations of three different scenarios of active learning in a stress-strain data pool.

*On-the-fly active learning.* Section 3.4 demonstrates the on-the-fly active learning where data generation and model training are performed interactively. During each round, only the optimal data which can minimise the prediction uncertainty is labelled. This part focuses on stress-strain predictions of granular materials under conventional triaxial testing scenarios. Special attention is paid to the advantage of active learning in identifying unreliable forecasts of a data-driven surrogate model and adaptively improving the model until it behaves satisfactorily in the desired stress-strain space.

### **3.3 A comprehensive examination of active learning-assisted data-driven constitutive modelling based on a data pool**

Although many simplifications are made in DEM, some research has shown that even the simplest sphere-based modelling can reproduce the primary behaviour (stress-strain relation, volumetric behaviour, and critical state) of real granular materials. To examine the role of active learning in constitutive modelling, the constant- $p$  data generated by DEM is used as a data pool. Three different training cases are considered:

Case 1: DNN models are initially trained with uniformly but relatively sparse sampled data. A total of 36 groups of specimens, as shown in Fig. 3.2, are used for the preliminary training. Then active learning is harnessed to detect blind predictions and improve the DNN models continuously by adding the most important points during each round. The workflow of Case 1 can be found in Fig. 3.3a.



Case 2: DNN models start from nothing, i.e. no data is used to train the initial DNNs. These DNNs are improved incrementally by adding the data points recognised by active learning. This case is designed to investigate whether active learning can select data judiciously at the very beginning of training. The workflow of Case 2 is exhibited in Fig. 3.3b.

Case 3: DNNs are trained with randomly selected stress-strain samples. This is the conventional training scenario without the involvement of active learning and thus can be regarded as a passive learning example.

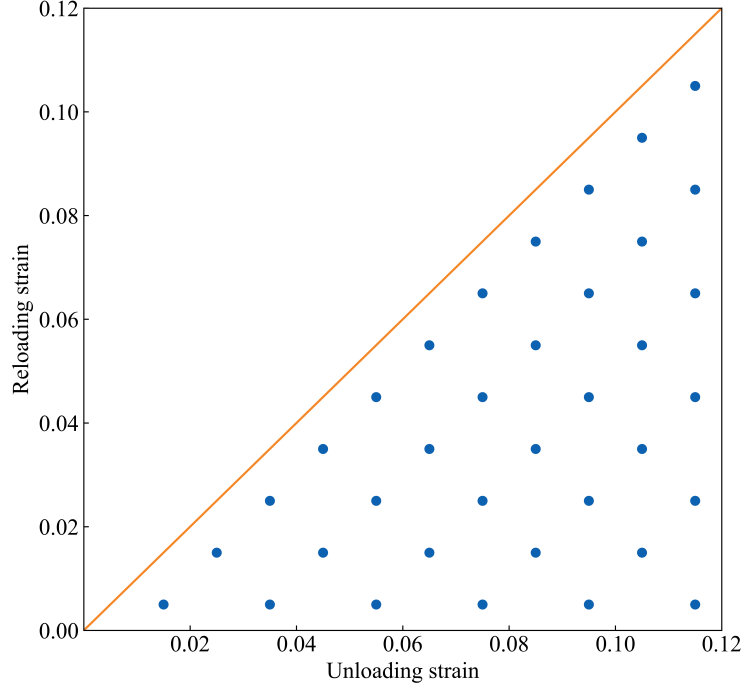
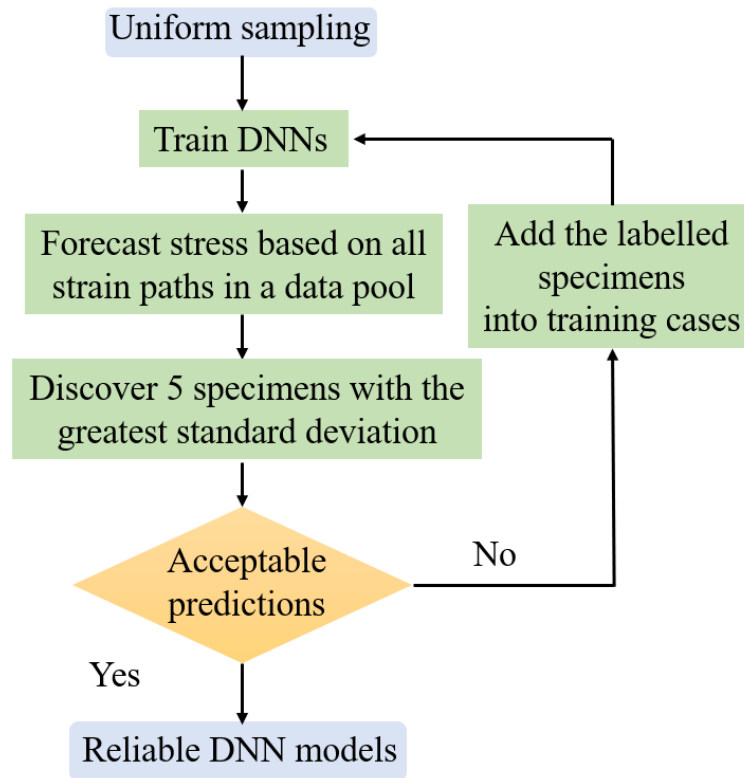


Figure 3.2: Specimen distribution used for preliminary training

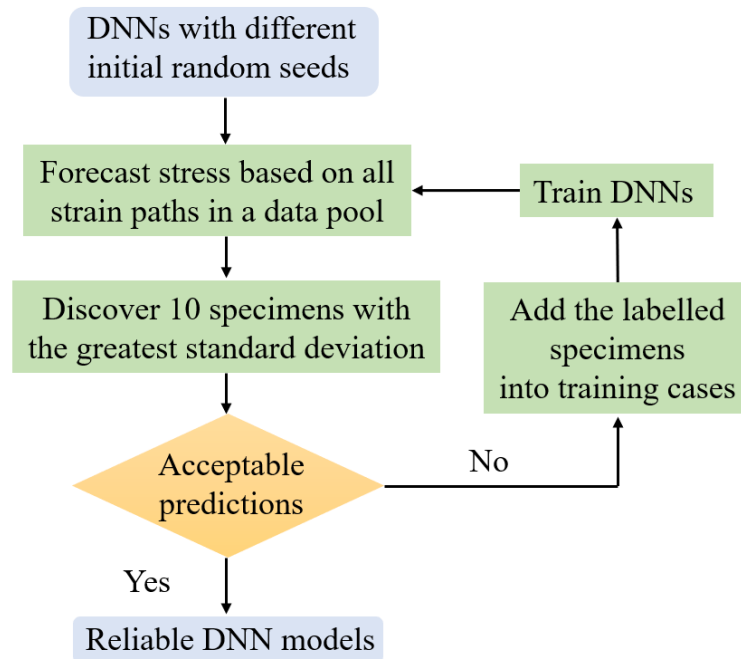
### 3.3.1 The adopted GRU neural network and accuracy evaluation

In this section, the GRU neural network is adopted as the base DNN model. In the previous chapter, this type of model has been proved to have an extraordinary fitting ability for path-dependent granular materials. Also, the GRU model is not very sensitive to network architectures and hyperparameters for relatively small-scale training data. Diverse architectures and hyperparameters can yield similar predictions owing to the strong fitting capability. In this work, the adopted architectures and hyperparameters are determined by the experience drawn from our previous work. The details can be found in Table 3.1. Such an architecture yields 5523 parameters to be trained. These GRU networks are built based

on Tensorflow and Keras.



(a) Case 1



(b) Case 2

Figure 3.3: Basic active learning procedures for Case 1 and Case 2

Table 3.1: The adopted network architecture and some key hyperparameters

Item	Value
Network architecture	GRU: 40
The length of moving windows	40
Batch size	64
Epoch number	200
Learning rate	0.001

The inference accuracy of a model is evaluated by quantifying the overall difference between the forecasts and the ground truth. Two evaluation metrics are employed in this study. One is the MAE and the other is the score metric as indicated in the previous chapter. The former can offer a relatively rigorous evaluation of the predictive ability while the latter is more intuitive to understand how good these predictions are as a whole.

Fig. 3.4 shows the loss values of training datasets in Case 1. The results show that the MAE values converge to a steady state after 200 epochs. In each active learning round, new data is added to training datasets. With increasing active learning rounds, the values of the loss function decrease gradually. Note that as the network architectures have been predetermined, no validation is conducted during training.

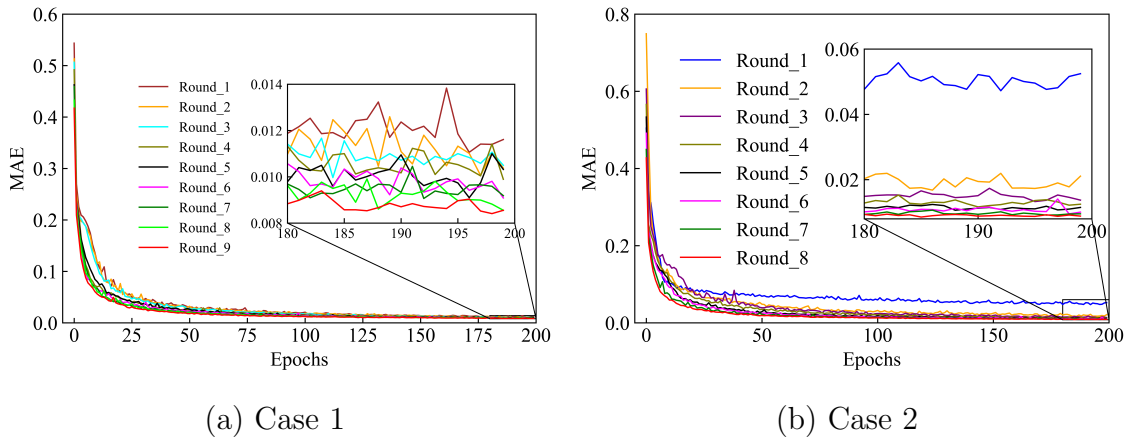


Figure 3.4: Learning curves during training

### 3.3.2 Examination and verification of the role of active learning in constitutive modelling

A series of parametric investigations are performed to understand the potential and limitations of deep active learning for data-driven constitutive modelling of granular materials. First, for Cases 1 and 2, four different DNNs are prepared and three committees with two, three, and four DNNs as members are considered. In Case 1, each DNN is pre-trained based on available datasets, while in Case 2, all the DNNs are simply initialised with different random seeds and no training is conducted at the beginning.

Active learning is utilised to detect all the forecast performance over the entire labelled data pool. Each committee member is employed to infer stress responses of all the strain paths in the data pool. Ten groups of data specimens with the largest standard deviations in Case 1 are listed in descending order in Table 3.2, where each index number in the table represents a certain strain path. To verify the capability of active learning, the actual predictive accuracy for each strain path is quantified by comparing it with the ground truth stress response. Then the ten worst predicted strain paths by each DNN are also listed.

Table 3.2: The first-round active learning-assisted forecasts and verifications for Case 1

Ranking	Estimated MAE ranking			Actual MAE ranking			
	2 DNNs	3 DNNs	4 DNNs	NN1	NN2	NN3	NN4
1	52	52	52	52	52	1	52
2	1	1	1	54	1	52	60
3	53	53	53	60	53	53	1
4	55	54	74	4	4	60	74
5	6	4	4	74	54	74	53
6	74	74	54	1	8	65	65
7	65	55	60	69	66	55	4
8	60	8	55	55	69	54	54
9	4	60	65	53	15	4	69
10	56	6	6	8	57	63	15

The table demonstrates that most estimations made by the active learning algorithm are consistent with the actual predictions. The ten worst predictions given by each committee member’s DNN vary slightly from each other, which confirms that different initialisations of weights and biases yield different DNN models. Among the ten groups of most unreliable

forecasts estimated by active learning with two, three and four committee DNNs, a total of eight, nine and nine groups of strain paths, respectively, are exactly within the actual MAE ranking list.

The results confirm that the active learning strategy is capable of discovering the worst predictions without knowing the ground truth. Furthermore, the results support that only two or three committee members are sufficient to serve as an error indicator, as using more committee member DNNs requires more computational costs. A group of three committee members are thus adopted throughout the study.

The results in Case 1 verify the effectiveness of active learning. However, in the first round of Case 2, the predictions given by active learning are irrelevant to the actual worst predicted strain paths, no matter if two, three or four committee members are employed. A big discrepancy between Cases 1 and 2 lies in that in Case 1, the initial committee DNNs have been trained based on uniformly distributed specimens and these models have learned sufficient knowledge about the constitutive relation (with average MAE: 0.03; average score: 0.944); while the initial committee DNNs in Case 2 are not trained and know nothing about the stress-strain mapping. The sharp contrast in the performance of active learning in Cases 1 and 2 indicates that only the DNNs which have learned sufficient knowledge are qualified to serve as committee members.

### 3.3.3 Batch-mode active learning scheme

For the committee-based active learning scheme, the time required to train DNN models cannot be ignored. If the most informative data is queried in serial, i.e., one at a time, the total cost of training DNN models will be practically unaffordable. To strike a balance between efficiency and labelling accuracy, batch-mode active learning is adopted by selecting a group of specimens with the largest uncertainty to label each round and thus reducing the number of training committee DNNs. However, one should consider the possible information redundancy among the selected “worst predicted” specimens, because labelling the most hard-to-predict specimen first may eliminate the need to label the “neighbouring” cases in a batch of selected specimens. To address this concern, a batch sampling scheme is proposed as shown in Fig. 3.5. When we have selected a certain point, its neighbouring points within an influence radius of  $R$  will be regarded as invalid and are not considered in the current batch. However, these invalid points in the current round can still be selected in the next round. In this study,  $R$  is empirically selected as  $1.5L_s$ , where  $L_s$  is the sampling interval.

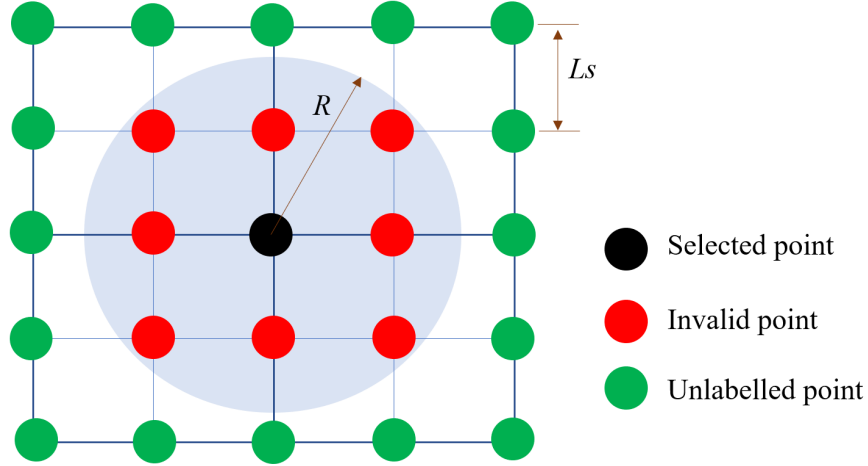


Figure 3.5: Illustration of data resampling in a batch-mode active learning scheme

### 3.3.4 Prediction performance of three training cases

The prediction performance of the three different training cases is shown in Fig. 3.6. To reduce the possible influence of randomness, three neural networks with the same architecture, hyperparameters, and training data but different initialisations in weights and biases are trained. The mean and standard deviation of these three different DNN predictions are considered. When the number of training specimens is less than 35 groups, the DNNs from Case 3 outperform those in Case 2, demonstrating that the committee-based active learning algorithm is not necessarily useful when each committee member DNN has not learnt sufficient knowledge.

When the training number is larger than 35, the prediction accuracy of Cases 1 and 2 outperforms those in Case 3. Figs. 3.7 and 3.8 demonstrate the worst and second worst forecasts given by a DNN in Cases 1 and 3 when 60 groups of training datasets are used. Although it appears that only a small discrepancy in the forecasted score and MAE is found, as shown in Fig. 3.6, a relatively large difference in the actual stress-strain predictions can be observed. Particularly, in the random inputted training datasets in Case 3, DNNs' generalisation error decreases relatively slowly and 30 groups of extra training specimens (almost  $\frac{1}{4}$  of the total number) are required to reach a similar predictive level in Cases 1 and 2. These results confirm that active learning can train a good predictive model by using less amount of data.

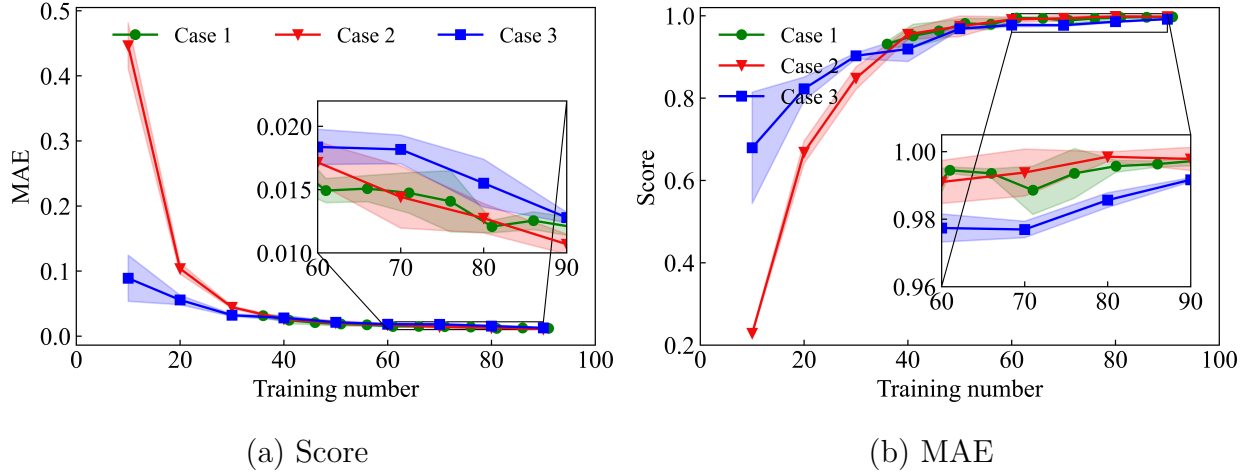


Figure 3.6: Learning curves during training

### 3.3.5 Active learning-informed data preparation

With the aid of active learning, DNNs can automatically prioritise labelling the most instructive data to maximise the inference performance of models. We summarise in Fig. 3.9 the intelligently sampled data recognised by active learning in each round for Cases 1 and 2. In Case 1, the top five groups of new specimens are selected in each round and a total of six rounds of active learning querying are included. In Case 2, the top ten groups of datasets in each round are selected because all the training data is obtained by active querying and labelling a relatively large number of specimens in each round can reduce the overall time costs. A total of seven active learning querying rounds are listed in Fig. 3.9b.

Fig. 3.9 may suggest that the specimens with reloading strain lower than 2% are probably the most informative data for training. These specimens represent a relatively larger unloading-reloading loop. The prediction is similar to extrapolation if we infer large unloading-reloading loops with only small-loop data because a large loop may include richer information than small loops. Furthermore, Fig. 3.9a indicates that the loops at the elastic-plastic transition phase (a major principal strain ranging from 0.03 to 0.07) may outweigh those at the critical stage where the stress remains constant with the increasing strain. The reason can be attributed to the fact that the adopted training data has a relatively long and steady critical state (see Fig. 3.8), and thus the information density carried by the data points at the post-peak state is lower than those at the elastic-plastic transition stage.

By observing the first and second rounds of selected data in Fig. 3.9b, it is found that the selected data always occur in clusters. This phenomenon may explain why the active learning

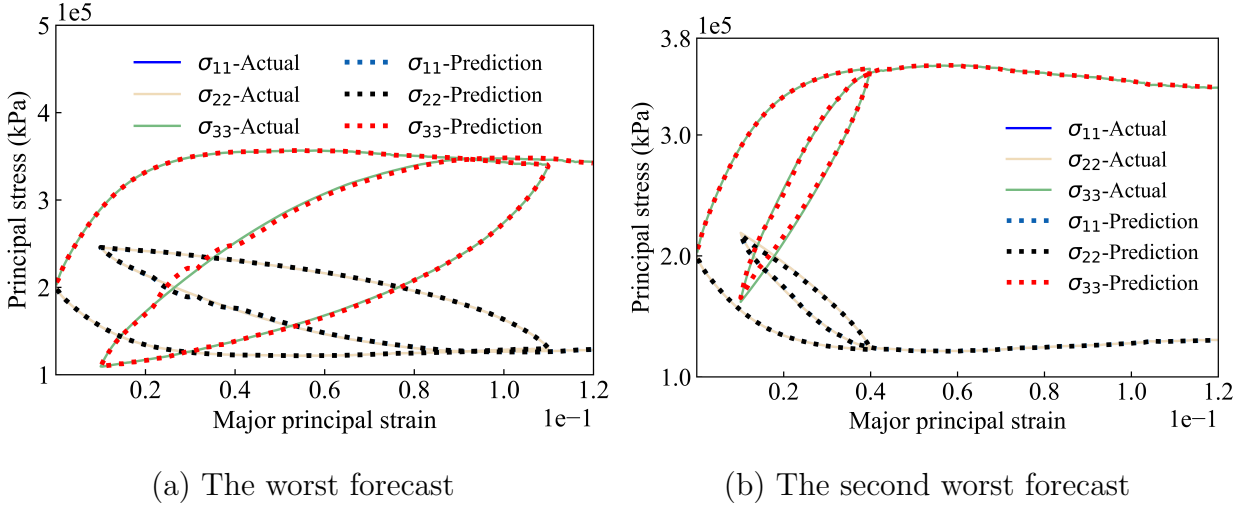


Figure 3.7: The worst predictions given by a DNN model in Case 1 with 60 groups of training data

algorithm may underperform conventional passive learning when the surrogate DNNs do not learn the real relation at all. The reasons can be that the active learning algorithm may converge to local or short-sighted optimal solutions due to the greedy nature of the algorithm. A greedy algorithm often fails to make the best decision although it always goes for the local best choice at each iteration.

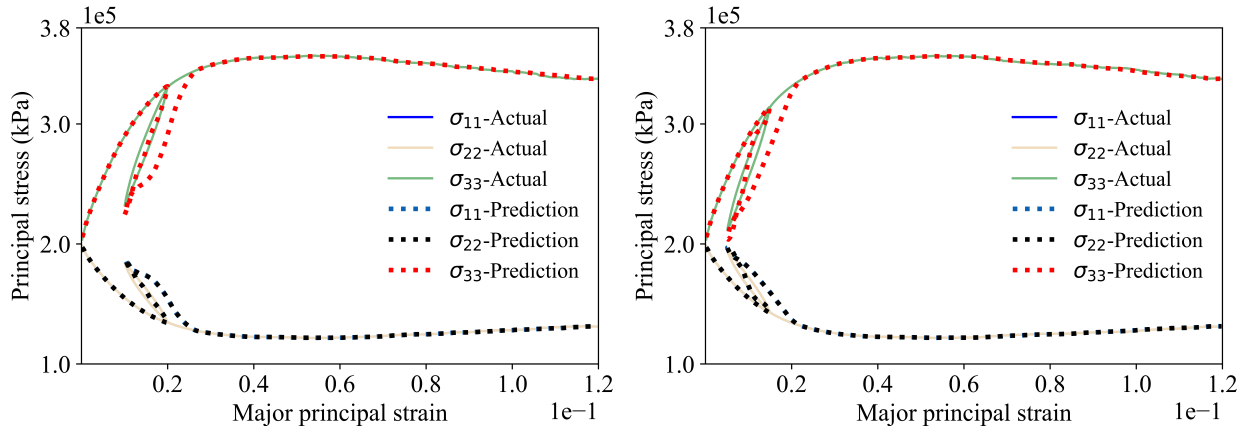
### 3.4 Interactive constitutive training and data labelling through active learning

In Section 3, all the stress-strain data is created before training models. The computational costs for preparing data are thus not reduced. In this section, the training of DNNs and the generation of data via microscale DEM simulations are performed in an interactive manner. Active learning is used to judiciously select data and only the most informative data is labelled, aiming to reduce the overall size of training datasets.

#### 3.4.1 Strain path falsification

In continuum-based numerical computation, the constitutive relation of a specific material receives a strain tensor and yields a corresponding stress tensor. The premise of using active learning is that inputs  $\mathbf{X}$  or unlabelled datasets  $\mathbf{D}_u$  are available. Thus, the primary concern of applying active learning in constitutive modelling is whether the strain sequences can be





(a) The worst forecast

(b) The second worst forecast

Figure 3.8: The worst predictions given by a DNN model in Case 3 with 60 groups of training data

constructed to generate an unlabelled data pool. Yet, granular material cannot undergo tensile and large compressive deformation. The admissible deformation scope for granular materials is restricted to a relatively narrow strain-stress space, compared to those of other solid materials (e.g., rubber and metal). In addition, the most common strain-stress path of granular materials experienced in a BVP is shear-type deformation, which is not easy to be artificially constructed through either proportional or random loading in the context of full-strain-dominated loading conditions.

Inspired by conventional triaxial experiments of granular soils, the in-situ stress condition of soils existing in the ground is approximated by imposing a constant confining pressure. Axial strain is applied to represent external disturbances on specimens. Such a hybrid boundary condition simplifies the complexity of artificially constructing admissible deformation for a granular specimen because the constant confining pressure ensures that the granular specimen does not undergo tensile or excessive isotropic compression loads. Also, only the one-dimensional axial strain needs to be fabricated. In this section, we adopt the most common experiment in geotechnical engineering as the prototype to perform deep active learning.

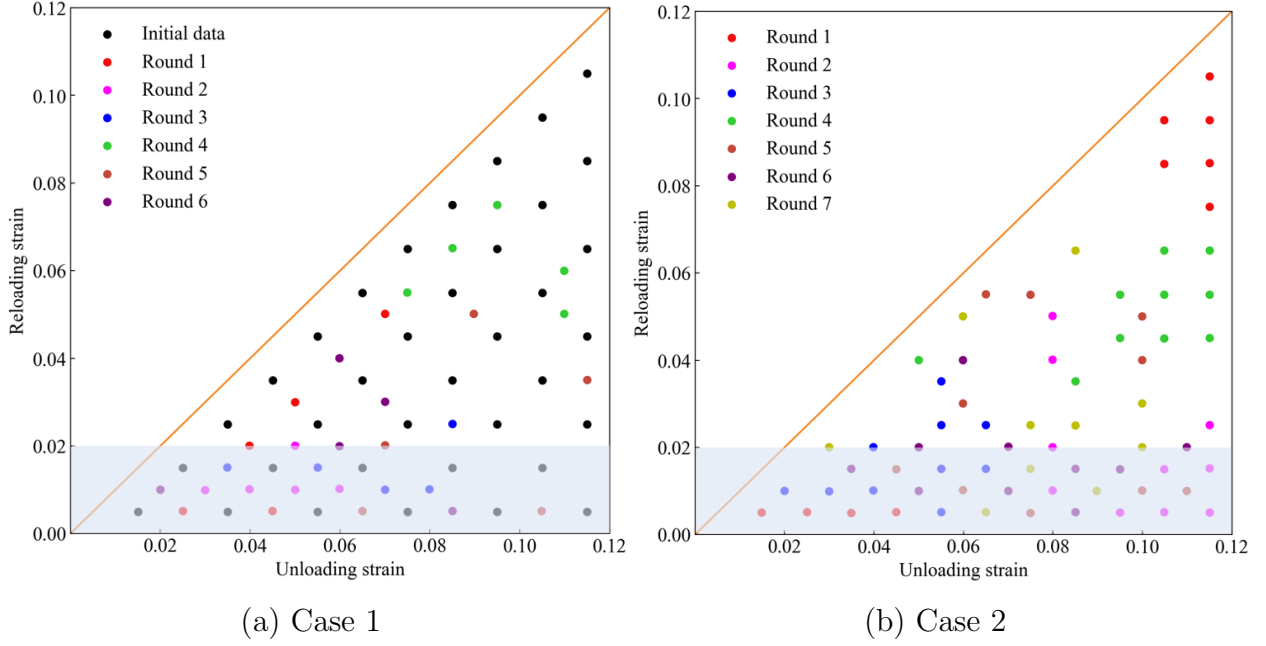


Figure 3.9: The added data specimens via active learning in each round

### 3.4.2 A surrogate error indicator for data-driven constitutive modelling

Although the granular specimen is not perfectly isotropic, especially during the loading process, we implicitly incorporate the isotropic assumption by homogenising stress responses in a triaxial testing condition. For an isotropic material, the strain state at a point in a 3D space should be depicted through at least three components (principal variables or invariants). When performing data-driven constitutive modelling, it is crucial to keep complete strain components (e.g. three principal strains) as inputs. For a conventional triaxial testing condition, the axial strain path can be falsified readily, but the lateral strain relates to the properties of materials and is thus difficult to construct artificially. Although the stress-strain prediction model should obey the relation shown in Fig. 3.10a: [Axial strain, lateral strain]  $\rightarrow$  [axial stress], we introduce a new mapping relation which links the axial strain and the axial stress directly (see Fig. 3.10b). The mapping requires only axial strain sequences as inputs thus one can generate extensive strain paths as unlabelled datasets.

The underlying idea is that both DNNs in Figs. 3.10 suffer from the same data scarcity issue, provided that they share the same training data. The predictions around the region where the training data is insufficient will have a larger variance than the region where data is sufficiently covered. Model B in Fig. 3.10b is not the desired candidate to forecast stress-

strain behaviour, but the active learning strategy introduced in Section 3.3 enables it to be a surrogate error indicator to identify the most helpful data for improving the current DNNs. Then these identified strain paths will be imposed on DEM specimens as the deformation of the boundaries to generate corresponding stress responses as new training data. Model A in Fig. 3.10a will make use of the resampled data to improve it. These procedures are repeated until a satisfactory predictive model has been fitted.

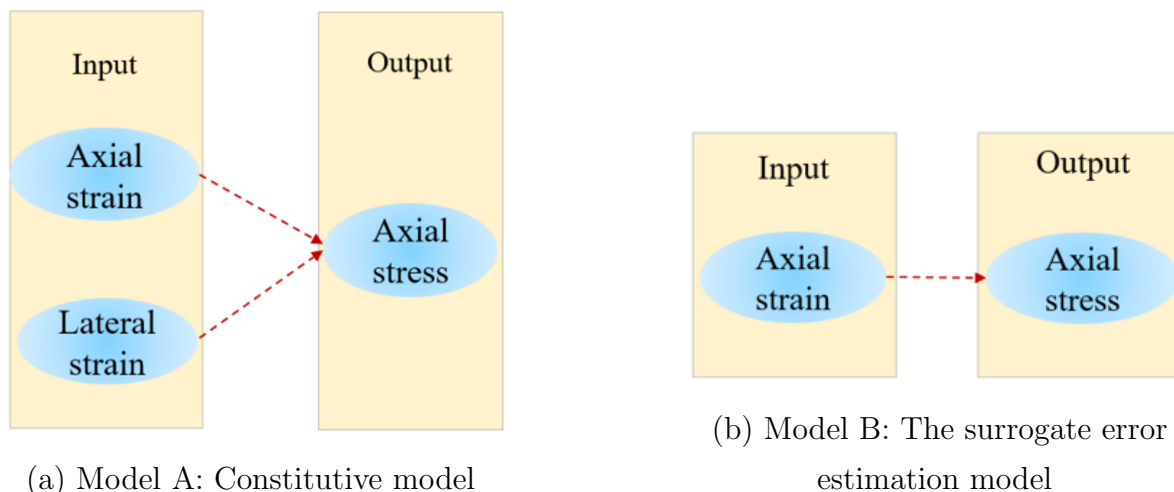


Figure 3.10: Inputs and outputs for data-driven stress-strain modelling in a conventional triaxial testing condition

### 3.4.3 Examination of the whole domain and adaptive resampling: an example with varied unloading-reloading cycles

The stress-strain curves with varied unloading-reloading loops are challenging to be predicted by traditional constitutive models. In this subsection, conventional triaxial loading cases with unloading-reloading loops are adopted to examine the capability of active learning in distinguishing complex strain paths. Constitutive modelling of path-dependent materials is a typical time series problem. To obtain unlabelled strain paths with unloading-reloading cycles, we artificially construct strain paths passing through selected unloading and reloading points, and then interpolate data points with equal spacing. These unloading and reloading strains can be described with horizontal and vertical coordinates in Cartesian coordinates, as shown in Section 3.3. For a strain-dominated loading condition, such a falsification of the axial strain path resembles the actual strain sequence in laboratory experiments, provided that a similar interval is selected. Through a large number of artificial strain paths, we can

develop a global domain examination and resampling strategy. The workflow is shown in Fig. 3.11 and the detailed procedure is described below:

- (1) Train a constitutive model (Model A) and three surrogate error models (Model B) based on the available stress-strain data pool.
- (2) Choose unloading-reloading strain points with a desired resolution in the whole domain, and then construct the strain sequence (unlabelled data) with even interpolation.
- (3) Utilise the three trained surrogate error indicators (Model B) with diverse initial weights and biases to predict stress values for these falsified strain paths.
- (4) Calculate the standard deviation (Eq.3.1) of each stress sequence prediction and rank the standard deviations of all strain sequences.
- (5) Select ten strain paths with the largest standard deviations from the previously ranked data pool using the batch-mode scheme (Fig. 3.4).
- (6) Perform DEM simulations using the ten selected strain paths as boundary conditions to generate corresponding stress responses. This procedure aims to tag the most instructive unlabelled data specimens.
- (7) Use the trained constitutive model (Model A) to forecast the ten newly generated DEM specimens and evaluate the prediction accuracy.

In the case that all the predictions have reached a satisfactory level, the current DNN model can be regarded as a reliable constitutive model. Otherwise, add these newly generated data to the training datasets, and repeat steps (1)-(7).

In the above workflow, Models A and B share the same network architecture and hyperparameters, but Model B only inherits part of the input features from Model A. Note that the number of network layers and neurons relates to the amount of training data and in principle, the DNN architecture and hyperparameters in each training phase for different models should be modified, because the optimal architecture may change slightly with a variation in training specimens. Yet the model architecture and hyperparameters are currently kept constant to reduce expensive tentative training costs when tuning these hyperparameters. In the future, some available packages, especially new algorithms in autoML, deserve to be explored to tune the network architecture automatically.

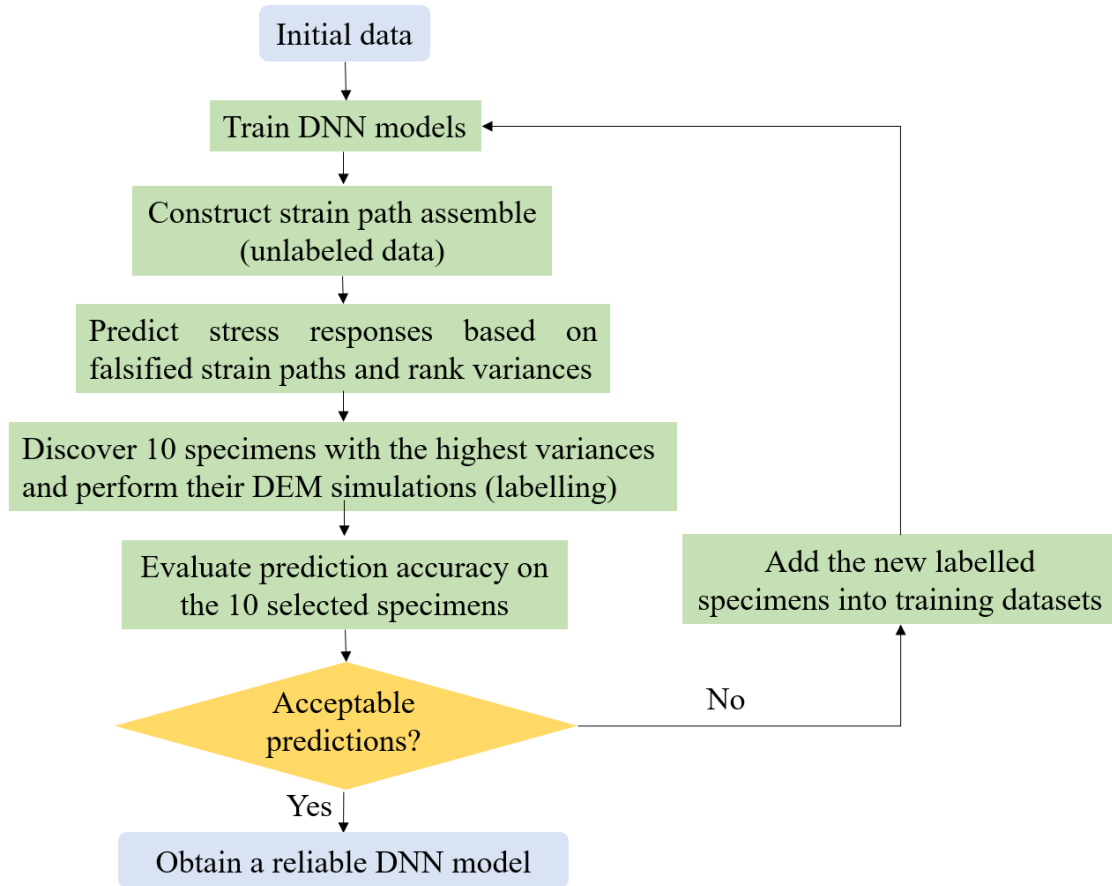


Figure 3.11: The workflow of global domain examination and resampling strategy

### 3.4.4 Verification of the interactive learning strategy

We consider training a data-driven model aiming to capture strain-stress mapping in a conventional testing condition with mutually different unloading-reloading points. The initial training data is randomly sampled as shown in Fig. 3.12a. Three DNNs fitted with these initial specimens are employed to examine unreliable predictions in the global domain with a strain interval of 0.003 shown in Fig. 3.12b.

All the training data is developed via DEM simulations of conventional triaxial testing. The simulated parameters are the same as the constant-p true triaxial testing in Section 3.3. The confining pressure is kept to 200 kPa during testing. A GRU network is used as a base DNN for constitutive training. Based on the initial random specimens shown in Fig. 3.12, the initial model, obtained by trial-and-error, includes two GRU hidden layers with 60 neurons and a dropout rate of 0.02 for each layer, resulting in a total of 33,843 weights and biases parameters. The two GRU layers adopt the tanh activation function while the output layer

uses a linear activation. The optimiser is the adaptive moment estimation (Adam) and the learning rate is 0.01. These networks are trained for 2000 epochs with a batch size of 256. MAE is used as the loss function.

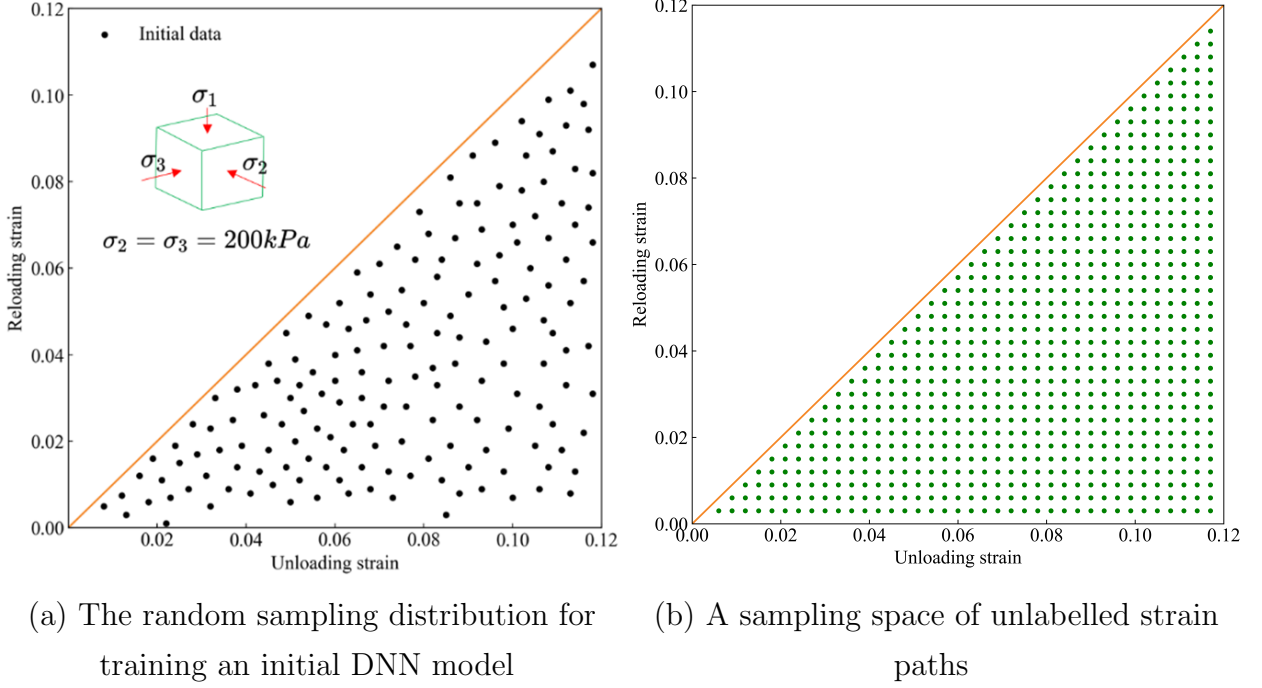


Figure 3.12: Sampling space for training and examination

Following the procedure described in Section 3.4.3, we train Model A for constitutive predictions while three error indication models (Model B) for seeking ten groups of most unreliable predictions. The ground-truth stress responses are obtained by performing DEM simulations of the selected strain paths. The forecasting accuracy of the selected samples is evaluated by comparing the predictions given by Model A and the ground truth from DEM simulations. The forecast results are shown in Table 3.3. Note that the first three columns are discovered by Model B while the last two columns are verified by Model A. The results demonstrate that almost all the predictions on these specimens are not sufficiently accurate and the average prediction score reaches only 0.71. In the next round, these new specimens will be added to the training samples and used for re-training DNNs.

To verify the proposed interactive constitutive training scheme, we also choose ten strain paths with the smallest standard deviations. The corresponding stress responses of these strain paths are also obtained via DEM simulations. The batch-mode scheme is used to create a wider range of representatives. Table 3.4 gives the prediction performances of these specimens. The results show that all the specimens are satisfactorily forecasted with a full

Table 3.3: The prediction performance of the selected ten samples with the greatest standard deviations but extracting neighbouring points

Unloading strain	Reloading strain	Standard deviation	Score	MAE
0.036	0.006	0.316	0.795	0.041
0.027	0.006	0.313	0.759	0.047
0.054	0.003	0.313	0.605	0.055
0.018	0.003	0.310	0.519	0.081
0.060	0.003	0.309	0.780	0.044
0.117	0.006	0.308	0.653	0.053
0.048	0.003	0.305	0.706	0.043
0.042	0.006	0.304	0.783	0.044
0.072	0.003	0.302	0.907	0.028
0.066	0.003	0.302	0.559	0.065

score.

The comparison between Tables 3.3 and 3.4 reveals that the proposed surrogate error indication model is useful for searching for the weakness of DNN models, even though actual stress-strain responses are unknown. For the committee-based active learning scheme, the main computational costs occur in training three DNNs as committee members (0.5 hours per DNN), while the querying process can be finished in a few seconds. In contrast, it takes a few hours to run a group of DEM specimens. By reducing the number of generating DEM specimens, the cost of training a reliable DNN model will be greatly saved.

By repeating the global domain examination and resampling scheme, the prediction accuracy of the data-driven constitutive model will be gradually enhanced. This repetition process terminates when the ten discovered specimens with the largest variances can be satisfactorily forecasted. In our current model, applying six rounds of resampling is found to be able to develop a sufficiently reliable data-driven constitutive model over the domain of interest. The worst forecasts after the sixth round of active learning are given as follows:

The newly added training specimens during each active learning round can be found in Fig. 3.14. The results show that the majority of newly added data are located in the shaded domain where the reloading strain is lower than 0.02. These points represent relatively large unloading-reloading loops. In contrast, only one specimen is selected as a reloading strain larger than 0.04. Again, the results confirm the importance of labelling the stress-strain

Table 3.4: The prediction performance on the ten specimens with the smallest standard deviations

Unloading strain	Reloading strain	Standard deviation	Score	MAE
0.075	0.066	0.041	1.0	0.011
0.063	0.054	0.044	1.0	0.013
0.069	0.057	0.046	1.0	0.013
0.081	0.069	0.061	1.0	0.009
0.060	0.048	0.062	1.0	0.011
0.069	0.063	0.064	1.0	0.017
0.051	0.042	0.064	1.0	0.010
0.081	0.075	0.067	1.0	0.012
0.057	0.042	0.069	1.0	0.017
0.063	0.060	0.069	1.0	0.016

curves with large unloading-reloading cycles.

## 3.5 Discussion

### 3.5.1 Significance of active learning

#### We know when our model does not know

When performing FEM simulations with data-driven constitutive models, a major concern is whether such a FEM simulation is sufficiently reliable. Following the idea of committee-based active learning, we can train multiple surrogate models simultaneously based on available datasets. If the predicted responses given by different DNNs vary from each other with a relatively large discrepancy, there might be a high risk of mispredictions. Then a procedure of labelling these strain paths is taken to enrich training datasets. The DNN model is refitted based on the new training dataset to improve the generalisation capability. Active learning enables DNN models to be self-learning with an intelligent sampling scheme.

#### Developing small-data-driven models with better generalisation capability

The data-hungry nature and the susceptibility to misprediction are two open challenges for a data-centric surrogate model. This study has confirmed the potential of active learning for



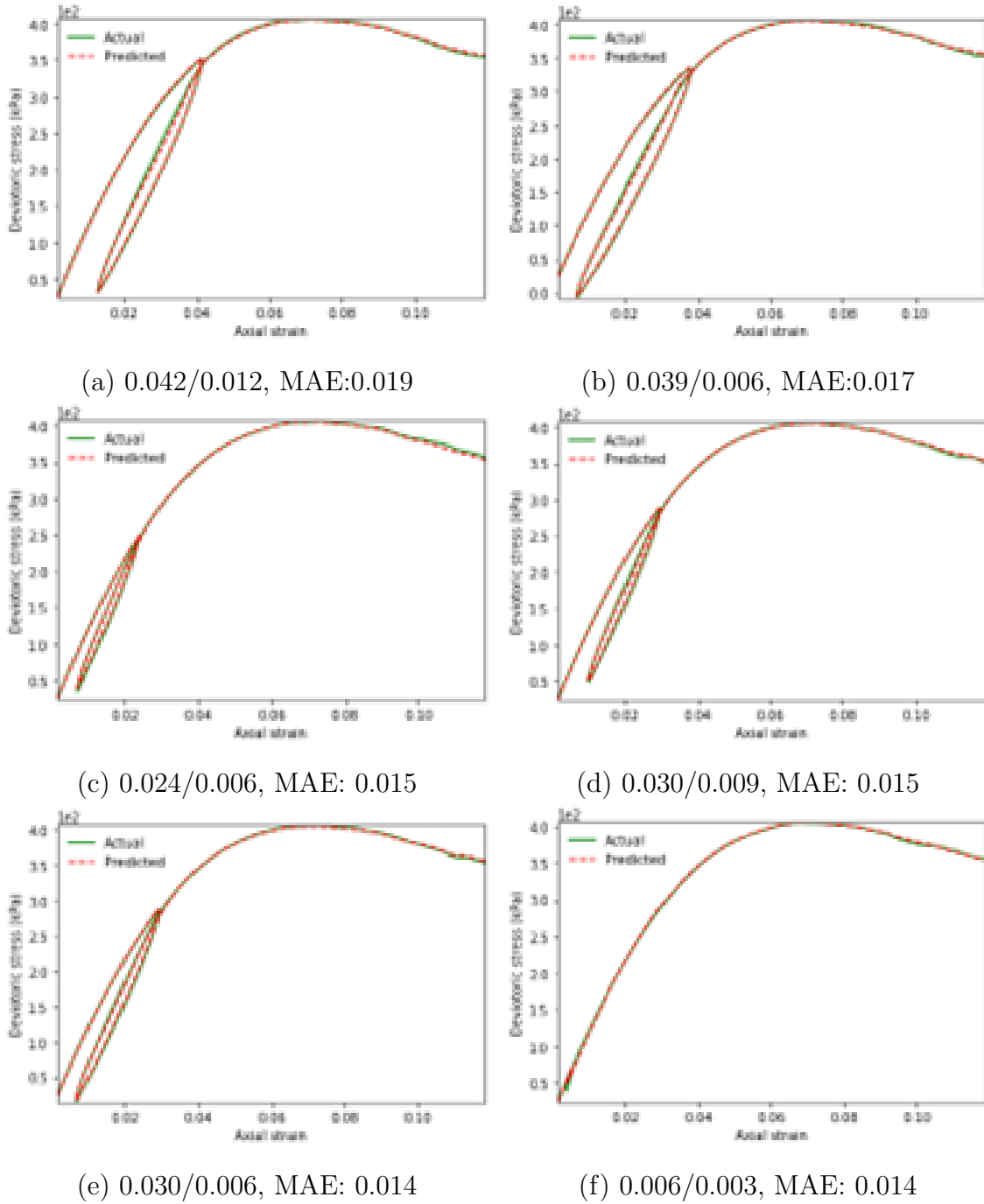


Figure 3.13: The worst forecasts discovered in the sixth round of active learning

developing a reliable and cost-effective data-driven constitutive model for granular materials. At each round of training, DNNs always discover and label the most hard-to-predict strain paths. This feature bypasses the need to label the less informative data. In addition, active learning can be a useful tool to combat imbalanced data, where most supervised learning

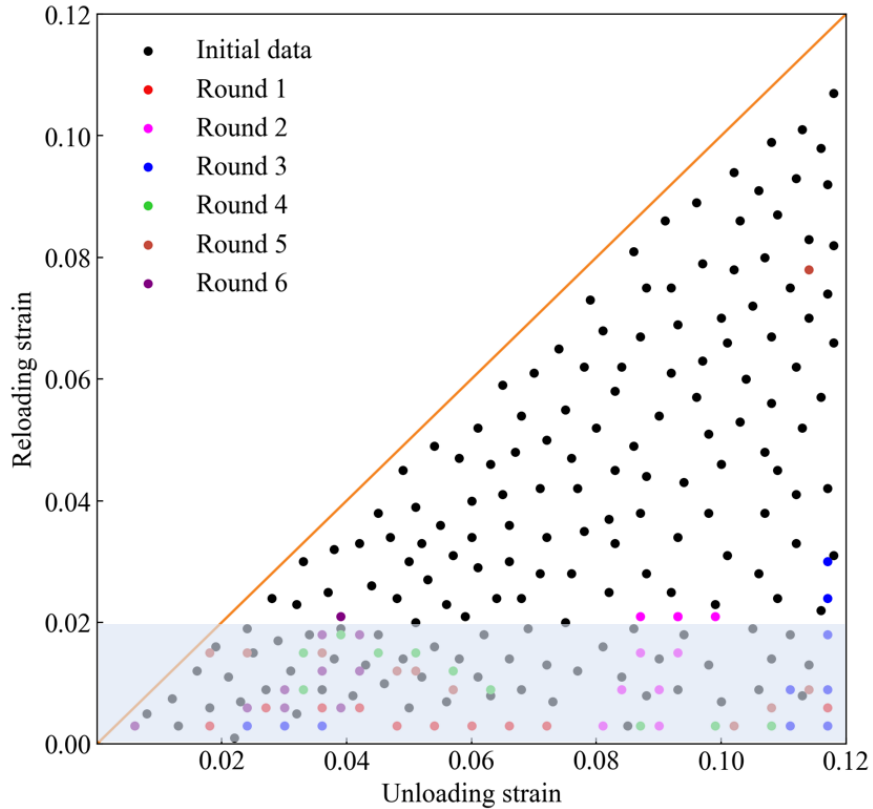


Figure 3.14: The newly added data specimens in each active learning round

models ignore and in turn tend to mis-predict the minority class, although the predictions of these minority datasets often dominate the success of the trained model. Active learning will automatically select these minority data if the trained DNNs mis-predict them.

### 3.5.2 Limitation of active learning and future work

#### Strain paths must be available to reduce labelling costs

A prerequisite of implementing active learning for data-driven constitutive modelling is that a large number of strain paths should be available. Although a fraction of them can be artificially constructed as demonstrated in Section 3.4, most strain paths experienced by Gauss points in a BVP are intertwined with stress responses and are thus hard to be falsified. This limitation no doubt restricts active learning from bringing possible disruptive progress for data-driven constitutive modelling. To address this issue, a possible remedy is by exploiting advanced machine learning tools, such as deep generative models, to learn the probability distribution of real strain-stress paths in typical BVPs. Then the learned generative model can produce massive strain paths that probably occur in engineering problems but never be

seen in existing datasets. The data pool can be used to verify the effectiveness of the model in a wider strain space through active learning.

### **Active learning tends to be sensitive to outliers in noisy datasets**

Active learning can identify the most unreliable forecasts while these recognised datasets might be outliers. If active learning deals with data with high-level noises or fluctuations, the algorithm may constantly add outliers to training datasets which will certainly deteriorate the predictive capability of models. For example, the post-peak stress-strain stage experienced by a granular RVE may exhibit remarkable “stick-slip” behaviour with locally up-and-down stress values because of the evolving breakage and reconstruction of strong force chains in deformed granular materials. The same situations apply equally to laboratory experiment-based data. In contrast, the datasets from analytical formulations are almost continuous and are easier to be fitted. However, closed-form expressions can simply provide approximated stress-strain data for certain materials. Thus one has to consider the value of training a surrogate model with such imperfect data.

To tackle the noisy data and enable a wider application of active learning, more research needs to be explored in the future. On the one hand, some advanced feature extraction measures should be harnessed to denoise or ‘wash’ datasets. On the other hand, some customised active learning algorithms should be designed to automatically remove the measurable fluctuated points.

## **3.6 Concluding remarks**

This study has developed a deep active learning-empowered data-driven constitutive modelling strategy, aiming to partially address two open challenges in the field: (1) use a small dataset to train a good predictive model, and (2) verify the reliability of a model without knowing ground truth. Three different application scenarios with RNNs and MLP-based DNN models are investigated in detail.

1. Committee-based active learning requires only a few committee members provided that each committee member has learned sufficient knowledge. The greatest advantage of active learning lies in its ability to detect inaccurate predictions without knowing the ground truth, rather than guiding the surrogate model to select data at the very beginning of the training phase.

2. The key to achieving interactive data-driven constitutive training, where training models and tagging data are dynamically performed, is that a pool of strain paths should be available beforehand. By using the proposed surrogate error indicator approach, some specific strain paths of certain proportional loadings can be artificially constructed.
3. The active learning algorithm is a useful tool to automatically discover the underlying order of importance for a pool of data. Such insight enables to develop a reliable model with as small datasets as possible by only preparing the most informative specimen for the current surrogate model.

# Chapter 4

## An FEM-NN framework for accelerating the multi-scale computation

In this chapter, our work introduces a neural network-based intrinsic structure model that serves as a proxy model for learning multiscale intrinsic structure relationships. The model is trained using raw data obtained from FEM-DEM multiscale simulations, enabling faster computations. Active learning is employed to estimate the significance of data points during network training and to establish an efficient resampling scheme that selects representative samples from the extensive dataset. To assess the performance of the proposed framework, biaxial simulations and retaining wall simulations are conducted. The simulations are carefully analyzed to identify simulation errors, and potential enhancements are discussed in detail.

### 4.1 FEM-DEM

Before introducing the FEM-ML framework, we introduce the FEM-DEM simulation method. The method was developed based on the open source partial differential equation solver Esyscript for the FEM part, and the open source software YADE for the DEM part. For the related procedures, please refer to FEMxDEM.

The FEM-DEM method for granular material calculation consists of two main parts: (1) the FEM calculation (macroscopic model) part; (2) the DEM calculation (fine material

RVE) part, which is carried out at the same time as the macroscopic calculation, and the procedure flow is shown in Fig. 4.1, with the steps as follows:

1. Macro-modelling and meshing in FEM;
2. Create a list of integration points  $\{G^{(n)}\}_{n=1}^{N_G}$  according to the element type corresponding and the total number of integration points  $N_G$ ;
3. Initialise a low-scale RVE list  $\{\text{RVE}^{(n)}\}_{n=1}^{N_G}$  that corresponds to the integration point list  $\{G^{(n)}\}_{n=1}^{N_G}$ ;
4. At every sub-step, apply boundary condition to the macro model, calculate the global force tensor and assemble the global stiffness matrix in FEM solver with the lower-scale information;
5. Obtain low-scale information such as initial stress  $\sigma$ , material matrix  $D$ , void ratio  $e$ , fabric tensor  $\Phi$ , *etc.* for each low-scale integration point;
6. Solve the FEM model for displacement increment  $\Delta u$  and the strain increment  $\Delta \epsilon$ ;
7. Apply the strain increment  $\Delta \epsilon$  to the lower-scale DEM  $\{\text{RVE}^{(n)}\}_{n=1}^{N_G}$ ;
8. Repeat (5)-(7) until the ratio of the displacement incremental paradigm to the total displacement incremental paradigm of the current load step  $\frac{\|\Delta u\|}{\|\Delta u_{total}\|}$  is less than the displacement error threshold,  $E_u = 0.01$ ;
9. Update the all of the lower-scale  $\{\text{RVE}^{(n)}\}_{n=1}^{N_G}$  and the configuration of the macro model  $\{x^{(n)}\}_{n=1}^{N_n}$  where  $N_n$  is the total number of nodes of the macro model;
10. Repeat (4)-(9) until the end of loading.

#### 4.1.1 Marco solver: FEM

**Control equation:** In the FEM part, the node displacement tensor  $\{u_j^{(n)}\}_{n=1}^{N_n}$  is taken as the basic unknown quantity and solved according to the equilibrium equations and co-ordination conditions [138]. Without considering the volume force, the control equation is expressed as:

$$\begin{cases} \sigma_{ij,i} + b_j = 0 & \text{in } \Omega \\ n_i \cdot \sigma_{ij} = \bar{t} & \text{on } \partial\Omega_t \end{cases} \quad (4.1)$$

where  $\sigma_{ij,i} = \nabla_{x_i} \cdot \sigma_{ij}$ ,  $b_j$  is the body force on in direction of  $j$ ,  $n_i$  is the out-normal direction on the traction boundary  $\partial\Omega_t$ , and  $\bar{t}$  represents the boundary traction.

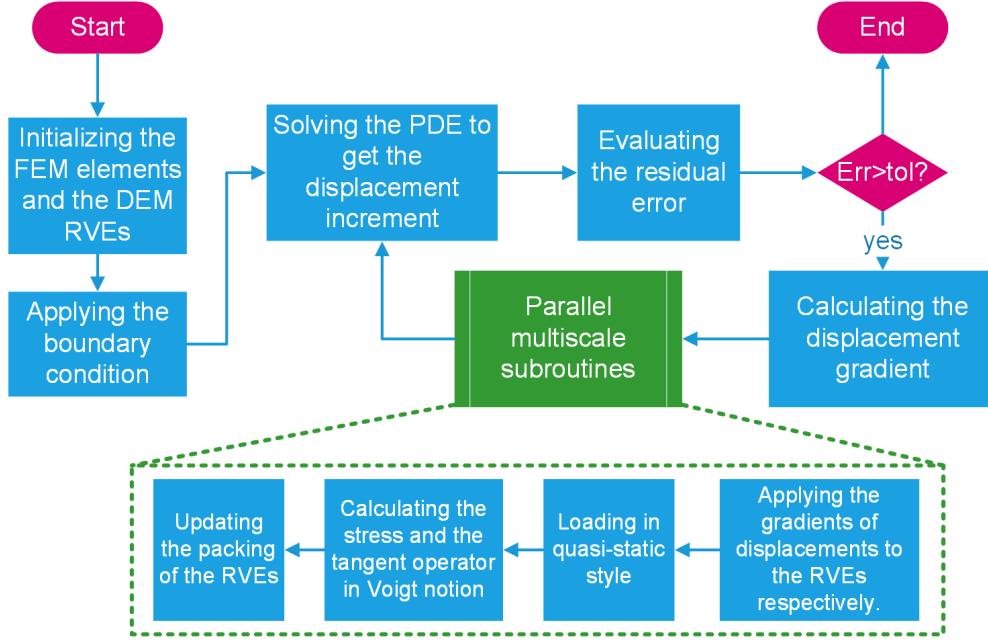


Figure 4.1: Flowchart of FEM-DEM multi-scale coupling calculation

The above is the strong form. In the Galerkin method, the weak form represented by the weight function and integration is shown as:

$$\int_{\Omega} w (\sigma_{ij,i} + b_j) d\Omega = 0 \quad (4.2)$$

where  $w$  represents the weight function (or the test function). The strong form is equivalent to having the above equation hold for any weight function  $w$ . The shape function  $N_n$  ( $n$  denotes the number of nodes in the element) is typically chosen as the weight function  $w$  because the shape function  $N_n$  automatically satisfies the displacement condition. Then Eq. can be written in the shape of:

$$\int_{\Omega} N_n (\sigma_{ij,i} + b_j) d\Omega = 0 \quad (4.3)$$

where  $N_n$  is the shape function, subscript  $n$  is the number of nodes used in this interpolation equation. In the case of isoparametric elements, the number of linear equations formed is equal to  $n * \text{Dof}$ , where Dof is each element's degree of freedom.

**Discretisation:** To achieve higher accuracy, the model is further discretised into individual elements connected by sharing nodes and the internal displacements of the elements are interpolated by means of element shape functions. By means of partition integration and divergence theorem, Eq. 4.3 can be expressed as an integral of the internal stresses and an

integral of the pressures on the boundaries as follows:

$$-\int_{\Omega^e} N_{n,i}\sigma_{ij}d\Omega + \int_{\partial\Omega_i^e} N_n n_i \sigma_{ij} d\Gamma + \int_{\Omega^e} N_n b_j d\Omega = 0 \quad (4.4)$$

where  $N_{n,i} = \nabla_{X_i} N_n$ . Note, because the model is discretised into a number of elements, the domain of the above formulas is within a single element  $\Omega^e$ . After substituting the traction boundary into upon equation we have:

$$-\int_{\Omega^e} N_{n,i}\sigma_{ij}d\Omega + \int_{\partial\Omega_i^e} N_n t_j d\Gamma + \int_{\Omega^e} N_n b_j d\Omega = 0 \quad (4.5)$$

Assume the nonlinear constitutive relationship as:

$$\sigma_{ij} = \sigma_{ij}^{(0)} + D_{ijkl}d\epsilon_{kl} \quad (4.6)$$

where  $\sigma_{ij}^{(0)}$  is the original stress before loading,  $d\epsilon_{kl}$  is the strain increment, and  $D_{ijkl}$  is the material tangent matrix.

Strain under the small deformation assumption is further introduced:

$$\epsilon_{kl} = \frac{u_{k,l} + u_{l,k}}{2} = \frac{N_{m,l}u_{mk} + N_{m,k}u_{ml}}{2} \quad (4.7)$$

where the gradient corresponds to the initial configuration  $u_{k,l} = \nabla_{X_l} u_k$  and  $N_{m,l} = \nabla_{X_l} N_m$ , and  $m$  indicates the number of node in one element.

Substituting Eq. 4.6 and 4.7 into Eq. 4.5, we have:

$$\begin{aligned} \int_{\Omega^e} N_{n,i}\sigma_{ij}d\Omega &= \left( du_{mk} \int_{\Omega^e} N_{n,i}D_{ijkl}N_{m,l}d\Omega + du_{ml} \int_{\Omega^e} N_{n,i}D_{ijkl}N_{m,k}d\Omega \right) / 2 \\ &+ \int_{\Omega^e} N_{n,i}\sigma_{ij}^{(0)}d\Omega \end{aligned} \quad (4.8)$$

Because of the symmetries of the tangent matrix  $D_{ijkl} = D_{ijlk}$ , we have:

$$\begin{aligned} du_{mk}N_{n,i}D_{ijkl}N_{m,l} &= du_{mk}N_{n,i}D_{ijlk}N_{m,l} \\ &= du_{ml}N_{n,i}D_{ijkl}N_{m,k} \end{aligned} \quad (4.9)$$

So we have:

$$-du_{mk}K_{n_jkm}^e - F_{n_j}^e + T_{n_j}^e + B_{n_j}^e = 0 \quad (4.10)$$

where the element stiffness  $K_{n_jkm}^e = \int_{\Omega^e} N_{n,i}D_{ijkl}N_{m,l}d\Omega$ , inner force  $F_{n_j}^e = \int_{\Omega^e} N_{n,i}\sigma_{ij}^{(0)}d\Omega$ , boundary force  $T_{n_j}^e = \int_{\partial\Omega_i^e} N_n t_j d\Gamma$  and the body force  $B_{n_j}^e = \int_{\Omega^e} N_n b_j d\Omega$ .



The element-specific Eq. 4.10 are assembled into global linear formulas through nodes shared between elements:

$$-du_{mk}K_{njkm} - F_{nj} + T_{nj} + B_{nj} = 0 \quad (4.11)$$

and can therefore be solved to obtain displacement increments  $du_{mk}$ .

### 4.1.2 Lower-scale solver: DEM

In the FEM-DEM framework, DEM calculations are utilised to determine the material's constitutive relationship and lower-scale structures at integration points. The strain of the macroscopic FEM solver at the integration point is inputted into the RVE model as a boundary condition for the lower-scale DEM computations. The information, such as the stress tensor  $\sigma_{ij}$  and the approximated tangent matrix  $D_{ijkl}$ , is obtained through parallel computation at each integration point and then passed back to the FEM solver for the calculation of nodal displacement increments.

To enhance simulation accuracy, particularly when dealing with a limited number of particles, periodic boundaries are employed in the RVE simulation to mitigate the impact of boundary effects. The contact between particles is governed by Hertz's contact model, and the calculation of normal and tangential contact forces is performed according to the following methodology:

$$\begin{cases} \mathbf{f}_n = k_n \mathbf{u}_n^c \\ \mathbf{f}_t = k_t \mathbf{u}_t^c \end{cases} \quad (4.12)$$

where  $\mathbf{u}_n^c$  and  $\mathbf{u}_t^c$  represent relative displacements in normal and tangent direction, respectively, and  $k_n$  and  $k_t$  are the normal and tangent contact stiffness, which can be presented as:

$$\begin{cases} k_n = \frac{G}{1-\nu} \sqrt{2\bar{r} - \delta_n^c} \\ k_t = \frac{2G}{2-\nu} \sqrt{2\bar{r} - \delta_n^c} \end{cases} \quad (4.13)$$

where  $G$  is the shear modulus,  $\nu$  is the Poisson's ratio,  $\delta_n^c$  is the normal overlap length between two particles and  $\bar{r} = 2(r_1 r_2)/(r_1 + r_2)$  represents the equivalent radius.

To facilitate the macroscopic calculations, the stress tensor needs to be transmitted to the FEM solver. This necessitates the utilisation of a homogenisation method that transforms the contact behaviour of the RVE into a stress tensor. The homogenisation formula is then

applied as follows, with reference to the works of Christoffersen et al. [125]:

$$\sigma_{ij} = \frac{1}{2V} \sum_{c=1}^{N_c} f_i^c d_j^c + f_j^c d_i^c \quad (4.14)$$

where  $V$  is the total volume,  $N_c$  is the total number of the inner contacts,  $f_i$  and  $f_j$  are the contact force vector and vector connecting centres of two particles. In this chapter, the FEM solver utilises the Newton-Ralphson non-linear iteration for the global balanced solution, so the tangent matrix is necessary for the global stiffness matrix evaluation as is shown in Eq. 4.10. Here, the tangent matrix is approximated via [28]:

$$D_{ijkl} = \frac{1}{V} \sum_{c=1}^{N_c} (k_n n_i^c d_j^c n_k^c f_l^c + k_t t_i^c d_j^c t_k^c f_l^c) \quad (4.15)$$

where  $n_i$  and  $t_j$  represent the normal and tangent directions, respectively.

To strike a balance between computational efficiency and the high fidelity of low-scale DEM simulations, each particle assembly should have a sufficient number of particles to replicate the mechanical properties of the granular material while minimizing computation time. Reference work [139] provides insights into the recommended number of particles.

Fig. 4.2 depicted the shear stress response of the particle assembly under simple shear. As the number of particles increases, the stress response gradually converges to a single value, resulting in decreased variance. When the number of particles reaches 700, the simulation achieves a fundamental convergence of stress results.

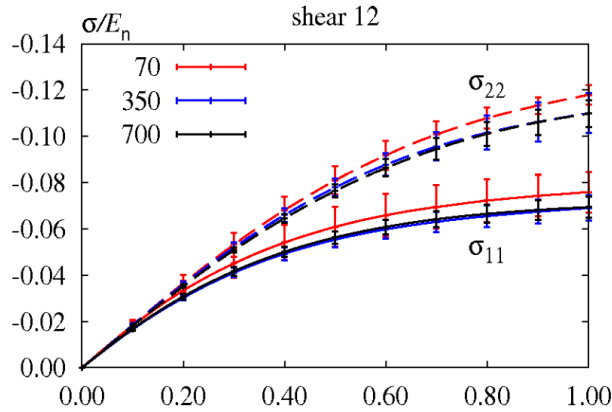


Figure 4.2: The shear stress with different numbers of particles in the assembly [3]

In the study conducted by Guo et al. [4], assemblies consisting of different numbers of particles (100, 200, and 400) were generated using the same particle gradation distribution.

These assemblies were subjected to isotropic loading until reaching an enclosing pressure of  $p = 100\text{kPa}$ . Fig. 4.3 illustrates the contact distributions and coordination number distributions for the three different particle number assemblies at the same pressure level.

Among them, the contact distribution rose diagram for the specimen with 400 particles closely resembles a circle. As the number of particles increases, the contact distribution becomes closer to a circular shape. However, due to chance, specimens with 100 and 200 particles exhibit varying contact distributions, with some directions having more contacts while others have fewer contacts.

The coordination number distribution plot demonstrates that the coordination number for particle assemblies with 400 particles is more concentrated around a mean value of 4.32. This indicates that the assembly with 400 particles mitigates the chance-induced contact anisotropy observed in lower particle number specimens.

In order to strike a balance between computational efficiency and the validity of the DEM simulation, a total of 500 particles per RVE are utilised in the calculations.

## 4.2 FEM-NN: neural network-based multi-scale method

Typically, traditional constitutive models based on assumptions can be adequately calibrated using conventional triaxial simulation experimental data. However, with the advancements in simulation techniques and measurements, a substantial increase in data availability has rendered these classical models insufficient to effectively handle the growing volume of high-quality data. In contrast to traditional constitutive models, data-driven models extract constitutive patterns directly from vast amounts of data, requiring minimal assumptions. This allows them to better accommodate and leverage the abundance of data in a more comprehensive manner.

By leveraging sophisticated network architectures and diverse nonlinear activation functions, the neural network model demonstrates excellent performance in nonlinear regression tasks involving high-dimensional data. Consequently, it becomes a valuable tool for reconstructing complex, high-dimensional nonlinear relationships in mechanical computations. This capability has been demonstrated in previous works such as Mozaffar et al. [98], Ghavamian et al. [105], Logarzo et al. [106], and Huang et al. [108]. Therefore, in the present study, the neural network model is employed to reproduce the macroscopic mechanical behaviour of granular materials.

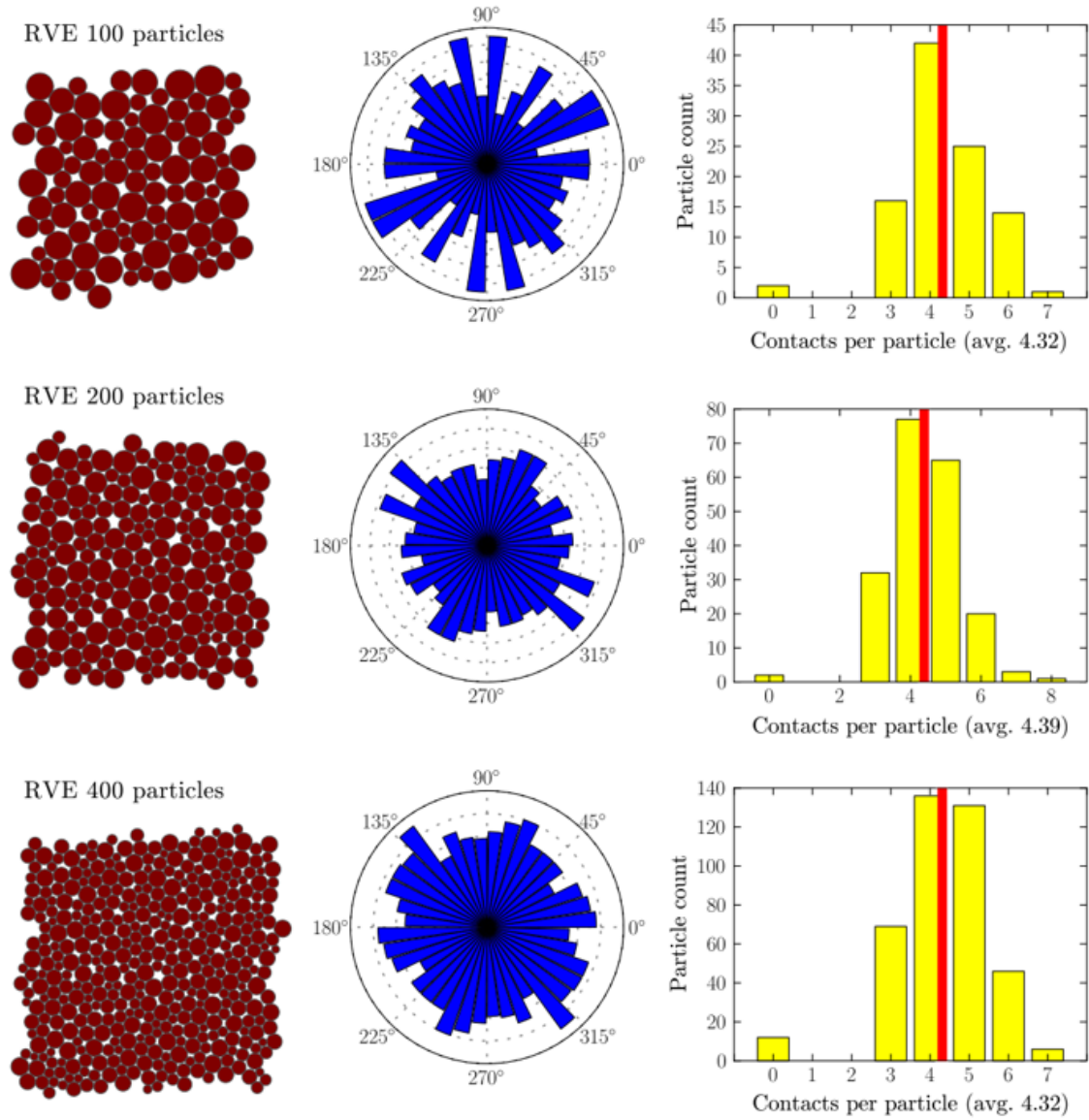


Figure 4.3: RVEs with different numbers of particles [4]

In FEM-DEM simulation, capturing the history effect of granular material involves storing the RVE model corresponding to the integration points in FEM model. However, this approach consumes a significant amount of memory. As the simulation progresses, the program’s memory usage continuously increases, eventually exceeding the memory limit and causing crashes.

To address this issue, the FEM-ML framework avoids the need to save the granular material RVE altogether, resulting in substantial memory savings. In this framework, a new variable must be introduced to represent the loading history of the granular material. Describing the plastic state of granular materials in DEM simulations using a single parameter is challenging because the division of strain into elastic and plastic components solely based on macroscopic quantities (such as stress, strain, void ratio, *etc.*) is not straightforward. Consequently, it is not feasible to characterise the historical state of granular materials solely in terms of plastic strain or cumulative plastic work, as typically done in traditional elastoplastic models.

#### 4.2.1 Neural network modelling of micro-RVE response

A crucial aspect for the further advancement of FEM-NN lies in effectively utilising a reduced number of features to comprehensively capture the influence of the loading history of granular materials. Neural networks offer a significant advantage in this regard, as their powerful mapping capabilities allow for the establishment of non-linear relationships between multiple physical variables, unconstrained by physical magnitudes. Leveraging neural networks’ capabilities, internal variables that represent the loading history of the granular material can be employed, even if they lack clear physical interpretations or fail to satisfy gauge requirements. As long as these variables can uniquely calibrate the historical state of the material, they can be utilised. In this study, the current state of the granular material is characterised using the cumulative value of the absolute strain increment during material loading:

$$\phi_{ij} = \sum_{k=0}^n |\Delta\epsilon_{ij}^{(k)}| \quad (4.16)$$

This set of variables is straightforward to implement and highly interpretable. Each element of this internal variable exhibits a monotonically increasing trend as the loading process advances. Consequently, these variables can be effectively employed to uniquely identify and characterise the current historical state of the granular material.

During the nonlinear iterations of FEM-DEM calculations, the approximate cut-line material matrix, calculated using Eq. 4.15, is utilised for evaluating the element stiffness matrix, as defined in Eq. 4.10. The computation of the low-scale RVE using dynamics is computationally intensive, with each integration point corresponding to a particle assembly. As a result, a substantial amount of data describing the state and mechanical properties of the granular material is stored in memory. However, only the macro-stress ( $\sigma_{ij}$ ) and the tangent matrix ( $D_{ijkl}$ ) are necessary for the FEM calculation.

In the neural network-based construction of the constitutive model, we compress the granular material information into a vector that contains only the essential features  $\{(\epsilon_{ij}, \phi_{ij}, \sigma_{ij}, D_{ijkl})^{(n)}\}_{n=1}^{N_G}$ , where  $N_G$  is the number of integration points. This compression reduces the complexity of training the network and enhances the trained network model's computational efficiency.

In the Newton-Ralphson method, the tangent matrix  $D_{ijkl}$  is necessary for nonlinear iterations. In network prediction, the tangent matrix can be evaluated via auto-gradient as:

$$D_{ijkl} = \frac{\partial \hat{\sigma}_{ij}}{\partial \epsilon_{kl}} \quad (4.17)$$

The Jacobian is automatically evaluated after the network predicts the stress. Whilst this technique of automatic differentiation has been employed in several studies [91, 105, 140], the author remains doubtful about its computational accuracy. As is shown in Fig. 4.4, without training the first order of the differentiation (Sobolev training), the network's predictions of the gradients are not obviously distinct from the ground truth.

The Jacobian is evaluated automatically once the network predicts the stress. Although this approach of automatic differentiation has been utilised in several studies [91, 105, 140], the author still harbours doubts regarding its computational accuracy. As depicted in Fig. 4.4, it is evident that without training the first order of differentiation (Sobolev training), the network's gradient predictions are significantly distinguishable from the actual values.

In the FEM-DEM simulation, the approximated secant matrix is actually computed based on Eq. 4.15 as an alternative to the tangent matrix. This is because the perturbation method used to evaluate the tangent matrix performs poor [4]. Fig. 4.5 shows the stress-strain curve obtained from the low-scale RVE simulation, which exhibits significant fluctuations. During the training process, the neural network model endeavours to capture all of the information even the noisy fluctuations. The network is insufficient to differentiate the constitutive patterns from the noise resulting from dynamic fluctuations. Even a well-trained neural

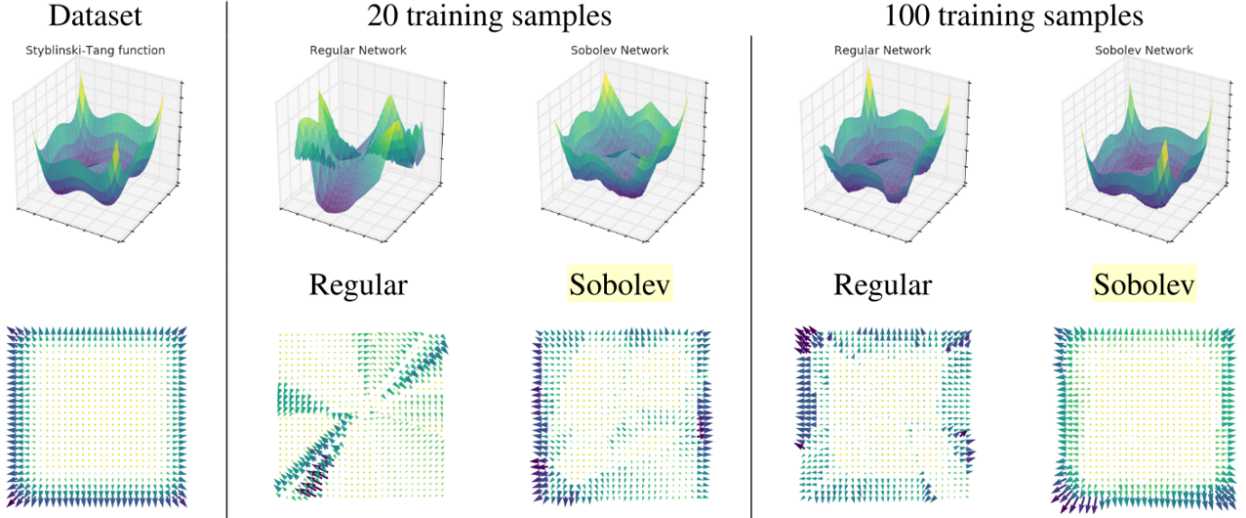


Figure 4.4: Comparison of network prediction under normal and Sobolev training [5]

network model yields a tangent matrix that lacks continuity or undergoes drastic changes at these fluctuating points. This behaviour deviates from the real tangent tensor, thereby significantly impacting the stability of the program calculations. Meanwhile, the secant matrix exhibits a relatively stable behaviour. It experiences some fluctuations, but its impact is relatively minor due to its larger base.

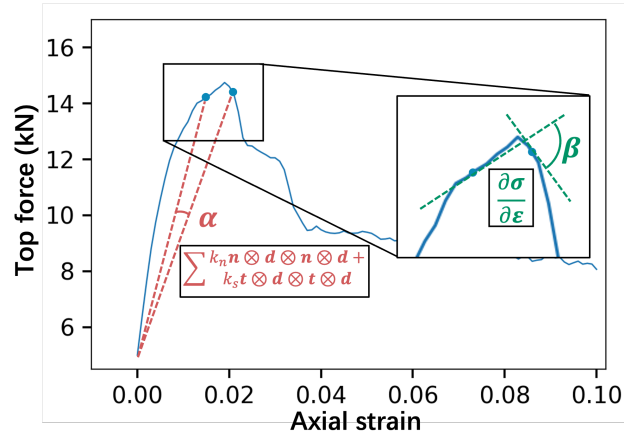


Figure 4.5: Comparison between tangent (green) and secant (red) matrices on the stress-strain curve

Since the strain tensor  $\epsilon_{ij}$  and the historical state quantity  $\phi_{ij}$  can be uniquely determined, the neural network is capable of establishing a mapping to this approximate secant matrix. Therefore, for the sake of computational stability, this work utilises the approximate secant matrix to perform FEM nonlinear iterations. The neural network is employed to establish

the mapping from the strain tensor and historical state to the stress tensor and secant matrix:

$$\hat{\sigma}_{ij}^{(n)}, \hat{D}_{ijkl}^{(n)} = \text{NN} \left( \epsilon_{kl}^{(n)}, \phi_{kl}^{(n)} \right) \quad (n = 1, 2, \dots, N) \quad (4.18)$$

where the superscript  $(n)$  indicates the step number.

Most of the data-driven research on stress-strain modelling tends to employ complex network structures, particularly those utilising deep learning methods [98, 101, 106, 107]. However, traditional elastoplastic models typically require only around 10 parameters to describe the stress-strain relationship within a specific framework, such as the Cambridge clay model [42] or the Nor-Sand model [49], where even only three or four key parameters are involved.

Using relatively simple neural networks can partially reduce the number of introduced parameters. In the work [141], as the network depth increases and the number of training iterations grows, the characteristic length of the corresponding Gaussian process diminishes. This implies that the network's predictions exhibit steeper variations. Even slight changes in inputs can result in significant deviations in outputs. On one hand, this allows the network to capture complex relationships more effectively. On the other hand, when training data is limited, this can lead to poor generalisation ability.

Considering feasibility, computational stability, and efficiency, we opted for a simple multilayer fully connected network to integrate with FEM computation. The neural network used consists of two hidden layers, each fully connected with 40 neural nodes. The ReLU activation function is applied, except for the output layer. The input layer comprises the current strain tensor  $\epsilon_{ij}$  and the historical variable  $\phi_{ij}$ . The output layer produces the corresponding stress tensor  $\sigma_{ij}$  and material matrix  $D_{ijkl}$  as is shown in Fig. 4.6.

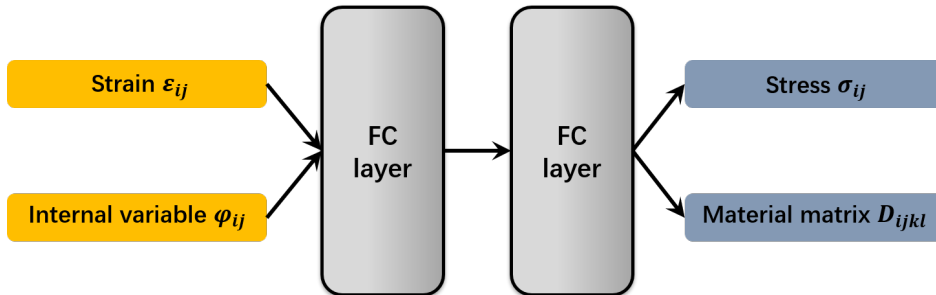


Figure 4.6: Network architecture



## 4.2.2 Training samples preparation

In general, machine learning models excel at interpolation but often struggle with extrapolation. To develop a machine learning-based constitutive model that is suitable for a wide range of strain paths, one approach is to include as many strain-stress data pairs as possible, effectively converting all "extrapolation" into "interpolation". However, this poses two challenges.

Firstly, generating a sufficient amount of training data to cover all possible strain-stress paths for a particular type of granular material is highly challenging. When considering all possible loading-unloading combinations, this task becomes practically impossible. This stands in stark contrast to traditional constitutive models, where model parameters can be calibrated using a relatively small number of conventional triaxial tests. Data-driven models do not rely on a predefined mechanics model but heavily depend on data.

Secondly, the existence of a large number of datasets also presents a challenge for training neural networks, as a large dataset requires expensive computational resources for training. It is crucial to utilise any useful conditions that can enhance the effectiveness of the training dataset, thereby reducing the need for extensive sampling and training costs.

For homogeneous materials, symmetry can be utilised to minimise the required material sampling in data-driven modelling [142–144]. By appropriately rotating the stress and strain tensors  $\sigma_{ij} - \epsilon_{ij}$  to their principal directions  $(\sigma_1, \sigma_2, \sigma_3) - (\epsilon_1, \epsilon_2, \epsilon_3)$ , the strain/stress sampling space can be reduced from six dimensions to three. This can also be presented as spectral decomposition:

$$t_{ij} = \sum_A^n t_{pr}^{(A)} n_i^{(A)} n_j^{(A)} \quad (4.19)$$

where,  $t_{ij}$  is a second-ordered tensor ( $t \in \mathbb{R}^{m \times m}$ ) with  $m$  dimensions, and  $t_{pr}^{(A)}$  and  $n_i^{(A)}$  are the  $A^{\text{th}}$  eigenvalue and eigenvector, respectively. Then we have the rotation matrix  $Q = [n^{(1)}, \dots, n^{(m)}]$ , where  $t_{ij} = Q^T \text{diag}([t_{pr}^{(1)}, \dots, t_{pr}^{(m)}])Q$ .

In the work of Tang et al. [109–111], this three-dimensional sampling space was further reduced to one dimension using the mapping method.

Note, that for non-coaxial granular materials, the strain and stress vectors cannot be simply mapped to the principal space. This is because the principal directions of the strains and stresses start to deviate since the granular material appears plastic (or non-affine defor-

mation). Due to the consideration of this non-coaxial nature, it is not proper to reduce the dimensions by spectral decomposition. In order to cover a sampling space as large as possible, this work introduces a Random Gaussian process to generate smooth random loading paths.

In some machine learning frameworks ([98, 106]), the random Gaussian process was used to generate microscale random loading paths. However, DEM simulations may return unreasonable results when the granular material assemblage is over-compressed or stretched under strain-controlled loading paths. Therefore, in our work, the random path is applied to a macroscopic model instead of a low-scale RVE to prevent the aforementioned overstretching or compression. Using this path for the macroscopic model will return a large number of integration point stress-strain data pairs  $(\sigma_{ij}, \varepsilon_{ij})$  at different locations in a single coupled FEM-DEM simulation, facilitating the generation of a large number of data sets.

**Gaussian Process:** A Gaussian process forms a sequence of associated random variables by linking these normally distributed random variables in space (or time), and this sequence is called a Gaussian process. Each point in this series of observations obeys a Gaussian normal distribution  $\mathcal{N}(0, \sigma)$ . The Gaussian process is determined by its mean vector  $m(x)$  and the covariance kernel function  $\kappa(x, x')$ .

The effect of the kernel function of the Gaussian process is shown in Fig. 4.7. Fig. 4.7a shows a diagonal matrix, indicating that the observations at each sample point are independent of each other, and in Fig. 4.7b, in addition to the values on the diagonal being non-zero, the values close to the diagonal are also non-zero, with the values of the elements decreasing as they move away from the diagonal. The values on this matrix represent the connection between the two points, and the more the two values are connected to 1, the higher the correlation. In the case where the diagonal matrix serves as the covariance matrix, a sequence of 20 independent variables is illustrated in Fig. 4.7c, while the associated random Gaussian process is illustrated in Fig. 4.7d.

The Gaussian process actually constructs multiple normal distributions into a sequence by means of a covariance matrix, which can be specified by the mean function  $m(x)$  and the covariance kernel function  $\kappa(x, x')$ , and can therefore be expressed as:

$$f(x) \sim \mathcal{GP}(m(x), \kappa(x, x')) \quad (4.20)$$

The mean value is kept as the initial confining pressure 100kPa, so  $m(x)$  is forming a vector with a constant value 1e5. Based on the description in Fig. 4.7, the curvature of the

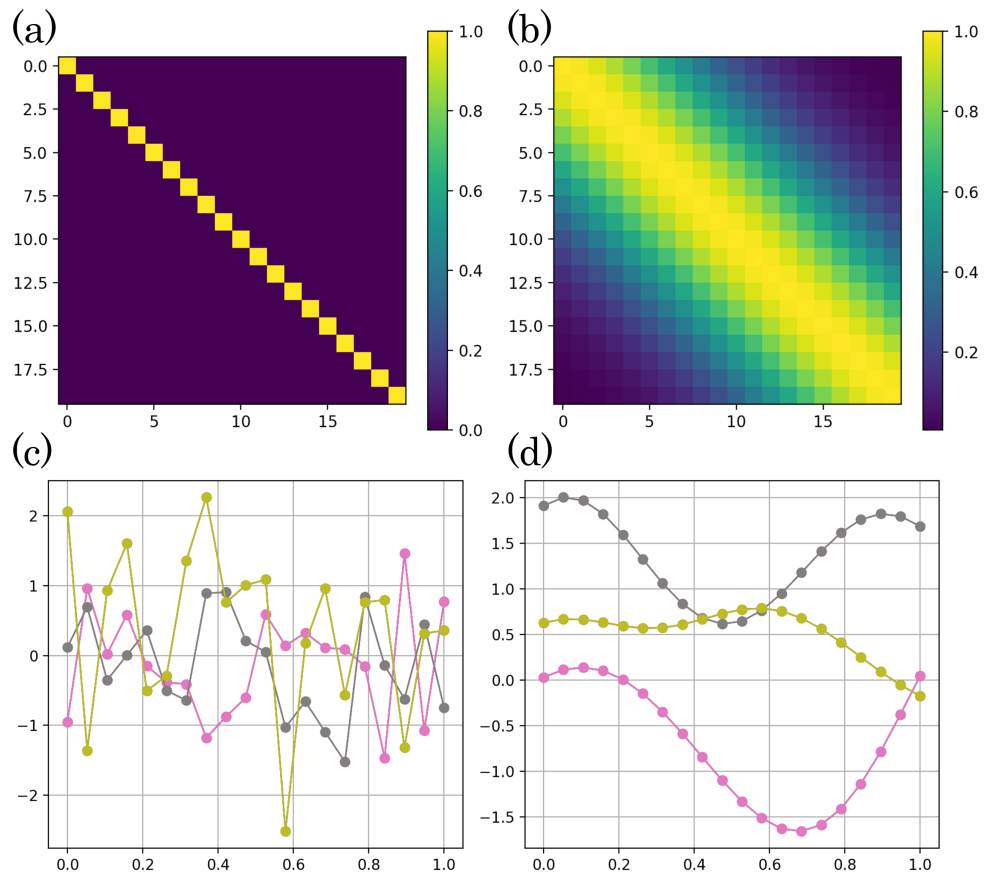


Figure 4.7: Gaussian processes corresponding to different covariance matrices are shown. Figures (a) and (b) display the covariance matrices. Figures (c) and (d) illustrate a series of random values sampled from Gaussian processes based on the covariance matrices in (a) and (b), respectively. In (c) and (d), the x-axis represents the points index (normalized to 0-1), while the y-axis represents their values.

generated random sequence can be controlled by specifying the parameters of the kernel function generation process. The covariance kernel function uses an exponential square function:

$$\kappa(x, x') = \exp(-v_c \theta_k(x) \|x - x'\|^2) \quad (4.21)$$

where  $x$  can be considered as the pseudo time can be expressed as  $x = [0.01, 0.02, \dots, 1]$  corresponding to 100 loading steps, and  $v_c$  is used to control the random path curvature. As shown in Fig. 4.8, when  $v_c$  increases, the band on the diagonal of the covariance matrix becomes narrower and the curvature becomes larger.  $\theta_k(x)$  is a function of  $x$ .  $\theta_k(0) = 0$  to ensure that the confining pressures start from an initial consolidation pressure of 100kPa, and then  $\theta_k$  is gradually increased to a certain value, the bigger the  $\theta_k$  the larger is the variance of the random path.

For this part of the dataset preparation, we used five different curvature coefficients  $v_c = [1.0, 2.0, 3.0, 4.0, 5.0]$ , and each curvature coefficient generates two sets of random paths as the macroscopic model confining pressures for the FEM-DEM biaxial simulations. The generated stress-strain sequences and tangent matrices will be saved and used as a network training dataset.

The total number of samples is summarised in Tab. 4.1, with the sum of generated data points approaching 17 million. Due to the nonlinear macro mechanical properties of granular materials, nonlinear iterations are required for each loading step. On average, each loading step requires about 20 iterations. Therefore, more than 2000 files of results are generated in 100 loading steps. Stress-strain and material matrix data for all integration points on each loading step are saved for building the network training database.

### 4.2.3 Active learning-based resampling

In the simulation we generated millions of sets of data, if we put that data into network training all at once, the training process will take up a lot of memory and it will be difficult for the optimiser to handle such a huge amount of data. Therefore we need to resample the data.

Random sampling, also known as passive learning is generally used to draw samples from large data sets. However, there are a large number of redundant data points in our data, such as neighbouring data points, which in fact contain basically the same information. It is expected to use a method that can screen the connections between data points, or continue to sample based on the information already learned by the network.

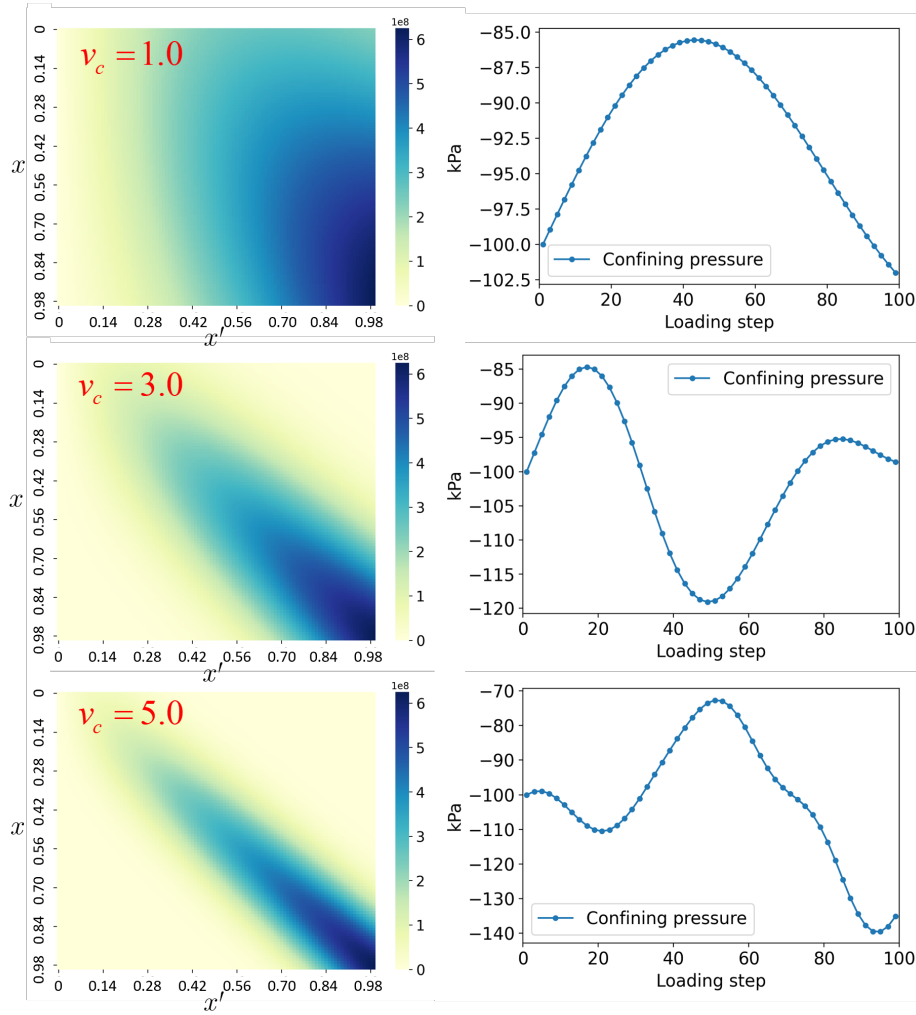


Figure 4.8: Random paths generated by the Gaussian process for  $v_c$ . The left column is the kernel and the right column is the generated random paths.

Table 4.1: Summary of datasets from FEM-DEM simulations

Random paths	Coarse grid	Medium grid	Fine grid
1	76,064	308,608	1,295,360
2	75,456	316,544	1,243,136
3	75,264	308,736	1,314,304
4	81,696	309,248	1,303,040
5	77,984	304,768	1,312,768
6	77,472	317,440	1,304,576
7	75,296	310,016	1,285,632
8	76,128	303,744	1,288,704
9	75,040	313,472	1,296,896
10	75,456	331,904	1,330,176
In total	16,864,928		

This approach to sampling is also known as the active learning resampling method [135]. Active learning models are introduced to effectively resample from massive data points. The key idea behind active learning resampling is to sample data points based on what has been learnt, using as few training samples as possible to achieve higher accuracy. In addition to resampling, active learning can also guide sample generation, especially if generating samples is costly. Since active learning methods are able to assess the predictive ability of the network on input without knowing the corresponding output, it is possible to determine the value of the data at that point for network training.

A large number of macroscopic stress-strain data pairs at integration points are stored in the dataset through the above coupled FEM-DEM multiscale simulations. Active learning is used to select samples efficiently, helping the optimiser to find the key dataset for network training, and mitigating the impact of redundant data points. The steps can be summarised as:

1. Randomly initialise the weights and biases of multiple networks with the same architecture and hyperparameters;
2. Training the network after initialisation using a partial dataset (i.e. the coarse FE grid here);
3. Evaluate the level of uncertainty on a new dataset (i.e., a fine FEM grid) based on

multiple networks that have been trained to obtain.

The active learning-based uncertainty is defined as:

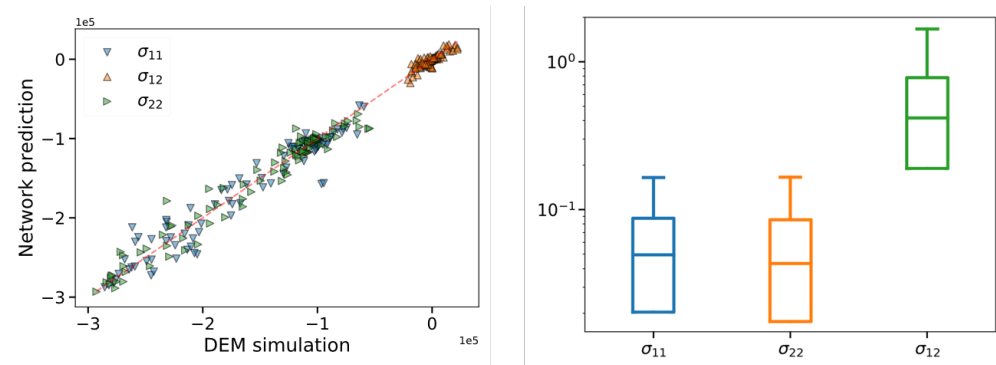
$$\begin{cases} \bar{y} = \frac{1}{M} \sum_{i=1}^M \text{NN}_i(x) \\ \xi = \sqrt{\frac{1}{M} \sum_{i=1}^M \|\text{NN}_i(x) - \bar{y}\|^2} \end{cases} \quad (4.22)$$

where the uncertainty  $\xi$  is defined as the standard deviation of the outputs from the parallel and randomly trained networks. The more confident the model predictions are, the closer the individual predictions here are to each other, and conversely, where the model is unsure, the predictions will vary widely. After evaluation using the above equation, points with high uncertainty are appended to the original training sets to retrain the network. The active learning method acts as a detector for finding locations where the randomly trained model performs poorly, and where the training dataset is insufficient.

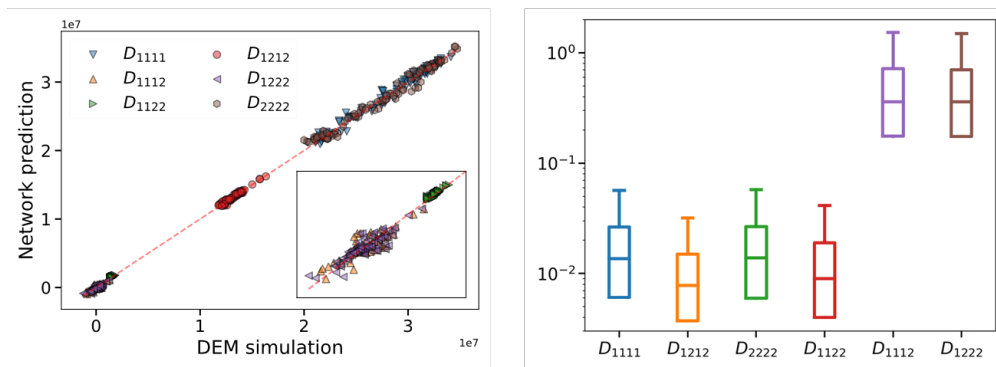
The neural network part of the model is built, trained and called in the PyTorch framework. After training, the network model was examined for prediction effectiveness and error, as shown in Fig. 4.9. In most cases, the trained neural network model is able to predict with satisfactory accuracy, especially in predicting the four components of the tangent matrix ( $D_{1111}, D_{2222}, D_{1212}, D_{1212}, D_{1122}$ ) and the two stress components ( $\sigma_{11}, \sigma_{22}$ ). However, the model performs poorly in the prediction of  $\sigma_{12}, D_{1112}, D_{1222}$ . In isotropic hyperelastic models or quasi-elastic phases of particle assemblages, the components of  $D_{1112}$  and  $D_{1222}$  should be 0 [145]. However, in the DEM simulations, the values in these two directions always remain as a smaller value of no regular noise due to the heterogeneity of the granular system and the effect of noise caused by larger components in other directions. In the FEM calculations, the prediction errors on these three components have less impact.

#### 4.2.4 FEM-NN coupling

The computational flow of the FEM-NN framework is shown in Alg. 2. There is no need to initialise or update the particle ensemble RVE during the computation process because the neural network will directly compute the stress tensor  $\sigma_{ij}$  and the approximate secant matrix  $D_{ijkl}$  based on the strain tensor  $\epsilon_{ij}$ , and the loading history  $\phi_{ij}$ . The proposed network-based constitutive model functions similarly to the conventional constitutive model in the FEM solver.



(a) Prediction of the stress vector



(b) Prediction of the material matrix

Figure 4.9: Comparison of neural network prediction results with DEM simulation



Typically, the nonlinear iteration step starts on the previous iteration step. Notice here that each iteration step starts with the solution of the previous load step instead of the previous iteration step. This is because starting at the previous iteration step requires the model to give an accurate tangent matrix. But the estimated tangent matrix given by the network is not sufficient to fulfil this condition. So in the FEM-NN framework, it will be more difficult for the nonlinear iteration to converge if the iteration step starts with the previous iteration step.

---

**Algorithm 2** The FEM-NN solver

---

**Require:** Discretized FEM model, well-trained network NN

```

1: Initialisation,  $\sigma_{ij}^{(0)}, D_{ijkl}^{(0)} = \text{NN} \left( \epsilon_{ij}^{(0)}, \phi_{ij}^{(0)} \right)$ 
2: for  $n = 1, 2, \dots, N$  do ▷ Load step
3:   Apply boundary conditions of step  $n$  to the FEM model
4:    $du_j^{(n,m)} \leftarrow$  solving Eq. 4.10 with  $\sigma_{ij}^{(n-1)}$  and  $D_{ijkl}^{(n-1)}$ 
5:    $m = 0, e_u = 1.0$  ▷ initialise the substep number and displacement error
6:   while  $e_u > e_{tol}$  do ▷ Iteration step
7:      $d\epsilon_{ij}^{(n,m)} = 0.5 \left( du_{i,j}^{(n,m)} + du_{j,i}^{(n,m)} \right)$  ▷ Under the small deformation assumption
8:      $\epsilon_{ij}^{(n,m)} = \epsilon_{ij}^{(n-1)} + d\epsilon_{ij}^{(n,m)}$ 
9:      $\phi_{ij}^{(n,m)} = \phi_{ij}^{(n-1)} + |d\epsilon_{ij}^{(n,m)}|$ 
10:     $\sigma_{ij}^{(n,m)}, D_{ijkl}^{(n,m)} = \text{NN} \left( \epsilon_{ij}^{(n,m)}, \phi_{ij}^{(n,m)} \right)$  ▷ Network prediction
11:     $du_j^{(n,m+1)} \leftarrow$  solving Eq. 4.10 with  $\sigma_{ij}^{(n,m)}$  and  $D_{ijkl}^{(n,m)}$ 
12:     $e_u = \frac{\|du_j^{(n,m+1)} - du_j^{(n,m)}\|}{\|du_j^{(n,m)}\|}$  ▷ Update the displacement error
13:     $m = m + 1$ 
14:  end while
15:   $u_j^{(n)} = u_j^{(n-1)} + du_j^{(n,m)}$  ▷ Update the variables
16:   $\sigma_{ij}^{(n)} = \sigma_{ij}^{(n,m)}$ 
17:   $D_{ijkl}^{(n)} = D_{ijkl}^{(n,m)}$ 
18:   $\epsilon_{ij}^{(n)} = \epsilon_{ij}^{(n,m)}$ 
19: end for

```

---

### 4.3 Numerical examples

This study employs two examples to showcase the accuracy, stability, and applicability of the FEM-NN method. The first example involves a biaxial compression test with a smooth

top surface. We vary the mesh densities and incorporate active learning for resampling. We compare the macroscopic computational results, such as top force and total volumetric strain, and analyse the stress-strain relationship at integration points. The second example involves simulating a retaining wall pushing the fill backwards. The objective of this case is to evaluate the generalisability and explore potential opportunities for further enhancement.

### 4.3.1 Bixial compression

To verify the effectiveness of the FEM-NN, biaxial simulations with different levels of mesh densities were conducted. Fig. 4.10 illustrates the division of the model into three meshes: coarse (2x4), medium (4x8), and fine (8x16). Each element within the meshes contains four integration points.

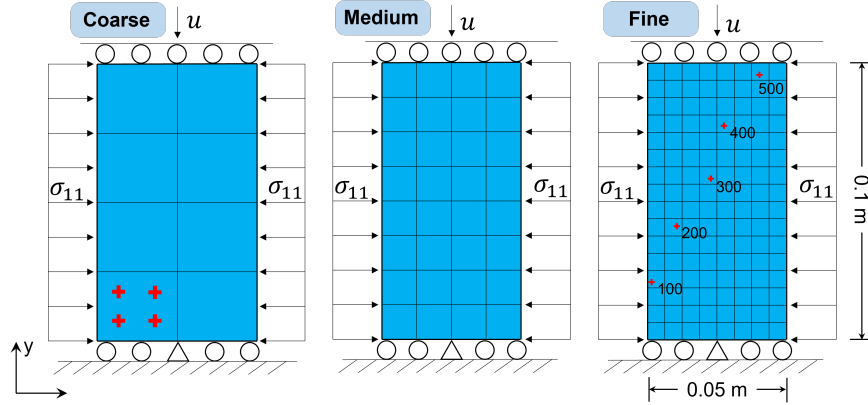


Figure 4.10: Biaxial compression simulation with different mesh densities

Linear vertical downward displacement-controlled loading was applied at the top boundary of the biaxial model until a macroscopic axial strain achieved 0.1. The random confining pressures for model loading were generated using the random Gaussian process described in Sec. 4.2.2. For each of the meshes, 10 simulations with random confining pressures were prepared. The parameters of the DEM-simulated granular materials are presented in Tab. 4.2.

Table 4.2: Parameters of the lower-scale DEM simulations

Density (kg/m <sup>3</sup> )	Young's modulus (MPa)	Poisson's ratio	Frictional coefficient	Damping
2650	600	0.8	0.5	0.1

Before training the networks with the complete FEM-DEM simulation datasets, several models were trained individually using biaxial simulation datasets from different grid resolutions and then evaluated using FEM-NN simulations. Although the networks were initialized with the same hyperparameters and weights, the training data came from coarse-grid simulations, fine-grid simulations, or a combination of both. Table 4.3 outlines six training and FEM-NN simulation scenarios, each involving different grid divisions for preparing the training data and conducting the simulations. The process for combining mixed data is illustrated in Fig. 4.11.

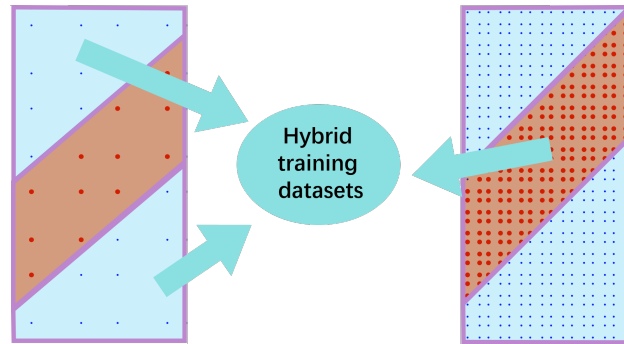


Figure 4.11: Mixed data set generation programme

Table 4.3: Summary for the different training and testing cases

Cases	Training dataset collected from	Level of mesh for FEM-NN simulation
A	Coarse	Coarse
B	Fine	Coarse
C	Mixed	Coarse
D	Fine	Fine
E	Coarse	Fine
F	Mixed	Fine

Fig. 4.12 indicates that all the results exhibit good agreement with the FEM-DEM simulations, except for Case E. The displacement comparison depicted in Fig. 4.13 reveals that the shear band in the FEM-DEM simulation is narrower when using a finer mesh. However, the displacement results for Case E resemble those of Case A. In both Cases A and E, only datasets collected from coarse grid simulations are utilised to train the network. Integration points in the coarse mesh are considerably sparse, resulting in insufficient patterns of the constitutive relationship for fine-meshed simulations, especially on the shearing band.

Conversely, due to the network’s excellent interpolation capabilities, the stress tensor and tangent matrices within the shear zone in the fine-grid simulation can be approximated using the network trained on the coarse dataset. Although Case E utilises a fine mesh for macroscopic simulation, the constitutive relationships are derived from the coarse dataset which was used to train the model for both Case A and Case E. As a result, the shear bands observed in Case E appear to be an interpolation of the results from Case A on a finer grid.

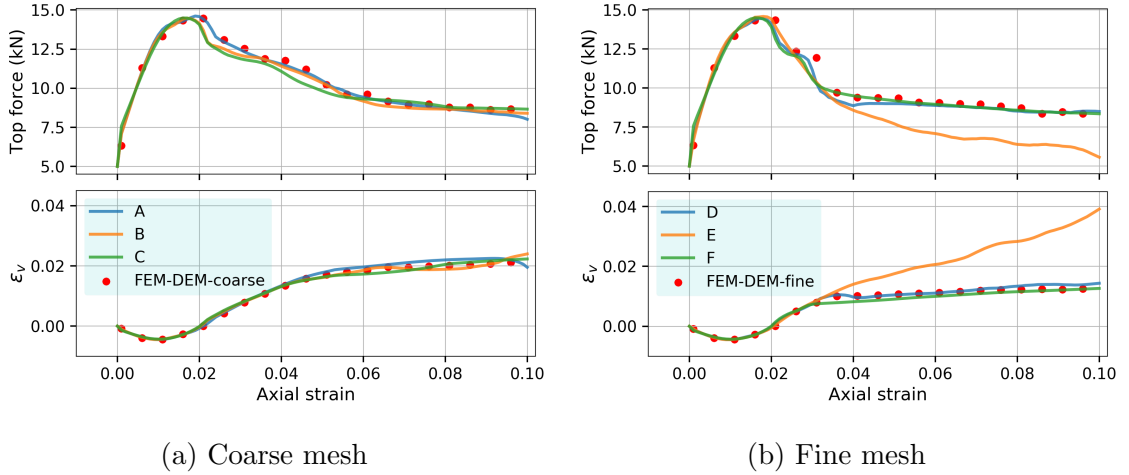


Figure 4.12: Curves of top forces and global volumetric strain corresponding to different Cases.

Since the strain distributions in the upper and lower triangular regions of the biaxial model were very similar for both coarse and fine meshes, we focused on integration points specifically within the shear zone. In Case F, the network model was trained using a hybrid sampling approach, as shown in Fig. 4.11. The dataset for the shear zone was extracted from the fine mesh simulation, while data for the upper and lower triangular regions of the domain was obtained from the coarse mesh simulation, due to the smooth and relatively low shear strains in these regions.

Fig.4.13 demonstrates a good agreement between the displacements obtained in Cases C and F and the FEM-DEM simulations. It emphasises the importance of utilising an appropriate training dataset with a well-designed sampling method to ensure effective network training.

The data points within the shear band region play a crucial role in addressing the current problem. Conversely, the constitutive relationships represented by data points in the upper and lower triangles of the fine-meshed model can be adequately captured by the data points

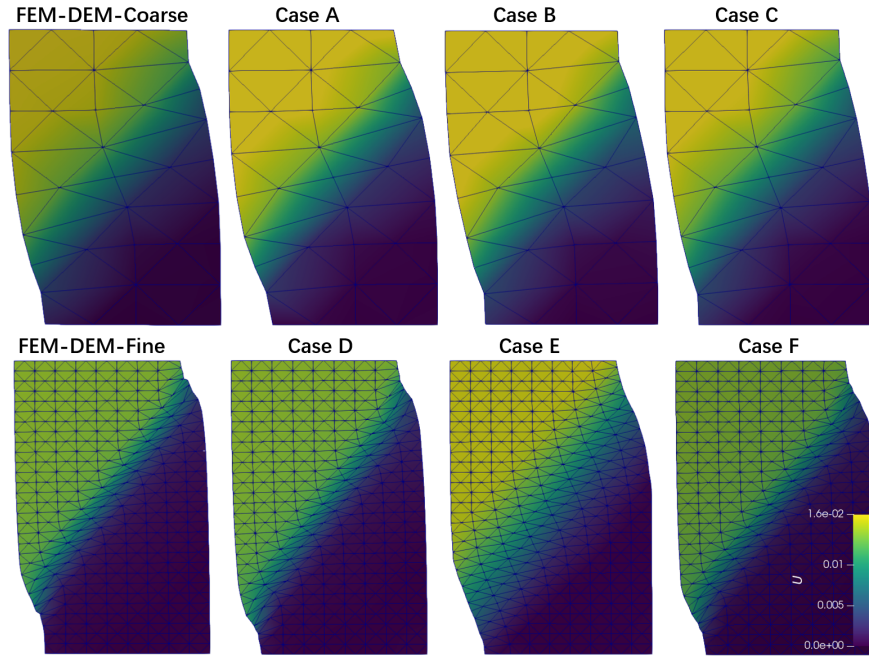


Figure 4.13: Comparison of displacement results corresponding to different Cases

from the coarse-meshed model. To better prove these characteristics, the uncertainty-based active learning scheme, as discussed in Sec. 4.2.3, is employed for automatic resampling. The detailed procedure is presented in Fig. 4.14 and explained below:

1. Choose five separate networks with the same architecture and hyperparameters as described in Sec. 4.2.3 but pre-trained based on the coarse datasets with different randomly initialised weights and biases;
2. Use the five pre-trained models to predict the mechanical responses of the data points of the fine mesh;
3. Evaluate the prediction uncertainty level at each data point based on the five predictions by Eq. 4.22.
4. Add the first 30% data points with the highest uncertainty levels to the training datasets used in the pre-training phase.
5. Re-train one pre-trained model on the enriched dataset and evaluate its performance in FEM-NN simulation.

Fig. 4.14 clearly illustrates prediction uncertainty, which is particularly pronounced within the shear band region. This suggests that data points in the shear band have a

greater impact on network predictions. The areas with high uncertainty align closely with the red data points in Fig. 4.11, which accounts for the improvement observed in Case F. Thus, active learning automatically identifies regions where the trained network model underperforms, indicating that additional training points should be added in these high-uncertainty locations.

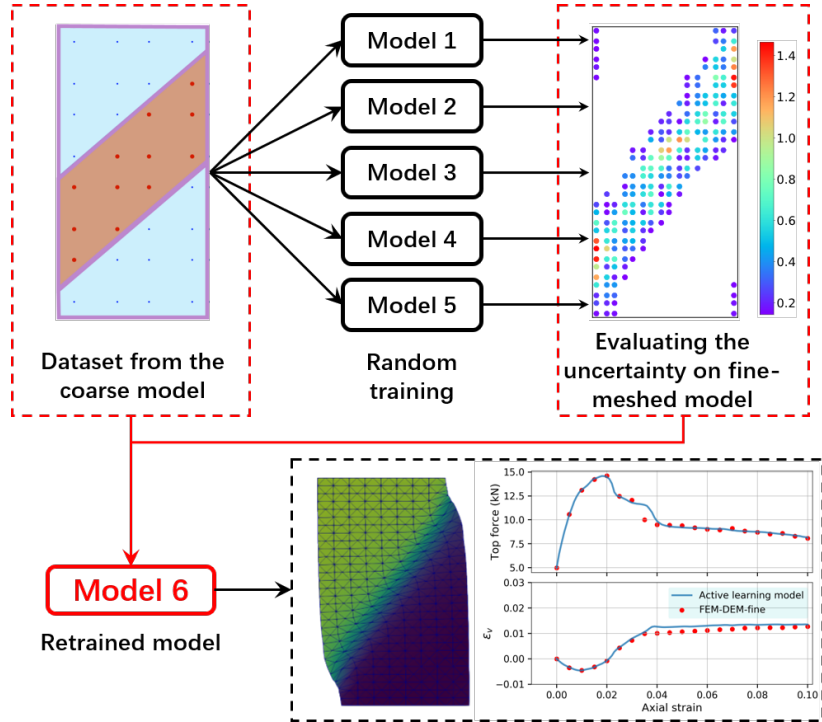


Figure 4.14: Flowchart depicting the process of network training and FEM-NN simulation using active learning for resampling.

In the active learning resampling process described above, newly added data points are primarily located within the shear band. This observation suggests that shear strain magnitude could serve as an alternative indicator for resampling. To assess the viability of using equivalent shear strain for active learning resampling, we calculate the equivalent shear strains at all integration points for each load step. The first 30% of the data points with the highest uncertainty levels are selected, as illustrated in the top row of Fig. 4.15.

The figure illustrates that the shear band or strain localisation begins to emerge around load step 29 and fully develops from load step 59 onwards. The integration points with the 30% highest shear strains for these load steps are concentrated around the shear band, which aligns with the points with higher uncertainty predicted by the active learning.

A specific indicator can be a straightforward and effective alternative for resampling,

provided the datasets are well-understood. However, identifying an appropriate indicator can be challenging, particularly when dealing with high-dimensional input data or diverse datasets. It's crucial to recognize that the shear strain indicator is problem-specific. For other types of problems, it may be necessary to explore problem-specific indicators for resampling within an indicator-based sampling scheme.

In contrast, the uncertainty-based active learning approach is a more versatile method that relies only on evaluating uncertainty levels in prediction results. It does not require prior knowledge of the specific datasets and can be applied across various scenarios.

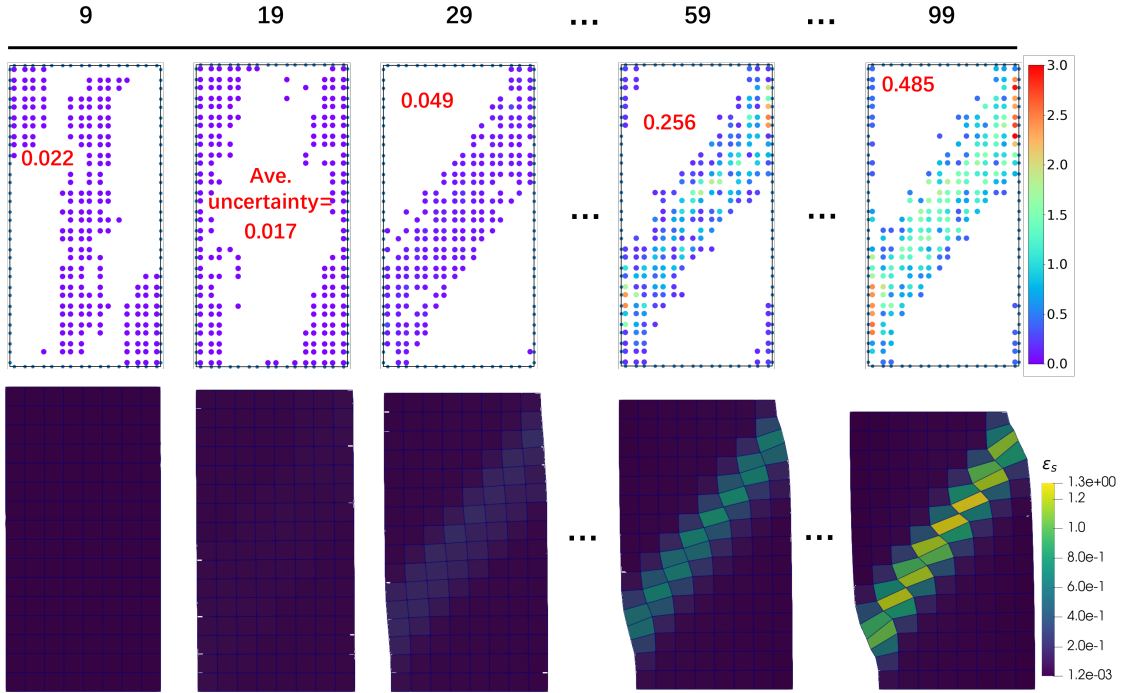
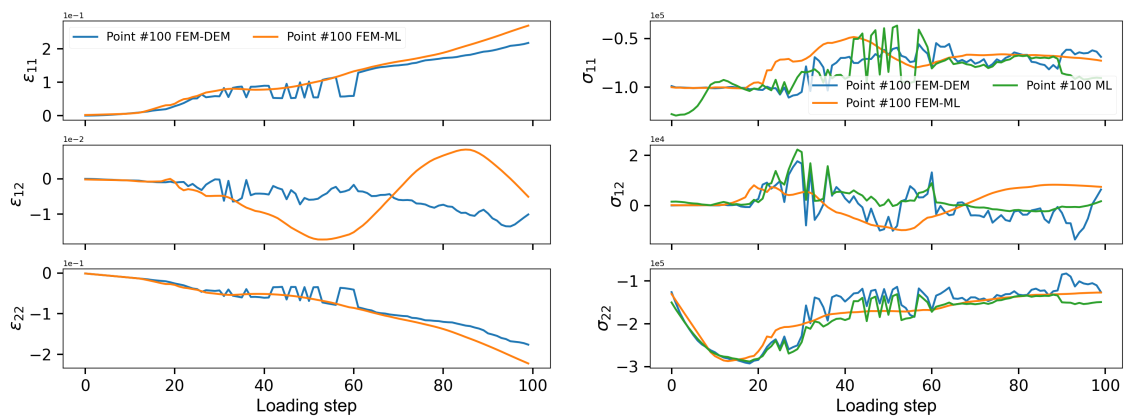


Figure 4.15: The shear strain-based active learning resampling at seven load steps: Top row – 30% of the data points with highest uncertainty levels (the number in red is the average uncertainty); Bottom row – equivalent shear strain distribution.

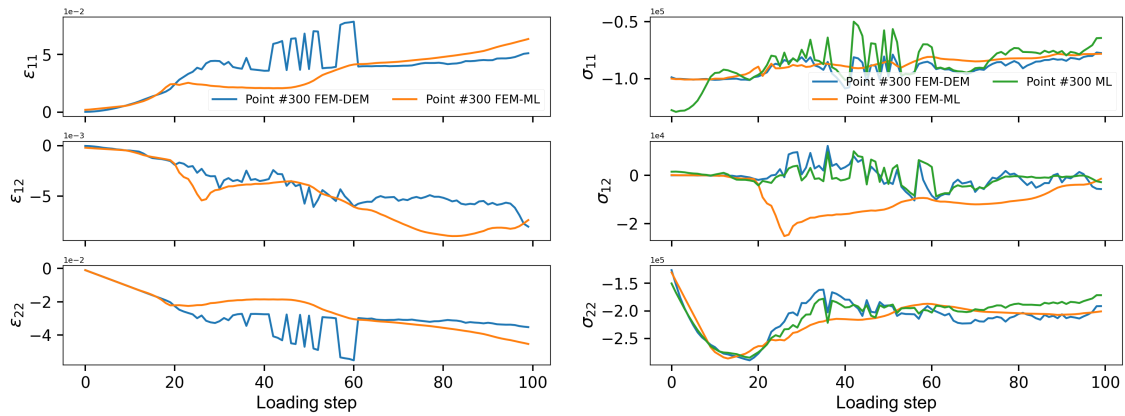
To further evaluate the effectiveness of the FEM-NN method, we examine the strain-stress responses obtained by various models at the integration point level. The strain and stress curves of two specific integration points labelled 100 and 300 (marked in Fig. 4.10), are shown in Fig. 4.16. The blue line represents the stress from the FEM-DEM simulation. The green line corresponds to the predicted stress after feeding the strain from FEM-DEM simulations directly to the trained network. The orange line represents the stress values obtained from the FEM-ML simulation.

The FEM-ML framework effectively predicts the overall trends of the macroscopic stress response in granular materials. However, it is important to note that the trained network model inherently produces a smoother output compared to the more oscillating output observed in the FEM-DEM results, which can be attributed to the transient nature of DEM simulations.

It is worth highlighting that the fluctuations and noises poses a significant challenge in network training. Such patterns can make it more difficult to distinguish valuable information from the noise. Additionally, large and sharp fluctuations in the predicted curve can render it non-differentiable, making it challenging to use automatic differentiation methods for obtaining tangent operators in granular material simulation.



(a) Integration point 100



(b) Integration point 300

Figure 4.16: Comparison of the local strain and stress responses from different solution schemes

Evaluating the efficiency of the FEM-NN framework is crucial. The performance of both



FEM-DEM and FEM-NN simulations for Case D was compared on a laptop computer with an i5-8500 6 Cores@3.00GHz processor. In the FEM-DEM simulation, all six cores were utilized, while only a single core was used in the FEM-NN simulation.

Tab. 4.4 demonstrates that the FEM-NN framework achieves an efficiency improvement of nearly 82 times per iteration compared to FEM-DEM. This significant enhancement in efficiency is accompanied by a substantial reduction in computer memory requirements. It greatly alleviates the memory demands of FEM-DEM multiscale computations.

The number of iterations required at each load step is also recorded in Fig. 4.17 for further comparison. It should be noted that the medium mesh with 4x8 elements, as depicted in Fig. 4.10, was used for simulation. Both the FEM-DEM and FEM-NN methods exhibit convergence at every loading step, particularly during the elastic stage. Subsequently. The number of iterations per loading step increases significantly until reaching a peak at step 20. Afterwards, the iteration number decreases, settling around 20.

In comparison to the FEM-DEM simulation, the FEM-NN simulation with all three meshes requires a slightly larger number of iterations. This results from the error between the network prediction and the lower-scale DEM simulation. The FEM-NN framework generally does not converge as rapidly as FEM-DEM but can still reach the final equilibrium state through iterative processes. This further supports that networks can serve as surrogate models for lower-scale DEM simulations.

Table 4.4: Efficiency improvement

	FEM-DEM	FEM-ML	Efficiency improvements
Time (h)	8.02	0.11	72.9
Total number of iteration	2510	2820	0.89
Single iteration (s)	11.50	0.14	82.1

### 4.3.2 Retaining wall

To evaluate the generality of the proposed neural network model, the well-trained neural network model in the previous biaxial compression case is employed in a retaining wall

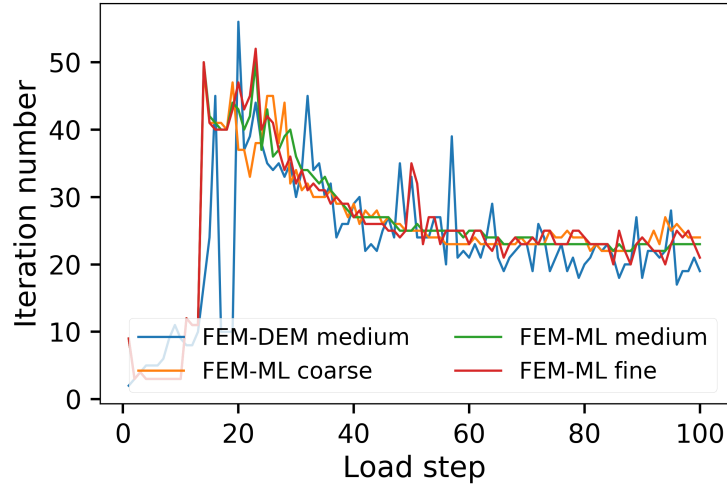


Figure 4.17: The number of iterations required for the FEM-DEM and FEM-NN simulations with the three levels of mesh

problem. The details of the problem are shown in Fig. 4.18, where the normal constraint is applied to the left boundary; the bottom is constrained in the  $x$  and  $y$  directions; and a prescribed displacement is applied to the right boundary, acting as the retaining wall, to compress the soil in the normal direction.

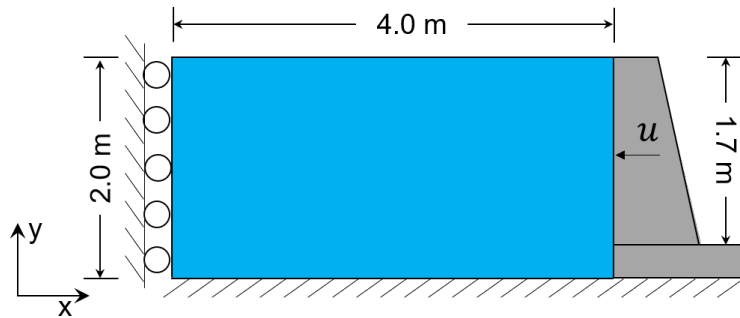


Figure 4.18: Schematic of the retaining wall simulation

To assess the generalisability, the well-trained network from the previous biaxial compression case is utilised in a retaining wall problem. Details of the simulation are illustrated in Fig. 4.18, where a normal constraint is applied to the left boundary, the bottom is constrained in the  $x$  and  $y$  directions, and a prescribed displacement is imposed on the right boundary to compress the soil, serving as the retaining wall.

Fig. 4.19a displays a prominent cambered shearing band observed in the FEM-DEM simulation. The relationship between the total force applied by the retaining wall and the

transverse strain is depicted in Fig. 4.20.

The FEM-NN with two different networks has been investigated separately. The first one, labelled **FEM-ML 1**, is trained solely on datasets collected from the biaxial simulations. The solution process encounters difficulties around the 80th load step. This issue arises due to accumulated errors in strain and internal variables when their values extend beyond the range covered by the network training data. The shearing band observed in the **FEM-ML 1** simulation approximately follows a straight line, primarily influenced by the training dataset obtained from the multiscale biaxial simulations, where the shearing band is also a straight line. The network model trained solely on the data from the biaxial compression test is unable to reproduce constitutive responses in the retaining wall simulation due to the insufficient training range.

The **FEM-ML 2** employs an enhanced network, wherein the network used in **FEM-ML 1** is retrained after incorporating datasets from the FEM-DEM retaining wall simulations. Fig. 4.19c and Fig. 4.20 demonstrate that the enhanced network exhibits significantly improved performance in both displacement and force calculations. This highlights the adaptability of the proposed methodology to upgrade the network model when new datasets are available.

## 4.4 Discussion

### 4.4.1 Limitations of Active Learning

Active learning resampling is quite sensitive to noise in the data. In the literature [146], the effect of noise in the data on active learning resampling is explained in detail. The expectation of error on prediction samples:

$$\mathbb{E}_e = \int_{x \in \mathcal{X}} \mathbb{E}((\hat{y} - y)^2 | x, \mathcal{D}) f(x) dx \quad (4.23)$$

where  $\hat{y}$  is the prediction,  $f(x)$  is the possibility distribution function.

## 4.5 Concluding remarks

This chapter focuses on developing a FEM-NN framework and training a network-based constitutive model to replace the lower-scale DEM simulations, thereby accelerating classical multiscale FEM-DEM simulations. A multi-layer fully connected neural network, cou-

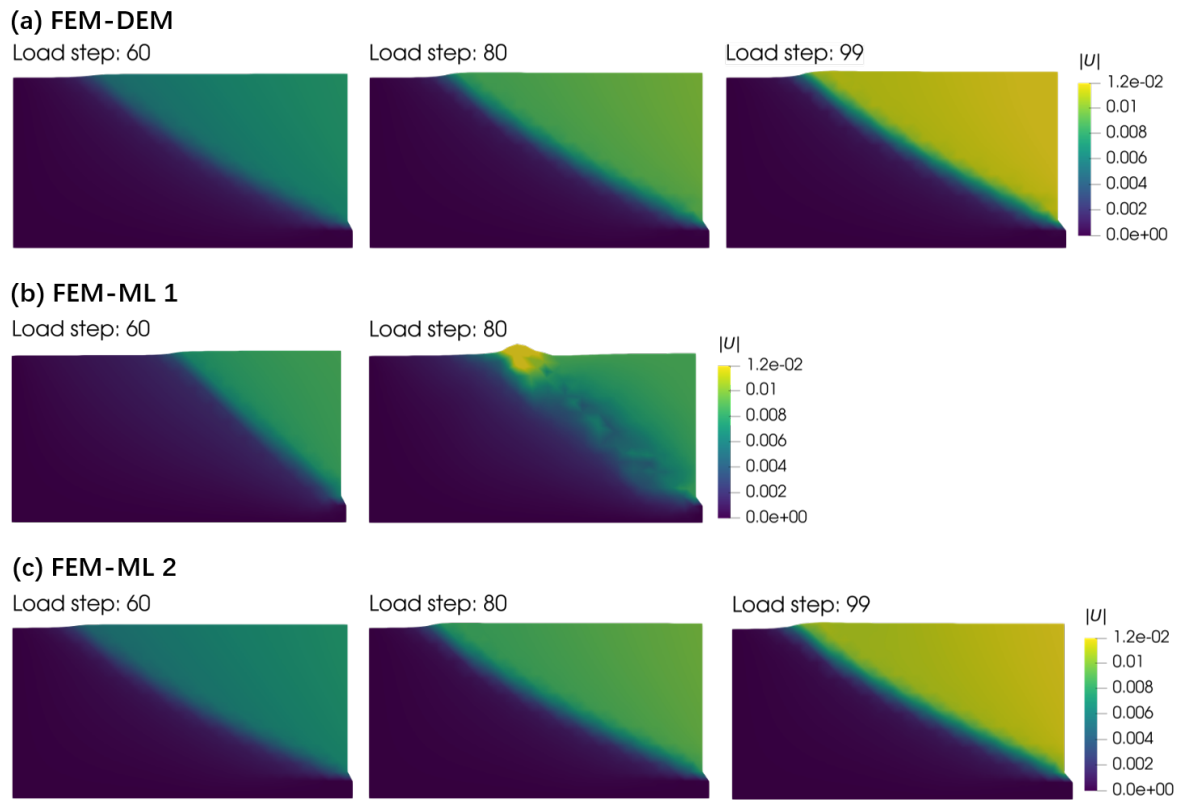


Figure 4.19: Displacement distributions of the soil at some load steps when compressed by the retaining wall and obtained by (a) FEM-DEM, (b) FEM-NN 1, and (c) FEM-NN 2

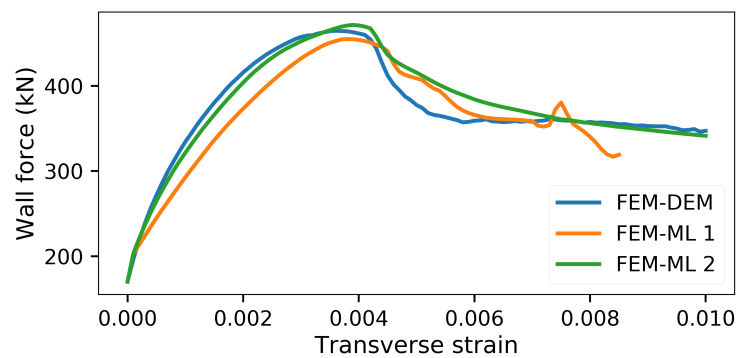


Figure 4.20: Comparisons of integrated wall forces

pled with the use of accumulated absolute strain increments as an explicit parametrization of loading histories, is chosen as the surrogate model. This simple network demonstrates a reasonable ability to reproduce the history-dependent constitutive responses of granular materials. Training samples are generated by applying random loading paths to biaxial compression simulations. An uncertainty-based active learning scheme is employed to select points with high uncertainty levels to enrich the training dataset. This resampling strategy is generic and proves to be highly effective.

The drained biaxial compression tests conducted demonstrate that the FEM-NN framework can accurately reproduce the micro-scale response of granular materials at significantly reduced CPU costs compared to the FEM-DEM approach. The generalizability is also investigated in the retaining wall example, where it exhibits flexibility to improve performance as long as the training datasets are enriched.

The numerical examples presented highlight the substantial improvement in computational efficiency achieved through trained surrogate networks. This improvement has the potential to extend multiscale computations to practical engineering problems.

# Chapter 5

## An explicit FEM-NN framework and analysis of error caused by NN-predicted stress

### 5.1 Methodology of the explicit FEM

#### 5.1.1 Governing equation and solving process

In implicit solver, we need to transform the governing equation  $\sigma_{ij}, i = 0$  in domain  $\Omega$ , to a weak form and discretise the domain to achieve  $K_{mjnl}^e = \int_{\Omega^e} N_{m,i} D_{ijkl} N_{n,k} d\Omega$ , where the global stiffness  $K_{mjnl}^g$  is given by the assembly of the element stiffness, where the tangential matrix  $D_{ijkl}$  is required. It is mentioned in the Introduction that avoiding predicting  $D_{ijkl}$  via neural networks can simplify the network architecture and avoid incompatibility.

FEM solver offers an alternative to bypass the use of the material matrix  $D_{ijkl}$ . In explicit FEM, the node's motion at time  $t$  is governed by Newton's 2nd law:

$$\begin{cases} \rho \ddot{u}_j = \sigma_{ij,i} & \text{in } \Omega \\ n_i \sigma_{ij} = t_j & \text{on } \partial\Omega \end{cases} \quad (5.1)$$

where  $\rho$  is the material density and  $t_j$  is the traction on the boundary  $\partial\Omega$ . In an element, the weak form of the equation can be represented as  $\int_{\Omega^e} \omega \rho \ddot{u}_j d\Omega = \int_{\Omega^e} \omega \sigma_{ij,i} d\Omega$ , where  $\omega$  is the arbitrary test function. After plugging the shape function into the test function and

integrating by parts, the weak form can be transformed into:

$$\int_{\Omega^e} N_n \rho \ddot{u}_j d\Omega = P_{nj}^e - I_{nj}^e \quad (5.2)$$

where  $P_{nj}^e = \int_{\partial\Omega^e} N_n t_j d\Gamma$  and  $I_{nj}^e = \int_{\partial\Omega^e} N_{n,i} \sigma_{ij} d\Omega$  are the external and inner node forces in the  $j$ th direction of the  $n$ th node of a single element, respectively. After substituting  $\ddot{u}_j = \sum_m \ddot{u}_{mj}$  into the RHS of the above equation, we can get  $M_{mn} = \int_{\Omega^e} N_m N_n \rho d\Omega$  is the elemental mass of the  $n$ th node of the element. With assembling, the global function can be written as:

$$\mathbf{M}\ddot{\mathbf{u}}|_t = (\mathbf{P} - \mathbf{I})|_t \quad (5.3)$$

where  $\mathbf{M}$  is the global mass matrix,  $\mathbf{P}$  is the global external force, and  $\mathbf{I}$  is the global inner node force.

Upon giving the nodal acceleration at time  $t$ , the central difference method is utilised to update the nodal displacements:

$$\begin{cases} \dot{\mathbf{u}}|_{t+0.5\Delta t} = \dot{\mathbf{u}}|_{t-0.5\Delta t} + \Delta t \ddot{\mathbf{u}}|_t \\ \mathbf{u}|_{t+\Delta t} = \mathbf{u}|_t + \Delta t \dot{\mathbf{u}}|_{t+\Delta t} \end{cases} \quad (5.4)$$

where the constant time step  $\Delta t$  is engaged. Then  $\Delta \mathbf{u}|_{t \sim t+\Delta t} = \Delta t \dot{\mathbf{u}}|_{t+0.5\Delta t}$  is calculated as the displacement increment from time  $t$  to time  $t + \Delta t$ . With the finite strain assumption, the strain increment is calculated as:

$$\Delta \epsilon_{ij}|_{t \sim t+\Delta t} = 0.5 (\Delta u_{i,j} + \Delta u_{j,i})|_{t \sim t+\Delta t} \quad (5.5)$$

Cast the strain increment from time  $t$  to time  $t + \Delta t$  into the constitutive model  $\mathcal{M}$  to renew the stress tensor:

$$\sigma_{ij}|_{t+\Delta t}, \mathcal{I}_m|_{t+\Delta t} = \mathcal{M}(\sigma_{ij}|_t, \mathcal{I}_m|_t, \Delta \epsilon_{ij}|_{t \sim t+\Delta t}) \quad (5.6)$$

where  $\mathcal{I}_m|_t$  is the vector of internal variables at time  $t$ . Thus, the explicit FEM solver, without the tangential material matrix, can move forward as shown in Algorithm 3.

### 5.1.2 Stable time increment

To ensure the accuracy of the explicit calculation, the time increment is carefully calculated according to the wave propagation velocity as the high-field solution can be obtained only when the time increment  $\Delta t$  is less than the stable time increment  $\Delta t_{\min}$ . If the time step  $\Delta t > \Delta t_{\min}$  or close to  $\Delta t_{\min}$ , the solution will diverge and results can be distorted.

---

**Algorithm 3** The explicit FEM solver (assuming the constant time step  $\Delta t$ )

---

**Require:** Discretised FEM model, and the constitutive model  $\mathcal{M}$

```

 $\sigma_{ij}^{(0)}, \mathcal{I}_m^{(0)} = \mathcal{M}(\sigma_{ij}^{(0)}, \mathcal{I}_m^{(0)})$  ▷ Initialise the stress
for  $n = 1, 2, \dots, N$  do ▷ Loading step
    Apply the loading condition  $\Delta u_j$  at the beginning of step  $n$  to the model
    Calculated the strain increment  $\Delta \epsilon_{ij} = \frac{1}{2} (\Delta u_{i,j}^{load} + \Delta u_{j,i}^{load} + \Delta u_{i,j} + \Delta u_{j,i})$ , due to
    the displacement load  $\Delta u_j^{load}$  and the displacement  $\Delta u_j$ 
    Renew the stress and internal variables  $\sigma_{ij}^{(n)}, \mathcal{I}_m^{(n)} = \mathcal{M}(\sigma_{ij}^{(n-1)}, \mathcal{I}_m^{(n-1)}, \Delta \epsilon_{ij})$  at step  $n$ 
    Solve Eq. 5.4 for the acceleration  $\ddot{u}_j^{(n)}$  at step  $n$ 
    Update the displacement to  $u_j^{(n)} = u_j^{(n-1)} + \Delta u_j|_{n-1 \sim n}$ 
end for

```

---

The dilatational wave speed  $c_d$  can be expressed for a linear elastic material as:

$$c_d = \sqrt{\frac{E}{\rho}} \quad (5.7)$$

where  $E$  is the Young's modulus and  $\rho$  is the material density.

Based on the current geometry, each element has a characteristic length  $L^e$ . Thus, the stable time increment can be expressed as:

$$\Delta t = \frac{L^e}{c_d} \quad (5.8)$$

In our simulation, the stable time increment is carefully evaluated before every time increment and a safety coefficient of 0.2 is utilised to ensure the calculation stability, i.e.  $\Delta t = 0.2 \Delta t_{\min}$ .

### 5.1.3 Damping

Damping is also an important topic in time-integral-based simulations. We added damping to the simulations to ensure that the calculations do not fluctuate and that the simulations are quasi-static. The selection of a suitable damping factor requires a combination of density and material stiffness. This is not an easy task for a multi-degree-of-freedom system. In the simulation, we used a damping ratio-like variable  $\gamma$  to relate the damping coefficient  $c$  to the density  $\rho$  and Young's modulus  $E$ , via:

$$c = 2\gamma\sqrt{\rho E} \quad (5.9)$$



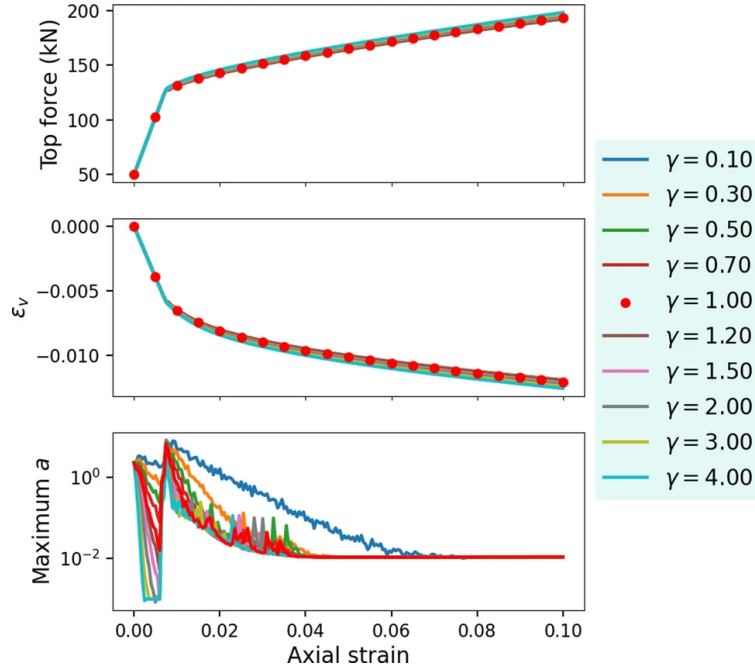


Figure 5.1: Contrasting various  $\gamma$  values to optimise damping coefficient selection

A sensitivity analysis was done for different  $\gamma$  as shown in Fig. 5.1. In this work, we use  $\gamma = 1.0$ .

## 5.2 Constitutive model and the network-constitutive model

Granular materials are the object of our study in this paper. The FEM-DEM multi-scale framework can accurately reproduce their constitutive responses. While, as the lower-scale DEM needs to be invoked thousands of times in the explicit FEM solver, using a lower-scale DEM for material stress calculations will result in a huge computational effort. Alternatively, stress integration along a prescribed strain path can be completed in a fraction of a second with a classical constitutive model. We, therefore, prefer to use mathematical formula-based classical models as the baseline for training dataset generation.

In this work, two classical constitutive models are selected as baseline models for training data generation. One of them is simple to formulate and the other is more complex. Neural networks are separately trained on the dataset generated by each of them and then embedded in the FEM solver for computation. It is found that the capability of the neural network varies in reproducing the constitutive relationships with different complexity.

### 5.2.1 Classical constitutive model

IME model (isotropic elasticity and von Mises plasticity with exponential hardening function): A simple elastoplastic model, comprised of isotropic elasticity with Young’s modulus  $E$  of 20 MPa and Poisson’s ratio  $\nu$  of 0.2, and the von Mises yielding with exponential hardening law, is introduced to show the feasibility of the ex-FEM-NN framework. The model is denoted as IME for convenience in subsequent reference. The IME model is used to initially verify that the neural network model can effectively reproduce the constitutive responses. The yield function of the model is shown as:

$$f = \sigma_v - H - \sigma_0 \quad (5.10)$$

where  $\sigma_v$  is the von-Mises stress,  $\sigma_0 = 1\text{e}3 \text{ Pa}$ ,  $H = A(\epsilon_0 + \bar{\epsilon}^p)^n$  represents the isotropic exponential hardening function using  $A = 0.3 \text{ MPa}$ ,  $n = 0.2$  and  $\epsilon_0 = 0.02$ . The associated flow rule is utilised in the plastic return mapping process.

CSUH model (Critical State Unified Hardening): The CSUH model is able to reproduce critical state theory and has a unified hardening rule for clays and sands based on relative void ratio variables. The material parameters of the CSUH model used in this work are listed in Table 5.1, detailed explanation of this model and the related parameters can be found in this paper [6].

Table 5.1: Parameters of the CSUH model

Material constants	Value
$\phi^\circ$	8.0
$\lambda$	0.214
$\kappa$	0.191
$\nu$	0.2
$N$	1.931
$Z$	0.2743
$\chi$	0
$m$	1.8
OCR	377

## 5.2.2 Neural network-based constitutive model

As mentioned at the beginning of the methodology, for implicit FEM-NN computation, the material matrix is necessary for the local elemental stiffness integration via:

$$K_{mjnl}^e = \int_{\Omega^e} N_{m,i} D_{ijkl} N_{n,k} d\Omega \quad (5.11)$$

where  $N_{m,i}$  is the shape function, with subscripts  $m$  and  $i$  representing the number of nodes in an element and direction of the solution space,  $D_{ijkl}$  is the material matrix or called the tangent matrix (ideally with  $D_{ijkl} = \partial\sigma_{ij}/\partial\epsilon_{kl}$ ) in the implicit solver.

In the exFEM-NN framework, the tangential matrix is no longer needed. In the NN part, with a tiny modification. Thus, for the NN part, the prediction of tangent matrix is no longer needed. The constitutive part of Eq. 5.6 can be changed to:

$$\sigma_{ij}|_{t+\Delta t} = \text{NN}(\epsilon_{ij}|_t + \Delta\epsilon_{ij}|_{t\sim t+\Delta t}, \varphi_{ij}|_t) \quad (5.12)$$

where the vector of internal variable  $\mathcal{I}_m$  is represented by the accumulation of the absolute strain value, with  $\varphi_{ij} = \sum |\Delta\epsilon_{ij}|$ . A similar internal variable for coding the loading history can be found in [147].

## 5.2.3 A novel MLP network enhanced by Fourier feature mask and the multiplied residual block

### Basic network architectures

The multi-layer neural network is used to predict stress via the strain and the state-related internal variable. During the computation, the strain calculated based on the previously predicted stress is fed to the network again to infer the stress at the next step as shown in Fig. 5.6, resembling the recurrent networks. The recurrent structure enables them to perform well in history-dependent issues, such as material constitutive response prediction [98, 105].

The multi-layer forward neural network is employed to represent the constitutive part. The input features are the strain tensor and the historical variables, and the outputs are stress tensors. The network's input and output are highlighted in Fig. 5.2.

### Fourier feature mask and multiplied residual blocks

The explicit FEM solver calls the constitutive model thousands of times throughout the process, unlike the implicit calculation, which only requires a few dozen load steps to finish

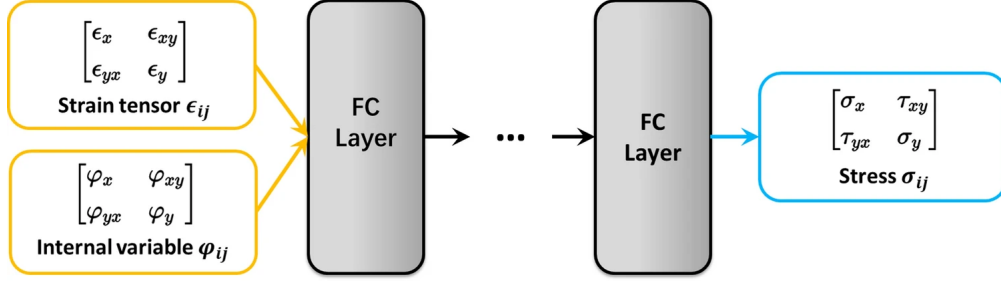


Figure 5.2: Architecture of the neural network used to reproduce the stress tensor

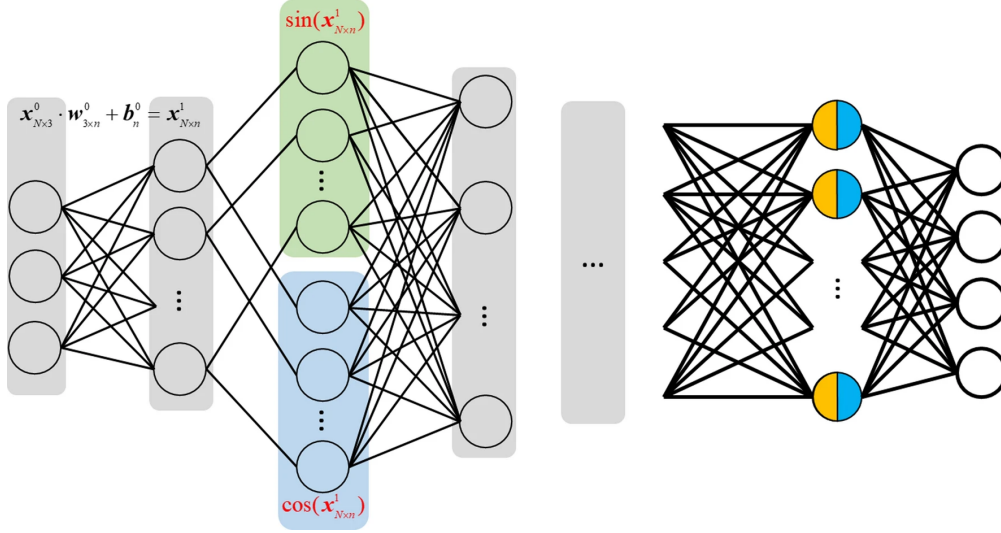


Figure 5.3: Fourier feature mask

the entire computation. Similar to the recurrent neural network results, the values predicted in the previous load step are used to predict the values in the next load step. Therefore, reducing the prediction error and slowing down error accumulation is a big challenge in exFEM-NN computation. Fourier features and the multiplied residual blocks are introduced for higher accuracy.

In terms of the Fourier features, before casting the inputs into the neural network, they are first processed by a Fourier feature mask, as shown in Fig. 5.3:

Tensors processed by the Fourier feature mask then flow into the multi-layer network. In addition to the general calculation related to weights and biases, the residual block is introduced to improve the network's performance.

Although the network is a black box, increasing the diversity of basis functions, for example by extending the basis functions from  $\phi_1(x) = \{1, x\}$  to  $\phi_3(x) = \{1, x, x^2, x^3\}$  would

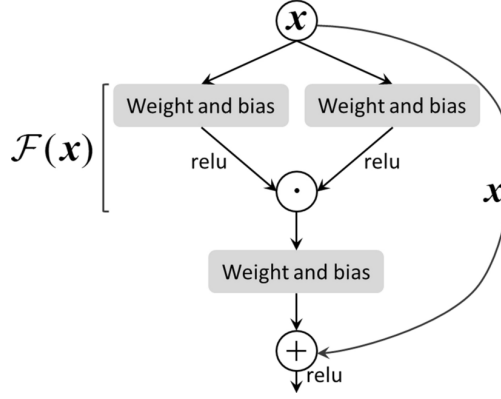


Figure 5.4: Multiplied residual block in the network architecture

enhance the regression model’s ability to perform non-linear regression. The multiplied operation  $F(\mathbf{x}) = \text{relu}(\mathbf{w}_1 \cdot \mathbf{x} + \mathbf{b}_1) \odot \text{relu}(\mathbf{w}_2 \cdot \mathbf{x} + \mathbf{b}_2)$ , which enables the second-order term to participate in the network tensor calculation, is involved in the residual block (Fig. 5.4).

By adding Fourier features and employing a multiplied residual block, we add sinusoidal functions and high-ordered basis functions to the network input.

Upon proposed methods can improve the network’s performance, as shown in Fig. 5.5a and b. In legends of the curves, ‘d’ means a dense layer in the network, ‘m’ means a multiplied residual layer and the number represents the number of neurons per layer. After the introduction of the multiplied residual block, the validation losses are remarkably decreased. In Fig. 5.5b, the Fourier features can also improve the network accuracy. However, the growth in the number of neurons and network layers increases the trainable parameters, and these can affect the computational efficiency of the network. Therefore, to strike a balance between efficiency and accuracy, we chose the network structure as **dmmd20**, which includes two fully connected layers and two multiplied residual blocks. Figure 5.5c shows satisfactory prediction results with the selected network architecture.

## 5.2.4 Check-and-revision method

Theoretically, the network model can behave as stable and universal as the classical constitutive model supposed the data is abundant and the training range is infinite. However, in most high-dimensional cases, the sampling range is far from covering all of the possible spaces. The network model performs poorer as the input features approaching the edge of the training range.

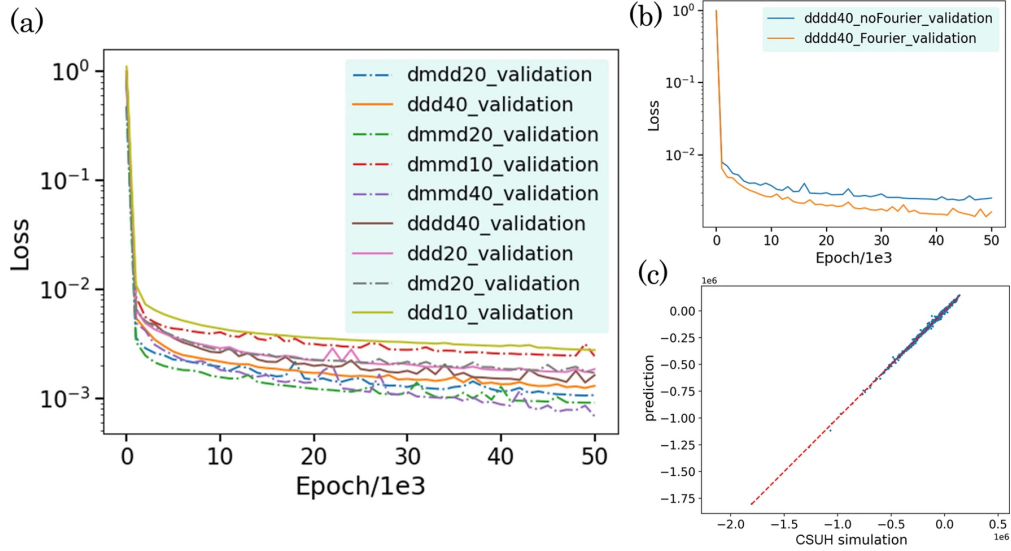


Figure 5.5: (a) Hyperparameters (layer type, layer number and the node number) sensitivity analysis. The dot-dashed curves represent the results with multiplied residual blocks; (b) Comparison of results with and without the Fourier mask (c) The normalised distribution of the network prediction on the 1:1 line

Due to the inevitable prediction errors, the error accumulation caused by the recurrent structure will shift the network from interpolation to extrapolation. This can lead to a rapid increase in prediction error and ultimately to computational distortion.

To address this problem, we propose a check-and-revision method, as shown in Fig. 5.6. The network prediction is compared with the classic constitutive model calculated result. If the discrepancy between them is unreasonably large, the latter will be employed and saved to enrich the training sets. In this way, we effectively expanded the training range. After iteration, points with remarkable errors are expected to be gradually reduced.

In the explicit FEM solver, as the equilibrium is governed by Newton's second law, the acceleration caused by the unbalanced force can be an indicator to demonstrate whether the NN-based constitutive model provides reliable stress predictions and the computation converges satisfactorily. After check-and-revision iteration, the maximum acceleration is also expected to gradually shrink. A decrease in the maximum acceleration indicates the can expand training range with informative training sets.

It is worth noting that, in Fig. 5.6, the processes in blue boxes will be performed at every load step. The operations in green boxes will only be done after the whole loading is finished. Points saved from the  $n$ th calculation are added to the  $(n+1)$ th dataset to retrain

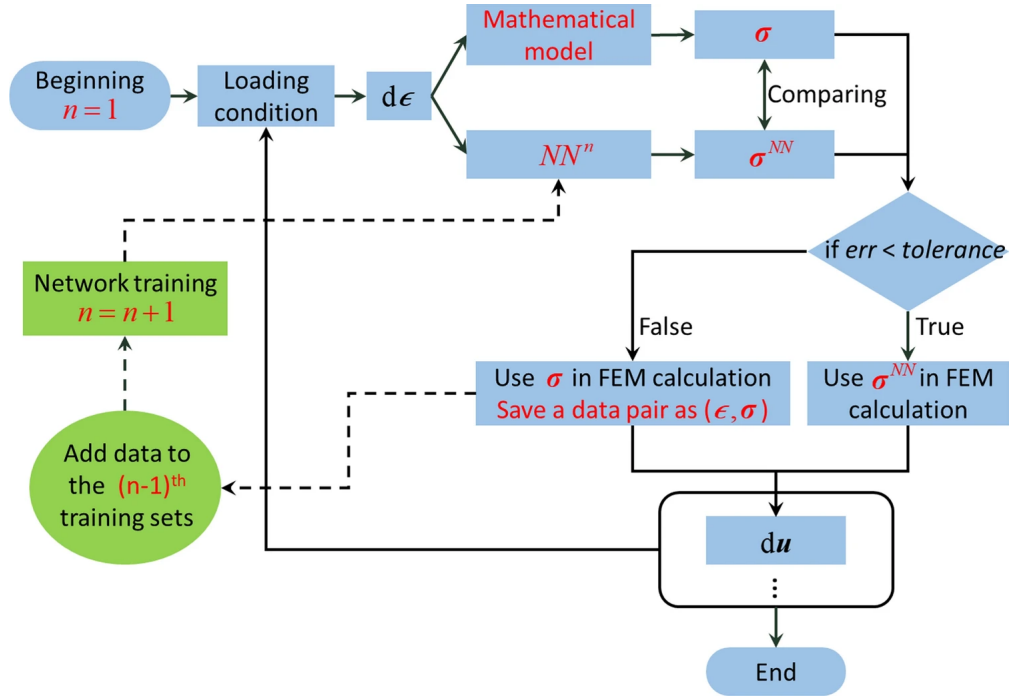


Figure 5.6: Schematic of the on-the-fly check-and-revision method

the network. After training, the updated network is used in the next check-and-revision iteration.

## 5.3 Numerical simulations

### 5.3.1 Biaxial compression

Biaxial compression tests (as shown in Fig. 5.7) with rough top and bottom constraints are first carried out to show the performance of the ex-FEM with an NN-based constitutive model. Two classical constitutive models with different complexity are employed to generate training datasets. As introduced in Sec. 5.2.1, one is the IME model, and the other is the CSUH model.

#### IME model datasets and the exFEM-NN simulations

The validating numerical test begins with the IME model. After FEM calculation with the IME as a constitutive model, strain–stress pairs on Gauss points are cast into the network training. Then, the trained model provides the constitutive responses for the exFEM-NN coupling simulations.

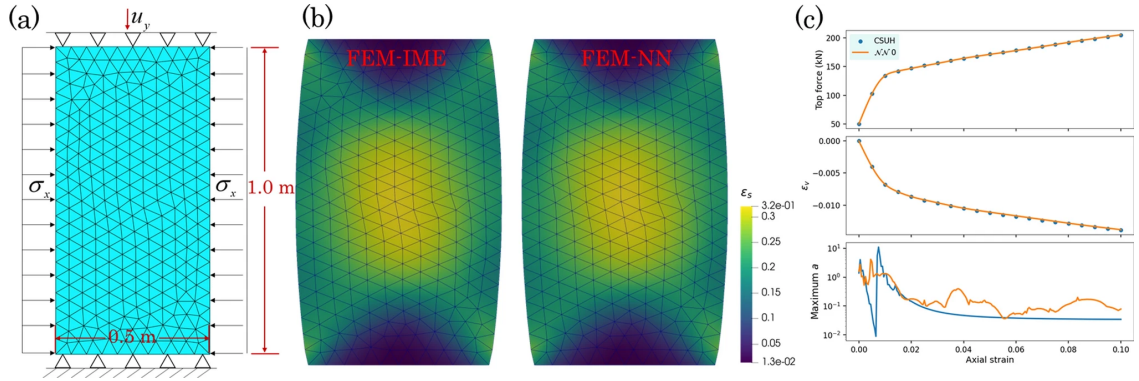


Figure 5.7: Biaxial compression simulation: (a) The model discretisation and boundary conditions; (b) Comparison of the shear-strain field; (c) Comparison of the top force, global volume strain and maximum nodal acceleration

Curves of the macro results, such as the top force, global volume strain, and maximum acceleration, are shown in Fig. 5.7c. The curves all agree very well except for curves of the maximum accelerations.

At the transformation from elasticity to the plastic stage, the accelerations increase because of the sudden change of stress rate. The maximum acceleration with the IME model is relatively low, especially when it comes to the end of the loading because the stiffness decreases. In contrast, in the NN-involved simulation, the maximum acceleration is slightly higher and keeps fluctuating. Different from the other macro results, the acceleration of each node can abruptly increase because of the sharp change in stress rate or error of the stress predictions. The difference in curves of the maximum acceleration in some extent represents the worst stress predictions.

To further check the computational performance of the exFEM-NN method, stress-strain predictions at four representative Gauss points are illustrated in Fig. 5.8. The strain sequences from the IME simulations are directly fed into the network to check the network's prediction accuracy without considering the recurrent structure. The stress predictions (inverted triangular point) are in perfect agreement with the simulated stresses, indicating the network is effectively trained and able to reproduce the constitutive relationship. There is some deviation in the stress extracted from the exFEM-NN simulations on Gauss point #0. The error is relatively small and can be accepted.



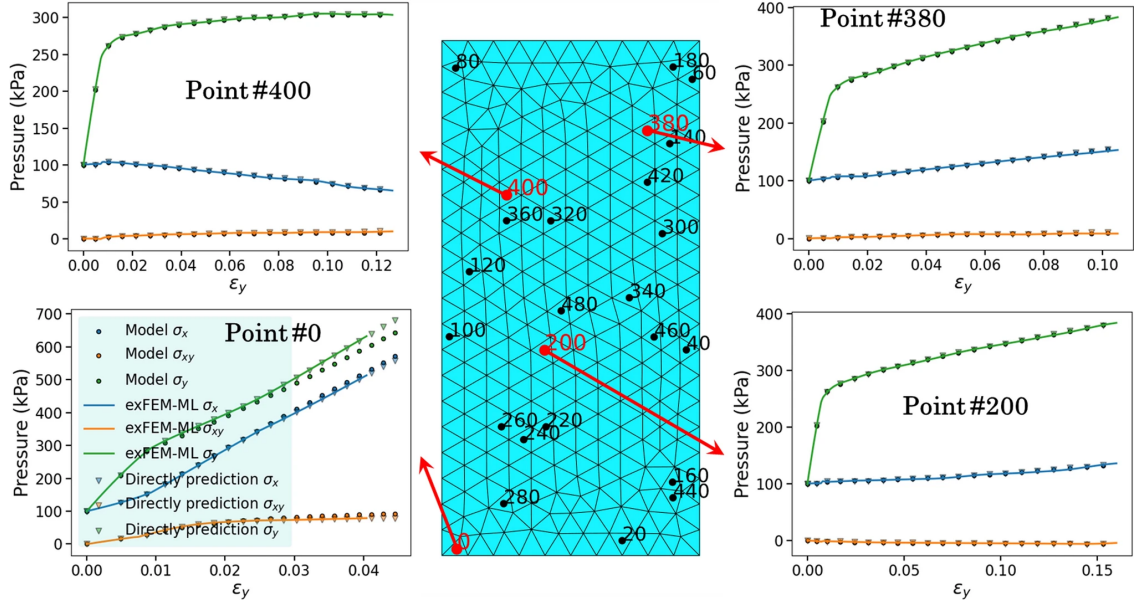


Figure 5.8: Curves of stresses belonging to Gauss points with datasets collected from the IME model simulations

### CSUH model-based datasets

After the validation based on the IME model, the biaxial compression is subsequently performed based on the CSUH model. Data pairs of the Gauss points in the exFEM-CSUH simulation are saved and subsequently fed to train the network. Then the network is employed to substitute the CSUH model in the exFEM solver.

Fig. 5.9 shows a comparison of the top force, global volume strain and maximum acceleration simulations based on the CSUH model and the NN constitutive model. The curves of the top force and the volume strain are smooth and stable. There are quite sharp rises and drops in the maximum acceleration, which is caused by the sudden changes in stress rate due to tensile or shear damage.

In the beginning, all the macro results agree well, however, some deviations occur close to the end of the loading. In the early stage, the NN-predicted stress is acceptably different from the CSUH model. Then the difference/error is passed to the acceleration, as shown in Fig. 5.6, and consequently, influences the displacements and strain field, followed by the increasing errors in the subsequent stress predictions. With the growth of error in the network predictions, the maximum node acceleration increases constantly.

Some stress–strain responses extracted from four Gauss points are displayed in Fig. 5.10.

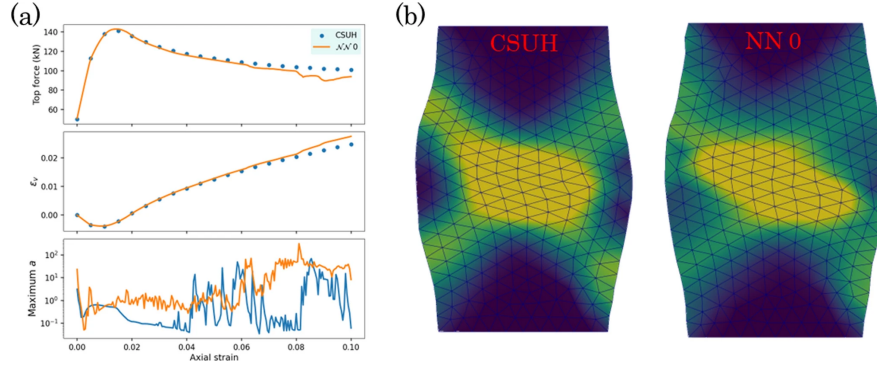


Figure 5.9: Comparison between the CSUH-based and NN-based exFEM simulations: (a) The top force, global volume strain and the maximum acceleration; (b) Shear strain field

Different from the IME model where the stress–strain curves consist roughly of two straight lines, representing the elastic phase and the elastoplastic phase respectively, the CSUH-based simulation yields far more complex stress–strain behaviour. In the elastic phase, non-linear elasticity is controlled by the normal consolidation line of the CSUH model, while isotropic linear elasticity is used to simply depict the strain–stress relationship in the IME model. In the plastic phase, the void ratio-related plastic flow rule of the CSUH model is utilised to better represent the plastic response, instead of the exponential isotropic hardening equation and associated flow rule in the IME model.

The higher complexity of the CSUH model compared to the IME model results in a more complicated CSUH dataset. The directly predicted results by the NN agree well with the offline dataset which is indicated by the almost exact overlap of the triangular and circular points in Fig. 5.12. However, once cooperating with the exFEM solver, the NN-based and CSUH-based computations experience two quite deviated strain–stress paths. This largely results from the error accumulation as the simulation goes on. The results also explain why the amplitudes of the acceleration in Fig. 5.9 grow increasingly larger.

In Fig. 5.10, since the x-axis in the graph is the axial strain, the strain obtained from the exFEM-NN calculation may be either greater or less than that in CSUH-based computation. This interprets why the curves in Fig. 5.10 are shorter or longer than the range covered by the data points.

In the purely data-driven framework, the errors between the training datasets and predictions are impossible to be completely removed, instead relatively decreased via improving the training accuracy. Once the initial error exists, it will influence the following computa-

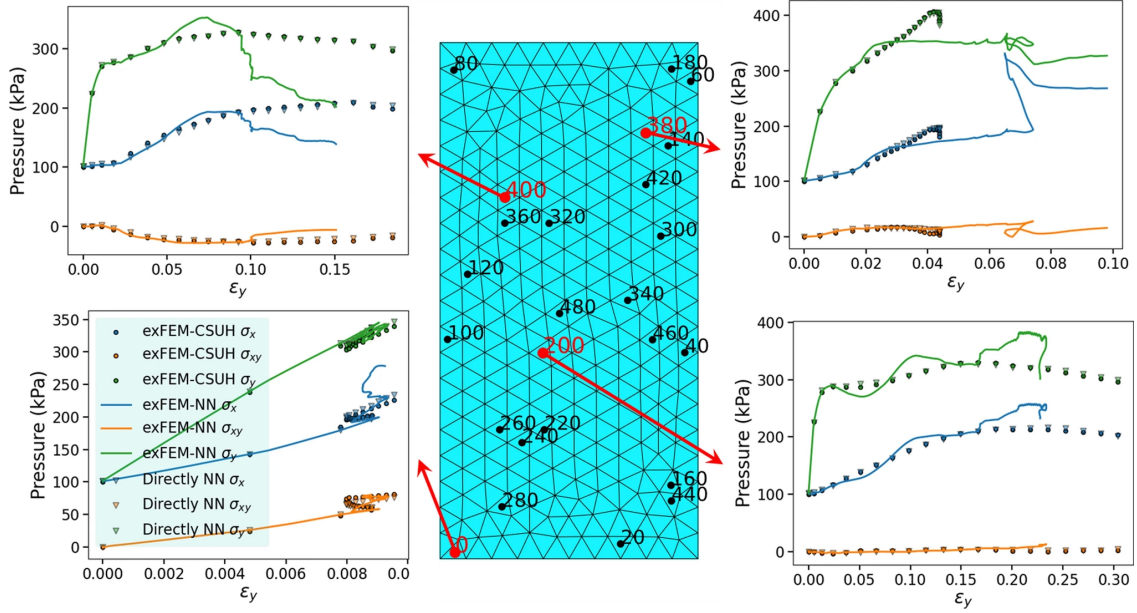


Figure 5.10: Stress–strain responses at representative Gauss points with datasets collected from the CSUH-based simulations

tions. Only if the predictions are sufficiently accurate or the NN-based model is capable of giving a sensible prediction when the loading path deviates from the original one, the error accumulation can be solved. Otherwise, the stress–strain path will gradually deviate from the ground truth.

In an explicit FEM solver, the constitutive part is invoked much more frequently than the implicit solver. So, it is more demanding to train a sufficiently accurate to relieve the error accumulation in thousands of steps. The Fourier features and the multiplied residual block is introduced to achieve higher accuracy. In terms of expanding the input range of the training samples, there is no proper method to evaluate how many samples are supposed to be generated and whether the training datasets are sufficient to cover all of the possible scenarios which will encounter in a boundary volume problem. A full sampling is quite challenging, even impossible.

### Model optimisation via the check-and-revision method

To get better results on the Gauss points, we use the on-the-fly check-and-revision method described in Sec. 5.2.3 to check, correct and save the data points. As shown in Fig. 5.6, after one check-and-revision calculation, the newly saved error data points are added to the original training data set and the network is retrained. The check-and-revision is invoked

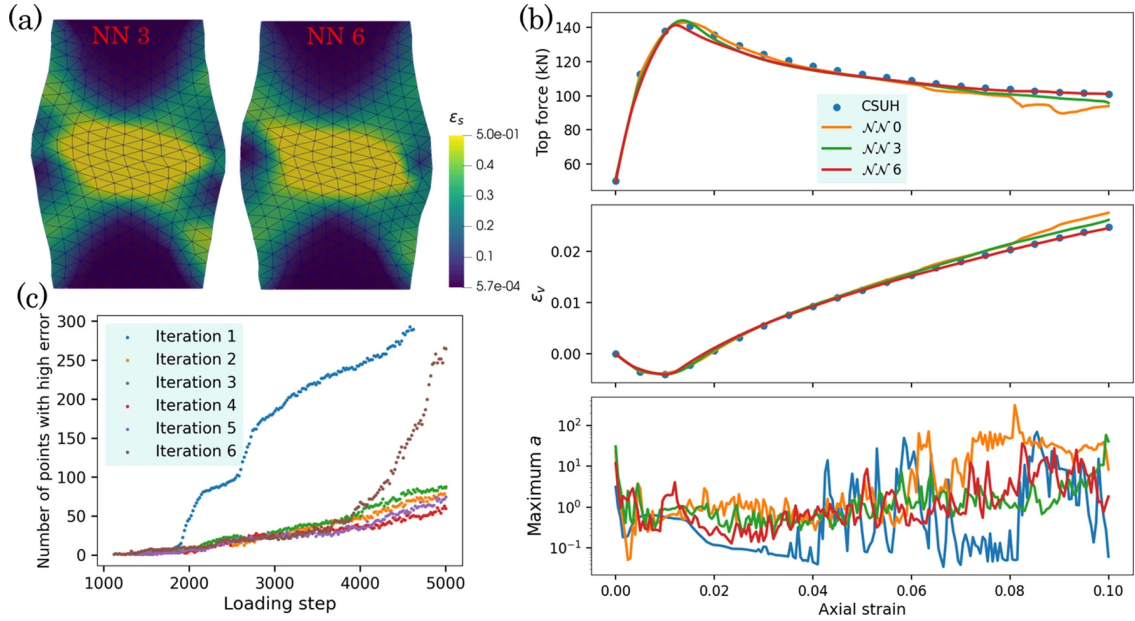


Figure 5.11: The exFEM-NN computation results with the check-and-revision iterations: (a) Strain field and (b) top force, total volume strain and maximum acceleration curves after three and six iterations of check-and-revision; (c) evolution of the number of Gauss points whose error of stress prediction greater than 50% with loading step

iteratively on the newly trained network to perform the calculation, and the number of error points evolves as shown in Fig. 5.11b. After the first expansion of the dataset via the check-and-revision iteration, the number of error points is significantly reduced. However, as we repeated this procedure, it turns out that the number of error points predicted by the network gradually stabilised at around 50.

This calculation suggests that the check-and-revision effectively enlarges the training range and enhances the network’s predictive capability on biased sample inputs, but this method does not completely eliminate the presence of error points. This may partly attribute to the nature of the data-driven model, as a surrogate simply approximates the ground truth and some loss of accuracy is unavoidable.

Fig. 5.11a and b show the results after three and six check-and-revision iterations, respectively. As the number of iterations increases, the top force and volumetric strain are closer to those from the CSUH-based simulation, and the acceleration has been reduced. This indicates that after iterations, the range of input strains to the network is broadened and the network performs better with unfamiliar input strains and therefore does not suffer from excessive acceleration and computational instability.

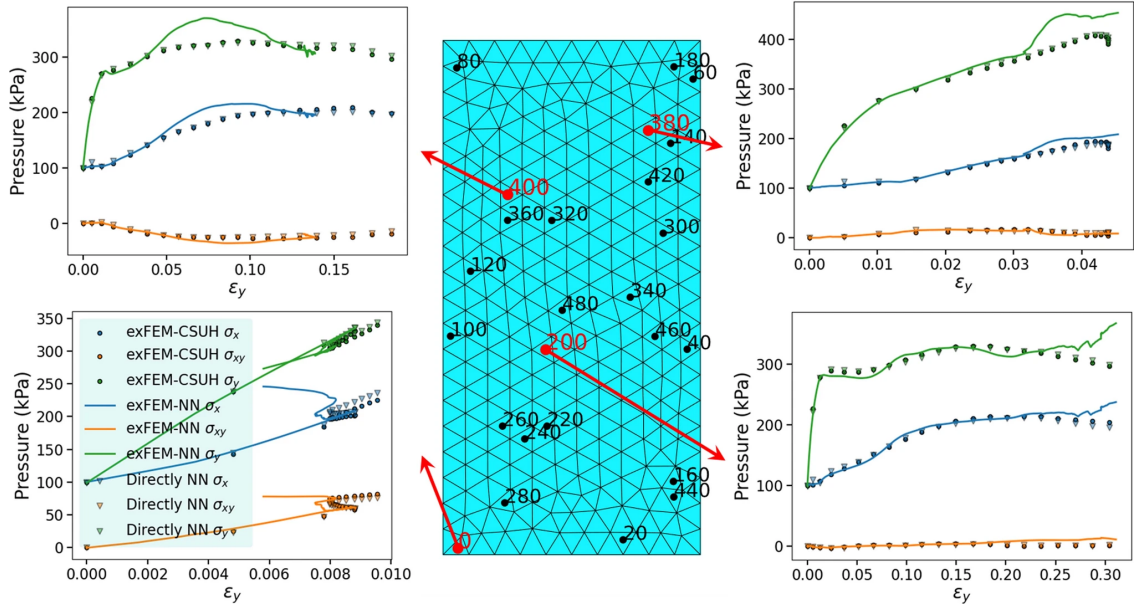


Figure 5.12: Stress curves on the Gauss points after six iterations of check-and-revision

Fig. 5.12 shows the stress–strain results at the gauss point in the exFEM-NN calculation after six check-and-revision iterations. Compared to the results in Fig. 5.10, the stresses at Gaussian points are closer to the CSUH-based dataset, but they do not match the dataset perfectly. Even though the macroscopic results of the exFEM-NN simulations in Fig. 5.11 agree well with the CSUH-based simulation, the results at the Gauss point still deviate considerably. This supports the view that errors and error accumulation processes cannot be completely removed.

### 5.3.2 Retaining wall

In parallel with the biaxial tests, we also carry out a retaining wall simulation, using the model schematic shown in Fig. 5.13a. The biaxial simulation and the retaining wall simulations are macroscopically loaded in the y-axis and x-axis directions, respectively. Therefore, the training data from the two are to some extent complementary and are proper to be merged together for network training.

As shown in Fig. 5.13b and c, the trained network used for the retaining wall simulation results in similar conclusions to the former biaxial case. As the number of check-and-revision iterative optimisation increases, the reaction force acting on the retaining wall and overall volumetric strain get closer to the ground truth, and the maximum acceleration in the second half of the loading process is gradually reduced. However, in the retaining wall simulations,



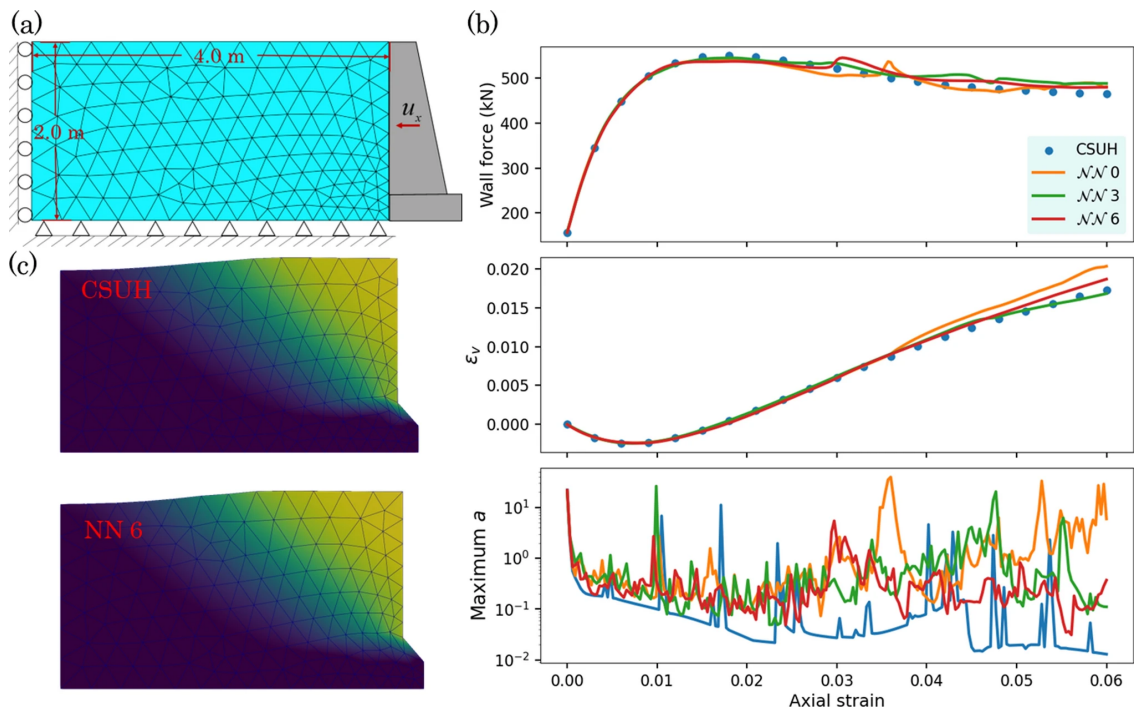


Figure 5.13: Retaining wall simulations with CSUH and NN-based model: (a) Discretisation and boundary condition; (b) Curves of the reaction force, macroscopic volumetric strain and the maximum acceleration; (c) Displacement field

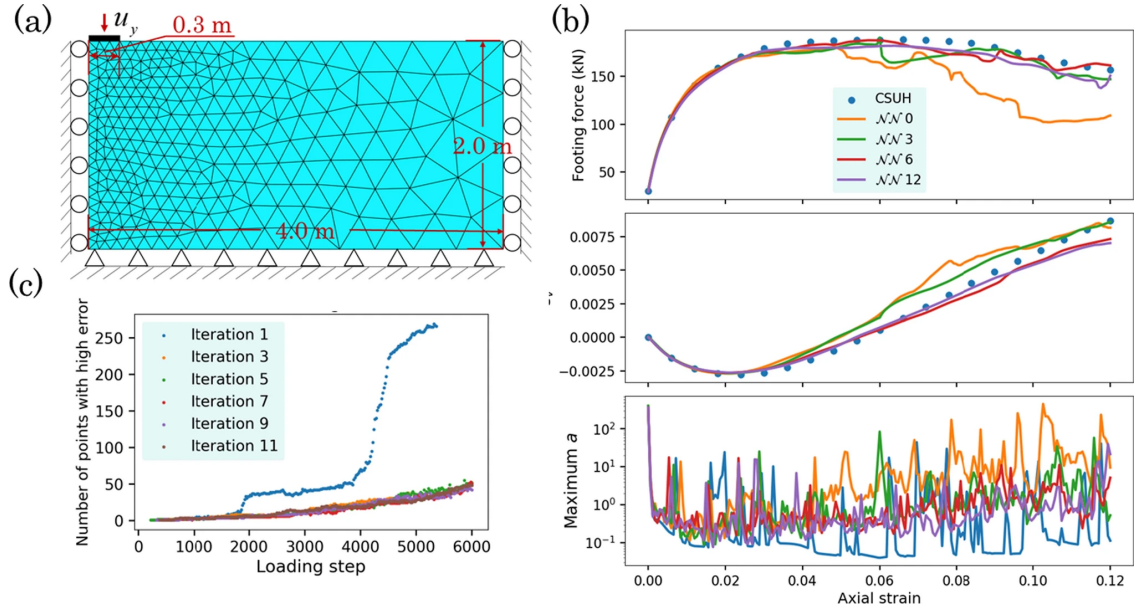


Figure 5.14: Rigid strip footing simulation results with CSUH and NN-based model (a) Discretisation and boundary condition; (b) Curves of the footing reaction force, global volumetric strain and maximum acceleration; (c) Evolution of the number of error points with loading steps

even after six iterations of check-and-revision, the reaction forces acting on the retaining wall still start to fluctuate and deviate after the loading exceeds 0.03, unlike the biaxial simulations where the results are in good agreement within the whole loading process.

### 5.3.3 Rigid strip footing

After performing the above calculations, we took the network after six check-and-revision iterations and put it directly into the computations for the strip footing case, with the model shown in Fig. 5.14a. Note that the network numbered 0 was trained without datasets from strip footing simulations. And therefore the case amounts to a wholly new problem for the network numbered 0.

Fig. 5.14b shows that before the iterations are performed, the calculation fails due to the network's misprediction, and the maximum acceleration increases gradually in the second half of the calculation, resulting in progressively more distorted stresses and strains. After three iterations, the network accuracy improved significantly to the point where it is able to give a relatively reasonable result. However, the simulations after the axial strain exceeded 0.05 still deviated from the ground truth. Repeating the check-and-revision iteration, from

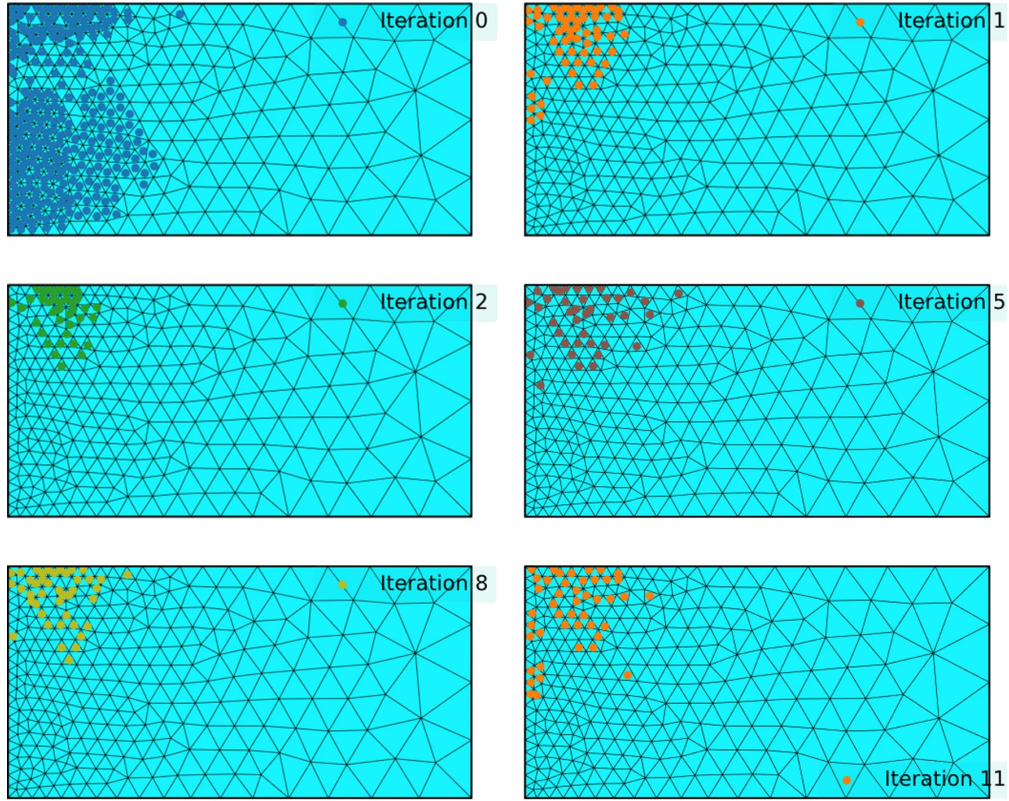


Figure 5.15: Error points selected at the last load step in each check-and-revision iteration

the third to the eleventh iteration, yields further improvements. However, some deviations in the macroscopic footing force and the whole volume strain remained.

The evolution of the number of error points with loading is shown in Fig. 5.14c. Since the training set for “NN 0” does not include any pairs of data from the strip footing simulation, the model is unable to eliminate the error after encountering it, and thus the number of error points increases rapidly. After performing a check-and-revision iteration, the number of error points in iteration 2 dropped significantly. After repeatedly widening the training range via iteration, we found that the number of error points stabilised at around 50 after 11 iterations.

Fig. 5.15 shows the positions of the Gauss points where the relative error is greater than 50% at each iteration step. The error points are mainly concentrated at the base compression area. As the check-and-revision iteration progress, the number of error points gradually decreases and eventually converges at around 50, with the distribution of error points remaining constant.

The strain results from exFEM-NN calculations on the network obtained during the



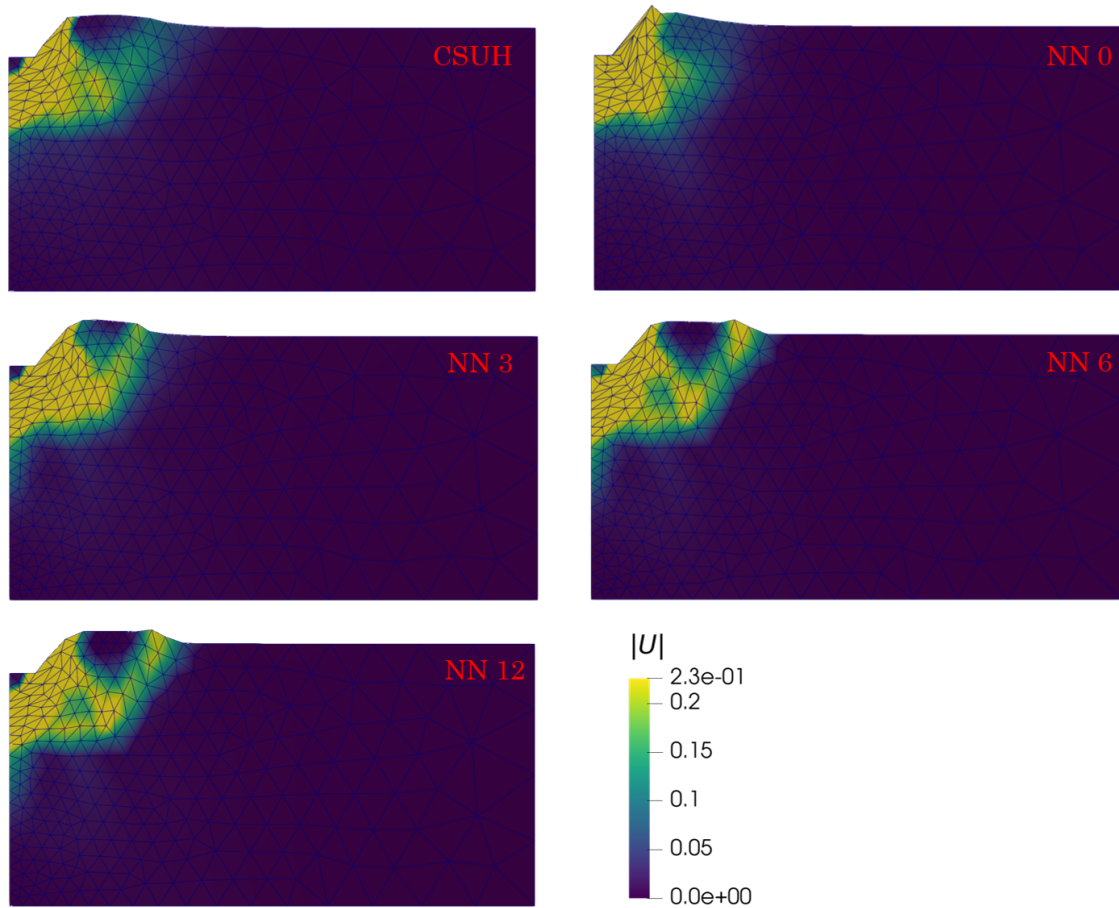


Figure 5.16: Shear strain field with different number of check-and-revision iterations

check-and-revision process are shown in Fig. 5.16. As the number of optimisations increases, the network becomes more stable at the location of maximum shear strain (top of the base). After three iterations, the network is able to reproduce the crash damage in the strip footing simulation, forming a lateral arc through the damage zone. After six iterations, the network simulation shows an oblique shear zone due to compression in the vertical direction, forming a triangular shear zone. After twelve iterations, the strain calculation does not improve further.

The above three simulations demonstrate that the check-and-revision method shown in Fig. 5.6 can be used to expand the training range. For both the biaxial and retaining wall simulations, the network is significantly improved by the check-and-revision procedure.

In the rigid step footing simulation, the network was first trained without inputting the stress-strain data from the strip footing simulation, and instead, the network is evaluated directly using check-and-revision on a completely new case. The network "NN 0" has a large

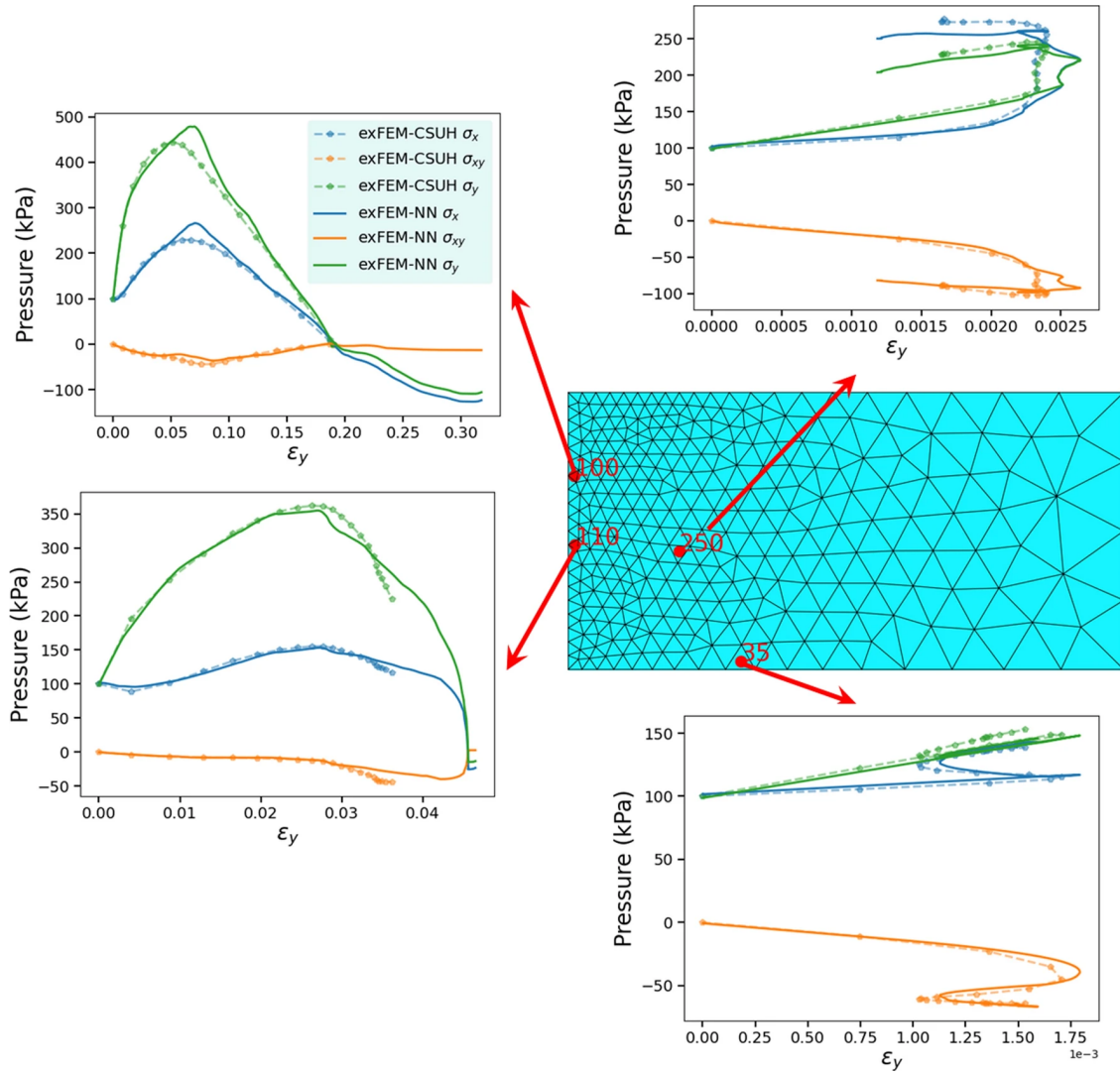


Figure 5.17: The strain-stress curves of four Gauss points of the rigid strip footing simulation

error in predicting stresses, especially in the compression and shear zones. After several iterations, the network is able to reproduce the results of the CSUH-based simulation, including the swelling and the shearing band.

However, after several iterations, there are still some error points which cannot be eliminated, indicating that the neural network, as a regressor, is always approximating the data set but cannot be completely free from mispredictions. The internal variable  $\varphi_{ij} = \sum |\Delta\epsilon_{ij}|$  was used to calibrate the historical influence in the training and predicting. When an implicit FEM solver is used, the simulation can be completed within 100 steps. So the error due to this internal variable will not be so pronounced. However, in this paper, when there is some error and instability in the NN predictions, the fluctuations of the strain increments can be accumulated in  $\varphi_{ij}$  due to its mono-increasing nature. In Fig. 5.14c, the number of error points gradually increases with loading steps, which is a vivid description of the accumulating errors.

The stress–strain curves comparison of the strip footing simulation at the four integration points is shown in Fig. 5.17. At integration points 100 and 110, the NN constitutive model is able to reproduce the failure of the material due to shear. At integration points 35 and 250, where unloading occurs, the NN model predictions deviate slightly from those of the CSUH model. In contrast, the results for point 35 are visually better than point 250 because the strain at this point is smaller and closer to the elastic state.

## 5.4 Stability of the network-based exFEM computation

In our work, the exFEM invokes the neural network to predict stresses at every Gauss point in each load step of the calculation. Any Gauss point with mispredicted stress will cause the neighbouring Gauss points to experience loading paths which is deviated from the training data. The error accumulation can lead to the overall computation collapse. Our work, therefore, requires a network with a high degree of prediction accuracy and robustness. When the predicted macroscopic results are close to the training data, we can assume that the method, to some extent, overcomes the problems of error accumulation and insufficient generalisability.

This section will primarily focus on analysing the stability of the network, with the training dataset from exFEM-IME simulations.

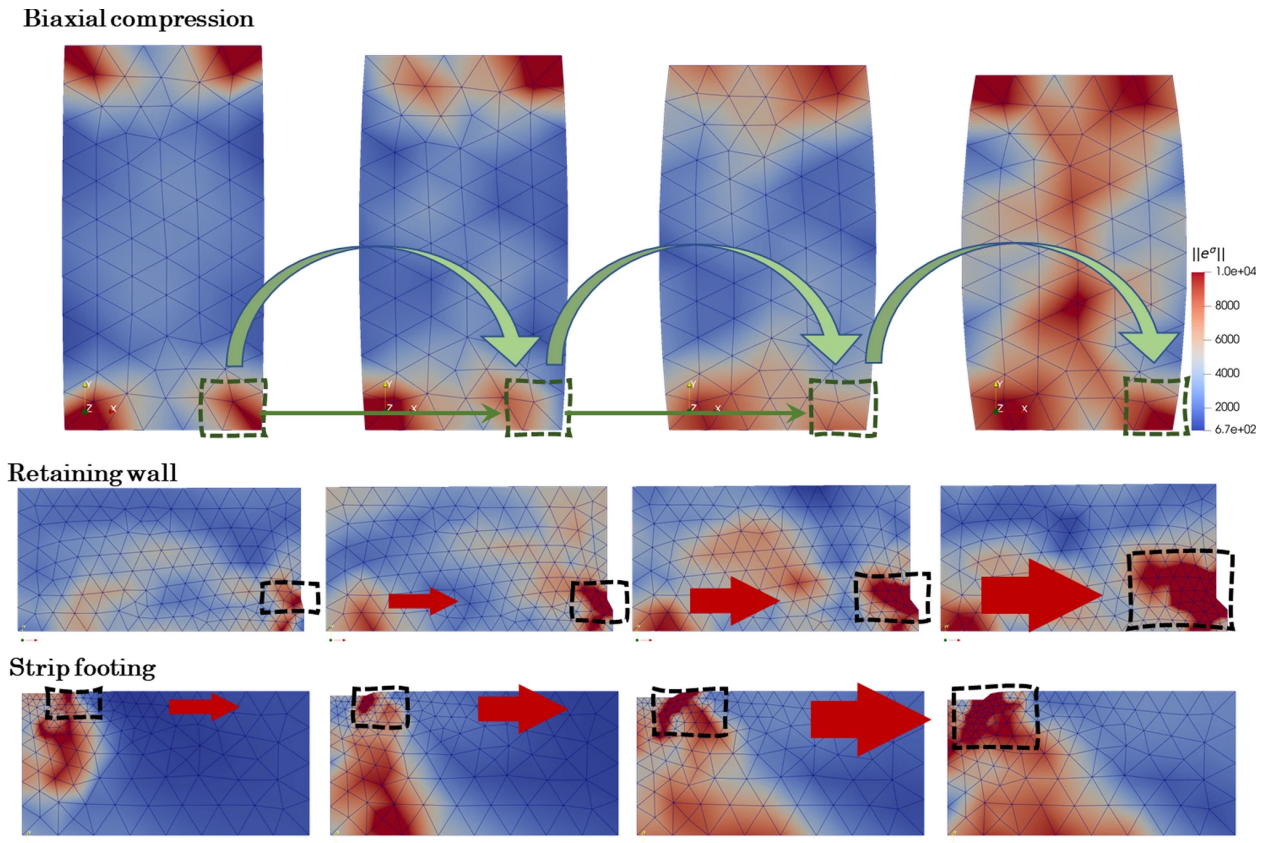


Figure 5.18: Evolution of the stress error during loading

#### 5.4.1 The emergence and propagation process of errors in the FEM calculation

In order to visualise the accumulation of errors in the computation of neural networks, we randomly train three neural networks and embedded all of them in one FEM computational model. The difference between their predicted stresses represents the prediction uncertainty. This idea is inspired by the active learning "query-by-committee" [135], where the difference between the predictions of the three networks is used to assess the accuracy of the prediction, with a smaller difference indicating that all three networks agree with the prediction and the prediction accuracy is higher, and vice versa (Fig. 5.18).

Fig. 5.18 illustrates the evolution of the prediction uncertainty in explicit loading. The errors in biaxial loading show that at the very beginning of loading (25% in the total loading steps), the network predictions already diverge at the four corners, but in subsequent loading (50%, 75%), the prediction errors at the four corners do not increase rapidly and spread, but gradually decrease or fluctuate within a certain range without spreading. This indicates that

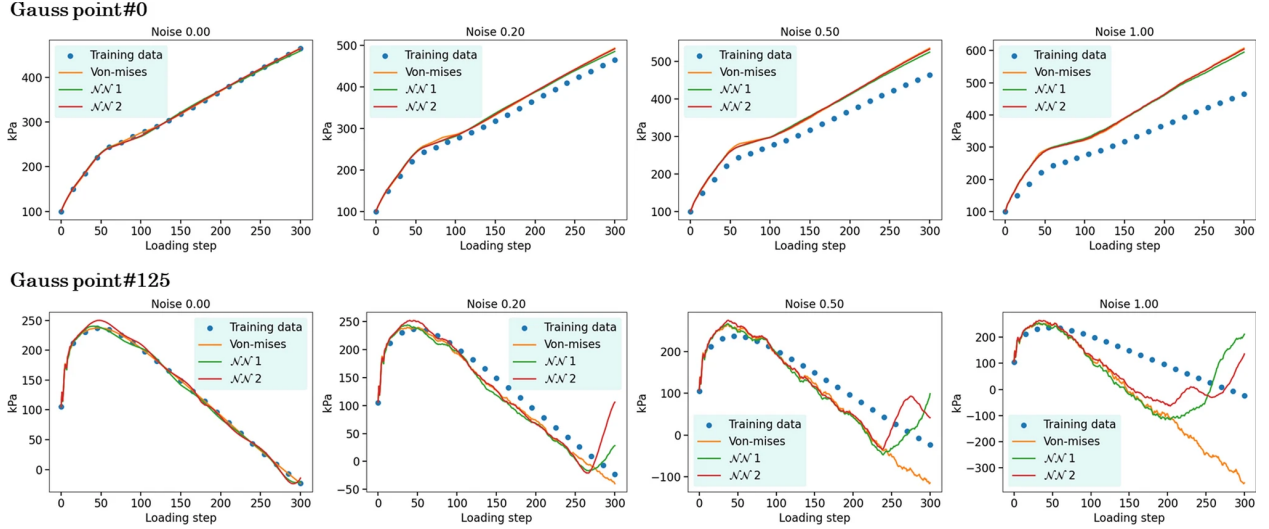


Figure 5.19: Network prediction results after introducing different levels of  $\xi$  (0, 0.2, 0.5, 1.0)

the network is able to adjust for fluctuations caused by its errors. For instance, if the next stress can be properly predicted in consideration of the previously biased strain increments, then the deviation can be gradually reduced.

The simulations of retaining wall and strip footing show that after the initial error occurs, the errors gradually accumulate and propagate. It is suggested that the error in stress deviates the strain from the trained loading path, which in turn causes the network to mispredict stresses increasingly. This vicious circle leads to a gradual increase in the number of Gauss points with large errors. The evolution of these regions illustrates the full process of computational instability appearing and propagating.

### 5.4.2 Stability analysis of NN-based predictions after adding noise

The decreasing or growing spread of errors indicates that networks vary in their ability to resist different levels of interference. We extracted the results of two representatives Gauss points in the strip footing simulation and observed the results of the neural network model prediction after applying different levels of noise (Fig. 5.19). The noisy strain increment is:

$$\Delta\epsilon_{\text{noised}} = (1 + \xi R)\Delta\epsilon \quad (5.13)$$

where  $\xi$  is the amplitude, and  $R$  is a random variable subject to a normal distribution  $N(0, 1)$ .

In the absence of noise, the network is able to reproduce the stress–strain relationship at these Gauss points with high accuracy. After introducing noise of  $\xi = 0.2$ , for Gauss point

#125, in the first half of the loading, although there are some wiggles, the predictions agree well with the model's calculation. The network predictions start to deviate significantly from the ground-truth data at the 250th loading step onwards. This proves again that the network prediction results get unstable due to the accumulation of this random error. As the amplitude of the noise gradually increases ( $\xi$  from 0.2 to 1.0), the network prediction results gradually shake heavier and the point of divergence between the predicted and calculated results gradually advances.

The predicted stress–strain curves at the selected Gauss points depicts that errors at some points in Fig. 5.18 (e.g. Gauss point #0) fade away or continue to fluctuate, while errors at some points increase and subsequently affect their neighbouring Gauss points (e.g. Gauss point #125). Similar to the butterfly effect, tiny errors at the beginning of the process can give rise to accumulated errors after several calls to the neural network, eventually leading to unacceptable deviations.

## 5.5 Discussion

For NN-based FEM computation, it is not the best-predicted data points that guarantee the computation's success but the worst data points that may result in failure. Although deep neural networks have made many successful applications, one issue we must face is that we may have overestimated their capability in reconstructing multivariate non-linear relationships for geomechanical materials, especially in terms of accuracy and generalisation because they cannot guarantee sufficiently good predictions at every point.

### 5.5.1 Challenges in network-based constitutive model development

Through the study, the limitations or challenges of rebuilding the constitutive model via data-driven method are summarised as following:

1. **Unavoidable prediction errors and the lack of generalisability.** We used classical constitutive models (CSUH and IME) as the ground-truth model to generate huge amount of training data, and introduced the Fourier series and multiplied residual blocks to improve the training accuracy. However, it is still impossible to eliminate the errors between the predicted values and the actual samples. The strain paths obtained from the NN-based FEM solution may significantly deviate from those in the training samples. The the lack of generalisability causes the poor stress predictions.



2. **The finite nature of the sampling space.** The NN's performance is dominated by the training data, so preparing training samples that are less noisy and cover as wide as possible is the primary task of the data-driven approach. However, we have to face the reality that the sampling space cannot cover the input space completely and there is no suitable method to assess the completeness of the sampling space. The CSUH and IME models require a small computational effort to prepare samples, and it will be more difficult to cover as large a strain–stress space as possible if the samples come from low-scale DEM simulations [147]. Furthermore, the loading history dependence of elastoplastic materials poses an even greater challenge for sample preparation, which requires not only strain states, but also previous loading history. The idea of covering the entire space by making a large number of samples is therefore difficult to realise. Even if we use the check-and-revision method, we are only able to train a model simply suitable for a certain BVP. If a brand new application scenario is computed using a trained neural network model, unexpected data points in the training space will still be encountered, see Sec. 5.3.3 for a strip footing simulation.
3. **Error accumulation in multi-step calculations.** FEM calculations, especially explicit solution methods, require invoking constitutive models intensively. The prediction errors at the previous loading steps influence the next strain increments, leading to input strain and state-related internal variables going beyond the original training space. If only a neural network is required for a single prediction, there is no need to consider the error accumulation. Figures 5.6 and 5.19 show the causes and effects of error accumulation during the computation, respectively.

## 5.5.2 Possible development

Machine learning, and in particular neural network methods for data regression, is a very easy thing to get started with, thanks to the development of various libraries for neural networks. But how to train a neural network model that can be used for BVP analysis is not actually a simple problem. At current stage, machine learning models for mechanical calculations exist only in academic research and do not appear in any commercial software or other distributions of wide-used open source software. The research is still very much in its infancy. After all, it is unacceptable to use an entirely black box for safety analysis of engineering. If one wants a machine learning constitutive model that can be used for generalisation, there are two routes:

1. A fully data-driven approach like the one in Otiz’s work [73, 143], without the use of a surrogate agent model, then the constitutive response extracted from the dataset can be perfectly acceptable. But the finite range of the datasets and the computational efficiency are other problems;
2. Introduce physics to interpret the machine learning model and help it to generalise.

### 5.5.3 Improvement in calculation efficiency

In numerical calculations especially when the number of integration points is sufficiently large, the speed of the constitutive model significantly affects the overall efficiency. In our previous work, we investigated the effect of ANN for accelerated FEM-DEM multiscale simulations. The computation of a mathematical equation-based constitutive model is significantly faster than that of a low-scale DEM simulation. The CSUH model requires plasticity correction after the material enters yield, and the loading step size needs to be reduced to a certain level in order to accurately reproduce the material non-linearity. In addition, the plasticity correction calculation, where the elasticity, yielding and hardening, is quite complex and time-consuming. Especially at locations on the verge of failure, the step size needs to be further reduced, otherwise, the average stress  $p < 0$  leads to a breakdown of the calculation. As shown in Tab. 5.2, we compare the computational speed of the different constitutive models in the FEM. The IME model is the fastest because it’s relatively simpler, and the NN-based model is faster than the CSUH model. The boost in the biaxial simulation is lower because, in the biaxial simulation, not as many material points are nearly damaged as in the other two simulations.

Table 5.2: Summary of time consumed in different simulations with different constitutive models

Simulations	Consumed time (min)			Number of Gauss points	Number of steps	Improvements (ANN vs CSUH)
	IME	CSUH	NN			
Biaxial compression	5.78	10.86	7.86	484	5000	27.62%
Retaining wall	-	7.48	5.23	271	6000	30.08%
Footing	-	16.30	10.70	546	6000	34.36%



## 5.6 Concluding remarks

This work proposes an explicit FEM computational framework coupled with a neural network, which circumvents the dependence on the tangential matrix during the non-linear iteration. The conditional stability in explicit FEM computation requires a sufficiently small time step size and thus a large number of load steps are needed, which gives rise to an intractable error accumulation problem. To improve the learning capability of neural networks, Fourier features and multiplied residual blocks are used to improve the network. The check-and-revision calculation method is proposed to check, correct and save the stress-strain data at the error points for expanding the network training range. After several iterations of the check-and-revision, the number of error points is significantly reduced.

In the three numerical tests presented in this paper, the neural network effectively reproduces the two constitutive models (IME and CSUH). Theoretically, the neural network can reproduce all the constitutive relationships contained in the data, provided that the state variables are suitably selected and the training data is adequate.

However, it is impossible to completely cover the infinite input space, especially for history-dependent and high-dimensional problems. By examining the emergence and accumulation process of error, it is found that noise perturbations can cause the inputs to the network to gradually exceed the original training range. The error accumulation and the lack of generalisability can lead to the failure of the exFEM-NN computation.

# Chapter 6

## Recurrent network-based constitutive model

### 6.1 Introduction

Since Hooke's Law, the development of models to describe material constitutive relationships has become an essential subject in fields such as materials science and geoen지니어ing. Numerous models based on the assumption of a continuous medium have been proposed [41, 49, 112, 148], yielding remarkable successes. However, as these constitutive models have been enhanced with an increasing number of parameters, they have become intricate and challenging to apply universally in engineering practice, despite their growing accuracy. Consequently, the question arises: How can we develop accurate models in a straightforward manner that are easy to apply broadly?

Machine learning appears to provide a potential solution to this issue. From an ML perspective, models can be derived once sufficient data is available. The utilization of ML in constitutive modelling was initiated by Ghaboussi et al. [90], where a computational and knowledge representation framework called neural networks were employed to train an ML-based model for concrete using experimental data. The encouraging results of accurately predicting stress responses using neural networks motivated researchers to further develop neural network-based constitutive models.

In light of the rapid advancement of ML, intensive efforts are currently being devoted to incorporating ML into the implementation of constitutive models. Unlike traditional

mathematical models that rely on pre-defined constitutive formulations, ML-based constitutive models leverage data sets to exploit stress-strain relationships with minimal or no prior assumptions about the material's behaviour. Notably, recent studies have demonstrated the competence of neural networks (NN) in reproducing complex constitutive responses [98, 101, 104, 149, 150]. Particularly noteworthy is the introduction of a Minimal State Cell in the recurrent structure by Bonatti et al. [103], aiming to train an ML-based model capable of capturing path-dependent material behaviour. With such a cell, the deep NN model exhibits satisfactory performance in terms of history dependency, even when trained with a limited number of loading paths.

Efforts have also been made to integrate ML-based constitutive models into finite element (FE) analysis of boundary value problems [94, 151]. In the study by Lefik et al., [151], the introduction of stress and strain tensor rotation, along with a scalar parameter, aimed to mitigate the influence of incremental step size during the training stage. Guan et al. [147] utilised a NN as a constitutive agent to replace computationally expensive sub-scale DEM simulations within a coupled FEM-DEM multiscale computational framework [4], thereby enhancing computational efficiency. Regarding the training of ML-based surrogate models, Recurrent Neural Networks (RNN) and its derivatives such as Long Short-Term Memory (LSTM) [127] and Gated Recurrent Unit (GRU) [152] have gained prominence. For example, Ghavamian et al. [105], Logarzo et al. [106], and Guan et al. [153] incorporated RNNs for sequence prediction, which were subsequently embedded within FEM simulations.

Alternatively, Gaussian Process Regression (GPR) can be utilised to implement the strain-stress mapping  $(\epsilon_{ij} - \sigma_{ij})$ . Rocha et al. [154] employed GPR to correct the trial stress, initially assessed based on linear elasticity assumptions, to accurately reproduce FE<sup>2</sup> calculations. However, the classic GPR has limitations in handling large datasets, as the covariance matrix becomes too large, making it challenging to find the inverse. Fuhg et al. [155] introduced Local approximation GPR (LaGPR), which predicts stress and tangent matrices by interpolating within neighbouring data points. Remarkably, a distinct data-driven searching method was proposed by Kirchdoerfer et al. [143]. Without constructing any regression or mapping agent, this method directly searches for the most suitable output within datasets prepared via offline simulations. The approach relies on two mapping operators: (i)  $P_E$ , which maps the trial state of  $(\epsilon^{(n)}, \epsilon^{(n)})$  to a state that matches the geometry and equilibrium condition, i.e.,  $(\epsilon^{(n)}, \epsilon^{(n)}) = P_E(\epsilon^{(n)}, \epsilon^{(n)})$ , and (ii)  $P_D$ , which searches for the closest data pairs in the offline prepared datasets to define the new trial state, i.e.,  $(\epsilon^{(n+1)}, \epsilon^{(n+1)}) = P_D(\epsilon^{(n)}, \epsilon^{(n)})$ .

It is important to emphasise that networks exhibit excellent performance in interpolation but are not as effective in extrapolation [147]. To address this limitation of network-based models, researchers have attempted to incorporate prior knowledge, such as principles of physics, to aid in the training of these models. One popular approach in this regard is known as Physics-Informed Neural Network (PINN) [156]. By incorporating physics-based constraints, the performance of NN-based models can be largely improved, even when dealing with inputs that are beyond the range of the training data. In the context of elastoplasticity, Vlassis et al. [118] aimed to train a network capable of reproducing hyperelastic behavior while implementing a level-set yield surface to govern plastic deformation and hardening. On a similar note, Fuhg et al. ([157]) proposed a convexity-constrained method that utilized various tools such as NN, GPR, and support vector machines (SVM) to ensure the convexity of data-driven yield surfaces. Jang et al. [158] provided the model with precise information regarding linear elasticity and the hardening function. They then utilised the network model for plastic return mapping, where the predicted stress  $\sigma$  is a function of the trial stress  $\sigma^{TR}$  and the hardening function  $\rho(\bar{\epsilon}^p)$ , represented as  $\sigma = \mathcal{NN}(\sigma^{TR}, \rho(\bar{\epsilon}^p))$ , where  $\bar{\epsilon}^p$  represents the accumulated plastic strain. However, it should be noted that introducing physics-based knowledge can make the training process more challenging and less efficient [159].

In addition to incorporating prior physics knowledge into the loss function, there is an approach that leverages the architecture or activation of neural networks to automatically satisfy certain physics-based constraints. This approach is known as Physics-Consistent Neural Network (PCNN) [160]. One commonly imposed constraint in training NN-based constitutive models is coaxiality. For example, Huang et al. ([142]) employed orthogonal decomposition to reduce the number of stress and strain tensor components from six to three using spectral decomposition. Then, only the components corresponding to the principal directions were involved in the training process. To further enforce the constraint of coordinate independence, Yang et al. ([112]) coerced the model to ensure that the predictions remain consistent regardless of the order of the input axes.

Despite the advancements in data-driven constitutive model development, several challenges still remain and require further attention.

- The coaxiality assumption can be used to simplify the dimension reduction process for elasticity models. Its introduction into plastic works, however, must be carried out with caution. For example, in the "MAP123" works from Tang's group ([109–111]), they detailed where and how the datasets with six components can be degraded to

volumetric and shear/effective directions i.e.  $\epsilon_v - \sigma_v$  and  $\epsilon_s - \sigma_s$ .

- Additionally, FE calculations typically involve multiple analysis steps as shown in Fig. 6.1. This implies, its output of a NN-based constitutive model will be used to compute the input for the next analysis step. Therefore, FE simulation usually possesses a recurrent structure regardless of the use of a recurrent NN. Notably, the results of a recurrent structure are significantly influenced by the step size. For instance, the outcomes can vary significantly depending on whether a load step is divided into ten or one hundred substeps. The recurrent structure can also lead to prediction error accumulation, which is a common limitation associated with such recurrent structures.
- Apart from these, the choice between the sequence-training and one-to-one training is plaguing. Sequence training can provide a better solution to the path dependency problem; it is hindered by the gradient explosion or vanishing. On the other hand, one-to-one training can achieve higher accuracy. However, it requires the explicit feeding of internal variables into the network, which can complicate the dataset preparation. Addressing these challenges will contribute to the development of data-driven constitutive models that can effectively capture path dependency while maintaining high accuracy in predictions.

The main focus of this work is to utilise networks for reproducing an elastoplastic model. In this chapter, we focus on general elastoplastic models, rather than models specifically designed for granular materials. To ensure accuracy, we did not rely on the dimension reduction method based on the coaxiality assumption. Instead, the physics-extended basis function and the isotropic swapping are introduced to apply constrained physics to the network-based material cell, thereby enhancing the model's generalization ability. The random Gaussian Process is employed for random loading path generation. Furthermore, an adaptive step size adjustment method was introduced to reduce the error resulting from mismatches between the step size in the FE simulation and the training dataset. The study also includes the reproduction of the classical model using both recurrent network-based and deep network-based material cells, with a thorough comparison between them. Overall, this work emphasizes the utilization of networks to accurately reproduce an elastoplastic model, incorporating physics-based constraints, random loading paths, adaptive step size adjustment, and a comparison between different network architectures.

## 6.2 Methodology

### 6.2.1 Governing equations and explicit FEM solver

Based on Newton's second law, the governing equation can be written as (for clarity, formulas in this section are expressed in Einstein summation form):

$$\begin{aligned}\rho\ddot{u}_j &= \sigma_{ij,i} + \rho b_j & \text{in } \Omega \\ u_j &= \bar{u}_j & \text{on } \partial\Omega_u \\ \sigma_{ij}n_i &= \bar{t}_j & \text{on } \partial\Omega_t\end{aligned}\tag{6.1}$$

where  $\ddot{u}_j$  is the acceleration, and  $\partial\Omega_u$  and  $\partial\Omega_t$  are the Dirichlet boundary and the Neumann boundary, respectively. After introducing the Galerkin method we obtain the weak form of Eq. 6.1 as follows:

$$\int_{\Omega} \rho\ddot{u}_j\phi_m dv = \int_{\Omega} (\sigma_{ij,i} + \rho b_j)\phi_m dv\tag{6.2}$$

where  $\phi_m$  is the basis function and the acceleration within an element is interpolated by  $\ddot{u}_j(x) = \sum_n^N \ddot{u}_j^{(n)}\phi_k(n)$ , where  $N$  is the total number of basis functions. Note that in the description here  $i$  and  $j$  denote the numbering of the dimensions of the solution space  $\mathbb{R}^2$ , and  $m$  and  $n$  represent the numbering of the basis functions. Utilising part integration and the divergence theorem, one can convert Eq. 6.2 into:

$$\ddot{u}_j^{(n)} \int_{\Omega} \rho\phi_n\phi_m dv = \int_{\partial\Omega} t_j\phi_m dS - \int_{\Omega} \sigma_{ij}\phi_{m,i} dv + \int_{\Omega} b_j\phi_m dv\tag{6.3}$$

The above formula can be expressed in a simplified form:

$$M_{mn}\ddot{u}_j^{(n)} = T_{mj} - F_{mj} + B_{mj}\tag{6.4}$$

where  $M_{mn} = \int_{\Omega} \rho\phi_n\phi_m dv$  is the mass matrix and  $T_{mj}, F_{mj}$  and  $B_{mj}$  correspond to the boundary force, the internal force and the body force tensor, respectively, of the right-hand sides of Eq. 6.3.

The finite element method discretises the entire model into a set of elements and applies the aforementioned Galerkin method to each element. Matrices like  $M_{mn}^e, T_{mj}^e, F_{mj}^e$ , and  $B_{mj}^e$  are first evaluated within each element, and then global matrices are assembled by combining these element matrices. The acceleration of all the nodes can then be obtained by solving a system of linear equations.

The central difference method is then employed to update the displacement according to the time integral:

$$\begin{cases} \dot{u}_j|_{t+0.5\Delta t} = \dot{u}_j|_{t-0.5\Delta t} + \Delta t \ddot{u}_j|_t \\ u_j|_{t+\Delta t} = u_j|_t + \Delta t \dot{u}_j|_{t+0.5\Delta t} \end{cases} \quad (6.5)$$

where the subscript indicates the moment. The first equation can be interpreted as the velocity at the moment  $(t + 0.5\Delta t)$  is equal to the velocity at the moment  $(t - 0.5\Delta t)$  plus  $\Delta t$  multiplied by the acceleration at the moment  $t$ .

The displacement and strain increment are evaluated assuming infinitesimal deformation as follows:

$$\begin{aligned} du_j|_{t+\Delta t} &= \Delta t \dot{u}_j|_{t+0.5\Delta t} \\ d\epsilon_{ij}|_{t+\Delta t} &= \frac{1}{2} (\Delta u_{j,i}|_{t+\Delta t} + \Delta u_{i,j}|_{t+\Delta t}) \end{aligned} \quad (6.6)$$

The coupling of the material cell to the FEM solver was accomplished through a Python interface. Alg. 4 is added to describe the coupling process. The finite element model uses eight-node planar elements, each containing four integration points.

## 6.2.2 Material cell

In this section, we introduce the concept of a material cell  $\mathcal{M}$  which is depicted in Fig. 6.1a. In the analysis of a BVP, the constitutive relationship can be generally expressed as

$$\mathcal{I} = \mathcal{M}(d\epsilon_{ij}, \mathcal{I}_0) \quad (6.7)$$

where  $\mathcal{I}_0$  and  $\mathcal{I}$  represent the internal variables before and after the loading of  $d\epsilon_{ij}$ , respectively. The internal variables encompass a range of quantities such as stress, strain, plastic deformation, plastic work, etc. The constitutive model is applied iteratively at Gauss points at each loading sub-step. The output  $\mathcal{I}$  serves as an input for the subsequent sub-step.

Similarly, in the recurrent neural network (RNN) cell shown in Fig. 6.1b, the hidden state  $h$  is updated by

$$h = \text{RNN}_{\text{Cell}}(x, h_0) \quad (6.8)$$

where  $x$  is the input, and  $h_0$  is the original hidden state. As a special type of RNN, GRU can alleviate the gradient vanishing and explosion problems that often occur during long-

sequence training. The tensor operations within the GRU cell can be expressed as

$$\begin{cases} r = \sigma_g(L_{xr}(x) + L_{hr}(h_0)) \\ z = \sigma_g(L_{xz}(x) + L_{hz}(h_0)) \\ n = \tanh(L_{xn}(x) + r \odot L_{hn}(h_0)) \\ h = (1 - z) \odot n + z \odot h_0 \end{cases} \quad (6.9)$$

where  $L(x) = wx + b$  is a fully-connected layer, the subscript denotes the where is the input from and where is the output going,  $r$  denotes the reset gate vector,  $z$  denotes the update gate vector, and  $n$  denotes the candidate activation vector,  $\sigma_g(\cdot)$  represents the Sigmoid activation function. Within each RNN cell, there involve six fully-connected layers. The computational procedure for GRU units has been extensively described in various literature and will not be detailed in this paper. The formula for the GRU unit can be viewed in the PyTorch Docs.

Due to the connection between the two we propose a network-based material cell  $\mathcal{M}_{NN}$ . By inputting the strain increment and internal state variables, the material cell can update the internal variables via the trained network, which can be expressed as:

$$\mathcal{I} = \mathcal{M}_{NN}(d\epsilon_{ij}, \mathcal{I}_0) \quad (6.10)$$

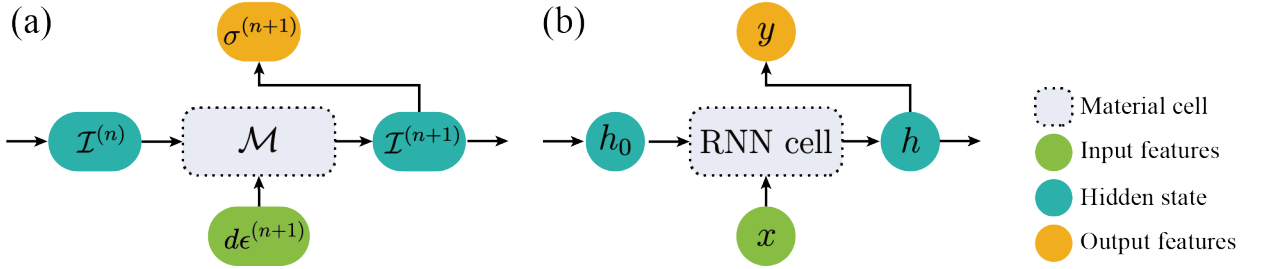


Figure 6.1: (a) Schematic of the material cell  $\mathcal{M}$ . (b) Tensor flow in the recurrent neural network cell.

After introducing the material cell, we would like to introduce one-to-one training and sequential training. Note that the *one* in *one-to-one* does not refer to a single input feature, but rather to a snap, which represents the input and output information obtained at a single load step. One-to-one training will only use the information from a single snap to calculate the error. Sequential training, on the other hand, iteratively uses the information from all the snaps in a single computation. It is also known as recurrent network training method.

As shown in Fig. 6.2, we use material cell to predict the stress responses. The required inputs are the current state, the internal variables and the strain increments. The loss function



for sequential training is:

$$L_{\text{seq.}} = \sum_i \left( \hat{\sigma}^{(i)} - \sigma^{(i)} \right)^2 \quad (6.11)$$

The loss function for one-to-one training is:

$$L_{\text{one.}} = \sum_i \left( \hat{\sigma}^{(i)} - \sigma^{(i)} \right)^2 + \sum_i \left( \hat{\mathbf{h}}^{(i)} - \mathbf{h}^{(i)} \right)^2 \quad (6.12)$$

If sequential training is used, there is no need to explicitly obtain the internal variable  $\mathbf{h}$ . With the output of one-to-one training as the input for the next prediction, the one-to-one training can also be turned into sequential training. It should be emphasised that in sequential training, there is the problem of gradient vanishing and gradient explosion. The special gate structures of GRU [152] or LSTM [127] are needed to alleviate this problem. In the subsequent works in Sec. 6.3 and 6.4, sequential training is used, except for DNN-based material cell in Sec. 6.4.6 where one-to-one training is used. The advantages of the two training methods are discussed in Sec. 6.5.

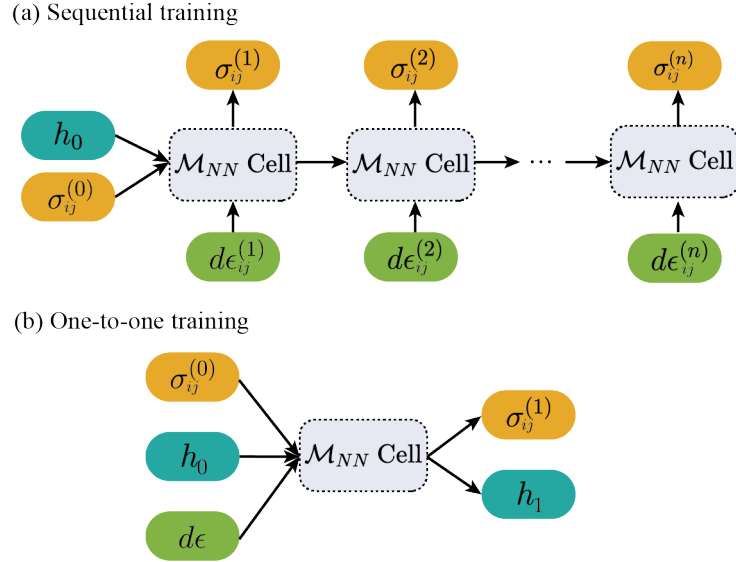


Figure 6.2: The sequential and one-to-one mapping method

The material cell is trained and then tested on the test set which has never been involved in training. Note that the loading path of the test set is generated using the same method as the training set (Sec. 6.2.3), but different from the training set. The test set and the training be described as independently-identically-distributed in mathematical statistics. After satisfactory results are obtained on the test set, the material cell is embedded into the explicit FE solver for further validation. Alg. 4 shows the FE solver calling material cell computation on a computational step.

---

**Algorithm 4** FE solver with the material cell at time  $t$ 

---

**Require:** Velocity of last half step  $\dot{\mathbf{u}}|_{t-0.5\Delta t}$  (for central difference), internal/hidden state at last step  $\mathbf{h}_0$

- 1: Compute the acceleration  $\ddot{\mathbf{u}}|_t$  via Eq. 6.4.
  - 2: Update  $\dot{\mathbf{u}}|_{t-0.5\Delta t}$  to  $\dot{\mathbf{u}}|_{t+0.5\Delta t}$
  - 3: Compute the displacement increment  $d\mathbf{u}$  and strain increment  $d\boldsymbol{\epsilon}$  via Eq. 6.6. Note, the displacement loading of this step is added to  $d\mathbf{u}$ .
  - 4: Get the constitutive prediction  $(\boldsymbol{\sigma}, \mathbf{h}) = \mathcal{M}_{NN}(d\boldsymbol{\epsilon}, \mathbf{h}_0)$
  - 5: Update the internal node force with the updated stress  $\boldsymbol{\sigma}$  via Eq. 6.3
- 

The neural networks are constructed using the Torch library for Python, version 1.13.1+cu117. The default parameters of the Adam optimiser used are "lr=0.001, betas=(0.9, 0.999), eps=1e-08, weight\_decay=0".

### 6.2.3 Loading path generating via random Gaussian Process

The strength of data-driven approaches lies in their powerful mapping ability, as they leverage the available data to its fullest extent. Theoretically, with a sufficient quantity of high-quality data, it becomes possible to train a model that is both precise and general ([147]). However, it is important to note that the potential solution space is infinite, while the datasets at hand typically have limited coverage as is shown in Fig. 6.3. To date, there has been no consensus on how to assess the completeness of training data coverage, although numerous methods have been developed to expand the sampling space. In this study, we employed a random GP to generate the smooth random loading paths ([106]).

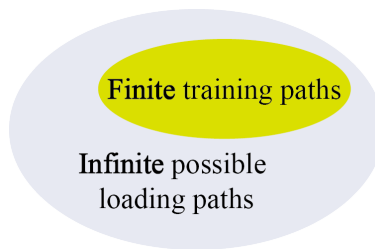


Figure 6.3: Finite training paths and infinite possible loading paths

The function value of a Gaussian process at some input points  $x$  are distributed as a Gaussian distribution with mean  $u$  and variance  $v^2$ . These points are linked by covariance. If the covariance matrix  $\boldsymbol{\Sigma}$  is diagonal, these points are distributed independently. The

Gaussian process can be expressed as:

$$f \sim \mathcal{GP}(\mathbf{u}, \mathbf{\Sigma}) \quad (6.13)$$

where  $\mathbf{u}$  is the mean vector and  $\mathbf{\Sigma}$  is the covariance matrix to define the curvature and amplitude. Details regarding the evaluation of  $\mathbf{u}$  and  $\mathbf{\Sigma}$  are provided in Appendix 6.6.1.

The random GP uses the Gaussian kernel, given by:

$$\mathbf{\Sigma}(x|v, l) = k(x, x'|v, l) = v^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) \quad (6.14)$$

where  $v$  is used to restrict the distribution range, and  $l$  represents the characteristic length of the correlation corresponding to the approximate distance between two points on the curve. Assuming a normal distribution, approximately two-thirds of the points fall within the range of  $u \pm v$ .

To ensure the diversity of loading paths, we assume that the components in different directions follow the same kernel but with different hyperparameters. Because strain components, such as  $\epsilon_{11}$  and  $\epsilon_{12}$ , may have different amplitudes, it is inappropriate to use the same values of  $v$  and  $l$  for all three components of the strain, namely  $[\epsilon_{11}, \epsilon_{12}, \epsilon_{22}]$ .

Instead of generating the strain components directly in space  $\epsilon \in \mathbb{R}^3$ , the two principle components and rotational angle are first generated and then transformed back to the original coordinate. If the components of the strain tensor are directly generated, the strain components in the 12 direction can be incompatible with those in the 11 and 22 directions, and it is difficult to imagine how the samples are distributed on the  $\pi$  plane.

We use distinct hyperparameters in the random GP to generate the principal components  $\{\epsilon_1^{PR}, \epsilon_2^{PR}\}$  and the rotation angle  $\theta_\epsilon$ . Once obtaining the sequences of  $\epsilon_1^{(PR)}, \epsilon_2^{(PR)}, \theta$ , tensor rotation, as described in Section 6.6.3, is performed to transform them back to the original space of  $\epsilon_{11}, \epsilon_{12}, \epsilon_{22}$ . This approach enables the generation of loading paths with diverse strain components while preserving the appropriate transformations between different coordinate systems.

In the random loading path generation, we adopt the following settings for the hyperparameters:

- For the strain components in principal directions:  $v_\epsilon$  is uniformly distributed between 0.1 and 0.18, and  $l_\epsilon$  is uniformly distributed between 0.1 and 0.5.
- For the rotation angle  $\theta + \theta_0$ :  $v_\theta$  is uniformly distributed between 0 and  $\pi/4$ .

- The characteristic length  $l_\theta$  is the same as  $l_\epsilon$ , and the initial rotation angle  $\theta_0$  follows a uniform distribution between  $-\pi$  and  $\pi$ .

Random paths can be easily generated based on the covariance matrix. Three paths generated under these settings are displayed in Fig. 6.4. The Gaussian process in this section generates a set of random strain paths, which is then applied to various elastoplastic constitutive models to obtain strain-stress datasets. Training for different constitutive models in Sec. 6.4 is based on these datasets.

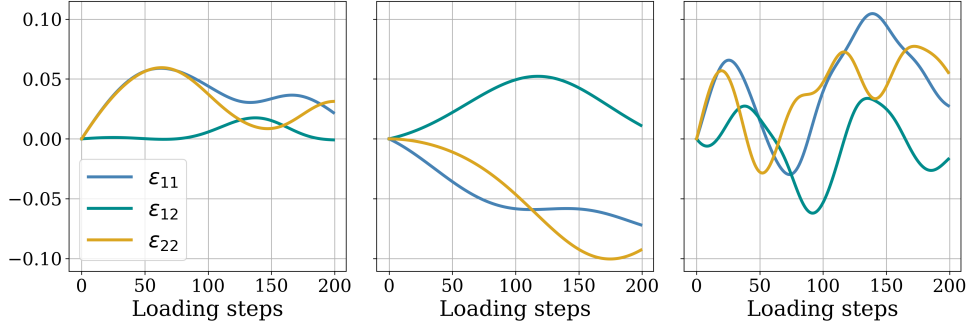


Figure 6.4: Three examples of the random loading paths. The curves represent each of the three components of the strain tensor in the two-dimensional case.

### 6.3 Material cell without physics: training and testing

We begin by constructing a material cell using the GRU architecture and stress sequences generated under the  $J_2$  model (Sec. 6.6.2). Before training, the values of strain and stress data are all normalised via:

$$\bar{x} = \frac{x - \mu_x}{\sigma_x} \quad (6.15)$$

where  $\mu_x$  is the mean value and  $\sigma_x$  represent the standard deviation.

The structure of the material cell is illustrated in Fig. 6.5a. This architecture enables the material cell to capture the temporal dependencies and patterns in the data, allowing it to learn the constitutive relationship between input and output sequences. The parameters in the material cell will be optimised during the training process to minimise the discrepancy between predicted and target stress values. Following training, we can evaluate its performance in reproducing the behaviour of the elastoplastic models on test sets, based solely on the data-driven approach provided by the GRU architecture.

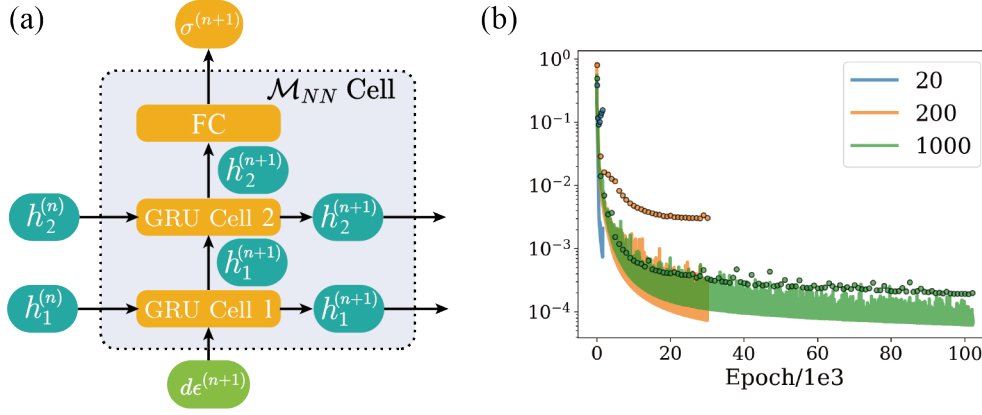


Figure 6.5:  $\mathcal{M}_{NN}$  consists of two layers of GRU and a fully connected (FC) layer and (b) the loss evaluation with different numbers of training sets, where the solid line indicates the training loss and the dots represent the validation loss.

The calculations in Fig. 6.5a can be described as follows:

$$\begin{cases} h_1^{(n)} = \text{GRU}_1(x^{(n-1)}, h_1^{(n-1)}) \\ h_2^{(n)} = \text{GRU}_2(h_1^{(n)}, h_2^{(n-1)}) \\ y^{(n)} = \text{FC}(h_2^{(n)}) \end{cases} \quad (6.16)$$

where the input  $x$  to the material cell represents the strain increment.  $x$  is fed into the first GRU layer denoted as  $\text{GRU}_1$ , which processes the input and updates its hidden state  $h_1$ . The hidden state from the first GRU layer is then passed to the second layer  $\text{GRU}_2$ , which further processes the input and updates its hidden state  $h_2$ . The final hidden state from the second GRU layer is fed into a fully-connected layer, which performs a linear transformation of the hidden state to obtain the predicted stress. These calculations are performed iteratively for input sequences to predict the corresponding output sequences. During the training process of the material cell, the hidden states in the GRU layers do not have explicit physical meanings and are initialized as null at the beginning.

According to Eq. 6.9, for one GRU, the number of trainable parameters is  $3 \times (n_{in} \times w_h + w_h) + 3 \times (w_h \times w_h + w_h)$ , where  $n_{in}$  is the number of input features and  $w_h$  is the width of the hidden state. The trainable parameters of the linear layer with biases is  $w_h \times n_{out} + n_{out}$ , where  $n_{out}$  is the number of the output features. The network in this section consists of two GRU layers and a fully-connected layer, resulting in a total of 8,823 trainable parameters.

To prevent overfitting and improve the model's generalisation, the early-stopping scheme, as introduced in previous works such as [101, 147], is adopted. In this scheme, a portion of

the dataset, typically around 20%, is randomly selected and reserved for validation purposes. This portion is not used during the training process but is used to evaluate the model's error after a certain number of epochs. The training process continues until the so-called validation error reaches a point where it no longer significantly decreases.

The material cell is trained on the complete sequences of  $(\epsilon, \sigma)$  data. To investigate the sensitivity of the material cell to the amount of training data, different numbers of datasets are used in the network training. Fig. 6.5b demonstrates that when only 20 sets of data are used for training, the network achieves good prediction performance on the training set but performs poorly on the validation set. This indicates that with a small amount of data, the model essentially memorises or "imitates" the training set without truly capturing the underlying physics. As the training data gradually increases, the errors on the training set and validation set gradually converge. This suggests that with a larger and more diverse training dataset, the network is able to learn the underlying physics and generalise better to unseen data.

Fig. 6.6 presents the results of the purely data-driven material cell using a total of 1000 sets of loadings. The error distribution exhibits a mean value of  $2.4 \times 10^{-3}$  and a maximum error of  $2.4 \times 10^{-1}$ . Fig. 6.6b illustrates the best prediction under a random loading path, demonstrating its capability to accurately capture the underlying physics. On the other hand, Fig. 6.6c displays the worst prediction, highlighting the limitations and potential challenges of purely data-driven approaches. Despite this, it is important to note that even in this case, the model's performance remains relatively good. With a sufficient amount of data, the purely data-driven model demonstrates a high level of accuracy in its predictions after training.

Even though the purely data-driven material cell achieves high accuracy, the discrepancy between the strain increments in training data and the FE analysis limits its effectiveness as a true constitutive model. The gap can be mitigated via linear transformation as mentioned by [103].

## 6.4 Material cell with physics extensions: training, test and FE analysis

In this section we introduce symmetry constraints and physical extensions for generalisability, and adaptive step size adjustment for step size dependence.

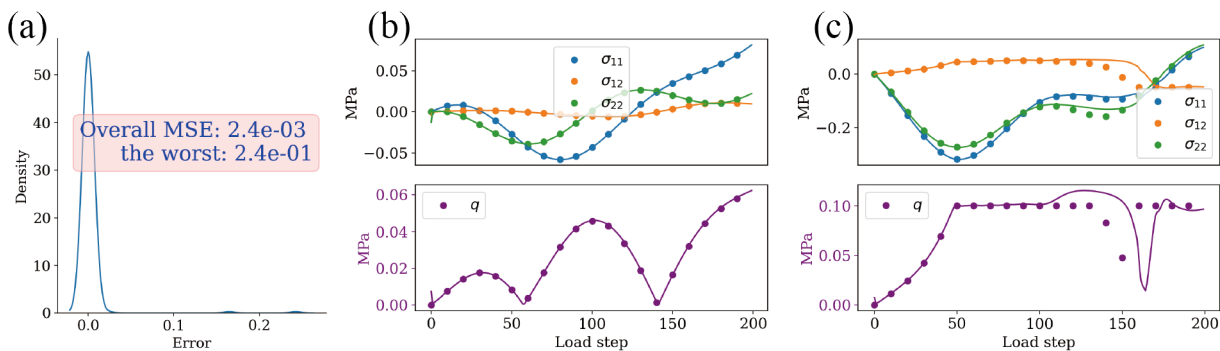


Figure 6.6: Results of the two-layer GRU model trained on data generated based on  $J_2$  model (a) Prediction error distribution. The vertical axis indicates the sample distribution’s density, with the curve representing an area integral equal to 1.

(b) Best prediction (c) Worst prediction. The curves corresponds to three components of the stress tensor. The dots indicate the training data and the lines are the predictions.

Material cells are trained using data from three different elastoplastic models, ranging from simple to complex. The objective is to assess the performance of the material cells in capturing the behaviour of these models.

By incorporating additional physics through the use of physical extensions, the material cells have the potential to enhance their accuracy and generalization capabilities.

Subsequently, we conducted a comparison between recurrent training and one-to-one training approaches. Recurrent training involves training the material cells using sequential data, where the output of one step is fed back as input to the next step. This approach takes into account the temporal dependencies and can capture path dependency. On the other hand, one-to-one training treats each step independently and does not consider the sequential nature of the data. By comparing the pros and cons of recurrent training and one-to-one training, we can gain insights into the strengths and limitations of each approach. Our goal is to understand the trade-offs between these two training approaches and determine which one is more suitable for capturing elastoplastic responses.

### 6.4.1 Adaptive step size adjustment

Developing a robust and universal network-based constitutive model for use in a BVP simulation is not a straightforward task compared to training a neural network alone. One significant challenge is the discrepancy in scales of strain increments. The strain increments used during training are typically much larger, sometimes even several orders of magnitude

larger, than those encountered in the incremental analysis of a BVP. As shown in 6.7, the median Frobenius norm for the strain increment is  $1.3\text{e-}3$  in the training; in the FE simulation, this value has a median of  $4.2\text{e-}06$ . There is a significant difference in magnitude.

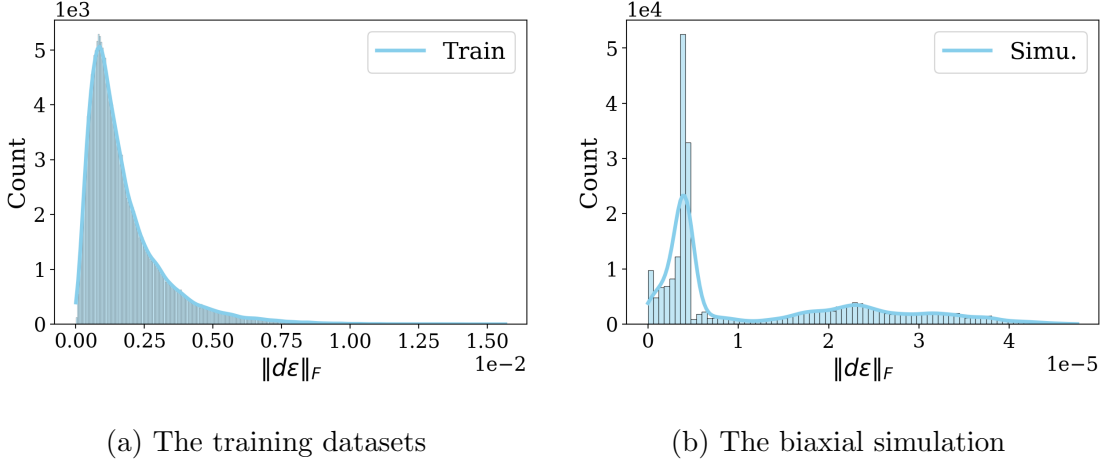


Figure 6.7: Distribution of the norm of the strain increment

A straightforward adaptive linear transformation, inspired by [103], is utilised to mitigate this problem, which can be presented as follows:

$$\begin{cases} s = \frac{\tau}{\|d\epsilon\|} \\ \tau = \text{Median}(\|d\epsilon^{(1)}\|, \dots, \|d\epsilon^{(N)}\|), d\epsilon^{(n)} \in \mathcal{D}^{(\text{train})} \\ \mathcal{I} = \frac{1}{s} (\mathcal{M}_{NN}(s d\epsilon, \mathcal{I}_0) - \mathcal{I}_0) + \mathcal{I}_0 \end{cases} \quad (6.17)$$

where  $\tau$  represents the median of the Frobenius norms of training strain increments of the training dataset. The scalar  $s$  is the linear adaptive scalar used for scaling the input and output variables. We set the threshold to the median of the Frobenius norm of the training strain increment. As a result, the input can be adaptively adjust to the size of the training median, which is the size with the highest prediction accuracy. For numerical stability, the above method directly returns the current state  $\mathcal{I}_0$  when the strain increment  $\|d\epsilon\|$  is very small and  $s$  is infinity. Similar with the self-consistent approach ([161]), the above method is integrated in the material cell architecture and involved in the offline training phase which is intended to mitigate the step size dependence.

It is important to note that in the tests presented in Fig. 6.6, 6.8, and 6.11, the test strain paths are generated using the same random GP approach employed for generating the loading paths of the training samples. As a result, the magnitude of the strain increments



( $d\epsilon$ ) in the test sets is similar to that in the training set. Hence, there is no need to adjust the size of the input strain increment. However, adaptive adjustment becomes necessary when the trained model is used in FE simulations, where the strain increments may differ in magnitude.

### 6.4.2 Utilisation of the symmetry

To address the curse of high dimensions without relying on co-axiality assumptions, we leverage the symmetry of the model instead of the co-axiality assumption. The co-axiality assumption states that the principal axis of the strain tensor aligns parallel to the principal axis of the stress tensor which is not strictly satisfied for the plasticity. However, by employing the symmetry based on the isotropic, we can enhance the model's generalisation while still satisfying the essential physics. The isotropy-based symmetry can be presented as:

$$(\sigma_{11}^{(t)}, \sigma_{12}^{(t)}, \sigma_{22}^{(t)}) = 0.5 \left( \begin{array}{l} \mathcal{M}_{NN} \left( d\epsilon_{11}, d\epsilon_{12}, d\epsilon_{22}, \sigma_{11}^{(t-1)}, \sigma_{12}^{(t-1)}, \sigma_{22}^{(t-1)} \right) + \\ \mathcal{M}_{NN} \left( d\epsilon_{22}, d\epsilon_{12}, d\epsilon_{11}, \sigma_{22}^{(t-1)}, \sigma_{12}^{(t-1)}, \sigma_{11}^{(t-1)} \right) [2, 1, 0] \end{array} \right) \quad (6.18)$$

In the expression  $(x, y, z)[2, 1, 0] = (z, y, x)$ , the index  $[2, 1, 0]$  is used to reorder the vector  $(x, y, z)$ , resulting in the new order  $(z, y, x)$ . This notation indicates a rearrangement of the vector elements based on the specified index values. It is worth noting that while the inclusion of prior knowledge has potential benefits, it is important to carefully design and validate its application in the specific context of the problem at hand.

In this work, the material cell is constructed under the isotropic assumption. Anisotropy is a very interesting topic. Data-driven methods generally normalise the training data and do not need to consider the gauge and size of the data when establishing relationships between them. Without employing the symmetry, the material cell would be able to be used directly on anisotropic materials.

### 6.4.3 Techniques for physical extensions

To enhance the interpretability and generalisability of network models, researchers have increasingly introduced physics terms as penalties [156] or coerced them to satisfy physics constraints [162, 163]. In training data-based constitutive models, physics has been incorporated for dimension reduction [109, 112, 142, 149, 158], yield surface reconstruction [118, 157], and plastic return mapping [158], or predicting the elastoplastic tangent matrix [164] etc.

For the  $J_2$  model with ideal plasticity (Appendix 6.6.2), the constitutive computation can be encapsulated as:

$$\sigma_{ij} = \mathcal{M}_{J_2Ideal}(d\epsilon_{ij}, \sigma_{ij}^{(0)}) \quad (6.19)$$

where the current stress  $\sigma_{ij}^{(0)}$  and the strain  $d\epsilon_{ij}$  increment are enough for stress updating.

Similar to Eq. (6.8), the network-based material cell can then be specified as:

$$\sigma_{ij} = \mathcal{M}_{NN}(d\epsilon_{ij}, \sigma_{ij}^{(0)}) \quad (6.20)$$

where the hidden state  $h$  in this case is tailored to the stress tensor  $\sigma_{ij}$  and the  $\mathcal{M}_{NN}$  is simplified to a single GRU layer. The calculations of the three gates in GRU can be written as:

$$\begin{cases} r = \sigma_g(L_{er}(d\epsilon) + L_{\mathcal{I}r}(\mathcal{I}_0)) \\ z = \sigma_g(L_{ez}(d\epsilon) + L_{\mathcal{I}z}(\mathcal{I}_0)) \\ n = \tanh(L_{en}(d\epsilon) + r \odot L_{\mathcal{I}n}(\mathcal{I}_0)) \\ \mathcal{I} = (1 - z) \odot n + z \odot \mathcal{I}_0 \end{cases} \quad (6.21)$$

where  $L$  indicates the FC layer  $L(x) = wx + b$  and the subscript denotes the input and the output. There are six FC layers in this material cell. Because the length of the hidden state equals the length of  $\sigma$  in the Voigt notion, the total number of trainable parameters for this material cell is  $3 \times 3 + 3 = 12$ .

The weights and biases are then optimised on the datasets generated via the  $J_2$  model. Afterwards, the trained model is evaluated on the test sets. As shown in Fig. 6.8, the average and worst prediction errors are  $4.9e - 2$  and  $6.2e - 1$ , respectively, which are much poorer than the predictions in Fig. 6.6. Despite using more datasets, the current model fails to produce satisfactory predictions. This is partly attributed to the complexity of calculating the shear stress  $q$ , as outlined in Appendix 6.6.2, which involves squaring and square rooting. Based on our testing results, it appears that the combination of the linear operator  $y = wx + b$  with the Sigmoid and Tanh activation functions is insufficient in capturing the nonlinearity inherent in the  $J_2$  model.

One possible approach to enhance the capacity of the network while maintaining the advantages of the GRU structure is to introduce additional basis functions  $\Phi(x) = \phi_1(x), \phi_2(x), \dots, \phi_n(x)$  to both the input and hidden state of the GRU cell. These basis functions can be designed to capture specific nonlinear relationships. For example, one basis function  $\phi_i$  could be defined as  $\phi = \hat{q}(\sigma_{ij})$ , where  $\hat{q}$  represents shear stress. By incorporating these additional basis functions, the GRU cell can capture more complex nonlinear

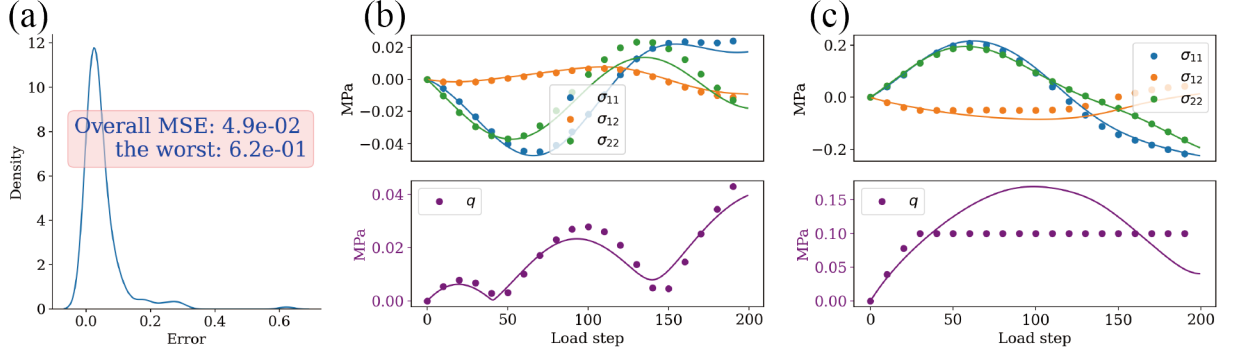


Figure 6.8: Results of the material cell without physical extensions trained on datasets generated with  $J_2$  model: only  $\sigma$  as the internal  $\mathcal{I}$  (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions.

patterns while keeping advantages in long-sequence training. It is important to carefully design and select these basis functions based on the specific problem and desired relationships to be captured. In essence, the physics extension is a kind of "feature engineering" based on our prior knowledge of the datasets.

The approach of incorporating additional basis functions to capture physical relationships is referred to as the physical extensions, as illustrated in Fig. 6.9. Based on the research of constitutive modelling, the material yielding is typically related to the shear stress, average stress, etc., and the plastic deformation is related to the direction of the current stress tensor and the direction of the strain increment. Therefore, these variables are fed into the model together and the model is expected to combine the appropriate physical quantities by itself.

In this case, the hidden state is expanded from  $\sigma_{ij}$  to include  $\{\sigma_{ij}, q, \theta_\sigma, \sigma_1^{PR}, \sigma_2^{PR}\}$ , where  $\sigma_{ij}$  represent the components of the stress tensor,  $q$  represents the shear stress,  $\theta_\sigma$  represents the rotation angle, and  $\sigma_1^{PR}$  and  $\sigma_2^{PR}$  represent the first and second principal stresses, respectively. Additionally, the input is expanded to include  $\{d\epsilon_{ij}, \theta_{de}, d\epsilon_1^{PR}, d\epsilon_2^{PR}\}$ , where  $\theta_{de}$  represents the Lode angle of the strain increment, and  $d\epsilon_1^{PR}$  and  $d\epsilon_2^{PR}$  represent the increments of the first and second principal strains, respectively.

It is worth noting that in Fig. 6.10, the model is trained on two different datasets, one consisting of 200 data series and the other consisting of 1000 data series. Interestingly, both training sets yield almost identical validation errors, with values of  $2.09e-3$  and  $2.04e-3$ , respectively. Compared to Fig. 6.5b, this observation suggests that incorporating physics

extensions into the material cell enables the model's performance to become less dependent on the quantity of the training data, which is a favourable characteristic for practical applications. And the physical extensions help mitigate the oscillations in the loss curve during training. In Fig. 6.5b, the training involves a considerable number of parameters, making the model highly capable but prone to getting stuck in local optima, resulting in significant fluctuations in the error curve. Conversely, Fig. 6.10 corresponds to a material cell with fewer parameters, and the physical extensions help to discern the optimal direction, resulting in reduced oscillations.

We have to admit the network is powerful enough to excavate knowledge in various scenarios. But, at the context of constitutive modelling, humans understanding based on the accumulated studies of the solid mechanics are worth to be fed to the network or even to coerce the network. Optimisation along the clues will be more effective and more likely to get the optimum than a blind one. Here the physical extensions are working as hints for the material cell training and the symmetry (in Sec. 6.4.2) is the rigorous condition that must be satisfy.

Fig. 6.11 shows the test results of the material cell with physics extensions. It can be observed that the prediction accuracy is improved compared to Fig. 6.8. The worst prediction accuracy achieved by the extended material cell decrease to  $1.9e-1$  from  $6.2e-1$ . In FEM simulations, the stress response of a large number of integration points at each loading step is predicted by the material cell model. The worst prediction contributes the most to the error accumulation. When embedded in the FEM, the one with physical extensions will result in much higher accuracy. Another advantage of the material cell with physics extensions is that the extended hidden state now has clear physical interpretations, allowing for better understanding and interpretation. With the physical interpretations, it is more comfortable to involve the hidden state and output in the adaptive step size adjustment mentioned in Sec. 6.4.1.

#### 6.4.4 Pressure independent material behaviour: $J_2$ model

After achieving satisfactory accuracy on the training data sets generated using the  $J_2$  yielding criterion with ideal plasticity, the trained material cell with physics extensions is employed in a biaxial compression FE analysis for further validation.

The mesh and boundary conditions of the simulation are illustrated in Fig. 6.12. The simulation results of the biaxial compression problem are depicted in Figure 6.13. Overall,

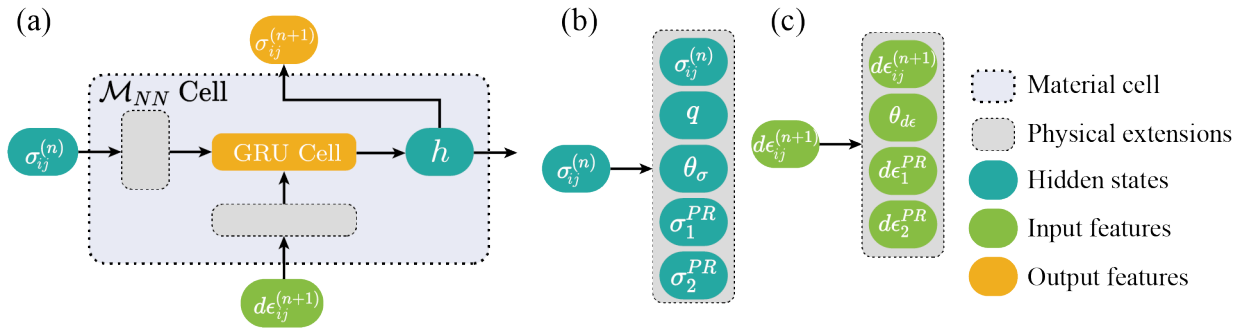


Figure 6.9: The material cell with physics extensions.

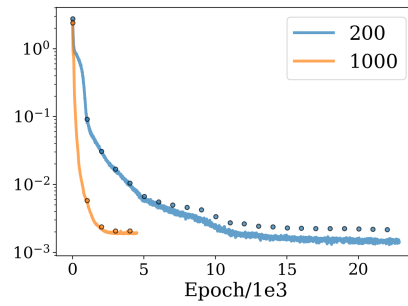


Figure 6.10: Training loss evaluation: material cell with physics extension trained on datasets generated via  $J_2$  model.

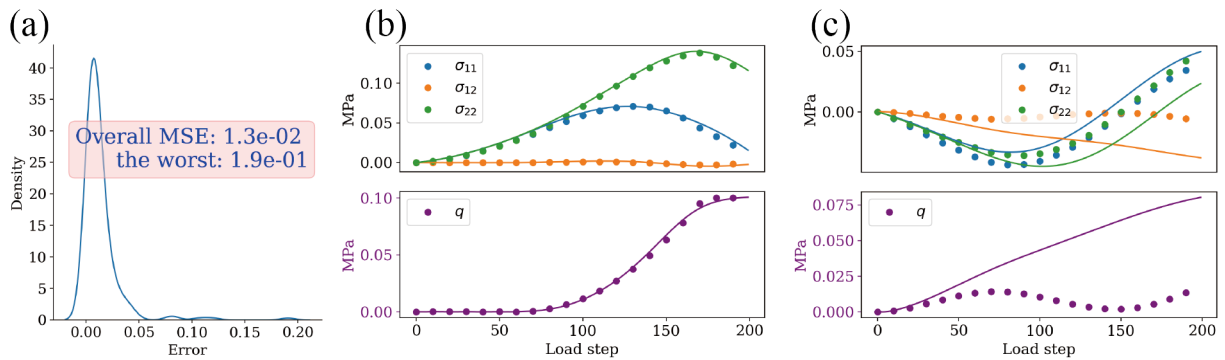


Figure 6.11: Predictions of material cell with physical extensions under  $J_2$  model. (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions.

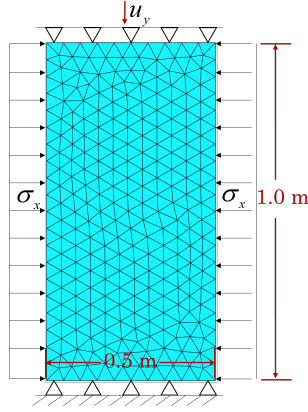


Figure 6.12: The meshes and boundary conditions of the biaxial compression simulation

the material cell, with its physics extensions, effectively captures the characteristics of the  $J_2$  model in biaxial simulations, encompassing both the elastic and plastic phases. In this simulation, various scaling factors ( $s = 1, 50, 60$ ) according to Eq. 6.17 were employed. Initially, without the introduction of the linear scalar (with  $s = 1$ ), the material cell failed to accurately reproduce the stress in explicit FE simulations, despite being well-trained (as seen in Fig. 6.11). As the scaling factor increased, the top pressure and overall volume approached the ground truth values. Furthermore, the maximum nodal acceleration gradually decreased, indicating improved stability of the predicted stresses. However, when the scaling factor is increased to  $s = 60$ , the simulation became unstable after reaching an axial strain loading of 0.07. The adaptive scalar outperformed the constant scalar, particularly in terms of the global strain curves.

#### 6.4.5 Pressure dependent material behaviour: Drucker-Prager model

In this section, we assess the effectiveness of the proposed material cell in capturing pressure-dependent material behaviour. To this end, we generate datasets using the ideal plasticity model that incorporates the Drucker-Prager yield criterion (refer to Appendix 6.6.2).

Upon completing the training phase, we plot the corresponding training loss evolution in Fig. 6.14 and test results in Fig. 6.15, respectively, to compare the performance of the network with and without including the mean pressure term  $p$  in the internal state  $\mathcal{I}$ . The inclusion of mean stress reduces the overall mean-square error (MSE) from  $1.8e-1$  to  $1.2e-1$ . More noteworthy is the reduction of the worst prediction error from 3.1 to 1.2. Since in explicit FEM simulations, each step of the prediction error accumulates along with the time integration steps. Therefore, the inclusion of  $p$  will significantly improve the FEM

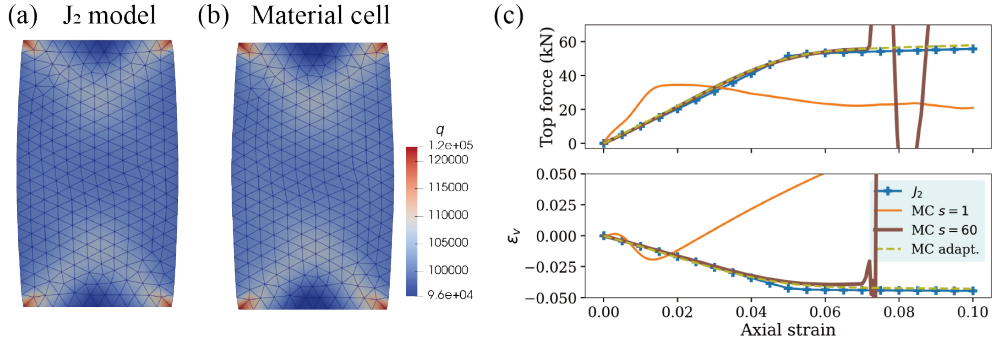


Figure 6.13: Biaxial monotonic compression:  $J_2$  model. (a) and (b) are the equivalent von-Mises stress of  $J_2$  model and material cell (with adaptive step size adjustment) simulations, respectively. (c) The curves of top force and the global volumetric strain for different cases, involving the ground truth  $J_2$  model, the material cell (Fig. 6.9) with  $s = 1$ ,  $s = 60$ , and adaptive step size adjustment. MC in the legend denotes material cell. The top force compresses the specimen in a downward direction. A volumetric strain less than zero indicates that the volume is undergoing compression.

simulation accuracy. While  $p$  can be easily calculated, providing this information to the material cell further enhances its performance. This finding emphasises the significance of incorporating physical insights to expand the input and hidden state via physics extensions.

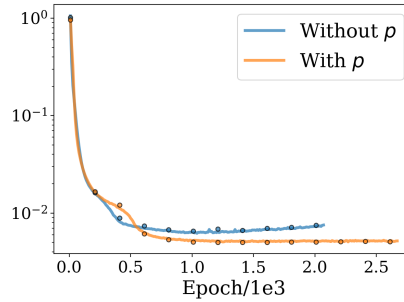


Figure 6.14: Training loss of material cell without/with  $p$  under Drucker-Prager model

Furthermore, We conducted a comprehensive evaluation of the trained material cell in FE analyses. The biaxial compression is focused and simulation results are compared with those using the classical Drucker-Prager model. The resulting curves for the top force, global volume strain, and maximum node acceleration are presented in Fig. 6.16, illustrating favourable agreement between the predictions of the two models.

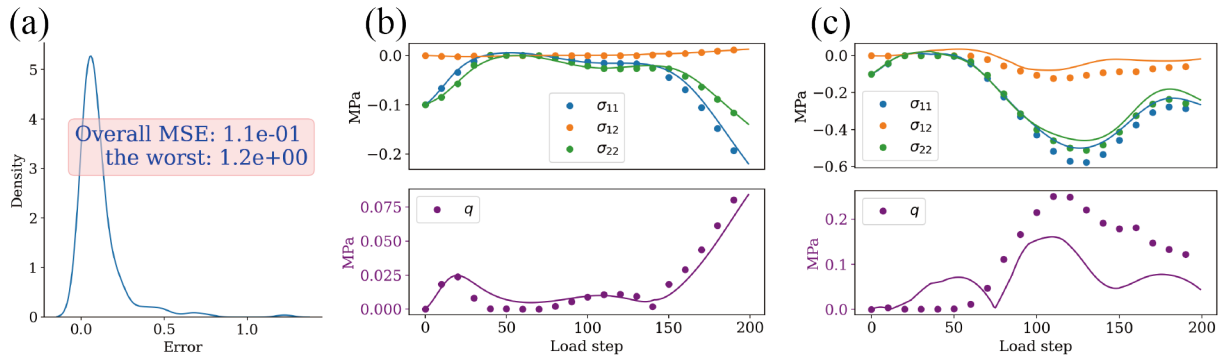


Figure 6.15: Predictions of material cell after training with  $p$  included. (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions.

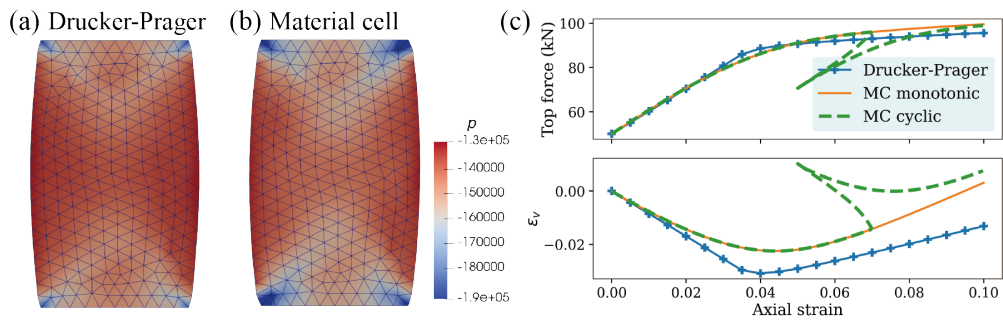


Figure 6.16: Biaxial simulation: Drucker-Prager model and the material cell shown in Fig. 6.9. (a) and (b) are the mean stress results of Drucker-Prager model and material cell (with adaptive step size adjustment) simulations, respectively. (c) The curves of top force and the global volumetric strain for different cases. MC in the legend denotes material cell.



### 6.4.6 Plastic hardening behaviour: $J_2$ -harden model

In this section, our focus is on evaluating the ability of the material cell to reproduce elasto-plastic behaviour with plastic hardening. Specifically, we consider a model that incorporates the  $J_2$  yield criterion and hardening plasticity, as described in Section 6.6.2.

Compared to the previous two constitutive models, the model in this section introduces a hardening function  $H(|\epsilon_{ij}^p|)$  into the yield function. Here,  $\epsilon_{ij}^p$  represents the plastic strain tensor. This means that the stress update depends not only on the stress tensor but also on the plastic strain tensor.

On the basis of the material cell in Fig. 6.9, the plastic strain  $\epsilon_{ij}^p$  is further included in the hidden state  $\mathcal{I}$ , and the norm of the plastic strain tensor is also included as physics extensions. However, the material cell, with only one GRU layer, encounters difficulties in capturing the non-linearity of the  $J_2$ -harden model. The material cell with physical extensions is demonstrated able to reproduce the stress in ideal plasticity. However, it performs poorly with the addition of plastic hardening. A possible reason is that the plastic strain increment should be zero in the unyielding case, but the material cell cannot give a prediction that is strictly zero. Without accurate prediction of the plastic strain at the beginning with unyielding stress, the model cannot effectively perform the iterative stress update process described in the plastic return mapping algorithm (Appendix 6.6.2), resulting in poor predictions shown in Fig. 6.17.

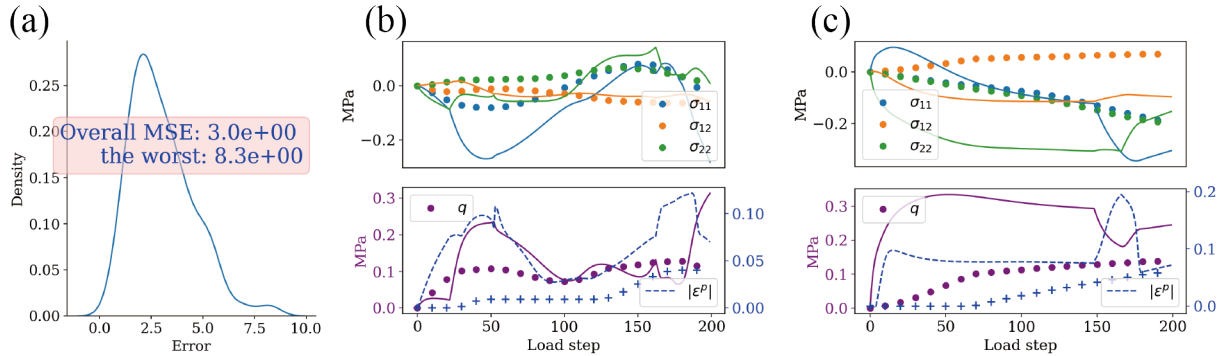


Figure 6.17: The material cell with a single layer of GRU:  $J_2$ -harden model (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions.

## Implicit internal variable-based material cell: sequential training

In an attempt to incorporate the influence of loading history on mechanical states, the material cell initially received physical parameters such as the plastic strain tensor  $\epsilon_{ij}^p$ . However, this approach did not yield satisfactory results in accurately updating the internal variables, which in turn affected the prediction of stress. One possible reason is that the plastic strain increment should be zero in the unyielding case, but the material cell cannot give a prediction that is strictly zero.

In this section, a data-driven internal variable  $\mathbf{h}$  is introduced as a replacement for  $\epsilon_{ij}^p$ . With this treatment, material cell training no longer requires explicit calibration of the plastic deformation which is also harder to obtain in experiments. The network structure employed for this approach is depicted in Fig. 6.18.

Compared with the purely data-driven material cell in Sec. 6.3, physical properties are artificially assigned to the internal variables for the utilisation of symmetry and the stress tensor is fed to the physics extensions before participating in the tensor computation within the material cell. According to Sec. 6.4.3, this is effective in improving generalisation and accuracy. The  $\mathbf{h}$  has a length of 30 and is divided into three data-driven internal variables, each of length 10, corresponding to three components of the 2D tensor. The data-driven internal variables can be denoted as  $\mathbf{h} = (\mathbf{h}^{(11)}, \mathbf{h}^{(12)}, \mathbf{h}^{(22)})$ . Subsequently, referring to the symmetry in Sec. 6.4.2, the internal variables can be swapped and averaged as follows:

$$\mathbf{h} = (\mathbf{h}^{(11)}, \mathbf{h}^{(12)}, \mathbf{h}^{(22)}) = \frac{1}{2} \left( \begin{array}{l} \mathcal{M}_{NN} \left( d\epsilon_{11}, d\epsilon_{12}, d\epsilon_{22}, \sigma_{11}, \sigma_{12}, \sigma_{22}, \mathbf{h}_0^{(11)}, \mathbf{h}_0^{(12)}, \mathbf{h}_0^{(22)} \right) + \\ \mathcal{M}_{NN} \left( d\epsilon_{22}, d\epsilon_{12}, d\epsilon_{11}, \sigma_{22}, \sigma_{12}, \sigma_{11}, \mathbf{h}_0^{(22)}, \mathbf{h}_0^{(12)}, \mathbf{h}_0^{(11)} \right) [2, 1, 0] \end{array} \right) \quad (6.22)$$

where the index manipulation of  $[2, 1, 0]$  is same with Eq. 6.18.

Under the assumption that  $\mathbf{h}_0^{(11)}$ ,  $\mathbf{h}_0^{(12)}$ , and  $\mathbf{h}_0^{(22)}$  represent stress and plastic strain-related internal variables in each of the three directions, the data-driven internal variable  $\mathbf{h}_0$  can be approximated linearly using the adaptive step method (Eq. 6.17). Except for predictions when calling the material cell, this adaptive scaling is also introduced into the training process. The forward process is shown in Alg. 5. In this way, the effect of the step size on the prediction is eliminated during the training.

After training, the model's predictions on the test set are shown in Fig. 6.19. Unlike the explicit plastic internal variable  $\epsilon_{ij}^p$ , the implicit internal variables enable the prediction of

---

**Algorithm 5** Forward process of the material cell shown in Fig. 6.18

---

**Require:** Strain increment  $d\epsilon_{ij}$ , data-driven hidden state at last step  $\mathbf{h}_0$ , current stress tensor  $\sigma_{ij}$

- 1: Compute the scalar  $s$  according to Eq. 6.17
  - 2: Normalise the strain increment  $d\epsilon'_{ij} = s d\epsilon_{ij}$
  - 3: Update the normalised hidden state  $\mathbf{h}'$  according to Eq. 6.22, but the strain increment is replaced by  $d\epsilon'_{ij}$
  - 4: Rescale the hidden state via  $\mathbf{h} = \frac{\mathbf{h}' - \mathbf{h}_0}{s} + \mathbf{h}_0$
  - 5: Predict the stress tensor  $\sigma_{ij}$  based on the updated hidden state  $\mathbf{h}$ .
  - 6: **return**  $\sigma_{ij}$
- 

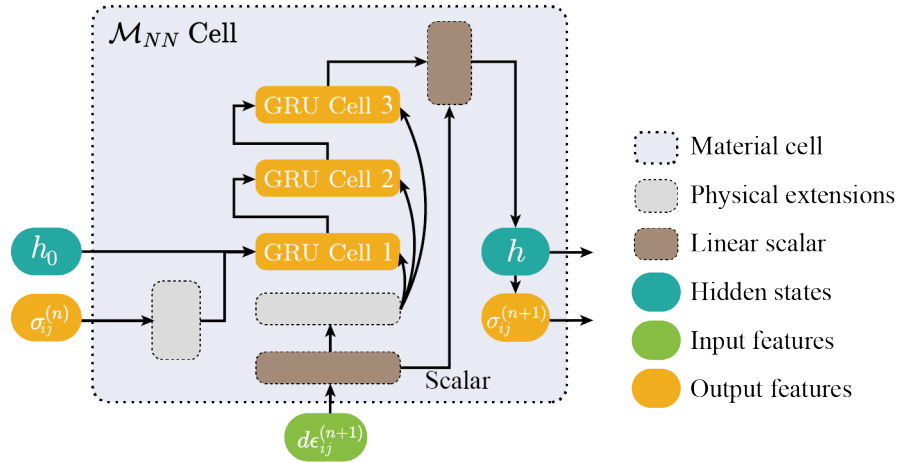


Figure 6.18: Material cell with implicit internal variables.

stress response without relying on precise plastic internal variable values. The optimiser automatically adjusts the output based on the difference between the predicted stress sequence  $\{\hat{\sigma}_{ij}^{(t)}\}_t^N$  and the training dataset  $\{\sigma_{ij}^{(t)}\}_t^N$ . During this process, the implicit internal variable is also optimised to represent the physical internal state.

The hidden state  $\mathbf{h}$  contains information on plastic deformation, but plastic deformation is not explicitly included within the material cell. So we no longer have access to the plastic strain tensor and therefore cannot check the yield and consistency conditions. This is a problem with the data-driven approach. Moving away from the elastic-plastic framework altogether and using data-coded loaded histories can reproduce the physics since the beginning of history, but the interpretability decreases.

On the flip side, because of circumventing the requirement of explicitly inputting internal variables during the training process, the material cell with an implicit hidden state can be trained on experimental datasets, such as conventional compression tests, where the plastic internal variables are not straightforwardly accessible. Data-driven models are expected to leverage this capability to learn plasticity-harden constitutive relationships from experimental data.

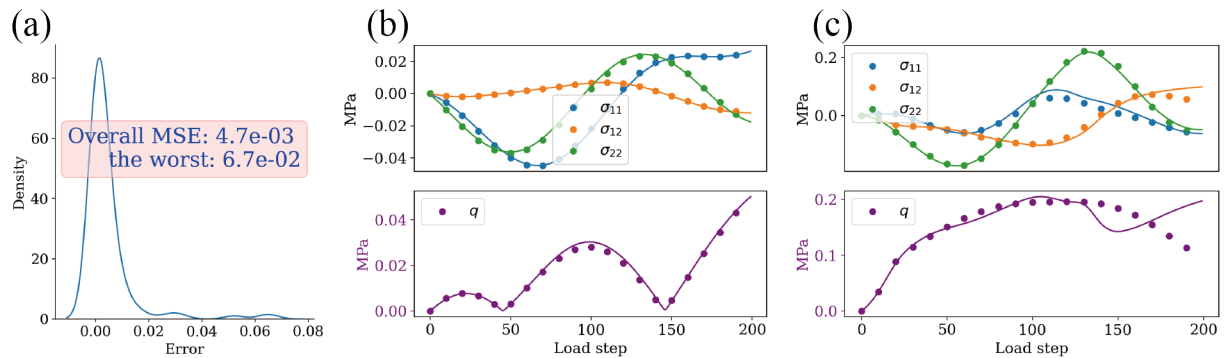


Figure 6.19: The material cell with implicit internal variables:  $J_2$ -harden (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions.

Figure 6.20 presents the results of the biaxial simulation after embedding the material cell in the FEM. The simulation exhibits good agreement with the true results during monotonic loading. However, when subjected to multiple cyclic loading, the results are somewhat less satisfactory, although they can capture fundamental properties such as distinguishing between elasticity and plasticity, and plastic harden. As the loading progresses, errors in stress and volume strain accumulate, which poses a significant challenge for the cyclic prediction

of structures.

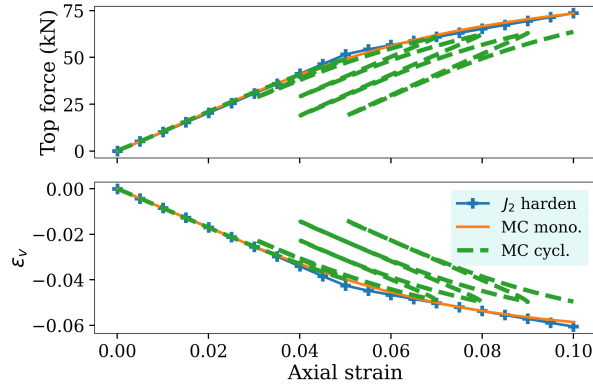


Figure 6.20: Results of biaxial calculations in FEM with the material cell based on the implicit internal variable (Fig. 6.18). MC in the legend denotes material cell.

Fig. 6.21 and 6.22 illustrate the outcomes of the two stretching simulations conducted using the trained material cell. Favourable computational results are obtained in both simulations, highlighting the versatility and accuracy of the material cell as well as demonstrating its capability in various loading simulations.

Expectations are held for the improved computational efficiency achieved by the neural network-based constitutive model. This material cell, as illustrated in Figure 6.19, has also been applied to the  $J_2$  model and Drucker-Prager model with ideal plasticity. To illustrate the improvements in computational efficiency, Fig. 6.23 presents a comparison in biaxial simulations. Notably, as the complexity of the constitutive model increases, the computational time rises for the three models. However, with the recurrent network-based material cell, a significant reduction in computational time is observed. Importantly, the computational time remains consistent across all different constitutive relationships due to the same network structure.

### Deep network-based material cell: one-to-one training

An alternative method, in addition to sequence training, employs a deep network in one-to-one training to build a material cell to reproduce the nonlinear properties of the  $J_2$ -harden model. The deep network is highly effective at nonlinear mapping due to its deep structure and nonlinear activation function. A similar methodology is employed in [164], where a deep network is used to model the plastic return mapping by taking plastic work and stress tensor as input and producing a hardening vector as output.

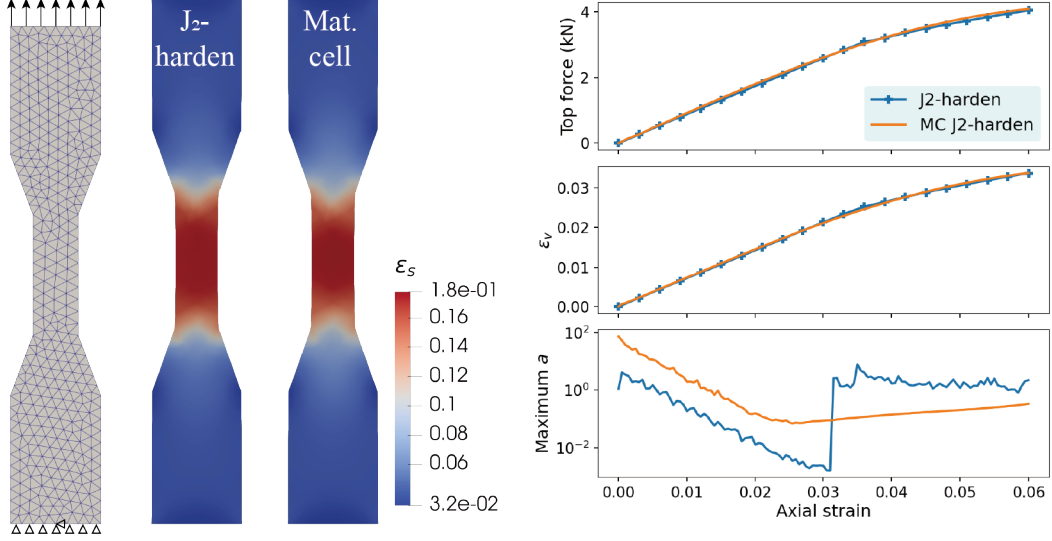


Figure 6.21: Dog bone stretching simulation:  $J_2$ -harden model with the material cell shown in Fig. 6.18. MC in the legend denotes material cell. The top force pulls the dog bone upward. In the second subplot, the global volumetric strain  $\epsilon_v$  is summarised over the simulated domain. In the subplot of the third row,  $a$  denotes the acceleration of nodes in the FEM simulation.

In this section, the deep network-based material cell is employed to predict stress by providing the strain increment and internal variables as inputs. As shown in Fig. 6.24, the structure remains similar to the one used for the GRU cell. However, the GRU cell is replaced by the deep neural network (DNN). Note the internal variable  $\mathbf{h}_0$  should be the plastic strain tensor and explicitly fed to the material cell for predicting the stress response  $\boldsymbol{\sigma}_1$  and update the hidden state to  $\mathbf{h}_1$ . This change prevents us from training the model using stress-strain sequences iteratively because the gradient vanishing or explosion can no longer be mitigated by DNN. The DNN consists of 10 hidden layers, each with 20 hidden neurons, yielding a total of 9,908 trainable parameters for the weights and biases.

After training, performances of the material cell on the test sets are depicted in Fig. 6.25. The prediction accuracy is significantly improved compared to the single GRU-based model. The overall MSE is  $7e-3$ , with the poorest prediction having an MSE of  $1.1e-1$ . The error distribution shown in Fig. 6.25a reveals that the majority of errors fall within the interval of less than  $2e-2$ .

By recognizing the advantages of one-to-one training, we utilised this approach to training a material cell using data from both the  $J_2$  and Drucker-Prager models for biaxial simula-

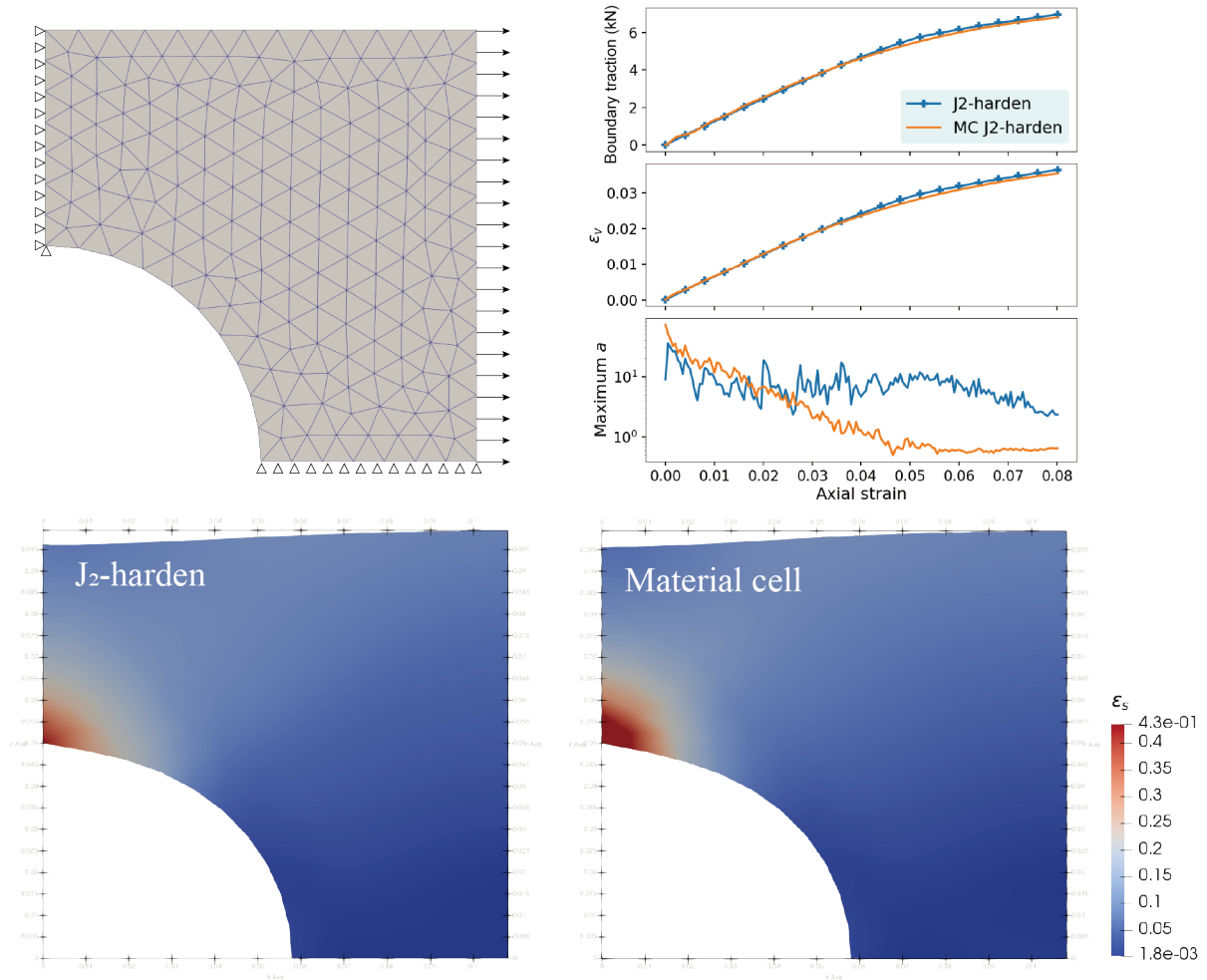


Figure 6.22: Stretching simulation with the quarter perforated plate:  $J_2$ -harden model with the material cell shown in Fig. 6.18. MC in the legend denotes material cell. Tension is considered positive, while compression is considered negative.

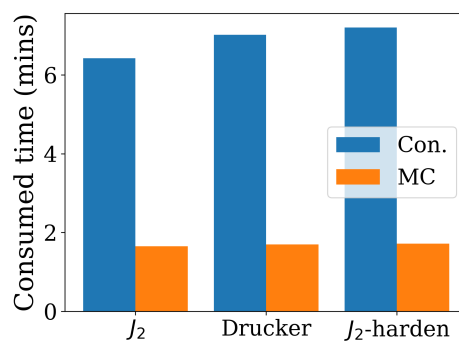


Figure 6.23: Comparison of the computational time required for biaxial simulations using the material cell and three conventional constitutive models

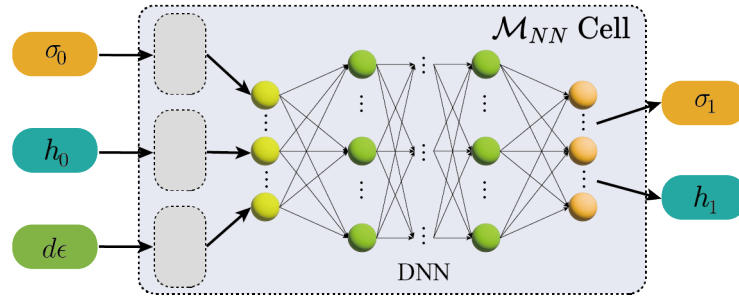


Figure 6.24: The DNN-based material cell

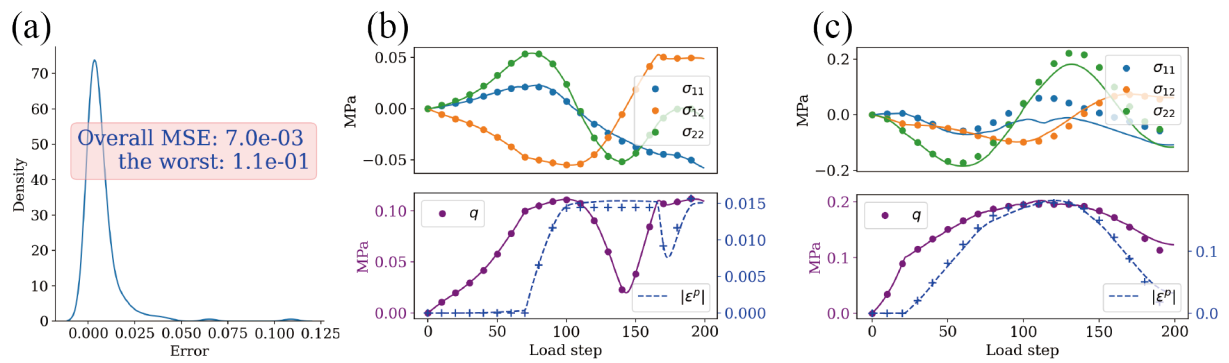


Figure 6.25: DNN-based material cell:  $J_2$ -harden model (a) Prediction error distribution (b) Best prediction (c) Worst prediction. The dots indicate the training data and the lines are the predictions.



tions. The DNN-based material cell performed well, enabling us to conduct multiple cyclic simulations, as depicted in Figure 6.26. Our findings indicate that the DNN-based material cell effectively reproduces the behaviour of the  $J_2$  and Drucker-Prager models, particularly under monotonic loading conditions where the predictions closely match the ground truth. Under cyclic loading paths, the model still struggles to fully capture the non-linearity at the inflexion points of the stress-strain curves, which may be attributed to the linear transformation used.

In the context of the  $J_2$ -harden model, while our predictions for unloading in the elastic zone are excellent (refer to Figure 6.26c), we were unable to fully match the ground truth for unloading in the plastic stage. These results highlight the challenge of accurately predicting the internal variables of a material, especially when a large number of recurrent calculations are involved. Due to the inherent recurrent properties, predictions from one step are used as inputs for the next step, leading to error accumulation over time.

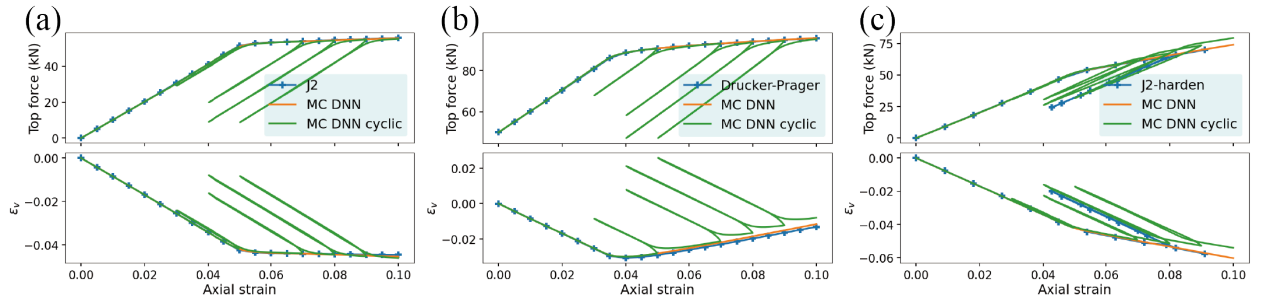


Figure 6.26: Results (top force and global volumetric strain) of biaxial simulations with the DNN-based material cell (Fig. 6.24) in multiple cyclic loading. (a):  $J_2$  yield surface with ideal plasticity; (b) Drucker-Prager yield surface with ideal plasticity and (c)  $J_2$  yield surface with hardening plasticity. MC in the legend denotes material cell.

## 6.5 Concluding remarks

In this study, we investigate the effectiveness of material cells with physical extensions and symmetry conditions in modelling three constitutive models of varying complexity. The main conclusions can be summarised as follows:

- Incorporating knowledge based on accumulated constitutive studies via physical extensions and symmetry constraint enables performance of the material cell to become less dependent on the quantity of the training data.

- In FE analysis, the size of strain increments can vary significantly, spanning several orders of magnitude (Fig. 6.7). However, RNNs in material cells are highly sensitive to step sizes. To address this issue, we employ an adaptive linear transformation approach (Eq. 6.17), which is effective to alleviate the error resulting from the magnitudes gaps between the strain increments in training sets and FEM simulations.
- Sequential training enables data-driven models to bypass the calibration of internal variables during reproducing path-dependency compared to one-to-one training. This opens up the possibility of directly optimising the data-driven elastoplastic model with experimental data, particularly considering that internal variables related to plasticity are often inaccessible in experimental settings.
- For the ideal elastic-plastic model, the single GRU-based material cell performs well. When plastic hardening is introduced and plastic strain becomes an additional internal variable, the single GRU-based material cell faces challenges in accurately updating the plastic strain. The material cell shown in Fig. 6.18 integrates multi-layer GRUs, physical extensions and adaptive step size adjustment, taking into account both sequential training and DNN advantages.

Meanwhile, the discussion mainly consists of:

- Traditional constitutive models, such as elastoplastic models, tend to be hard to be understand and implement by engineers. As a result, the more advanced the model, the more difficult to be widely used. Data-driven methods, which require only data and training, are potential to fully reproduce the constitutive response contained in the data. This simple, convenient, and accurate method will revolutionise engineering computation.
- The data-driven constitutive models have two emphases: (1) physics (2) data. The former is conducive to improving generalisation and resisting the influence of noises and outliers. The latter is conducive to improving the accuracy and truly reproducing the constitutive responses contained in the data. Traditional constitutive models are constructed entirely on the basis of physics and phenomenological assumptions. The best models need to find a balance between physical priors and data-driven.
- During training, sequences of 200 in length were used. However, in FE simulations, we computed load steps for more than 4,000, where the material cell performed well for such extensive loading. Yet, cyclic loading revealed noticeable error accumulation,

which calls for further exploration and resolution.

- In this work, as shown in Fig. 6.7, the strain increment in the explicit FE case is much smaller, whereas using the implicit FE solver, the strain increment is larger. In this case, the strain may be split into multiple small increments and the material cell is called multiple times to integrate the constitutive response. But feasibility needs to be tested in future work.
- Uncertainty quantification should be crucial for the black-box to be used in practical engineering computations. The level of confidence can be quantified only when the results of uncertainty analyses are provided alongside the mechanical analysis results.

## 6.6 Appendix

### 6.6.1 Regression via Gaussian Process

From the perspective of the weight, GP starts with the linear regression:

$$f(x) = x^T w \quad (6.23)$$

where the weight  $w$  is assumed to follow a normal distribution with  $w \sim \mathcal{N}(0, \Sigma_p)$ . To extend the regression to high dimensions, basis function  $\phi(x)$  needs to be introduced. The kernel calculation is simplified using a technique known as the kernel trick:

$$k(x, x') = \phi(x)^T \Sigma_p \phi(x') \quad (6.24)$$

where  $k$  is the kernel function of the basis  $\phi$ . Given a dataset of  $\mathcal{D} = \{(x, y)^{(i)}\}_{i=1}^N$  where  $N$  is the total number of data points,  $x$  represents the input and  $y$  represents the output. The Gaussian processes regression is extensively described in [165]. The mean and covariance can be evaluated at the test input points  $x_*$  as:

$$\begin{cases} \mathbf{u}_* = k(x_*, X^T) (k(X, X^T) + \sigma_n^2 I)^{-1} y, \\ \Sigma_* = k(x_*, x'_*) - k(x_*, X^T) (k(X, X^T) + \sigma_n^2 I)^{-1} k(X, x'_*). \end{cases} \quad (6.25)$$

where  $\sigma_n^2$  is the magnitude of the noise which is null in our random loading path generation, and  $I$  is the Kronecker operator. In this way, the generated random loading path strictly satisfies  $\mathcal{D}$ , where the data points at the beginning are involved to ensure the loading starts from 0. Apart from this, for the subsequent points, the loading follows a joint random Gaussian distribution.

## 6.6.2 Classic constitutive model

### Ideal plasticity with J2 yield criterion

The linear elastic perfectly plastic model with J2 yield criterion used in this study is described in this section. The linear elastic relationship is  $\sigma_{ij} = K\epsilon_v\delta_{ij} + 2Ge_{ij}$  where  $e_{ij} = \epsilon_{ij} - \delta_{ij}\epsilon_v/2$  is the deviatoric strain, the elastic volume modulus is  $K = 1.25\text{e}6\text{Pa}$  and the elastic shear modulus is  $G = 8.33\text{e}5\text{Pa}$ . The J2 yield surface can be expressed as

$$f = q - \sigma_y \quad (6.26)$$

where we have  $q = \sqrt{s_{ij}s_{ij}}$  with  $s_{ij} = \sigma_{ij} - p\delta_{ij}$  being the deviatoric stress tensor. The yield stress is  $\sigma_y = 1\text{e}5 \text{ Pa}$ .

If the trial value of  $q$ , denoted by  $q^{TR}$ , is greater than the yield stress  $\sigma_y$ , the stress is then updated as:

$$\begin{cases} p = p^{TR} \\ s_{ij} = s_{ij}^{TR} \cdot \frac{\sigma_y}{q^{TR}} \\ \sigma_{ij} = \delta_{ij}p + s_{ij} \end{cases} \quad (6.27)$$

Afterwards, the elastic volume strain and elastic deviatoric strain are calculated as  $\epsilon_v^e = p/K$  and  $e_{ij}^e = \frac{s_{ij}}{2G}$ , respectively.

### Ideal plasticity with Drucker-Prager yield criterion

Here we introduce an elastoplastic model with a Drucker-Prager (DP) yield surface and ideal plasticity. In contrast to the J2 yield surface, which is commonly employed to represent purely cohesive material behaviour, the DP yield surface is typically used to characterize pressure-dependent material behaviour, such as the frictional behaviour of dry sands. The yield surface is

$$f = q - Mp \quad (6.28)$$

where  $M$  is related to the friction angle  $\theta_f = 15^\circ$ . The stress ratio can be calculated as  $M = 2 \sin \theta_f$ . Non-associated flow rule is employed here to implement the plastic return calculation. The derivative of the yield function is presented as:

$$\begin{aligned} \frac{\partial f}{\partial \sigma_{ij}} &= \frac{\partial q}{\partial \sigma_{ij}} - M \frac{\partial p}{\partial \sigma_{ij}} \\ &= \frac{2(\sigma_{ij} - p\delta_{ij})}{q} - \frac{1}{2}M\delta_{ij} \end{aligned} \quad (6.29)$$

The derivative of the plastic function as  $\frac{\partial g}{\partial \sigma_{ij}} = \frac{2(\sigma_{ij} - p\delta_{ij})}{q} - \frac{1}{2}M_d\delta_{ij}$  where  $M_d = 2 \sin \theta_d$ , and  $\theta_d = 8^\circ$

In the plastic return mapping process, the consistent condition can be displayed as:

$$f_0 + df = \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} + \frac{\partial f}{\partial H} \frac{dH}{d\epsilon_{ij}^p} d\epsilon_{ij}^p \equiv 0 \quad (6.30)$$

where  $f_0$  is used to correct the plastic return mapping calculation of the former step,  $df$  is the full differentiation with regard to the increment of stress tensor and plastic strain with current derives,  $H$  is the hardening variable used to describe the evolution of the yield surface. In perfect plasticity, the hardening variable  $H \equiv 0$  and  $\frac{\partial H}{\partial \epsilon_{ij}^p} = \mathbf{0}$ . After substituting  $d\sigma_{ij} = D_{ijkl}(d\epsilon_{kl} - d\epsilon_{ij}^p)$  and  $d\epsilon_{ij}^p = d\lambda \frac{\partial g}{\partial \sigma_{kl}}$  into Eq. 6.30, the plastic multiplier can be explicitly calculated as:

$$d\lambda = \frac{\frac{\partial f}{\partial \sigma_{ij}} D_{ijkl} d\epsilon_{kl} + f_0}{\frac{\partial f}{\partial \sigma_{ij}} D_{ijkl} \frac{\partial g}{\partial \sigma_{kl}} - \frac{\partial f}{\partial H} \frac{dH}{d\epsilon_{ij}^p} \frac{\partial g}{\partial \sigma_{ij}}} \quad (6.31)$$

Then, the model plastic deformation and the stress can be updated after the calculation of  $d\lambda$ .

## J2 plasticity with linear hardening plasticity

Here we introduce the model of the J2 yield surface and isotropic exponential hardening function. The elasticity is the same as the model in Section (6.6.2). But the hardening function:

$$H = A(\|\epsilon_{ij}^p\|_2 + \epsilon_0)^B \quad (6.32)$$

Where material parameters  $A = 4e5$ ,  $B = 0.5$  and  $A\epsilon_0^B = \sigma_y$ . In the plastic return mapping,  $\frac{\partial f}{\partial H} \frac{\partial H}{\partial \epsilon_{ij}^p}$  needs to be considered. After differentiation, we have:

$$\frac{\partial H}{\partial \epsilon_{ij}^p} = AB(\|\epsilon_{ij}^p\|_2 + \epsilon_0)^{B-1} \cdot \frac{\epsilon_{ij}^p}{\|\epsilon_{ij}^p\|} \quad (6.33)$$

Substituting Eq. (6.33) into Eq. (6.31), plastic multiplier  $d\lambda$  can be solved.

### 6.6.3 Tensor transformation

#### Rotation of a tensor

In the two-dimensional case, the tensor can be rotated via the following transformation:

$$\begin{cases} t' = Q^T t Q \\ Q = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \end{cases} \quad (6.34)$$

where  $Q$  is the rotation matrix satisfying  $Q^T Q = 1$ . Then the rotated tensor can be presented as:

$$t' = \begin{bmatrix} \cos^2 \theta t_{00} + 2 \sin \theta \cos \theta t_{01} + \sin^2 \theta t_{11} & (\cos^2 \theta - \sin^2 \theta) t_{01} + \sin \theta \cos \theta (t_{11} - t_{00}) \\ (\cos^2 \theta - \sin^2 \theta) t_{01} + \sin \theta \cos \theta (t_{11} - t_{00}) & \sin^2 \theta t_{00} - 2 \sin \theta \cos \theta t_{01} + \cos^2 \theta t_{11} \end{bmatrix} \quad (6.35)$$

Thus, given the rotation angle  $\theta$ , the rotated tensor  $t'$  can be calculated as above; Meanwhile, its angle to the principal direction can be calculated by assigning the shear component of the rotated tensor  $t' = 0$ :

$$\theta = \frac{1}{2} \tan^{-1} \frac{2t_{01}}{t_{00} - t_{11}} \quad (6.36)$$

After rotating the tensor with a proper angle of  $\theta$ , the tensor  $t$  can be converted to the principal direction with the shear components equal to 0. Under the isotropic assumption, the strain and stress tensors are coordinate independent, i.e. the strain and stress tensor retains its original constitutive relationship after rotation in the reference coordinate system.

#### Spectral decomposition of a tensor

The diagonal components can also be calculated via the spectral decomposition as follows:

$$t_{ij} = \sum_A^n t_{pr}^{(A)} n_i^{(A)} n_j^{(A)} \quad (6.37)$$

where,  $t_{ij}$  is a second-ordered tensor ( $t \in \mathbb{R}^{m \times m}$ ) with  $m$  dimensions, and  $t_{pr}^{(A)}$  and  $n_i^{(A)}$  are the  $A^{\text{th}}$  eigenvalue and eigenvector, respectively. Then we have the rotation matrix as  $Q = [n^{(1)}, \dots, n^{(m)}]$ .

# Chapter 7

## A universal machine learning-based material cell

### 7.1 Introduction

After being trained on the FEM-DEM dataset, the network-based agent model can accurately replicate the macroscopic responses of granular materials and significantly accelerate the classical multiscale computation. However, due to the limited generalisation ability of the network, developing a universal network-based constitutive model for various loading paths poses a challenge. This chapter proposes a modified version of the machine learning-based constitutive model, referred to as a material cell. By combining machine learning with the principles of elasticity, yield, hardening, and plastic flow, the material cell adheres to physical laws and exhibits greater generality.

### 7.2 Components of the elastoplastic constitutive model

The elastic-plastic model consists of a yield function, a hardening function and a plastic flow rule.

## 7.2.1 Yield function

The yield function serves as a criterion to determine whether a material undergoes irreversible plastic deformation.

$$f = f(\sigma_{ij}, H) \quad (7.1)$$

where  $\sigma_{ij}$  is the stress and  $H$  is the hardening value. If the yield value of the trial stress is larger than 0, plasticity calculations are performed; otherwise, the material remains in the recoverable elastic phase. For cohesion-controlled materials (Fig 7.1a-b), the projection of the yield function in the  $\pi$  plane typically takes a circular shape, with its magnitude remaining constant regardless of the hydrostatic pressure axis. For friction materials (Fig 7.1c-f), the projection of the yield function in the  $\pi$  plane is influenced by the coefficient  $b = \frac{\sigma_2 - \sigma_3}{\sigma_1 - \sigma_3}$ . When  $b = 0$ , the material is subjected to compression, when  $b = 1.0$ , the material is under tension. The elastic zone exhibits reduced bias stress in this direction, leading to an approximately triangular projection on the  $\pi$  plane. The yield surface gradually expands along the positive direction of the hydrostatic pressure axis.

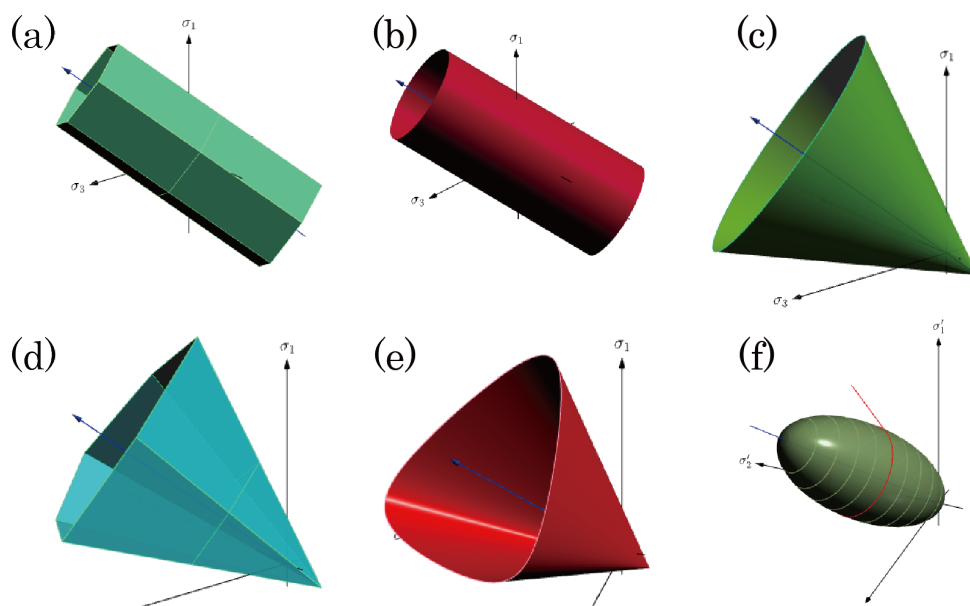


Figure 7.1: Yield surfaces in stress space. (a) Tresca yield surface; (b) von Mises yield surface (c) Drucker-Prager Yield surface; (d) Mohr-Coulomb yield surface; (e) Spatial mobilised plane; (f) modified Cam-Clay surface.



## 7.2.2 Hardening function

The hardening function describes changes in the yield surface with loading. As the material experiences plastic deformation during loading, the yield surface gradually expands. In the case of geo-materials, isotropic hardening is typically used to model material hardening. The hardening function can incorporate various quantities related to plastic deformation, such as total plastic strain  $\bar{\epsilon}^p$ , plastic volumetric strain  $\epsilon_v^p$ , and plastic shear strain  $\epsilon_s^p$  and the plastic work  $W^p = \int \sigma d\epsilon^p$  to describe the hardening process. In this chapter, the hardening value is assumed to be a non-linear function of plastic strain:

$$dH = dH(d\epsilon_{ij}^p, H^0, \mathcal{I}) \quad (7.2)$$

where  $H^0$  is the original hardening value, and  $\mathcal{I}$  is certain internal variables influencing the relationship between  $dH$  and  $d\epsilon$ . For example, the relative density indicator  $\xi$  (Eq. 7.3) in the Unified Hardening rule can be  $\mathcal{I}$ .

## 7.2.3 Plastic flow rule

A plastic flow law determines the direction of plastic deformation once it has occurred. There are two categories: the associated flow rule and the non-associated flow rule. In Associated flow, the plastic potential function and yield function always maintain the same direction as the outer normal. This can be expressed by Eq. 7.3. In Non-associated flow, the directions of the plastic potential function and yield function are not the same. This can be represented by Eq. 7.4.

$$\frac{\partial f}{\partial \sigma_{ij}} = \frac{\partial g}{\partial \sigma_{ij}} \quad (7.3)$$

$$\frac{\partial f}{\partial \sigma_{ij}} \neq \frac{\partial g}{\partial \sigma_{ij}} \quad (7.4)$$

In an ideal elastoplastic material, the hardening function is constant, i.e., the hardening function does not change with plastic strain or plastic work. Therefore, an ideal elastic-plastic material will not strengthen when loading continues after reaching plasticity. The stress remains constant while the deformation keeps increasing.

In numerical calculations, one crucial aspect is to incorporate the plastic strain increments modification, commonly known as plastic return mapping. The objective of this step is to

ensure that the stresses consistently reside on the yield surface. The stress is adjusted to return to the yield surface by return mapping, which expands the yield surface and corrects the stresses accordingly.

In the elastoplastic model, the calculation can be typically presented as follows:

1. Given a tensor of strain increment  $d\epsilon_{kl}$ ;
2. Evaluated the trial stress  $\sigma'_{ij} = \sigma_{ij}^0 + D_{ijkl}^e d\epsilon_{kl}$ , where  $\sigma_{ij}^0$  is the original stress, and  $D_{ijkl}^e$  is the elastic matrix;
3. Check if the model is plastically deformed by Eq. 7.1;
4. If not, the update  $\sigma_{ij}^0$  to  $\sigma'_{ij}$ . If yes, enter the plastic return mapping process and modify the stress and hardening function.

In the return mapping process, both the stress and hardening value undergo modifications according to the consistent condition. This condition ensures that the stress remains on the yield surface after yielding. Mathematically, it can be expressed as follows:

$$\begin{aligned} f(\sigma_{ij}^0 + d\sigma_{ij}, H^0 + dH) &= 0 \\ \text{or as : } df &= \frac{\partial f}{\partial \sigma_{ij}^0} d\sigma_{ij} + \frac{\partial f}{\partial H^0} dH = 0 \end{aligned} \quad (7.5)$$

where it is assumed that the yield function is fully differentiable at point  $(\sigma_{ij}^0, H^0)$ .

The stress increment can be expressed according to the tangent matrix:

$$d\sigma_{ij} = D_{ijkl}^e d\epsilon_{kl}^e = D_{ijkl}^e (d\epsilon_{kl} - d\epsilon_{kl}^p) \quad (7.6)$$

where  $D_{ijkl}^e$  is the elastic material matrix, and the plastic strain increment can be calculated as:

$$d\epsilon_{kl}^p = d\lambda \frac{\partial g}{\partial \sigma_{kl}} \quad (7.7)$$

where  $d\lambda$  is the plastic factor which can be solved later.

The increment of the hardening parameter can be expressed as Eq. 7.2. For example, if the hardening function is set to be the plastic volume strain, then  $dH = d\epsilon_{kk}^p = d\lambda \frac{\partial g}{\partial \text{sigma}_{kk}}$

Substituting Eq. 7.6, 7.7 and 7.2 into Eq. 7.5, the plastic factor can be derived as:

$$\frac{\partial f}{\partial \sigma_{ij}} D_{ijkl}^e (d\epsilon_{kl} - d\lambda \frac{\partial g}{\partial \sigma_{kl}}) + \frac{\partial f}{\partial H} \frac{\partial H}{\partial \epsilon_v^p} d\lambda \frac{\partial g}{\partial \sigma_{kk}} = 0 \quad (7.8)$$

So the plastic factor can be calculated as:

$$d\lambda = \left( \frac{\partial f}{\sigma_{ij}} D_{ijkl}^e d\epsilon_{kl} \right) / \left( \frac{\partial f}{\sigma_{ij}} D_{ijkl}^e \frac{\partial g}{\partial \sigma_{kl}} - \frac{\partial f}{\partial H} \frac{\partial H}{\partial \epsilon_v^p} \frac{\partial g}{\partial \sigma_{kk}} \right) \quad (7.9)$$

Then, the stress tensor  $\sigma_{ij}$ , plastic strain tensor  $\epsilon_{ij}^p$  and the hardening value can be updated according to the plastic factor. After substituting these variables, the yield function should be 0 to keep the stress state on the yield surface. Substituting upon equation into Eq. 7.6, the stress increment can be expressed as:

$$d\sigma_{ij} = D_{ijkl}^e \left[ d\epsilon_{kl} - \left( \frac{\partial f}{\sigma_{pq}} D_{pqmn}^e d\epsilon_{mn} \frac{\partial g}{\partial \sigma_{kl}} \right) / \left( \frac{\partial f}{\sigma_{pq}} D_{pqmn}^e \frac{\partial g}{\partial \sigma_{mn}} - \frac{\partial f}{\partial H} \frac{\partial H}{\partial \epsilon_v^p} \frac{\partial g}{\partial \sigma_{kk}} \right) \right] \quad (7.10)$$

In some works [6, 166, 167], the stress increment is expressed as:

$$\begin{aligned} d\sigma_{ij} &= D_{ijkl}^{ep} d\epsilon_{kl} \\ &= D_{ijkl}^e d\epsilon_{kl} - \left( \frac{\partial f}{\sigma_{pq}} D_{pqmn}^e d\epsilon_{mn} \frac{\partial g}{\partial \sigma_{kl}} D_{ijkl}^e \right) / \left( \frac{\partial f}{\sigma_{pq}} D_{pqmn}^e \frac{\partial g}{\partial \sigma_{mn}} - \frac{\partial f}{\partial H} \frac{\partial H}{\partial \epsilon_v^p} \frac{\partial g}{\partial \sigma_{kk}} \right) \\ &= D_{ijkl}^e d\epsilon_{kl} - \left( \frac{\partial f}{\sigma_{pq}} D_{pqmn}^e d\epsilon_{kl} \frac{\partial g}{\partial \sigma_{mn}} D_{ijkl}^e \right) / \left( \frac{\partial f}{\sigma_{pq}} D_{pqmn}^e \frac{\partial g}{\partial \sigma_{mn}} - \frac{\partial f}{\partial H} \frac{\partial H}{\partial \epsilon_v^p} \frac{\partial g}{\partial \sigma_{kk}} \right) \\ &= \left[ D_{ijkl}^e - \left( \frac{\partial f}{\sigma_{pq}} D_{pqmn}^e \frac{\partial g}{\partial \sigma_{mn}} D_{ijkl}^e \right) / \left( \frac{\partial f}{\sigma_{pq}} D_{pqmn}^e \frac{\partial g}{\partial \sigma_{mn}} - \frac{\partial f}{\partial H} \frac{\partial H}{\partial \epsilon_v^p} \frac{\partial g}{\partial \sigma_{kk}} \right) \right] d\epsilon_{kl} \end{aligned} \quad (7.11)$$

where the subscripts highlighted in red fonts are exchanged. Specifically,  $d\epsilon_{mn}$  is changed into  $d\epsilon_{kl}$ , and  $\partial g/\partial \sigma_{kl}$  is changed to  $\partial g/\partial \sigma_{mn}$ , which does not obey the tensor calculation rule.

## 7.3 Enhanced IME model

### 7.3.1 Original IME model

The original IME model consists of Isotropic elasticity, von Mises yield surface and the Exponential hardening function. For simplicity this model is referred to as the IME model.

The constitutive model can be expressed as following formulas:

$$\left\{ \begin{aligned} \sigma_{ij} &= p\delta_{ij} + s_{ij} = K\epsilon_{kk}\delta_{ij} + 2Ge_{ij} \\ &= \left[ K\delta_{ij}\delta_{kl} + 2G(\delta_{ik}\delta_{jl} - \frac{1}{3}\delta_{ij}\delta_{kl}) \right] \epsilon_{kl} \\ &= D_{ijkl}^e \epsilon_{kl} \\ f &= \sigma_v - H - \sigma_0 \\ H &= A(\epsilon_0 + \|\epsilon_{ij}^p\|_F)^n \end{aligned} \right. \quad (7.12)$$

where  $\sigma_v = \sqrt{\frac{3}{2}s_{ij}s_{ij}}$  is the von Mises stress,  $\|\epsilon_{ij}^p\|_F$  is the the Frobenius norm of plastic strain tensor, which can be expressed as  $\|\epsilon_{ij}^p\|_F = \sqrt{\sum_{i,j} |\epsilon_{ij}|^2}$ . The plastic shear strain, which is evaluated as  $\epsilon_s^p = \sqrt{\frac{2}{3}e_{ij}^p e_{ij}^p}$  where  $e_{ij}^p = \epsilon_{ij}^p - \delta_{ij}\epsilon_{kk}^p/3$ , is always used for metals. The von-Mises stress  $\sigma_v$  can be represented as:

$$\begin{cases} \sigma_v = \sqrt{3J_2} \\ J_2 = \frac{s_{ij}}{s_{ij}} \\ s_{ij} = \sigma_{ij} - p\delta_{ij} \end{cases} \quad (7.13)$$

where  $J_2$  is the second invariant of the derivative stress.

In Eq. 7.9,  $\frac{\partial f}{\partial \sigma_{ij}}$  and  $\frac{\partial H}{\partial \epsilon_{ij}^p}$  are needed for the plastic return mapping. The differentiation of  $\sigma_v$  to stress can be expressed as:

$$\frac{\sigma_v}{\sigma_{ij}} = \frac{3(\sigma_{ij} - p\delta_{ij})}{2\sigma_v} \quad (7.14)$$

And the differentiation of the Frobeniusnorm of the plastic tensor can be shown as:

$$\frac{\partial \|\epsilon_{ij}^p\|_F}{\partial \epsilon_{ij}^p} = \frac{\epsilon_{ij}^p}{\|\epsilon_{ij}^p\|_F} \quad (7.15)$$

We also have  $\partial f/\partial \sigma_v = 1$ ,  $\partial f/\partial H = -1$  and  $dH/d\|\epsilon_{ij}^p\|_F = nA(\epsilon_0 + \|\epsilon_{ij}^p\|_F)$ . According to the chain rule, after substituting all of the derivatives to Eq. 7.9, the plastic factor  $d\lambda$  is obtained. The stress, material tangent matrix and plastic strain can be fed to the FEM solver.

Parameters used in the IME model are summarised as following table 7.1.

Table 7.1: Material constants of IME model

Young's modulus $E$	20 MPa
Poisson's ratio $\nu$	0.2
$\sigma_0$	0.1 MPa
$A$	0.3 MPa
$\epsilon_0$	0.02
$n$	0.2

### 7.3.2 Enhance the original model

The original IME model is rather straightforward. It exhibits isotropic linear elasticity and relies solely on the von Mises stress to govern the yield surface of plasticity. Due to these basic functions, its capabilities are limited. As a result, the model is incapable of reproducing nonlinear elasticity or yield criteria associated with mean stress, etc.

Consequently, we introduce several modifications to the original IME model in order to augment its capabilities. For the elastic part, the material remains isotropic and is governed by Young's modulus  $E$  and Poisson's ratio  $\nu$ . However, we adjust Young's modulus  $E$  to become a nonlinear function, which now correlates with the plastic volume strain.

$$E = E_0(1 + \epsilon_v^p)^{n_E} \quad (7.16)$$

where  $E_0$  is the initial Young's modulus, and  $n_E$  is an added material constant to describe the changes of the elastic modulus. If  $n_E = 0$  then the elastic part degenerates to isotropic linear elasticity.

And change the yield surface to mean stress-related:

$$f(p, \sigma_v) = C_p p + \sigma_v - H - \sigma_0 \quad (7.17)$$

where  $C_p$  is the correlation coefficient between the yield surface and the mean stress, and if  $C_p = 0$ , the yield surface degenerates to the original IME model yield surface.

In the original IME model, the hardening function is related to the Frobenius norm of the plastic strain tensor, which in this case we divide into plastic volumetric strain and plastic shear strain. Modify the hardening function into:

$$H(\epsilon_s^p, \epsilon_v^p) = A(\epsilon_0 + \epsilon_s^p)^B + C(\epsilon_{0p} + \epsilon_v^p)^D \quad (7.18)$$

where  $C$ ,  $\epsilon_{0p}$  and  $D$  are added material constants to consider the influence of plastic volumetric deformation on the material hardening.

At the same time, the expansion coefficient  $C_d$  is used to change the associated plastic flow into the non-associated flow:

$$\frac{\partial g}{\partial p} = C_d + \frac{\partial f}{\partial p} = C_d + C_p \quad (7.19)$$

where  $C_d$  is an added material constant to consider the plastic volumetric deformation.

With the above modifications, the enhanced IME model is able to take into account the nonlinear elasticity, the mean stress-dependent yield function and the non-associated plastic flow.

## 7.4 CSUH model

Yao introduced the CSUH (Critical State Unified Hardening) model [6] based on the UH (unified hardening model) [168] as an advancement to the modified Cam-Clay (MCC) model. The key components of this model are the critical state-related internal parameter  $\xi$  and the unified hardening function, which allows for a unified representation of clay and sand behaviours.

### 7.4.1 Yield surface and hardening function

The CSUH model's yield function comprises two main formulas: the reference yield surface Eq. 7.20 and the current yield surface Eq. 7.21. Fig. 7.2 provides a visual representation of these yield surfaces.

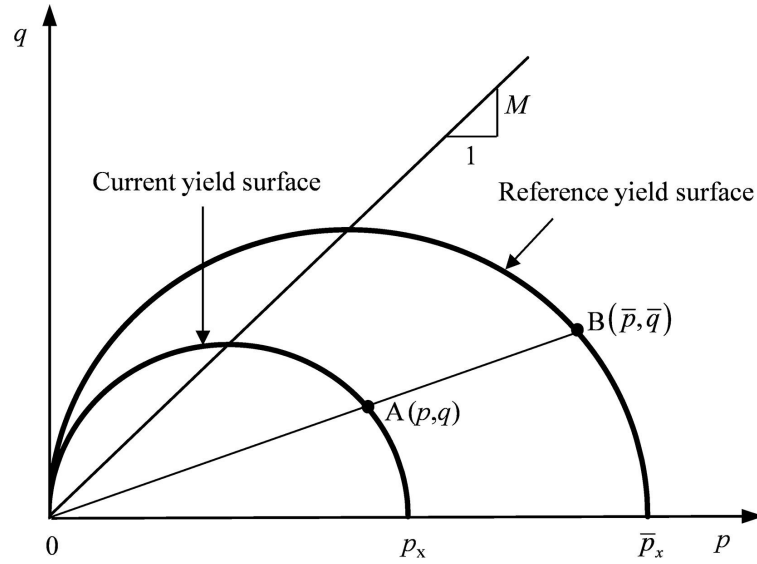


Figure 7.2: The reference and current yield surface of the CSUH model [6]

$$f = \ln \frac{\bar{p}}{\bar{p}_{x0}} + \ln \left( 1 + \frac{\bar{q}^2}{M^2 \bar{p}^2} \right) - \frac{\epsilon_v^p}{c_p} \quad (7.20)$$

$$f = \ln \frac{p}{p_{x0}} + \ln \left( 1 + \frac{q^2}{M^2 p^2} \right) - \frac{H}{c_p} \quad (7.21)$$

where  $p_{x0}$  is the where,  $p_{x0}$  is the value corresponding to the intersection point of the initial yield surface and the axis of the mean stress  $p$ , and the unit is generally kPa;  $c_p = \frac{\lambda - \kappa}{1 + e_0}$ ,  $e_0$  is the initial void ratio, and the unified hardening function is as follow:

$$H = \int \frac{M_f^4 - \eta^4}{M_c^4 - \eta^4} d\epsilon_v^p \quad (7.22)$$

where  $M_f$  is the potential failure stress ratio corresponding to current state and  $M_c$  is the characteristic state stress ratio. These two variables are related with the current state  $\xi$ .

## 7.4.2 Current state representation

Before introducing the calculation of  $\xi$ , the over-consolidation ratio can be defined as  $\text{OCR} = \frac{\bar{p}}{p}$ . We introduce the over-consolidation parameter  $R \in (0, 1]$ :

$$R = \frac{p}{\bar{p}} \quad (7.23)$$

where  $\bar{p}$  is the reference stress or the pre-consolidation pressure if the sample is isotropically compressed as is shown in Fig. 7.3. From the figure, we can have the relationship of the current stress and the reference stress:

$$\ln \bar{p} - \ln p = \frac{\xi}{\lambda - \kappa} \quad (7.24)$$

Then the over-consolidation parameter can be calculated as:

$$R = \exp \frac{-\xi}{\lambda - \kappa} \quad (7.25)$$

where  $\xi$  is the distance between the current void ratio and the corresponding void ratio at critical state as is shown in Fig. 7.3:

$$\xi = e_\eta - e \quad (7.26)$$

where  $e_\eta$  is the void ratio according to the anisotropic critical state line, which can be calculated as:

$$e_\eta = N - \lambda \ln p - (\lambda - \kappa) \ln \left(1 + \frac{\eta^2}{M^2}\right) \quad (7.27)$$

And in this model, the potential failure stress ratio and the characteristic stress ratio are calculated as:

$$\begin{cases} M_f = 6 \left( \sqrt{\frac{k}{R} \left(1 + \frac{k}{R}\right)} - \frac{k}{R} \right), & k = \frac{M^2}{12(3 - M)} \\ M_c = M \exp(-m\xi) \end{cases} \quad (7.28)$$

where  $M$  is the critical stress ratio, and  $m$  is a coefficient of dilatation which can be calibrated by experiment. The potential failure stress ratio is related with the Hvorslev envelope and different types of Hvorslev envelopes are compared in Zhou's work [7]. The potential failure ratio should not be larger than 3 so the piece-wise Hvorslev envelope is introduced to apply

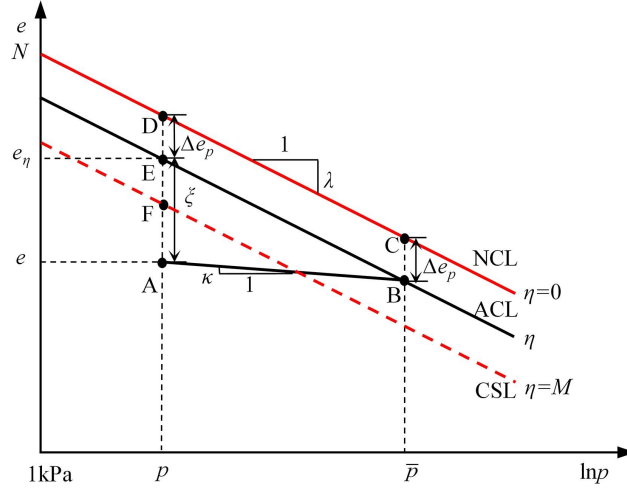


Figure 7.3: Normal consolidation line (NCL), anisotropic compression line (ACL), and the critical state line (CSL) on the  $e - \ln p$  plane

this constraint. In Fig. 7.4, we can find the parabolic Hvorslev envelope and the Hermite Hvorslev envelope performs well. The potential failure stress ratio  $M_f$  in Eq. 7.28 is based on the parabolic envelope.

As shown in Fig. 7.5, when the sand material is compressed isotropically, its curve is composed of two straight lines on the logarithmic coordinate. At low confining pressure, the material void ratio changes slightly with the consolidation pressure. Entering high confining pressure, the curves gradually approach the NCL and finally coincide. Therefore, the isotropic compression line after introducing the inflection point is assumed to be:

$$e = Z - \lambda \ln \frac{p + p_s}{1 + p_s} \quad (7.29)$$

The curve automatically satisfies the constraint condition of the first point, at  $p = 1e3\text{kPa}$ , the void ratio is  $Z$ ; the second condition, when the confining pressure tends to infinity, the curve coincides with the NCL, and the above formula and  $e = N - \lambda \ln p$  can be solved simultaneously have to:

$$p_s = \exp\left(\frac{N - Z}{\lambda}\right) - 1 \quad (7.30)$$

As shown in Fig. 7.5, the logarithmic coordinates of the curve are composed of approximately straight lines at both ends, where the initial point is  $(1e3, Z)$  and the inflection point is  $(p_s, Z - \lambda \ln \frac{p_s}{1+p_s})$ . Note that if then the curved NCL degenerates into the original NCL formula (6.131).

After modifying NCL, the material yield function should be accordingly modified to the



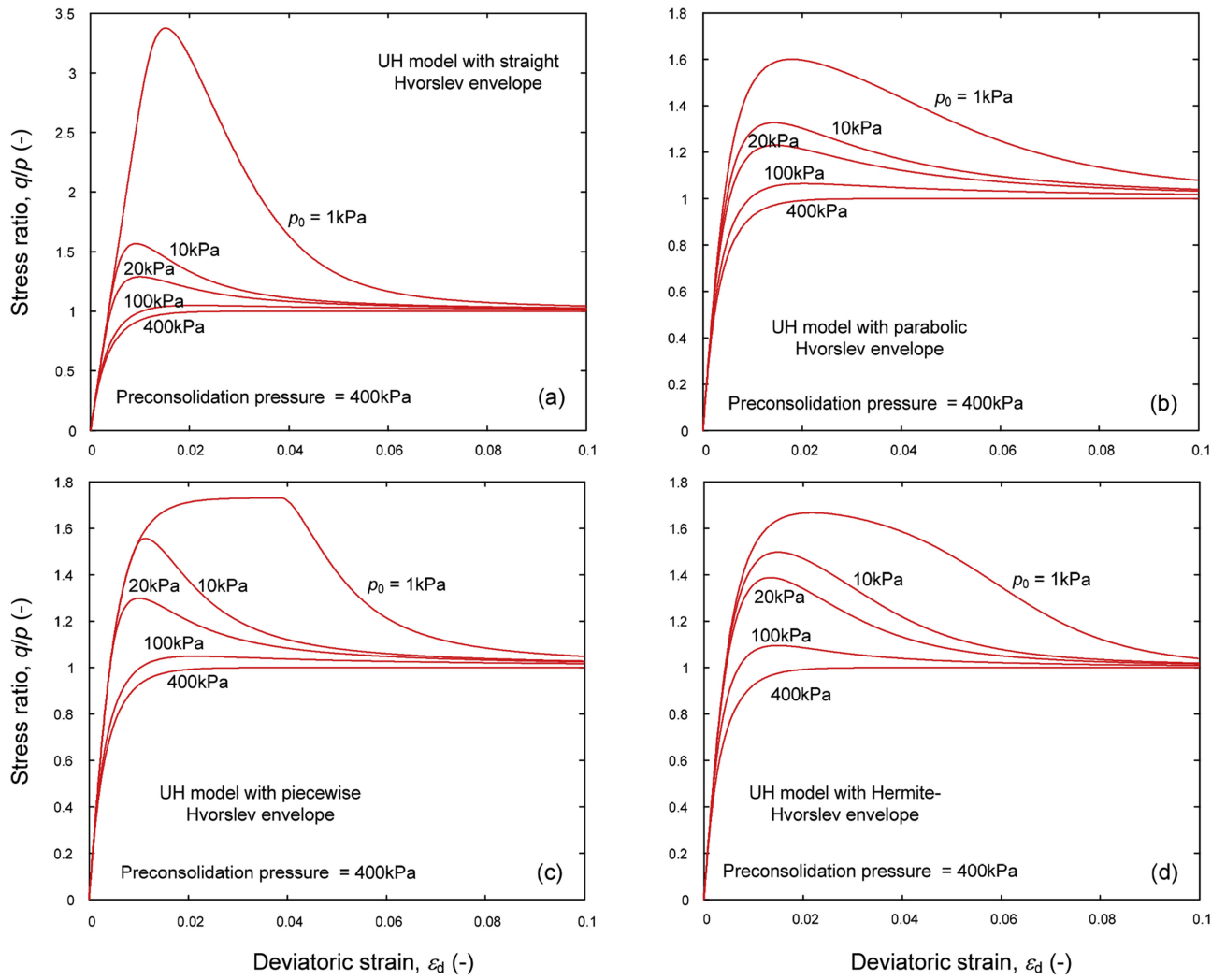


Figure 7.4: Shear stress ratio according to different kinds of Hvorslev envelope [7]

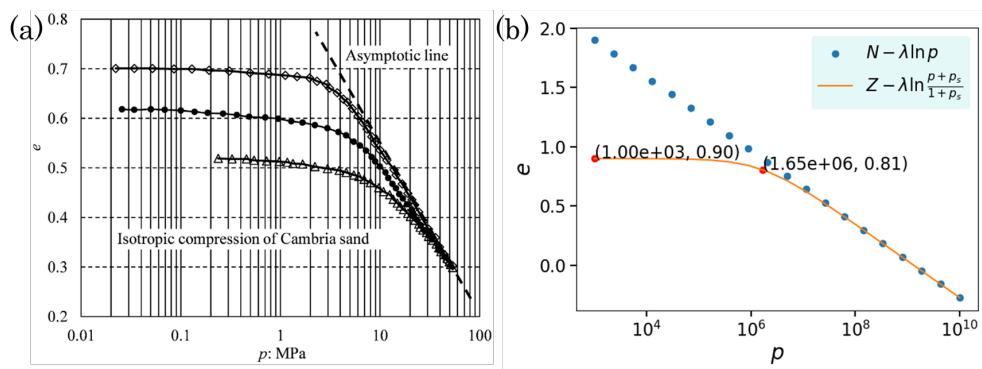


Figure 7.5: The logarithmic coordinate ICL (Isotropic compression line) of Cambria sand and the asymptote on the logarithmic coordinate [8]; (b) The two forms of the isotropic compression line on the logarithmic coordinate with  $N = 1.9$ ,  $Z = 0.9$ .

following form:

$$f = \ln \left[ 1 + \frac{(1 + \chi)q^2}{M^2 p^2 - \chi q^2} \right] p + p_s - \ln(p_{x0} + p_s) - \frac{H}{c_p} \quad (7.31)$$

where  $\chi$  is the parameter represents the distance between NCL and CSL, which should be less than non-negative and less than 1. Before modification the distance between NCL and CSL is  $\ln 2$ . Then the distance is changed to  $\frac{1+\chi}{1-\chi} \ln 2$

And the plastic potential function is:

$$g = \ln \frac{p}{p_y} + \ln \left( 1 + \frac{q^2}{M_c^2 p^2} \right) \quad (7.32)$$

where the character stress ratio  $M_c$  is introduced the implement the non-associated flow rule and the dilatation, which is critical stress ratio  $M$  in the UH model. So in CSUH model, the direction of plastic deformation is:

$$\frac{d\epsilon_v^p}{d\epsilon_s^p} = \frac{\partial g / \partial p}{\partial g / \partial q} = \frac{M_c^2 - \eta^2}{p(M_c^2 + \eta^2)} / \frac{2\eta}{p(M_c^2 + \eta^2)} = \frac{M_c^2 - \eta^2}{2\eta} \quad (7.33)$$

### 7.4.3 Influence of the medium principal stress ratio

The medium principal stress coefficient  $b = \frac{\sigma_2 - \sigma_3}{\sigma_1 - \sigma_3}$  in loading affects the critical state stress ratio.  $b = 0$  represents the compression and  $b = 1$  indicates the extension loading. In the CSUH model, the distinction between compression and extension is realised by using the transformed stress space [169] as follows:

$$\begin{cases} \tilde{\sigma}_{ij} = \sigma_{ij} & q = 0 \\ \tilde{\sigma}_{ij} = p\delta_{ij} + \frac{q_c}{q} (\sigma_{ij} - p\delta_{ij}) & q \neq 0 \end{cases} \quad (7.34)$$

where the stress  $q_c = \frac{2I_1}{3\sqrt{(I_1 I_2 - I_3)/(I_1 I_2 - 9I_3)} - 1}$ , and  $I_1$ ,  $I_2$  and  $I_3$  are the invariants for stress tensor. The above method can transform the stresses from the original space to the transformed space. Through this, it is possible to consider the effects of the tensile and compressive loading conditions of the material by means of the circular yield surface on the  $\pi$  plane in the transformed space. However, the multiple squaring and divisions involved in this method make this calculation less robust in numerical calculations.

Here we refer to Pastor's work [148], and use the following formula to consider the influence of the medium principal stress ratio:

$$M = \frac{18M_{\text{com.}}}{18 + 3(1 - \sin 3\theta)} \quad (7.35)$$

where  $\theta$  is the Lode angle,  $M_{\text{com.}}$  is the critical stress ratio under compression loading path. The Lode angle ranges from  $-\frac{\pi}{6}$  to  $\frac{\pi}{6}$ , while  $\theta = -\frac{\pi}{6}$  represents extension and  $\theta = \frac{\pi}{6}$  is extension.

According to the Mohr-Coulomb criterion, the stress ratio under compression can be expressed as:

$$M_{\text{com.}} = \frac{6 \sin \phi}{3 - \sin \phi} \quad (7.36)$$

where  $\phi$  is the frictional angle.

## 7.5 Specificity of mechanical properties of granular materials

For granular materials, firstly we cannot obtain a stable one-to-one mapping relationship between shear stress  $\sigma_s$  and shear strain  $\epsilon_s$ , as shown in Fig. 7.6. The material exhibits nonlinear elasticity in the small strain stage, where the material shear stress is basically determined by the shear strain. Moreover, once the loaded shear strain increases, the shear stress is no longer a unique value. In addition, as shown in Fig. 7.6b, there is an angle between the stress tensor direction and the strain tensor direction, and it changes with loading conditions. The angle is calculated based on Sec. 6.6.3.

Therefore, for granular materials, we are unable to introduce this co-axiality which assumes the principal direction of the strain tensor and stress tensor aligns parallel. This assumption is utilised in several papers [118, 142] to reduce the dimension of input and output features. This assumption is further developed by Tang et al. [109] to map the one-dimensional dataset to three dimensions.

It is worth mentioning that when we use the CSUH model for biaxial simulation, the results obtained for this material at the Gaussian point are shown in Fig. 7.7. The relationship between shear strain and shear stress is affected by the medium principal stress coefficient  $b$  and the relative compactness of the material; the difference in the angle between the principal direction of stress and strain tensor changes gradually with the loading. The difference is not significant in the CSUH model, whereas the difference obtained from the exFEM-DEM simulation is much larger and noisy. Due to the introduction of state-dependent shear expansion properties, the CSUH model has the potential to reflect the nature of the strain-stress relationship and deflection angle between the principal directions of strain and stress tensor.

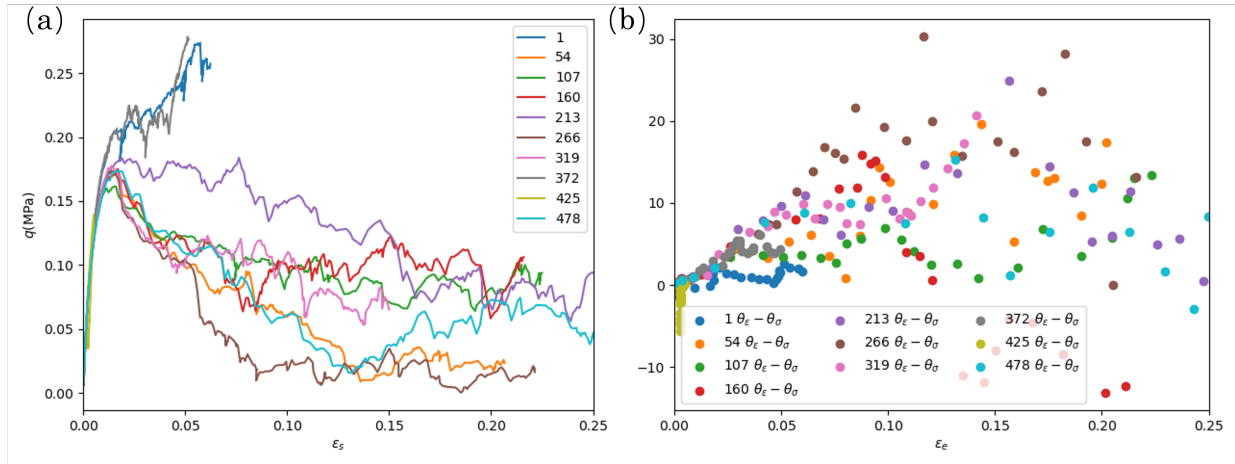


Figure 7.6: Mechanical responses on Gauss points in exFEM-DEM biaxial simulations (a) Shear stress and shear strain relationship; (b) Difference in angle between strain tensor and stress tensor  $\theta_\epsilon - \theta_\sigma$  ( $^\circ$ )

Therefore, we construct a constitutive template based on the CSUH constitutive model. And it is optimised by sequential training as is depicted in the recurrent material cell training.

## 7.6 Optimisation of constitutive models based on the datasets collected from exFEM-DEM simulations

The theory of elastoplasticity of granular materials, especially geotechnical granular materials, has been widely studied [170]. The elastic-plasticity theory mainly includes the definition of the elasticity, yield function, hardening function and plastic potential function as is mentioned in Sec. 7.2.

The study of FEM-NN shows that the neural network-based constitutive model is more similar to an interpolator, learning the constitutive patterns from the data and completing the stress prediction by mapping the inputs and outputs. The performance of the network in FEM is completely controlled by the training samples. When the training samples cover enough sampling space with high accuracy, the network-based constitutive model is able to fully reproduce the constitutive knowledge; however, when the inputs exceed the training range, the network's accuracy is quite poor.

We have to face the problem that for high dimensional cases, it is very difficult to guarantee that the trained network can be general and robust in any problem by completely

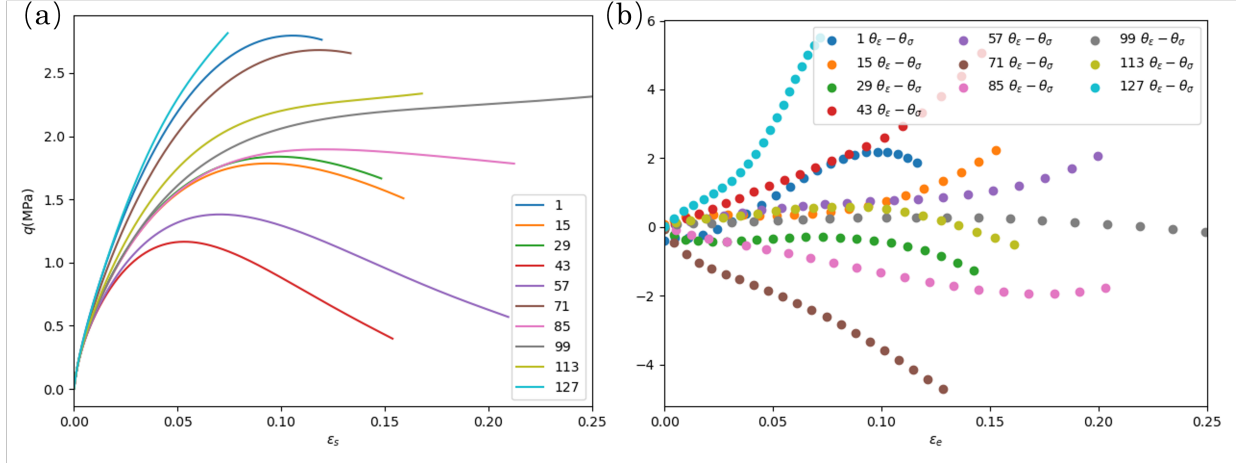


Figure 7.7: Mechanical responses on Gauss points in CSUH model-based biaxial simulations (a) Shear stress and shear strain relationship; (b) Difference in angle between the principal direction of strain and stress tensor  $\theta_\epsilon - \theta_\sigma$ ( $^\circ$ )

covering the input space by sampling, as the possible loading paths are infinite but the samples are finite.

The generalisation ability and accuracy of network-based constitutive models are two critical points for networks to reproduce constitutive relationships. Neural networks with thousands of parameters have considerable mapping ability to excavate potential knowledge. Yet poorly generalised models are unable to cope with a wide variety of inputs when embedded in FEM, and therefore cannot guarantee prediction accuracy at every Gauss point. Errors at individual Gauss points arise, accumulate and propagate leading to the failure of the global computation.

Therefore, it is not wise to adopt a purely data-driven approach and completely discard prior knowledge about elasticity and plasticity; we have to find a balance between the model's generalisation ability and its accuracy. By sacrificing some of the neural network's degrees of freedom, a generalised model is obtained. To implement this, the optimisable parameters in the neural network are restricted by introducing some constraints.

As shown in Fig. 7.8, the calculation process of a constitutive model is basically the same as that of the recurrent neural network calculation process. Therefore, we establish the constitutive model based on prior knowledge in the framework of PyTorch. And optimised the model on the strain stress sequences from FEM-DEM multi-scale simulations. Once the stress and strain sequences are available, the material constants of the recurrent constitutive

model  $\mathcal{M}_{cons}$  can be optimised.

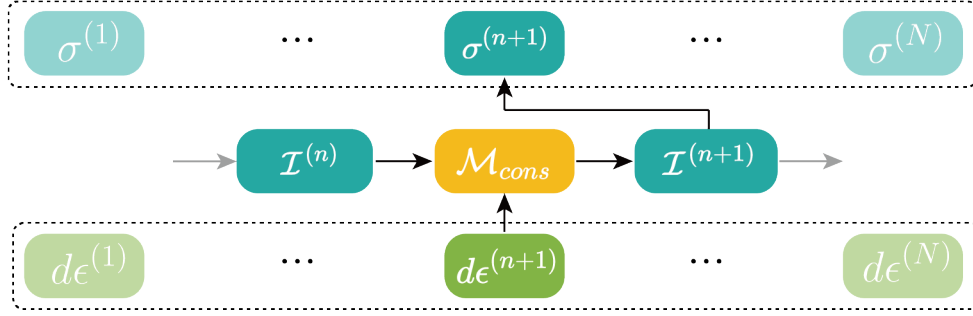


Figure 7.8: Constitutive model calculation process, from the strain sequence to the stress sequence

### 7.6.1 Baseline: $J_2$ model

Before training the model using the exFEM-DEM data, we validate the model using the  $J_2$  model with ideal plasticity under the assumption of planar stresses, as shown below:

$$\begin{cases} \sigma_{ij}^{TR} = 2G * e_{ij} + \delta_{ij}K\epsilon_v \\ p = \sigma_{kk}^{TR}/2 \\ q^{TR} = \sqrt{2s_{ij}^{TR}s_{ij}^{TR}} \\ \sigma_{ij} = \delta_{ij}p + s_{ij}^{TR} \min\left(\frac{\sigma_y}{q^{TR}}, 1\right) \end{cases} \quad (7.37)$$

where  $K$  and  $G$  are the bulk and shear elastic modulus, respectively, and  $\sigma_y$  is the yield stress. There is no hardening in this model, so the model is defined on the three trainable parameters  $\{K, G, \sigma_y\}$ .

In order to obtain sufficient training data, we adopt a stochastic Gaussian process for a large number of strain-controlled loading paths. This is done in the following way, the Gaussian process is used to generate two principal strains and the Lode angle  $\theta$  which are then rotated from the principal space to the tensor space.

The constitutive model constructed via the trainable parameters  $\{K, G, \sigma_y\}$  was trained by error back-propagation on the datasets generated under the  $J_2$  model. In this case, the error back-propagation optimisation corrects these parameters to the values used in the training data preparation. The training process is shown in Fig. 7.9, where trainable parameters perfectly align with the original values. Getting perfect results is not a surprise, the reason

is straightforward. The physical mechanism used to construct the model matches exactly the constitutive relationship contained in the dataset. Finally parameters are optimised to the right values  $K \rightarrow 1.25e6$ ,  $G \rightarrow 8.3e5$  and  $\sigma_y \rightarrow 1e5$ .

The example illustrates the feasibility of the error back-propagation method for calibrating material parameters. As long as the physical mechanisms introduced match the constitutive knowledge of the dataset, a satisfactory constitutive model can be obtained.

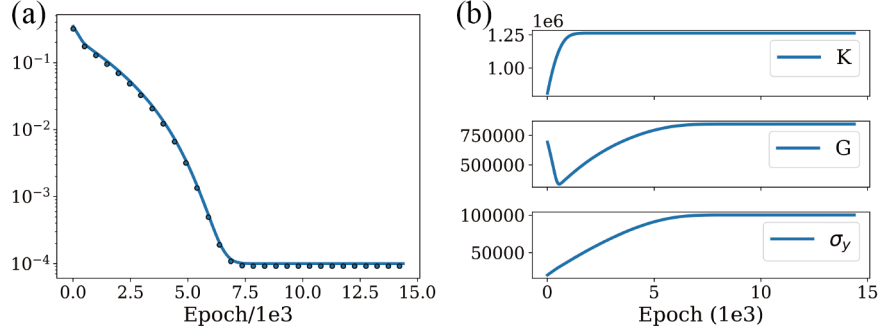


Figure 7.9: Optimising process of the  $J_2$  model. (a) The calculation error. (b) The trainable model parameters.

## 7.6.2 Enhanced IME model

Instead of using mathematical models to generate the data sets like the optimisation of the  $J_2$  model, we extract the data sets from the explicit FEM-DEM simulation, which is more similar to the granular material's nature. There are no exact mathematical models competent to reproduce the constitutive relationship contained in these data sets fully. As depicted before, the complexity is caused by history-dependency, friction, density-dependency, etc.

Then this constitutive knowledge is assumed to match the elastoplasticity constraints of the enhanced IME model. Specifically, the constitutive relationship is assumed to be mean stress-dependent nonlinear elasticity, isotropic exponential hardening and non-associated flow rule as is shown in Sec. 7.3. The material constants of the enhanced IME model are optimised on these data sets. The loss is shown in Fig. 7.10. Finally, the training loss stopped at  $3.5e-2$ . The original and optimised parameters are listed in Tab. 7.2.

It's important to take note that the parameter  $C_p$  has been optimised to -0.512, signifying that the behaviour of friction governs the yield surface. This mirrors the yield surfaces of Drucker-Prager and Mohr-Coulomb. Parameter  $A$  and parameter  $C$  have been optimised

very close to 0. This implies that the optimised model approaches the ideal plasticity, where deformation occurs without altering the yield surface. The parameter  $n_E$  captures the softening of the model, where the average stress  $p$  increases while Young's modulus of the material decreases during loading. Typically, an increase in pressure or material stress would result in a higher Young's modulus. This phenomenon is intricate to explain due to the interconnectedness of parameters within the model.

Table 7.2: The optimised material constants of the enhanced IME model

Parameters	$E$	$\nu$	$A$	$B$	$\epsilon_0$	$\sigma_y$
Original	2e7	0.2	3e5	0.2	0.02	1e4
Optimised	4.81e8	0.216	34.5	-2.23	2.79e-2	5.17e-6
Parameters	$C_d$	$C_p$	$C$	$D$	$\epsilon_{0p}$	$n_E$
Original	0.1	-0.1	3e5	0.2	0.02	0.1
Optimised	0.155	-0.512	8.35e-5	0.424	8.83e-4	-3.54

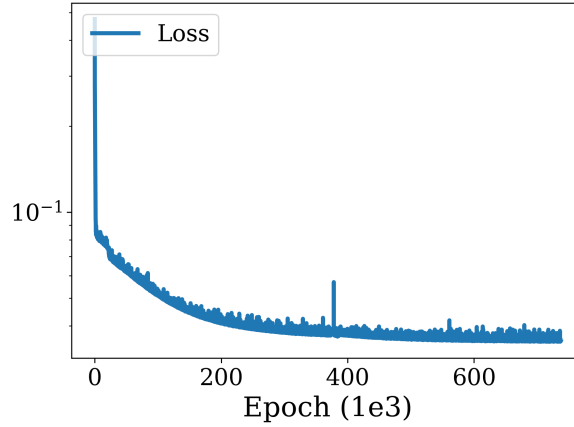


Figure 7.10: Evolution of the loss during the optimisation process: enhanced IME model

The stress values predicted after optimising the Enhanced IME model are depicted in Fig. 7.11. The optimised model successfully captures the characteristics of nonlinear elasticity and the critical state of the material. While the peak stresses calculated by the Enhanced IME model are slightly lower than the actual peak stresses, the stresses at the initial stages of model calculations are slightly higher than those in the training set. This discrepancy arises because the model doesn't account for a peak failure stress ratio. Most post-peak stresses align well with the actual values, although the stress at point 192 exhibits a notably



abrupt post-peak surge, and the post-peak stresses at points 0 and 48 are somewhat lower. In summary, the Enhanced IME model better replicates the attributes of the FEM-DEM simulation dataset, achieving good agreement in terms of nonlinear elasticity and post-peak critical states.

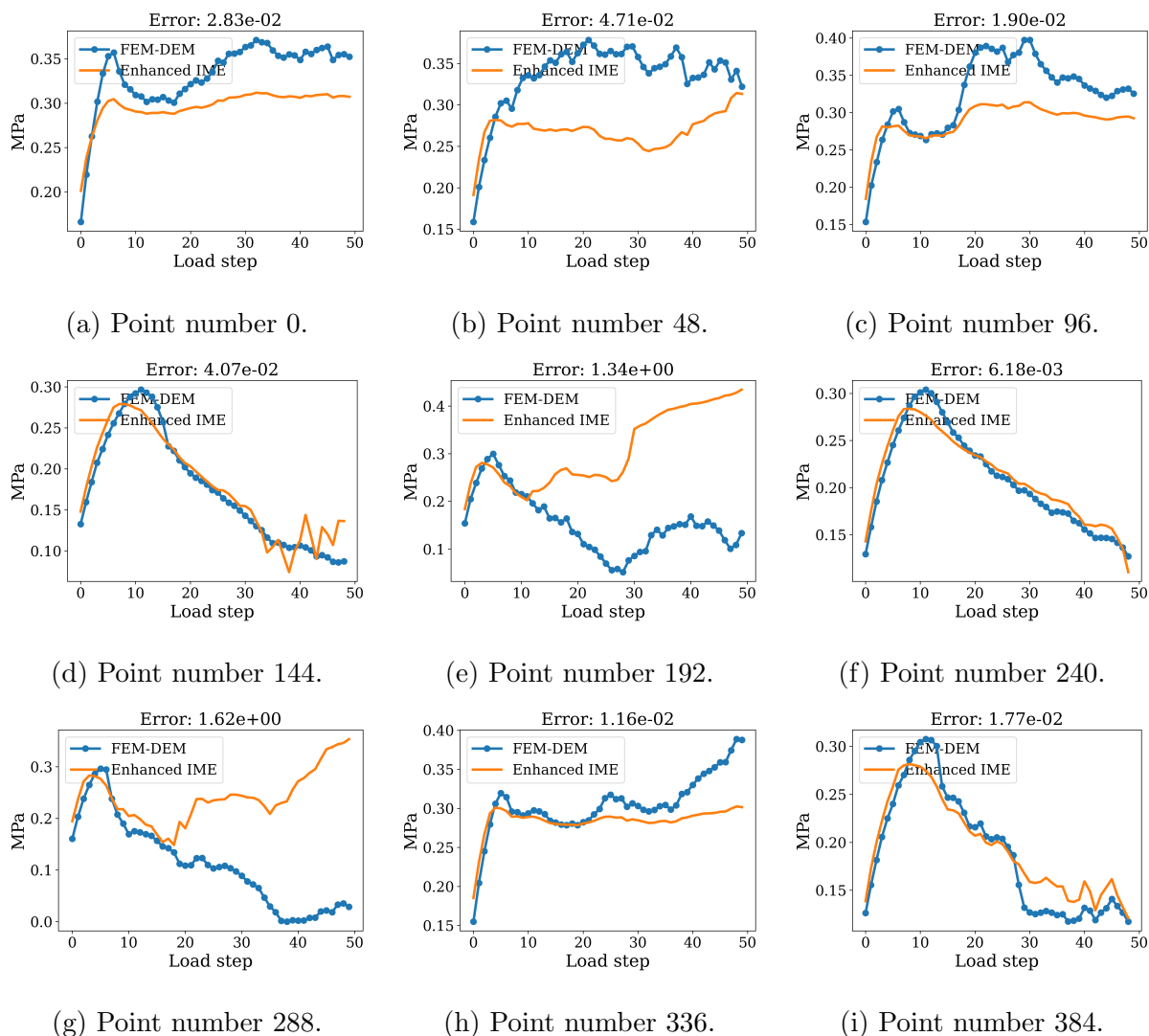


Figure 7.11: Predicted  $\sigma_{yy}$  curves after optimisation of Enhanced IME model.

### 7.6.3 CSUH model

Through the training of the Enhanced IME model, we gather insights that the FEM-DEM dataset showcases attributes like nonlinear elasticity, yield surface related with mean stress  $p$ , and peak stress surpassing critical state stress. In contrast, the CSUH model considers aspects such as nonlinear elasticity, the Cambridge yield surface, peak stress in dense sands,

the non-associative plastic flow rule, etc. Consequently, the CSUH model aligns more closely with the constitutive knowledge contained within the FEM-DEM dataset.

The CSUH model underwent optimisation using the FEM-DEM dataset. The parameters of the CSUH model before and after optimisation are outlined in Tab. 7.3. Prior to optimisation, the slope of the rebound line stood at  $\kappa = 0.04$ , while post-optimisation, it became  $\kappa = 0.329$ . Consequently, the material's elasticity modulus decreased, rendering the material softer. Simultaneously, the over-consolidation ratio  $ocr$  experienced an increment, aligning with the material's elevated yield stress to reproduce its peak stress levels. The critical stress ratio  $M$  was reduced from 1.25 to 0.63, aimed at diminishing shear stress during the critical state. Pre-optimization, the parameter  $p_s = \exp\left(\frac{N-Z}{\lambda}\right) - 1 = 0$ . Here,  $N = Z$ , signifying the degeneration of the natural consolidation line with inflexion points into a straight line within the  $e - \ln p$  space. Through optimisation, the parameter  $Z$  was determined to be  $-0.769$ , indicating that natural compression corresponds to a pore ratio  $e$  of  $-0.96$  when the mean pressure  $p = 1$  kPa. Additionally, the parameter  $m$  escalated from 1.8 to 29.26. This parameter,  $m$ , exerts control over the characteristic state stress ratio via  $M_c = M \exp(-m\xi)$ , thereby governing shear expansion through the hardening equation.

The interplay of material parameters within the elastic and plastic components is important but makes it harder for us to analyse the influence of the material constants changes. In the optimised model, a clear demarcation between elastic and plastic behaviour, as outlined in the original theory, will be influenced by each other. For instance, the parameter  $\kappa$  can be adjusted to modify the model's elastic characteristics, given that the modulus of elasticity  $K = \frac{(1+e)p}{\kappa}$ . Manipulating the elastic modulus can influence the peak stress by tuning the over-consolidation ratio  $ocr$ , which in turn impacts elasticity. Due to these intricate interactions, a singular solution to the optimisation is not guaranteed.

Nonetheless, this approach offers enhanced interpretability in comparison to utilising a black-box method like a neural network. The optimisation process of the parameters facilitates a clearer understanding of the underlying knowledge embedded within the dataset. For example, the optimisation shifted the critical state stress  $M$  ratio from 1.25 to 0.63, indicating that the initial model overestimated shear resistance attributed to material friction. In the critical state, the shear stress to mean stress ratio  $q/p$  should ideally reduce to 0.63.

The model's predictions on the training dataset (from the footing simulation) following optimisation are depicted in Fig. 7.12. The model demonstrates robust performance across the majority of data points, particularly those featuring higher stress levels. Notably,

Table 7.3: The CSUH parameters optimised on the exFEM-DEM footing simulation

Parameters	$\kappa$	$\lambda$	$N$	$Z$
Original	0.04	0.135	1.9	1.9
Optimised	0.329	0.342	2.708	-0.769
Parameters	$m$	$\nu$	ocr	$M$
Original	1.8	0.2	120	1.25
Optimised	29.26	0.263	787.8	0.630

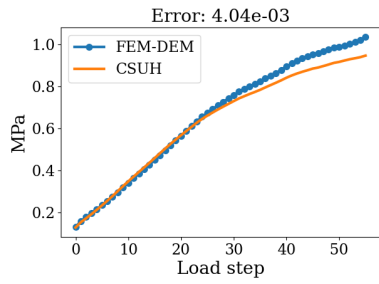
at integration points 48 and 288, the relative error appears substantial visually; however, the absolute error remains minimal. These specific points are situated near the boundary, resulting in stress levels approximating the designated simulated confining pressure of 1 kPa.

Comparing Fig. 7.11 (training results of the IME model) with Fig. 7.12 (training results of the CSUH model), the average prediction error for the IME model is 7.81% after removing outliers (points with excessive error), while the CSUH model's prediction error is 0.42%. This suggests that the priors/assumptions in the CSUH model align more closely with the FEM-DEM data.

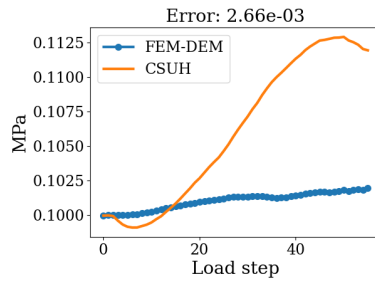
Following the optimisation of the CSUH model using footing simulation datasets, its performance is assessed against the biaxial simulation dataset. When compared to Fig. 7.11, the predictions from the optimised CSUH model performs much better in at the beginning of the loading, and successfully capturing the nonlinear elastic phase and the peak state at all test points. The optimised IME model performs poor in capturing the peak state. At point 0, 96 and 336, results of the computation with optimised CSUH model perfectly agree with the FEM-DEM simulation.

This result highlights the alignment between the CSUH model's constitutive representation of nonlinear elasticity, which is captured by the natural compression line, as well as the properties defined by the yield and hardening function.

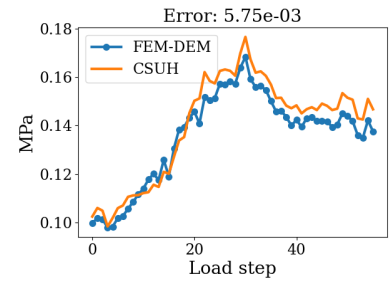
Subsequently, the optimised model is incorporated into the explicit FEM solver. The results of macroscopic calculations are displayed in Fig. 7.14. On a macroscopic scale, the optimised CSUH model adequately reproduces the outcomes of FEM-DEM multiscale simulations. This includes the macroscopic peak stresses and the emergence of shear zones during biaxial shear. The experiment illustrates that despite the CSUH model's parameters not being determined through standard physical experiments, the parameters derived



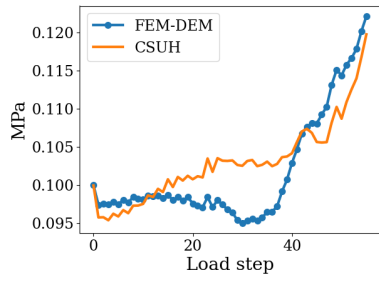
(a) Point number 0.



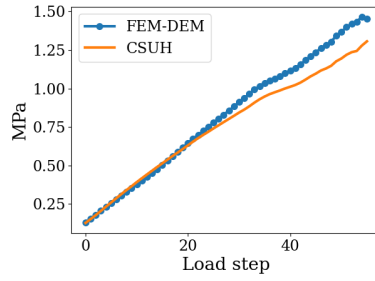
(b) Point number 48.



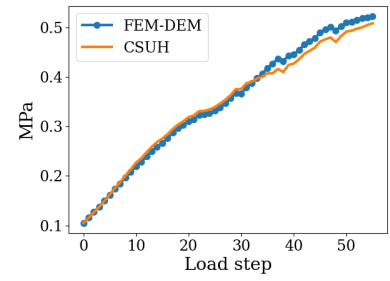
(c) Point number 96.



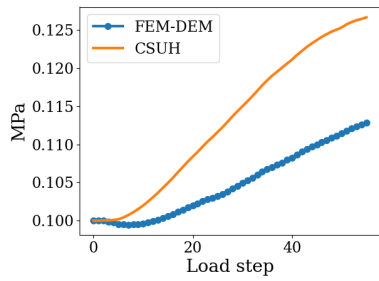
(d) Point number 144.



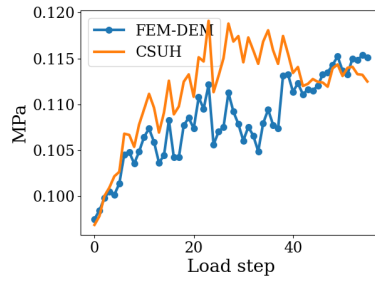
(e) Point number 192.



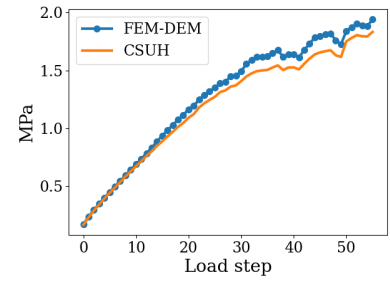
(f) Point number 240.



(g) Point number 288.



(h) Point number 336.



(i) Point number 384.

Figure 7.12: Predicted stress component  $\sigma_{yy}$  of the CSUH model after optimisation on the data sets collected from exFEM-DEM footing simulation.

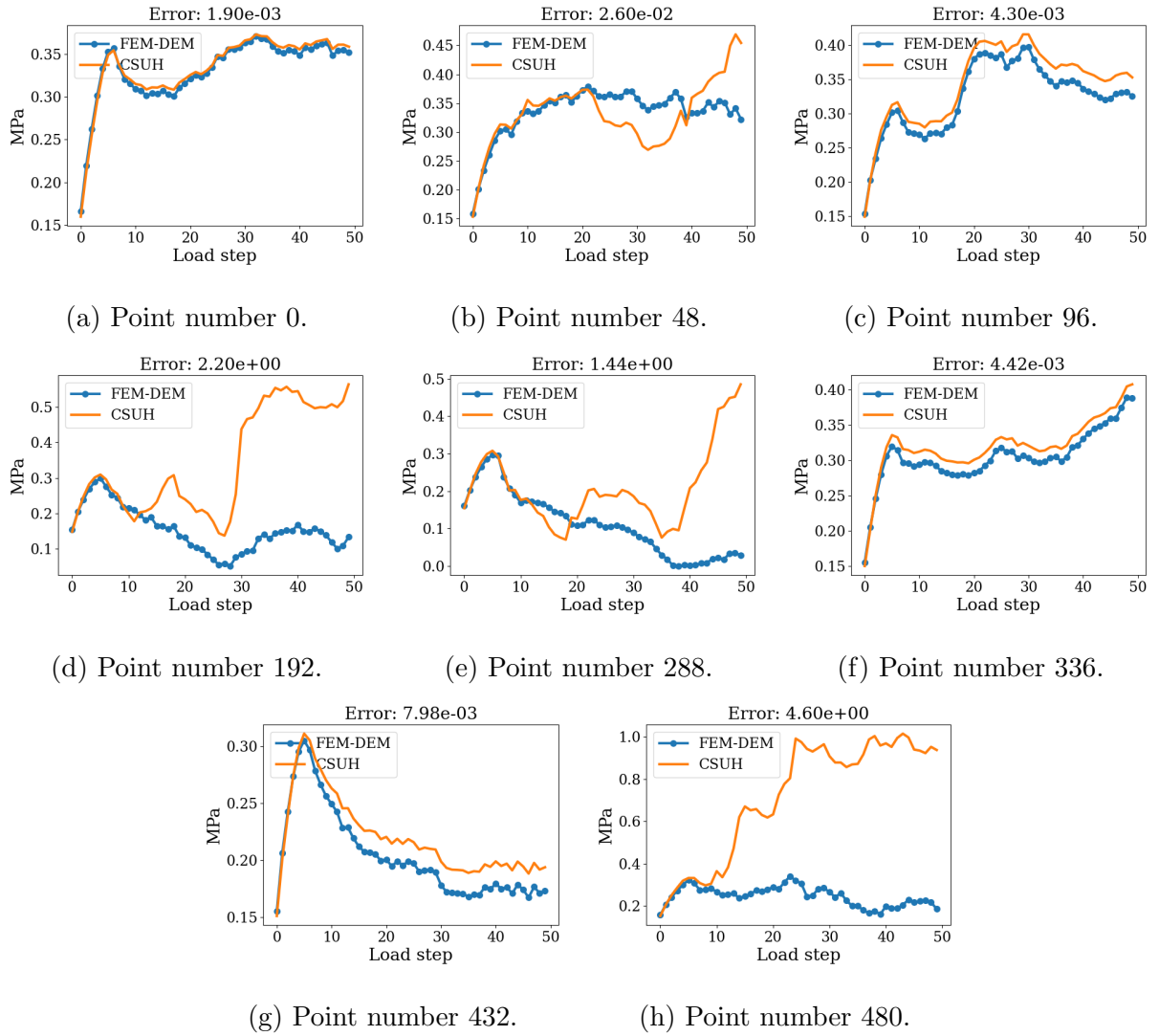


Figure 7.13: Test on data sets collected from the biaxial simulation: predicted stress component  $\sigma_{yy}$  after the CSUH model optimised on the footing simulation. The test data sets are from the exFEM-DEM biaxial simulation.

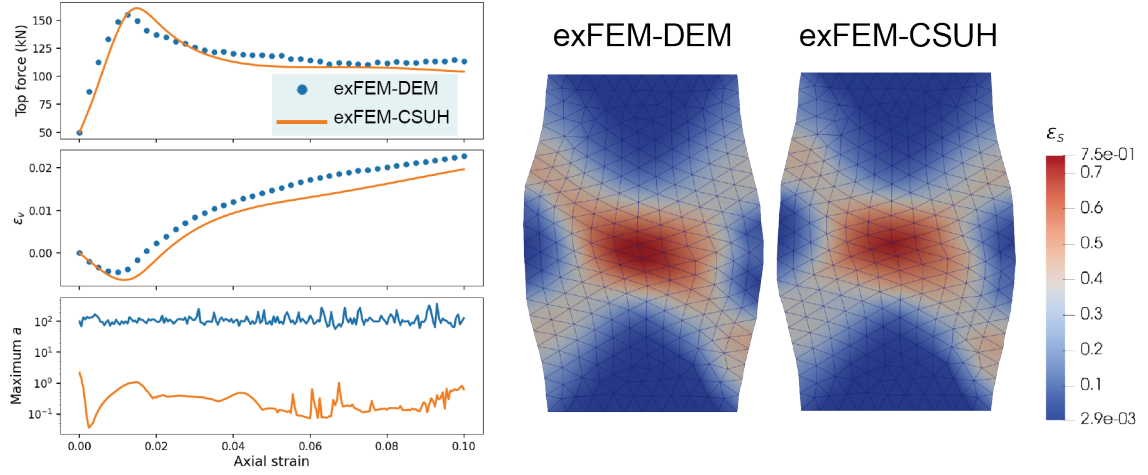


Figure 7.14: Macroscopic results of biaxial compression based on the optimised CSUH model

from this optimisation technique, based on error back-propagation, effectively replicate the macroscopic mechanical response of geotechnical materials.

Analyzing the constitutive behaviour of the optimised CSUH model at integration points, the outcomes are depicted in Fig. 7.15. The top row of the figure illustrates the divergence between the strain tensor Lode angle ( $\theta_\epsilon$ ) and the stress tensor Lode angle ( $\theta_\sigma$ ). The second line is the shear stress. Within FEM-DEM computations, a substantial and unstable discrepancy emerges between the principal directions of tensor  $\epsilon_{ij}$  and tensor  $\sigma_{ij}$ . In low-scale DEM calculations, stress exhibits fluctuations as particles within the aggregate experience events like collision or sliding. At point 0, the disparity between the principal directions of these two tensors remains minimal due to the material's deformation closely resembling elastic deformation. This can be elucidated through calculations involving elastic stress. Assuming isotropic linear elasticity, stress is computed as  $\sigma_{ij} = D_{ijkl}\epsilon_{kl}^e = K\epsilon_v^e\delta_{ij} + 2Gs_{ij}^e$  in cases of purely elastic deformation. Consequently, in cases of elastic deformation only, the principal direction of the elastic strain tensor coincides with that of stress. Thus, the deflection angle at point 0 remains modest.

Contrastingly, points 265 and 477 correspond to larger deflection angles. The shear stress curve illustrates a peak followed by a decline, indicating plastic deformation or the occurrence of damage. During such instances, the principal directions of the material's strain and stress tensors undergo significant deflection. Stress calculation at these points follows the aforementioned equation, yet plastic deformation transitions from  $\epsilon^p_{ij} = \mathbf{0}$  to a non-zero

tensor. Plastic strain is determined as  $d\epsilon_{ij}^p = d\lambda \frac{\partial g}{\partial \sigma_{ij}}$ . Consequently, the direction of plastic strain typically diverges from that of the elastic strain tensor  $\epsilon_{ij}^e$ . Hence, in cases of plastic strain, the principal axes of the strain and stress tensors no longer align. As observed in the figure, for points 265 and 477, the deflection angle remains small prior to the material's onset of plastic strain. However, after plastic strain initiates, the deflection angle increases significantly in both magnitude and prominence. The stress predictions of the optimised CSUH model closely resemble those obtained from FEM-DEM biaxial simulations. The optimised CSUH model adeptly captures the trend of deflection angle alterations.

At integration point 0, the shear stresses exhibit behaviour more akin to elasticity during loading. There is no plastic deformation in the CSUH model-based simulation at point 0. Despite the higher magnitudes of shear stresses at this point, they are overshadowed by the greater compressive stresses. This phenomenon is interpreted by the CSUH model as indicative of shear strengthening. This inference is supported by insights from the yield function, revealing that as the mean stress elevates, the shear stress on the yield surface also increases.

At integration point 477, the shear stress experiences a rapid decline to zero after reaching its peak. Within the DEM simulation, the calculation of shear stress follows the expression  $\sigma_{ij} = \frac{1}{V} \sum_c f_i^c l_j^c$ , where  $f_i^c$  represents the vector of contact forces, and  $l_j^c$  is the vector connecting the centres of the two spheres [125, 171]. In the context of the DEM simulation, when all contact forces become zero, the stress tensor of the particle assembly also becomes zero, indicating a shear failure within the particle assembly. This phenomenon is commonly observed in experiments involving shear damage of granular materials. The stress distribution observed at integration point 477 provides evidence that the optimised CSUH model is proficient in simulating this specific type of shear-induced damage encountered in granular materials.

Overall, at a macroscopic level, the optimised CSUH model demonstrates the capacity to closely replicate concentrated forces and macroscopic shear strain as observed in FEM-DEM simulations. At the integration points, the optimised CSUH model aligns with the FEM-DEM well in aspects of stress magnitude and stress-strain deviation angles. On a localised scale, the error of the optimised CSUH model is evident. However, this discrepancy diminishes on a macroscopic scale where it aligns well with the concentrated forces. This phenomenon can be attributed to the presence of substantial noise within the data. The optimised model adeptly identifies patterns within stress-strain data sequences at integra-

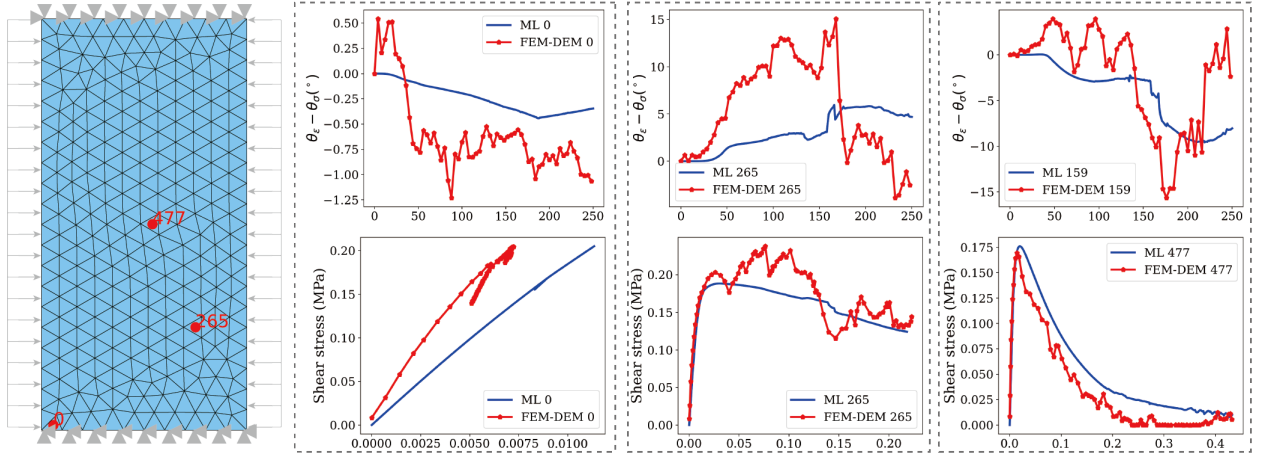


Figure 7.15: Comparison of the constitutive response at integration points of the biaxial compression simulation. The  $x$  coordinate is the strain in the vertical direction.

tion points, automatically filtering out the influence of such noise. The errors observed in stress-strain curves at integration points 7.15 primarily stem from noise interference. During the computation of the overall concentrated force, this noise tends to cancel each other out, resulting in a favourable match with FEM-DEM simulation outcomes concerning the magnitude of the overall concentrated force. This underscores the effectiveness of the optimised CSUH model.

To further validate the efficacy of the optimised CSUH model, we apply it to the assessment of a retaining wall scenario. The macroscopic structural analysis is portrayed in Fig. 7.16. The optimised CSUH model closely approximates the multi-scale FEM-DEM simulated results for the retaining wall. Particularly noteworthy is the fact that simulations founded on the optimised CSUH model yield maximum node acceleration values orders of magnitude lower than those obtained through multi-scale FEM-DEM simulations. The inherent limitations associated with the usage of a limited number of particles in low-scale DEM simulations give rise to evident interactions (e.g., friction, collision) between particles, exerting a substantial impact on stress outcomes. The subsequent oscillations in node acceleration, stemming from the fluctuating stress response in the explicit FEM solver, are effectively minimised through the optimised CSUH model. By deriving the constitutive relation under static conditions from the data, the optimised model achieves relatively reduced maximum node accelerations. This reduction reinforces heightened stability in explicit FEM simulations.



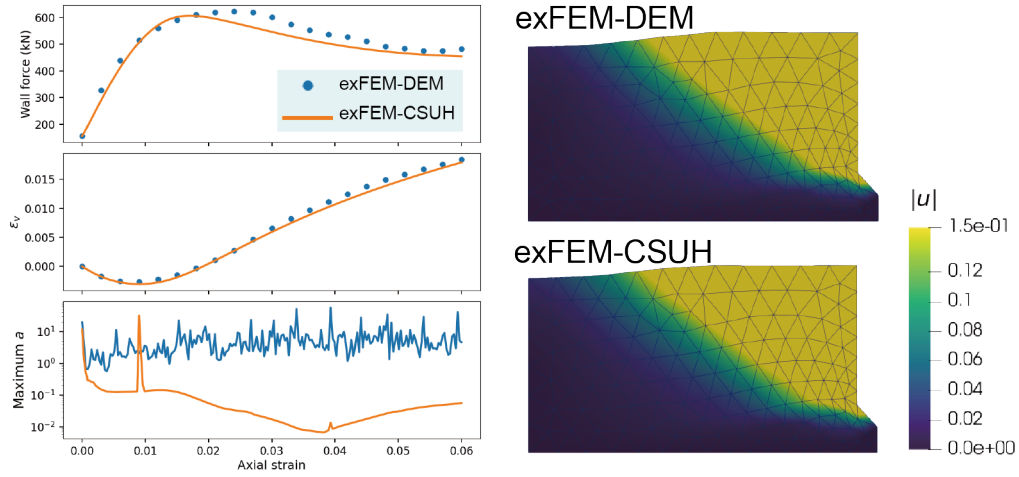


Figure 7.16: Macroscopic results of the optimised CSUH model in the retaining simulation

## 7.7 Concluding remarks

In this chapter, we construct an Enhanced IME model by integrating concepts from linear elasticity, the von Mises yield surface and exponential hardening. Additionally, we introduce the CSUH model and modified it based on the work of Pastor et al., specifically adjusting the transform stress component. The parameters of these constitutive models, which are implemented using the PyTorch framework, are optimised using the Adam optimisation algorithm.

To develop a versatile constitutive model, we incorporate traditional elastoplastic model constraints and devise a machine learning optimisation approach based on conventional constitutive frameworks. Following a methodology akin to training recurrent neural networks, stress-strain sequences are utilised as training data, and the error backpropagation technique is applied to optimise the model parameters.

We begin by demonstrating the capacity of recurrent structures, under the J2 ideal plastic constraint, to optimise model parameters in a manner reminiscent of the optimisation mechanisms employed by recurrent neural networks. Subsequently, the Enhanced IME model is trained using strain stress sequences derived from FEM-DEM simulations. The evolution of the model’s training parameters reveals fundamental physical principles, including non-linear elasticity and the mean stress-related yield surface, reflected within the training dataset. We then introduce the CSUH model, tailored for geomaterials, to improve the model’s ability to replicate FEM-DEM results. During the optimisation process, the CSUH model fine-tunes parameters such as increasing the slope of the elastic unloading line ( $\kappa$ ) to adjust the elastic

modulus, enhancing the over consolidation ratio OCR to amplify peak stress, and reducing the critical state stress ratio  $M$  to modulate the magnitude of critical shear stress.

The optimised CSUH model exhibits strong agreement both at the macroscopic and integration point levels with FEM-DEM simulations, thereby discerning patterns of compression hardening and shear failure from the dataset. Notably, FEM-DEM simulations exhibit fluctuations in deviatoric angles due to the stochastic nature of low-scale DEM simulations. This effect of chance is amplified by the insufficient number of particles in the assemblage. However, the optimised CSUH model adeptly captures these patterns of deviatoric angles.

The CSUH model approached as a guiding constraint for constructing surrogate models, is systematically tailored to align with the inherent constitutive patterns embedded in the FEM-DEM datasets. This approach produces a refined surrogate model that faithfully reproduces FEM-DEM simulation outcomes, thereby advancing precision and computational efficiency.

# Chapter 8

## Conclusion

Many scholars begin to introduce machine learning methods into computational mechanics in recent years. Due to the development of measurement and simulation techniques, a large amount of high-precision data can be used for the study of the data-driven constitutive modelling of rock-fill materials.

In this study, machine learning methods are introduced to learn and reproduce the constitutive knowledge. Machine learning constitutive models are established in different manners. They are subsequently implemented with the implicit and explicit FEM solvers. The network structure, training cost, prediction accuracy, generalisation, and error propagation and propagation are analysed. The main conclusions are summarised as follows:

- In geotechnical computations, first-principle modelling is computationally demanding. Multiscale simulations like FEM-DEM aim to boost accuracy and speed. However, as the number of CPUs increases in parallel computing, the efficiency gains slow down and reach a plateau due to the cost of inter-node communication.
- Recurrent neural networks, particularly those based on LSTM or GRU, are able to accurately capture and reproduce the memory effects observed in granular materials.
- Active learning resampling can identify the most informative data for network training from massive datasets. Nevertheless, the presence of noise within the dataset can significantly impact the efficacy.
- In BVP analysis, achieving high accuracy at specific integration points doesn't necessarily ensure overall simulation accuracy. Instead, errors that occur at one or multiple

integration points tend to accumulate and propagate, ultimately causing a decline in the accuracy of the entire calculation or even causing it to fail. Hence, it becomes more important to maintain an acceptable level of accuracy across all integration points, rather than solely focusing on improving the accuracy of select ones.

- Neural network-based constitutive modelling can extract constitutive knowledge directly from data, bypassing the need for phenomenological assumptions. However, achieving generalisation with this approach remains challenging. It is crucial to balance data-driven methods with physical priors/assumptions. By incorporating physics, a more generalised constitutive model can be developed. Furthermore, refining the physical priors according to both of the training process and constitutive knowledge can significantly improve the model's performance.

It is essential to strike a balance between data requirements, generalisation, and accuracy, which requires ongoing exploration to achieve optimal results. Data-driven methods hold the potential to address larger-scale and more complex engineering challenges, offering more efficient and reliable solutions in the field of geotechnical engineering.

# Publications and contributions to conferences

## Publications:

1. **Guan, S.**, Qu, T., Feng, Y. T., Ma, G., and Zhou, W. (2023). A machine learning-based multi-scale computational framework for granular materials. *Acta Geotechnica*, 18(4), 1681–1698.
2. **Guan, S.**, Feng, Y. T., Ma, G., Qu, T., Wang, M., and Zhou, W. (2023). An explicit FEM-NN framework and the analysis of error caused by NN-predicted stress. *Acta Geotechnica*.
3. **Guan, S.**, and Ranftl, S. (2023). A recurrent machine learning structure for few-shot constitutive model optimization: Application to Geomechanics. 1(June), 1–2.
4. **Guan, S.**, Zhang, X., Ranftl, S. and Qu T., A neural network-based material cell for elastoplasticity and its performance in FE analyses of boundary value problems, *International Journal of Plasticity*, under review.
5. Ma, G., **Guan, S.**, Wang, Q., Feng, Y. T., and Zhou, W. (2022). A predictive deep learning framework for path-dependent mechanical behavior of granular materials. *Acta Geotechnica*, 17(8), 3463–3478.
6. Qu, T., **Guan, S.**, Feng, Y. T., Ma, G., Zhou, W., and Zhao, J. (2023). Deep active learning for constitutive modelling of granular materials: From representative volume elements to implicit finite element modelling. *International Journal of Plasticity*, 164, 103576.
7. Wang, M., Qu, T., **Guan, S.**, Zhao, T., Liu, B., and Feng, Y. T. (2022). Data-driven strain–stress modelling of granular materials via temporal convolution neural network.

*Computers and Geotechnics*, 152(March).

8. Wang, M., Feng, Y., **Guan, S.**, and Qu, T., Multi-layer perceptron-based data-driven multiscale modelling of granular materials with a novel Frobenius norm-based internal variable, *Journal of Rock Mechanics and Geotechnical Engineering*, submitted.
9. Qu, T., Zhao, J., **Guan, S.**, and Feng, Y., Data-driven multiscale modelling of granular materials via transfer learning, *International Journal of Plasticity*, under review.
10. Yao, F. hai, **Guan, S.**, Yang, H., Chen, Y., Qiu, H. feng, Ma, G., and Liu, Q. wen. (2019). Long-term deformation analysis of Shuibuya concrete face rockfill dam based on response surface method and improved genetic algorithm. *Water Science and Engineering*, 12(3), 196–204.
11. Wang, M., Liang, L. xun, **Shaoheng Guan**, Ma, G., Lai, Z. qiang, Niu, X. qiang, Zhang, S. fan, Tian, W. xiang, and Zhou, W. (2023). Experimental and numerical investigation of the collapse of binary mixture of particles with different densities. *Powder Technology*, 415.

#### **Contributions to conferences:**

1. **Poster:** "An Energy-based discrete element method": 9th International Conference on Discrete Element Methods (DEM9), taking place from September 17, 2023 to September 21, 2023 in Erlangen, Germany.
2. **Presentation:** "A recurrent machine learning structure for few-shot constitutive model optimization: Application to Geomechanics": The ECCOMAS Young Investigators Conference YIC2023, Faculty of Engineering of the University of Porto, Porto, 19-21 June 2023
3. **Presentation:** "NN-based constitutive model implemented in explicit FEM solver and prediction error analysis: elastoplastic models in Geomechanics": 2023 Annual Conference of the UK Association for Computational Mechanics, The University of Warwick, UK, 19-22 April 2023
4. **Presentation:** "Machine learning based multi-scale computation framework for granular materials": 2022 Annual Conference of the UK Association for Computational Mechanics, The University of Nottingham, UK, 20-22 April 2022
5. **Presentation:** "A predictive deep learning framework for path-dependent mechanical behavior of granular materials". The Fifth National Conference on Computational

Mechanics of Granular Materials Wuhan University, China, 26-28, April 2021.

# Bibliography

- [1] T. L. Youd, “Factors Controlling Maximum and Minimum Densities of Sands.,” *ASTM Special Technical Publication*, pp. 98–112, 1972.
- [2] J. C. Lopera Perez, C. Y. Kwok, C. O’Sullivan, X. Huang, and K. J. Hanley, “Assessing the quasi-static conditions for shearing in granular media within the critical state soil mechanics framework,” *Soils and Foundations*, vol. 56, no. 1, pp. 152–159, 2016.
- [3] H. A. Meier, P. Steinmann, and E. Kuhl, “On the multiscale computation of confined granular media,” *Computational Methods in Applied Sciences*, vol. 14, pp. 121–133, 2009.
- [4] N. Guo and J. Zhao, “A coupled FEM/DEM approach for hierarchical multiscale modelling of granular media,” *International Journal for Numerical Methods in Engineering*, no. 1, pp. 789–818, 2014.
- [5] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu, “Sobolev training for neural networks,” *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 4279–4288, 2017.
- [6] Y. Yao, Y. Tian, A. Zhou, and D. Sun, “Unified hardening law for soils and its construction,” *Zhongguo Kexue Jishu Kexue/Scientia Sinica Technologica*, vol. 49, no. 1, pp. 26–34, 2019.
- [7] A. Zhou and Y. Yao, “Revising the unified hardening model by using a smoothed Hvorslev envelope,” *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 10, no. 4, pp. 778–790, 2018.
- [8] P. V. Ladeo and P. A. Boppli, “Relative density effects on drained sand behavior at high pressures,” *Soils and Foundations*, vol. 45, no. 1, pp. 1–13, 2005.



- [9] P. Richard, M. Nicodemi, R. Delannay, P. Ribière, and D. Bideau, “Slow relaxation and compaction of granular systems,” 2005.
- [10] G. H. DARWIN, “on the Horizontal Thrust of a Mass of Sand.,” *Minutes of the Proceedings of the Institution of Civil Engineers*, vol. 71, no. 1883, pp. 350–378, 1883.
- [11] O. Reynolds, “LVII. On the dilatancy of media composed of rigid particles in contact. With experimental illustrations ,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 20, no. 127, pp. 469–481, 1885.
- [12] Z. Gao and J. Zhao, “Constitutive Modeling of Anisotropic Sand Behavior in Monotonic and Cyclic Loading,” *Journal of Engineering Mechanics*, vol. 141, no. 8, p. 04015017, 2015.
- [13] W. Wu, G. Ma, W. Zhou, D. Wang, and X. Chang, “Force transmission and anisotropic characteristics of sheared granular materials with rolling resistance,” *Granular Matter*, vol. 21, no. 4, 2019.
- [14] T. Qu, M. Wang, and Y. Feng, “Applicability of discrete element method with spherical and clumped particles for constitutive study of granular materials,” *Journal of Rock Mechanics and Geotechnical Engineering*, no. xxxx, 2021.
- [15] P. A. Cundall and O. D. Strack, “A discrete numerical model for granular assemblies,” *Geotechnique*, vol. 29, no. 1, pp. 47–65, 1979.
- [16] P. A. Cundall, “Distinct element models of rock and soil structure,” *In Analytical and Computational Methods in Engineering Rock Mechanics*, p. 129–163, 1987.
- [17] R. Hart, P. A. Cundall, and J. Lemos, “Formulation of a three-dimensional distinct element model-Part II. Mechanical calculations for motion and interaction of a system composed of many polyhedral blocks,” *International Journal of Rock Mechanics and Mining Sciences and*, vol. 25, pp. 117–125, 6 1988.
- [18] Z. X. Yang and Y. Wu, “Critical State for Anisotropic Granular Materials: A Discrete Element Perspective,” *International Journal of Geomechanics*, vol. 17, no. 2, p. 04016054, 2017.
- [19] G. Ma, R. A. Regueiro, W. Zhou, Q. Wang, and J. Liu, “Role of particle crushing on particle kinematics and shear banding in granular materials,” *Acta Geotechnica*, vol. 13, pp. 601–618, 6 2018.

- [20] S. Zhao, T. M. Evans, and X. Zhou, “Shear-induced anisotropy of granular materials with rolling resistance and particle shape effects,” *International Journal of Solids and Structures*, vol. 150, pp. 268–281, 2018.
- [21] G. Ma, W. Zhou, X. L. Chang, T. T. Ng, and L. F. Yang, “Formation of shear bands in crushable and irregularly shaped granular materials and the associated microstructural evolution,” *Powder Technology*, vol. 301, pp. 118–130, 2016.
- [22] W. Liu, H. Ke, J. Xie, H. Tan, G. Luo, B. Xu, and G. Abakari, “Constitutive modelling of natural sands using a deep learning approach accounting for particle shape effects,” *Geoderma Regional*, p. e00484, 2020.
- [23] R. J. Bathurst and L. Rothenburg, “Observations on stress-force-fabric relationships in idealized granular materials,” *Mechanics of Materials*, vol. 9, no. 1, pp. 65–80, 1990.
- [24] N. Guo and J. Zhao, “The signature of shear-induced anisotropy in granular media,” *Computers and Geotechnics*, vol. 47, pp. 1–15, 2013.
- [25] G. Ma, Y. Zhang, W. Zhou, T. T. Ng, Q. Wang, and X. Chen, “The effect of different fracture mechanisms on impact fragmentation of brittle heterogeneous solid,” *International Journal of Impact Engineering*, vol. 113, no. December, pp. 132–143, 2018.
- [26] W. Zhou, X.-l. Chang, and W. Yuan, “Combined FEM/DEM Modeling of Triaxial Compression Tests for Rockfills with Polyhedral Particles,” vol. 14, no. 4, pp. 1–12, 2014.
- [27] W. Zhou, J. Liu, G. Ma, and X. Chang, “Three-dimensional DEM investigation of critical state and dilatancy behaviors of granular materials,” *Acta Geotechnica*, vol. 12, no. 3, pp. 527–540, 2017.
- [28] R. I. Borja and J. R. Wren, “Micromechanics of granular media Part I: Generation of overall constitutive equation for assemblies of circular disks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 127, no. 1-4, pp. 13–36, 1995.
- [29] J. R. Wren and R. I. Borja, “Micromechanics of granular media Part II: Overall tangential moduli and localization model for periodic assemblies of circular disks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 141, no. 3-4, pp. 221–246, 1997.
- [30] L. Rothenburg and R. J. Bathurst, “Discussion: Analytical study of induced anisotropy in idealized granular material,” *Géotechnique*, vol. 40, no. 4, pp. 665–667, 1990.

- [31] J. Shi, P. Guo, and D. Stolle, “Noncoaxiality between Fabric and Stress in Two-Dimensional Granular Materials,” *Journal of Engineering Mechanics*, vol. 144, no. 9, p. 04018092, 2018.
- [32] E. Azéma, S. Linero, N. Estrada, and A. Lizcano, “Shear strength and microstructure of polydisperse packings: The effect of size span and shape of particle size distribution,” *Physical Review E*, vol. 96, no. 2, pp. 1–10, 2017.
- [33] T. T. Ng, W. Zhou, G. Ma, and X. L. Chang, “Macroscopic and microscopic behaviors of binary mixtures of different particle shapes and particle sizes,” *International Journal of Solids and Structures*, vol. 135, pp. 74–84, 2018.
- [34] K. Iwashita and M. Oda, “Micro-deformation mechanism of shear banding process based on modified distinct element method,” *Powder Technology*, vol. 109, pp. 192–205, 4 2000.
- [35] M. Oda, “Fabric Tensor for Discontinuous Geological Materials.,” *Soils and Foundations*, vol. 22, no. 4, pp. 96–108, 1982.
- [36] K. Ken-Ichi, “Distribution of directional data and fabric tensors,” *International Journal of Engineering Science*, vol. 22, pp. 149–164, 1 1984.
- [37] X. S. Li, Y. F. Dafalias, and Z. L. Wang, “State-dependent dilatancy in critical-state constitutive modelling of sand,” *Canadian Geotechnical Journal*, vol. 36, no. 4, pp. 599–611, 1999.
- [38] X. S. Li and Y. F. Dafalias, “Anisotropic Critical State Theory: Role of Fabric,” *Journal of Engineering Mechanics*, vol. 138, no. 3, pp. 263–275, 2012.
- [39] R. L. Kondner, “Hyperbolic Stress-Strain Response: Cohesive Soils,” *Journal of the Soil Mechanics and Foundations Division*, vol. 89, no. 1, pp. 115–143, 1963.
- [40] J. M. Duncan and C. Y. Chang, “Nonlinear analysis of stress and strain in soils,” in *Geotechnical Special Publication*, no. 118 II, pp. 1347–1371, 1970.
- [41] K. H. Roscoe, A. N. Schofield, and C. P. Wroth, “On the yielding of soils,” *Geotechnique*, vol. 8, no. 1, pp. 22–53, 1958.
- [42] K.H. Roscoe and J. B. Burland, “On the generalized stress-strain behavior of “wet” clay,” *Journal of Terramechanics*, vol. 7, no. 2, pp. 107–108, 1970.
- [43] A. Schofield and P. Wroth, “Critical State Soil Mechanics,” tech. rep.

- [44] D. M. Wood, *Soil behaviour and critical state soil mechanics*. 1990.
- [45] Y. P. Yao, “Advanced UH models for soils,” *Yantu Gongcheng Xuebao/Chinese Journal of Geotechnical Engineering*, vol. 37, no. 2, pp. 193–217, 2015.
- [46] Y. P. Yao, W. Hou, and A. N. Zhou, “UH model: Three-dimensional unified hardening model for overconsolidated clays,” *Geotechnique*, vol. 59, no. 5, pp. 451–469, 2009.
- [47] Y. Yao, Z. Gao, J. Zhao, and Z. Wan, “Modified UH Model: Constitutive Modeling of Overconsolidated Clays Based on a Parabolic Hvorslev Envelope,” *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 138, no. 7, pp. 860–868, 2012.
- [48] H. B. Poorooshasb, I. Holubec, and A. N. Sherbourne, “Yielding and Flow of Sand in Triaxial Compression: Part I,” *Canadian Geotechnical Journal*, vol. 3, no. 4, pp. 179–190, 1966.
- [49] M. G. Jefferies, “Nor-Sand: A simple critical state model for sand,” *Geotechnique*, vol. 43, no. 1, pp. 91–103, 1993.
- [50] M. Jefferies, “NORSAND : DESCRIPTION , CALIBRATION , VALIDATION AND APPLICATIONS DESCRIPTION , CALIBRATION , VALIDATION AND Dawn Shuttle & Michael Jefferies,” no. January 2005, 2016.
- [51] X. S. Li and Y. F. Dafalias, “A constitutive, framework for anisotropic sand including non-proportional loading,” *Geotechnique*, vol. 54, no. 1, pp. 41–55, 2004.
- [52] X. S. Li and Y. F. Dafalias, “Dilatancy for cohesionless soils,” *Geotechnique*, vol. 50, no. 4, pp. 449–460, 2000.
- [53] H. S. Yu, “CASM: a unified state parameter model for clay and sand,” *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 22, no. 8, pp. 621–653, 1998.
- [54] J. M. Pestana and A. J. Whittle, “Formulation of a unified constitutive model for clays and sands,” *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 23, no. 12, pp. 1215–1243, 1999.
- [55] Y. P. Yao, D. A. Sun, and H. Matsuoka, “A unified constitutive model for both clay and sand with hardening parameter independent on stress path,” *Computers and Geotechnics*, vol. 35, no. 2, pp. 210–222, 2008.

- [56] A. Asaoka, T. Noda, E. Yamada, K. Kaneda, and M. Nakano, “An elasto-plastic description of two distinct volume change mechanisms of soils,” *Soils and Foundations*, vol. 42, no. 5, pp. 47–57, 2002.
- [57] Y. P. Yao, L. Liu, T. Luo, Y. Tian, and J. M. Zhang, “Unified hardening (UH) model for clays and sands,” *Computers and Geotechnics*, vol. 110, no. March, pp. 326–343, 2019.
- [58] H. Matsuoka, “on the Significance of the “ Spatial Mobilized Plane” .,” *Soils and Foundations*, vol. 16, no. 1, pp. 91–100, 1976.
- [59] O. C. Zienkiewicz and Z. Mroz, “Generalized Plasticity Formulation and Applications To Geomechanics.,” no. April, pp. 655–679, 1984.
- [60] Z. Mroz and O. C. Zienkiewicz, “Uniform Formulation of constitutive equations for clays and sands,” no. April, pp. 655–679, 1984.
- [61] M. Pastor, O. C. Zienkiewicz, and A. H. C. Chan, “Theme / Feature Paper Generalized Plasticity and the Modelling of,” *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 14, no. March 1988, pp. 151–190, 1990.
- [62] Y. Xiao, H. Liu, Q. Chen, L. Long, and J. Xiang, “Evolution of particle breakage and volumetric deformation of binary granular soils under impact load,” *Granular Matter*, vol. 19, no. 4, pp. 1–10, 2017.
- [63] Y. Xiao, H. Liu, Q. Chen, Q. Ma, Y. Xiang, and Y. Zheng, “Particle breakage and deformation of carbonate sands with wide range of densities during compression loading process,” *Acta Geotechnica*, vol. 12, no. 5, pp. 1177–1184, 2017.
- [64] Y. Xiao, L. Long, T. Matthew Evans, H. Zhou, H. Liu, and A. W. Stuedlein, “Effect of Particle Shape on Stress-Dilatancy Responses of Medium-Dense Sands,” *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 145, no. 2, p. 04018105, 2019.
- [65] R. Kawamoto, E. Andò, G. Viggiani, and J. E. Andrade, “All you need is shape: Predicting shear banding in sand with LS-DEM,” *Journal of the Mechanics and Physics of Solids*, vol. 111, pp. 375–392, 2018.
- [66] J. E. Andrade and X. Tu, “Multiscale framework for behavior prediction in granular media,” *Mechanics of Materials*, vol. 41, no. 6, pp. 652–669, 2009.

- [67] J. E. Andrade, C. F. Avila, S. A. Hall, N. Lenoir, and G. Viggiani, “Multiscale modeling and characterization of granular matter: From grain kinematics to continuum mechanics,” *Journal of the Mechanics and Physics of Solids*, vol. 59, no. 2, pp. 237–250, 2011.
- [68] M. Nitka, G. Combe, C. Dascalu, and J. Desrues, “Two-scale modeling of granular materials: A DEM-FEM approach,” *Granular Matter*, vol. 13, no. 3, pp. 277–281, 2011.
- [69] P. Guo and W. C. Li, “Development and implementation of Duncan-Chang constitutive model in GeoStudio2007,” *Procedia Engineering*, vol. 31, no. December, pp. 395–402, 2012.
- [70] N. Guo and J. Zhao, “Parallel hierarchical multiscale modelling of hydro-mechanical problems for saturated granular soils,” *Computer Methods in Applied Mechanics and Engineering*, vol. 305, pp. 37–61, 2016.
- [71] J. D. Zhao and N. Guo, “Bridging the micro and macro for granular media: A computational multi-scale paradigm,” *Geomechanics from Micro to Macro - Proceedings of the TC105 ISSMGE International Symposium on Geomechanics from Micro to Macro, IS-Cambridge 2014*, vol. 2, pp. 747–752, 2015.
- [72] H. A. Meier, P. Steinmann, and E. Kuhl, “On the multiscale computation of confined granular media,” *Computational Methods in Applied Sciences*, vol. 14, pp. 121–133, 2009.
- [73] K. Karapiperis, L. Stainier, M. Ortiz, and J. E. Andrade, “Data-Driven multiscale modeling in mechanics,” *Journal of the Mechanics and Physics of Solids*, vol. 147, p. 104239, 2021.
- [74] W. Liang and J. Zhao, “Multiscale modeling of large deformation in geomechanics,” *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 43, no. 5, pp. 1080–1114, 2019.
- [75] W. Liang, H. Wu, S. Zhao, W. Zhou, and J. Zhao, “Scalable three-dimensional hybrid continuum-discrete multiscale modeling of granular media,” *International Journal for Numerical Methods in Engineering*, vol. 123, no. 12, pp. 2872–2893, 2022.
- [76] Y. Lian, F. Zhang, Y. Liu, and X. Zhang, “Material point method and its applications,” *Advances in Mechanics*, vol. 43, no. 2, pp. 237–264, 2013.

- [77] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang, “14. MLS-MPM supplementary document,” *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–5, 2018.
- [78] V. P. Nguyen, “Material point method: basics and applications,” no. May, p. 207, 2014.
- [79] G. Remmerswaal, M. Bolognin, P. J. Vardon, M. A. Hicks, and A. Rohe, “Implementation of non-trivial boundary conditions in MPM for geotechnical applications,” *Proceedings of the 2nd International Conference on the Material Point Method for Modelling Soil-Water-Structure Interaction*, no. November, pp. 61–67, 2019.
- [80] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [81] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Van Den Driessche, T. Graepel, and D. Hassabis, “Mastering the game of Go without human knowledge,” *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.
- [82] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of Artificial Intelligence Research*, vol. 4, no. 19, pp. 237–285, 1996.
- [83] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. Kohli, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, “Highly accurate protein structure prediction with AlphaFold,” *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.
- [84] M. Abadi and J. Z. A. J. M. S. G. M. M. J. R. S. D. G. B. P. V. P. X. Barham, PaulChen, “TensorFlow: A System for Large-Scale Machine Learning,” *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.

- [85] D. Mazza and M. Pagani, “Automatic differentiation in PCF,” *Proceedings of the ACM on Programming Languages*, vol. 5, no. POPL, pp. 1–4, 2021.
- [86] T. K. Ho, “Random Decision Forests,” in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, pp. 278–282, 1995.
- [87] M. Seeger, *Gaussian processes for machine learning.*, vol. 14. 2004.
- [88] T. Michalis K., “Variational Learning of Inducing Variables in Sparse Gaussian Processes,” *Russian Journal of Genetics*, 2009.
- [89] R. B. Gramacy, “LaGP: Large-scale spatial modeling via local approximate Gaussian processes in R,” *Journal of Statistical Software*, vol. 72, 2016.
- [90] J. Ghaboussi, J. H. Garrett, and X. Wu, “Knowledge-Based Modeling of Material Behavior with Neural Networks,” *Journal of Engineering Mechanics*, vol. 117, no. 1, pp. 132–153, 1991.
- [91] J. Ghaboussi and D. E. Sidarta, “New Nested Adaptive Neural Networks (NANN) for Constitutive Modeling,” *Computers and Geotechnics*, vol. 22, no. 1, pp. 29–52, 1998.
- [92] 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*, vol. 42, no. 1, pp. 105–126, 1998.
- [93] D. E. Sidarta and J. Ghaboussi, “Constitutive Modeling of Geomaterials from Non-uniform Material Tests,” *Computers and Geotechnics*, vol. 22, no. 1, pp. 53–71, 1998.
- [94] C. Hoerig, J. Ghaboussi, and M. F. Insana, “Cartesian Neural Network Constitutive Models for Data-driven Elasticity Imaging,” *arXiv*, pp. 1–22, 2018.
- [95] S. Jung and J. Ghaboussi, “Neural network constitutive model for rate-dependent materials,” *Computers and Structures*, vol. 84, no. 15-16, pp. 955–963, 2006.
- [96] Y. M. Hashash, S. Jung, and J. Ghaboussi, “Numerical implementation of a neural network based material model in finite element analysis,” *International Journal for Numerical Methods in Engineering*, vol. 59, no. 7, pp. 989–1005, 2004.
- [97] B. Kou, Y. Cao, J. Li, C. Xia, Z. Li, H. Dong, A. Zhang, J. Zhang, W. Kob, and Y. Wang, “Granular materials flow like complex fluids,” *Nature*, vol. 551, no. 7680, pp. 360–363, 2017.



- [98] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, and M. A. Bessa, “Deep learning predicts path-dependent plasticity,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, no. 52, pp. 26414–26420, 2019.
- [99] P. Zhang, Z. Y. Yin, Y. Zheng, and F. P. Gao, “A LSTM surrogate modelling approach for caisson foundations,” *Ocean Engineering*, vol. 204, no. April, p. 107263, 2020.
- [100] P. Zhang, Z. Y. Yin, Y. F. Jin, and X. F. Liu, “Modelling the mechanical behaviour of soils using machine learning algorithms with explicit formulations,” *Acta Geotechnica*, vol. 4, no. 3, 2021.
- [101] G. Ma, S. Guan, Q. Wang, Y. T. Feng, and W. Zhou, “A predictive deep learning framework for path-dependent mechanical behavior of granular materials,” *Acta Geotechnica*, vol. 0123456789, 2022.
- [102] T. Qu, Y. Feng, M. Wang, T. Zhao, and S. Di, “Constitutive Relations of Granular Materials By Integrating Micromechanical Knowledge With Deep Learning,” *Lixue Xuebao/Chinese Journal of Theoretical and Applied Mechanics*, vol. 53, no. 9, pp. 2404–2415, 2021.
- [103] C. Bonatti and D. Mohr, “One for all: Universal material model based on minimal state-space neural networks,” *Science Advances*, vol. 7, no. 26, pp. 1–9, 2021.
- [104] M. Wang, T. Qu, S. Guan, T. Zhao, B. Liu, and Y. T. Feng, “Data-driven strain–stress modelling of granular materials via temporal convolution neural network,” *Computers and Geotechnics*, vol. 152, no. March, 2022.
- [105] F. Ghavamian and A. Simone, “Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network,” *Computer Methods in Applied Mechanics and Engineering*, vol. 357, p. 112594, 2019.
- [106] H. J. Logarzo, G. Capuano, and J. J. Rimoli, “Smart constitutive laws: Inelastic homogenization through machine learning,” *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113482, 2021.
- [107] K. Xu, D. Z. Huang, and E. Darve, “Learning constitutive relations using symmetric positive definite neural networks,” *Journal of Computational Physics*, vol. 428, p. 110072, 2021.

- [108] D. Z. Huang, K. Xu, C. Farhat, and E. Darve, “Learning constitutive relations from indirect observations using deep neural networks,” *Journal of Computational Physics*, vol. 416, p. 109491, 2020.
- [109] S. Tang, G. Zhang, H. Yang, Y. Li, W. K. Liu, and X. Guo, “MAP123: A data-driven approach to use 1D data for 3D nonlinear elastic materials modeling,” *Computer Methods in Applied Mechanics and Engineering*, vol. 357, no. September, p. 112587, 2019.
- [110] S. Tang, Y. Li, H. Qiu, H. Yang, S. Saha, S. Mojumder, W. K. Liu, and X. Guo, “MAP123-EP: A mechanistic-based data-driven approach for numerical elastoplastic analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 364, no. March, p. 112955, 2020.
- [111] S. Tang, H. Yang, H. Qiu, M. Fleming, W. K. Liu, and X. Guo, “MAP123-EPF: A mechanistic-based data-driven approach for numerical elastoplastic modeling at finite strain,” *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113484, 2021.
- [112] H. Yang, X. Guo, S. Tang, and W. K. Liu, “Derivation of heterogeneous material laws via data-driven principal component expansions,” *Computational Mechanics*, vol. 64, no. 2, pp. 365–379, 2019.
- [113] R. Eggersmann, T. Kirchdoerfer, S. Reese, L. Stainier, and M. Ortiz, “Model-Free Data-Driven inelasticity,” *Computer Methods in Applied Mechanics and Engineering*, vol. 350, pp. 81–99, 2019.
- [114] S. Conti, S. Müller, and M. Ortiz, “Data-Driven Problems in Elasticity,” *Archive for Rational Mechanics and Analysis*, vol. 229, no. 1, pp. 79–123, 2018.
- [115] T. Kirchdoerfer and M. Ortiz, “Data-driven computing in dynamics,” *International Journal for Numerical Methods in Engineering*, vol. 113, no. 11, pp. 1697–1710, 2018.
- [116] J. N. Fuhg, M. Marino, and N. Bouklas, “Local approximate Gaussian process regression for data-driven constitutive models: development and comparison with neural networks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 388, p. 114217, 2022.
- [117] J. N. Fuhg, A. Fau, N. Bouklas, and M. Marino, “Elasto-plasticity with convex model-data-driven yield functions,” no. March, 2022.

- [118] N. N. Vlassis and W. C. Sun, “Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening,” *Computer Methods in Applied Mechanics and Engineering*, vol. 377, p. 113695, 2021.
- [119] K. Wang and W. C. Sun, “Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning,” *Computer Methods in Applied Mechanics and Engineering*, vol. 346, pp. 216–241, 2019.
- [120] K. Wang, W. C. Sun, and Q. Du, “A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with AI-guided experimentation,” *Computational Mechanics*, vol. 64, no. 2, pp. 467–499, 2019.
- [121] G. Guida, I. Einav, B. Marks, and F. Casini, “Linking micro grainsize polydispersity to macro porosity,” *International Journal of Solids and Structures*, vol. 187, no. xxxx, pp. 75–84, 2020.
- [122] “Contact models of liggghts.” [https://www.cfdem.com/media/DEM/docu/Section\\_gran\\_models.html?highlight=contact](https://www.cfdem.com/media/DEM/docu/Section_gran_models.html?highlight=contact), 2019.
- [123] C. Kloss, C. Goniva, A. Hager, S. Amberger, and S. Pirker, “Models, algorithms and validation for opensource DEM and CFD-DEM,” *Progress in Computational Fluid Dynamics*, vol. 12, no. 2-3, pp. 140–152, 2012.
- [124] I. Einav, “Breakage mechanics-Part I: Theory,” *Journal of the Mechanics and Physics of Solids*, vol. 55, no. 6, pp. 1274–1297, 2007.
- [125] J. Christoffersen, M. M. Mehrabadi, and S. Nemat-Nasser, “A micromechanical description of granular material behavior,” *Journal of Applied Mechanics, Transactions ASME*, vol. 48, no. 2, pp. 339–344, 1981.
- [126] A. Jensen, K. Fraser, and G. Laird, “Improving the precision of discrete element simulations through calibration models,” in *13th International LS-DYNA users conference*, pp. 1–12, 2014.
- [127] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [128] P. Conti, M. Guo, A. Manzoni, and J. S. Hesthaven, “Multi-fidelity surrogate modeling using long short-term memory networks,” 2022.

- [129] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” *Proceedings of the IEEE Conference on Decision and Control*, vol. 12, pp. 5442–5444, 2012.
- [130] T. Tieleman and G. Hinton, “Divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw,” *Mach. Learn.*, vol. 6, pp. 26–31, 2012.
- [131] G. Raskutti, M. J. Wainwright, and B. Yu, “Early stopping and non-parametric regression: An optimal data-dependent stopping rule,” *Journal of Machine Learning Research*, vol. 15, pp. 335–366, 2014.
- [132] C. C. Wang, “Stress relaxation and the principle of fading memory,” *Archive for Rational Mechanics and Analysis*, vol. 18, no. 2, pp. 117–126, 1965.
- [133] M. Oeser and S. Freitag, “Modeling of materials with fading memory using neural networks,” *International Journal for Numerical Methods in Engineering*, vol. 78, no. 7, pp. 843–862, 2009.
- [134] D. D. Lewis and J. Catlett, “Heterogeneous uncertainty sampling for supervised learning,” in *Machine Learning Proceedings 1994* (W. W. Cohen and H. Hirsh, eds.), pp. 148–156, San Francisco (CA): Morgan Kaufmann, 1994.
- [135] B. Settles, “Active Learning Literature Survey,” *Materials Letters*, vol. 65, no. 5, pp. 854–856, 2011.
- [136] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pp. 287–294, 1992.
- [137] R. Burbidge, J. Rowland, and R. King, “Active Learning for Regression Based on Query by Committee,” *Engineering and Automated Learning*, vol. 4881, pp. 209–218, 2007.
- [138] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method for Solid and Structural Mechanics*. 2014.
- [139] M. H. A. and K. E., “Toward multiscale computation of confined granular media,” 2008.

- [140] Z. Huang, Y. Tian, C. Li, G. Lin, L. Wu, Y. Wang, and H. Jiang, “Data-driven automated discovery of variational laws hidden in physical systems,” *Journal of the Mechanics and Physics of Solids*, vol. 137, 4 2020.
- [141] M. Y. Li, E. Grant, and T. L. Griffiths, “Gaussian process surrogate models for neural networks,” pp. 1–20, 2022.
- [142] D. Huang, J. N. Fuhg, C. Weißenfels, and P. Wriggers, “A machine learning based plasticity model using proper orthogonal decomposition,” *Computer Methods in Applied Mechanics and Engineering*, vol. 365, p. 113008, 2020.
- [143] T. Kirchdoerfer and M. Ortiz, “Data-driven computational mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 304, pp. 81–101, 2016.
- [144] N. Vlassis, R. Ma, and W. Sun, “Geometric deep learning for computational mechanics Part I: Anisotropic Hyperelasticity,” pp. 1–36, 2020.
- [145] T. Qu, Y. T. Feng, T. Zhao, and M. Wang, “Calibration of linear contact stiffnesses in discrete element models using a hybrid analytical-computational framework,” *Powder Technology*, vol. 356, pp. 795–807, 2019.
- [146] R. Burbidge, J. J. Rowland, and R. D. King, “Active learning for regression based on query by committee,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4881 LNCS, pp. 209–218, 2007.
- [147] S. Guan, T. Qu, Y. Feng, and G. Ma, “A machine learning-based multi-scale computational framework for granular materials,” *Acta Geotechnica*, vol. 0123456789, 2022.
- [148] M. Pastor, O. C. Zienkiewicz, and A. H. C. Chan, “Generalized Plasticity and the Modelling of soil behaviour,” *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 14, no. March 1988, pp. 151–190, 1990.
- [149] T. Qu, S. Di, Y. T. Feng, M. Wang, and T. Zhao, “Towards data-driven constitutive modelling for granular materials via micromechanics-informed deep learning,” *International Journal of Plasticity*, vol. 144, 2021.
- [150] P. Zhang, Z. Y. Yin, and Y. F. Jin, “Machine Learning-Based Modelling of Soil Properties for Geotechnical Design: Review, Tool Development and Comparison,” *Archives of Computational Methods in Engineering*, vol. 29, no. 2, pp. 1229–1245, 2022.

- [151] M. Lefik and B. A. Schrefler, “Artificial neural network as an incremental non-linear constitutive model for a finite element code,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 28-30, pp. 3265–3283, 2003.
- [152] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” pp. 1–9, 2014.
- [153] Q. Guan, Z. Yang, N. Guo, and Z. Hu, “Finite element geotechnical analysis incorporating deep learning-based soil model,” *Computers and Geotechnics*, vol. 154, no. November 2022, p. 105120, 2023.
- [154] I. B. Rocha, P. Kerfriden, and F. P. van der Meer, “On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning,” *Journal of Computational Physics: X*, vol. 9, p. 100083, 2021.
- [155] J. N. Fuhg, M. Marino, and N. Bouklas, “Local approximate Gaussian process regression for data-driven constitutive models: development and comparison with neural networks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 388, p. 114217, 2022.
- [156] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.
- [157] J. N. Fuhg, A. Fau, N. Bouklas, and M. Marino, “Elasto-plasticity with convex model-data-driven yield functions,” no. March, 2022.
- [158] D. P. Jang, P. Fazily, and J. W. Yoon, “Machine learning-based constitutive model for J2- plasticity,” *International Journal of Plasticity*, vol. 138, no. December 2020, p. 102919, 2021.
- [159] S. Wang, X. Yu, and P. Perdikaris, “When and why PINNs fail to train: A neural tangent kernel perspective,” *Journal of Computational Physics*, vol. 449, p. 110768, 2022.
- [160] S. Ranftl, “A connection between probability, physics and neural networks,” 2022.
- [161] C. Bonatti and D. Mohr, “On the importance of self-consistency in recurrent neural network models representing elasto-plastic solids,” *Journal of the Mechanics and Physics of Solids*, vol. 158, no. June 2021, p. 104697, 2022.

- [162] S. Ranftl, “A connection between probability, physics and neural networks,” *Physical Sciences Forum*, vol. 5, no. 1, 2022.
- [163] J. Abbasi and P. Østebø Andersen, “Physical activation functions (pafs): An approach for more efficient induction of physics into physics-informed neural networks (pinns),” 2022.
- [164] A. Zhang and D. Mohr, “Using neural networks to represent von Mises plasticity with isotropic hardening,” *International Journal of Plasticity*, vol. 132, no. February, p. 102732, 2020.
- [165] C. E. Rasmussen, C. K. Williams, *et al.*, *Gaussian processes for machine learning*, vol. 1. Springer, 2006.
- [166] N.-H. Kim. <https://mae.ufl.edu/nkim/egm6352/Chap4.pdf>.
- [167] M. Ristinmaa, “Consistent stiffness matrix in fe calculations of elasto-plastic bodies,” *Computers Structures*, vol. 53, no. 1, pp. 93–103, 1994.
- [168] Y. P. Yao, W. Hou, and A. N. Zhou, “UH model: Three-dimensional unified hardening model for overconsolidated clays,” *Geotechnique*, vol. 59, no. 5, pp. 451–469, 2009.
- [169] H. Matsuoka and T. Nakai, “Stress-Deformation and Strength Characteristics of Soil Under Three Different Principal Stresses.,” *Proc Jap Soc Civ Eng*, no. 232, pp. 59–70, 1974.
- [170] D. M. Wood, *Soil behaviour and critical state soil mechanics*. 199.
- [171] F. Nicot, N. Hadda, M. Guessasma, J. Fortin, and O. Millet, “On the definition of the stress tensor in granular media,” *International Journal of Solids and Structures*, vol. 50, pp. 2508–2517, 7 2013.