A machine learning based multi-scale computation framework for granular materials

Shaoheng Guan[a,b,c,1] , Tongming Qu[c], Y.T. Feng[c,2] , Gang Ma[a,b,3] , Wei Zhou[a,b]

[a] State Key Laboratory of Water Resources and Hydropower Engineering Science, Wuhan University, Wuhan 430072, China

[b] Key Laboratory of Rock Mechanics in Hydraulic Structural Engineering of Ministry of Education, Wuhan University, Wuhan 430072, China

[c] Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Swansea, Wales, SA1 8EP, UK

Abstract

With the development of experimental measurement technology and high-fidelity numerical simulations of granular materials, empirical-based classical constitutive models may not be able to take full advantage of the rapidly increasing available datasets. Machine learning-based models can inherently avoid phenomenological assumptions to directly learn the constitutive relationship from the datasets, and the trained model is sufficiently flexible to be reconstructed once new training samples are added. In this work, a coupled finite element method and machine learning (FEM-ML) computational framework is proposed for simulating granular materials. Gaussian process-based random loading paths and coupled FEM-DEM simulations are used to generate training samples. A parametrisation of the material deformation history is used to represent the historical influence of granular materials. An uncertainty-level based active learning is utilised to evaluate the informativeness of data points for network training and then to establish an effective resampling scheme from a massive dataset. Two examples are provided to show the applicability of the implemented FEM-ML framework. The performance of the proposed framework is also evaluated, the error is systematically analysed, and possible improvements are discussed. The results demonstrate that the FEM-ML framework offers considerable improvements in terms of computational efficiency and competence to simulate the mechanical responses of granular materials.

[1] Email: shaohengguan@gmail.com

[2] Corresponding author 1: y.feng@swansea.ac.uk

[3] Corresponding author 2: magang630@whu.edu.cn

1 Introduction

Granular materials such as gravel, sand, and powders are ubiquitous in industry and geotechnical applications, and they are considered to be the second most abundant material on Earth after fluids [1]. The discrete nature of granular materials and dissipative interactions among particles give rise to a rich and complex bulk behaviour, making them differ significantly from solids, liquids, and gases. The complexity of granular media can be partially attributed to its unique features, such as inherent anisotropy and heterogeneity [2, 3], pressure and rate-dependence [4–6], continuous evolving microstructure and complicated strain localisation phenomenon within unstable granular materials [7–11]. Accurately reproducing the mechanical behaviour of granular materials subject to various external loads through mathematical equations is intricate but crucial for engineering-scale numerical simulations.

In general, the mechanical behaviour of history-dependent granular materials has been described by empirical constitutive laws formulated within continuum thermodynamics and elastoplastic theory. Internal variables characterise the material state according to phenomenological assumptions. Advancements in micro-mechanical simulations and the internal fabric statistical description [12–15] have inspired physics-based internal variables that encapsulate the micro-structure and motivated the bottom-up mathematical modelling of granular materials by linking contact fabric tensor to macro-mechanical responses [6, 15–18]. Nearly in parallel, another class of bottom-up modelling approaches, the hierarchical multi-scale modelling, which directly bridges the macroscopic response of granular materials and the macroscopic boundary conditions [19–24], has also been developed due to computer hardware development. In the framework of hierarchical multi-scale modelling, the models of two scales are solved concurrently, and the results are exchanged on-the-fly. Both the two conventional modelling paradigms have achieved quite a success.

However, there are still considerable challenges in the way of their further progress. For the mathematical-based phenomenal approach, uncertainty arises due to the imperfect knowledge of the functional form of the constitutive laws and the parameter calibration is subject to the

increasing complexity. On the other hand, the classical hierarchical multi-scale computation is indecently time-consuming because a large number of lower-scale simulations are concurrently required during the computation of a macroscopic model.

In addition to the continuum mechanics modelling and hierarchical multi-scale modelling, the use of neural-network-based constitutive models can be traced back to the 1990s and the beginning of the 21st century [25–29]. Data-science-based model is a general term used for the recently arising new modelling method related to massive datasets, statistics, machine learning and data mining. Some related work in the fields of computational mechanics, geomechanics and geotechnical engineering can be found in recent reviews [30–34] and references therein.

In the case of the surrogate model constructed by the neural network, [27, 35–41], material informatics uses database techniques to identify parameter correlations and then uses machine-learning regression techniques to provide the quantitative predictive model. Due to the powerful high-dimensional mapping capability of the neural network, this method can reproduce the constitutive response of materials more precisely than the traditional mathematical based models, provided that sufficient data is available. For hierarchical multi-scale computation, a promising application of the neural network-based approach is to accelerate multi-scale mechanical calculations because of the higher computational efficiency over the conventional framework [37, 42, 43].

Unlike the regression-based modelling approach, another kind of data-driven method directly chooses the samples which best satisfy the compatibility and equilibrium constraints from the prepared database composed of strain stress pairs [44–47]. Instead of calibrating the parameters of the pre-constructed model through the datasets, this pure data-based method, which is solely based on the datasets, transforms the solution problem into a minimum distance optimisation, further avoiding the errors associated with the regression process, and almost wholly reproduces the results of the conventional multi-scale method. However, the particularity and computational efficiency of this *ad hoc* method may hinder its further development.

History dependence, fundamentally the intrinsic characteristic of granular materials [48], is another challenge for the data-driven modelling of granular materials. Internal variables based on plasticity mechanics approaches are often used to describe the historical state of the material in traditional elastoplastic constitutive models.

In the data-driven modelling work, some scholars use RNN (recurrent neural networks) to deal with the material history dependence and their works have demonstrated that stress-strain responses of granular materials can be predicted satisfactorily in certain stress-strain spaces [37, 38, 49–52]. However, the training is normally quite time-consuming as this model is not suitable for GPU parallel training because of its inherent sequential nature. Additionally, using different step sizes not included in the training samples will deteriorate the computational stability once the model is employed in boundary value problems [49, 50]. Due to the development of granular material simulation and the emergence of internal fabric description methods, some scholars use energy-based variables or fabric tensors to calibrate the current state of granular materials [46, 47]. This approach requires encoding the loading history of the material as the describer of the current state with as few parameters as possible, but how to take the sequence of macroscopic information (e.g. strain, stress, etc.) and encode it into historical state variables is again a tricky problem. Also, some preceding work attempts to discover some parameters to encode the history strain-stress path experienced by materials. In one example [39], the accumulation of the absolute values of the strain increments is used to uniquely represent the history state and appears to perform well in boundary value problem simulations.

The tangent operator of the strain-stress relation is necessary for the nonlinear iteration using the Newton-Raphson method in FEM simulations. In contrast to the FEM simulation of continuous materials, the particle simulation of granular materials, such as DEM (discrete element method), exhibits a random nature. The stress-strain curve can rise and drop sharply after entering the plastic regime, so the Jacobians of the network computed stress to the input strain cannot represent a proper tangent operator of the material. The approach that uses the perturbation method to numerically compute the Jacobian performs poorly in nonlinear iteration [22]. Therefore, the partial derivative tangent operator [28, 37] used for continuous materials may not work well for

granular materials. A viable solution, to be introduced in Section 3, is to replace the tangent matrix with a scant matrix in the non-linear iteration.

This work aims to propose a neural-network-based constitutive model as the surrogate model that directly learns the multi-scale mechanical relationship from the raw data collected from the consistent FEM-DEM multi-scale simulation. This model can be integrated into the FEM program with minimal intrusion and is considerably more computationally efficient than the classical multi-scale procedure. The history effect of the strain-stress relationship is considered by an internal variable-based parameterisation. The secant matrix is adopted to replace the Jacobian operator in the nonlinear iteration.

The rest of the paper is organised as follows. We first introduce the FEM-DEM framework in Section 2, followed by the description of the neural network-based constitutive model, a self-consistent sampling method, a general procedure of the uncertainty level-based active learning method and the model training process in Section 3. We first present the baseline numerical simulation of a biaxial compression problem to validate our implemented framework in Section 4. This is subsequently employed in a retaining wall simulation to evaluate its generalisability. The advantages and disadvantages of the FEM-ML framework and possible further improvements are summarised in the conclusion.

2 Multi-scale FEM-DEM framework

The multi-scale FEM-DEM framework for solving a granular problem consists of two main parts: (1) macro model which is implemented in FEM; (2) representative volume element (RVE) which is concurrently simulated through DEM. The main procedures in the FEM-DEM framework are shown in Figure 1.

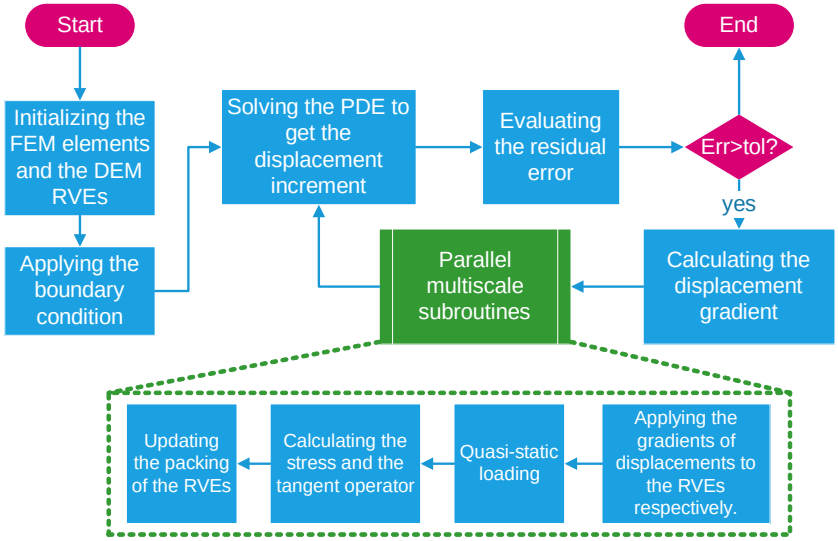


Figure 1 Flow chart of the FEM-DEM multiscale scheme

2.1 Macro model

Governing equation. In the FEM solver, the nodal displacements are solved as the basic unknown quantity constrained by the equilibrium equation and the compatible condition. Without considering the body force, the macro equilibrium is governed by:

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} = 0 & \text{in} \quad \Omega \\ \boldsymbol{n} \cdot \boldsymbol{\sigma} = \boldsymbol{t} & \text{on} \quad \partial\Omega \end{cases} \tag{1}$$

where $\nabla \cdot \boldsymbol{\sigma}$ is the gradient of the stress tensor, $\boldsymbol{n}$ is the unit vector of the outer normal of the boundary, $\boldsymbol{t}$ is the boundary traction, and Ω and $\partial\Omega$ are the domain and the boundary of the macro model, respectively. The governing equation can be written in the weak form as:

$$\int_{\Omega} \boldsymbol{\omega} \nabla \cdot \boldsymbol{\sigma} \mathrm{d}\Omega = 0 \tag{2}$$

where $\boldsymbol{\omega}$ is the test function that is often chosen as the shape function in the Galerkin method.

Discretisation. The domain is discretised into elements where the shape functions $\boldsymbol{N}$ are defined. With integration by parts and some routine operations, Eq. (2) is changed to:

$$\int_{\partial\Omega} \boldsymbol{N}^T t \mathrm{d}\Gamma - \int_{\Omega} \boldsymbol{B}^T \boldsymbol{\sigma} \mathrm{d}\Omega = \boldsymbol{0} \tag{3}$$

where $\boldsymbol{B} = \left(\nabla \boldsymbol{N} + \nabla \boldsymbol{N}^T \right)$. Or simply as:

$$\boldsymbol{T} - \boldsymbol{F} = \boldsymbol{0} \tag{4}$$

with the external traction vector $\boldsymbol{T} = \int_{\partial\Omega} \boldsymbol{N}^T \boldsymbol{t} \mathrm{d}\Gamma$ and the internal force vector $\boldsymbol{F} = \int_{\Omega} \boldsymbol{B}^T \boldsymbol{\sigma} \mathrm{d}\Omega$.

Nonlinear solution. Due to the plasticity of granular materials, nonlinear iteration should be introduced to obtain the solution. For a given initial solution $\boldsymbol{u}_0$, the internal force $\boldsymbol{F}$ as a function of the solution $\boldsymbol{u}$ can be approximated as

$$\boldsymbol{F}(\boldsymbol{u}) = \boldsymbol{F}(\boldsymbol{u}_0) + \boldsymbol{K}\Delta\boldsymbol{u} \tag{5}$$

where $\boldsymbol{F}(\boldsymbol{u}_0) = \int_{\Omega} \boldsymbol{B}^T \boldsymbol{\sigma}_0 \mathrm{d}\Omega$, and $\boldsymbol{K}$ is the stiffness matrix with a general expression:

$$\boldsymbol{K} = \int_{\Omega} \boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^T \mathrm{d}\Omega \tag{6}$$

where the matrix $\boldsymbol{D}$ depends on the material constitutive model used. Then the displacement increment $\Delta\boldsymbol{u}$ can be obtained by solving

$$\boldsymbol{K}\Delta\boldsymbol{u} = \boldsymbol{R}(\boldsymbol{u}) \tag{7}$$

where $\boldsymbol{R}(\boldsymbol{u}) = \boldsymbol{T} - \boldsymbol{F}$ is the residual force vector of the macro model.

The matrix $\boldsymbol{D}$ can be a tangent or secant operator, depending on which solution procedure is used, but the former may lead to a faster convergent rate. In the Newton-Raphson method, the tangent operator $\boldsymbol{D}$ is needed to construct the tangential element stiffness matrix in order to achieve a 2nd order convergence. In multi-scale calculations, the perturbation method [21] is often used to calculate $\boldsymbol{D}$. In Guo's FEM-DEM framework [53], an approximate secant elastic matrix is used.

2.2 Micro model

In the FEM-DEM framework, a particle assembly solved by DEM serves as the RVE at each Gauss point to capture the local material responses. The RVE used in the current work has 500 particles with their diameters uniformly distributed from 0.006 to 0.014. This particle number is chosen without conducting a convergence test in terms of accuracy for the RVE but is in line with some previous studies. For instance in [22], 400 particles are used in the RVE which seems sufficient to reproduce the homogenisation of the contact fabric tensor at the initial state. More importantly, because the accuracy of the FEM-DEM framework is not the main concern, but rather the machine-learning representation effectiveness of the FEM-DEM model, so choosing a

reasonably large number of particles for each RVE should suffice. The boundary condition (deformation) for each RVE is imposed based on the macro strain $\boldsymbol{\epsilon}$ obtained at the Gauss point by the macro FEM solver. Then the corresponding stress $\boldsymbol{\sigma}$ and matrix $\boldsymbol{D}$ are derived from the RVE via DEM and passed to the FEM solver to update the macro solution.

The periodic boundary and Hertz's contact model are employed in the DEM simulation. The normal and tangential contact forces between two contacting particles are calculated as:

$$\begin{cases} \boldsymbol{f}_n = k_n \boldsymbol{u}_n^c \\ \boldsymbol{f}_t = k_t \boldsymbol{u}_t^c \end{cases} \tag{8}$$

where $\boldsymbol{u}_n^c$ and $\boldsymbol{u}_t^c$ are the relative normal and tangential displacement vectors between the two particles; and k_n and k_n are the normal and tangential stiffness respectively and defined by

$$\begin{cases} k_n = \dfrac{G}{1-\mu}\sqrt{2r^* u_n^c} \\ k_t = \dfrac{2G}{2-\mu}\sqrt{2r^* u_n^c} \end{cases} \tag{9}$$

where G and μ are the shear modulus and Poisson's ratio of the particle material respectively; the relative normal displacement is commonly referred to as the overlapping of the two particles; $r^* = \left(2r_1 r_2\right)/\left(r_1 + r_2\right)$ is the equivalent radius of the particles with r_1 and r_2 being their radii.

In the multi-scale simulation, the contact forces in an RVE should be transformed into the average stress tensor. The homogenised stress tensor of the particle assembly is calculated as [54–56]:

$$\boldsymbol{\sigma} = \frac{1}{2V}\sum_{c=1}^{N_c}\left(\boldsymbol{f}^c \otimes \boldsymbol{d}^c + \boldsymbol{d}^c \otimes \boldsymbol{f}^c\right) \tag{10}$$

where V is the total volume of the particle assembly, N_c is the total number of contacts within the volume, $\boldsymbol{f}$ and $\boldsymbol{d}$ are the contact force vector and the branch vector connecting the centres of the two contacted particles, respectively.

In the FEM-DEM framework, the matrix $\boldsymbol{D}$ is statistically approximated from the contacts and homogenised elastic modulus of the particle assembly. The analytical form based on the assumption of a uniform strain field is given by [57]:

$$\boldsymbol{D} = \frac{1}{V}\sum_{c=1}^{N_c}(k_n \boldsymbol{n}^c \otimes \boldsymbol{d}^c \otimes \boldsymbol{n}^c \otimes \boldsymbol{f}^c + k_t \boldsymbol{t}^c \otimes \boldsymbol{d}^c \otimes \boldsymbol{t}^c \otimes \boldsymbol{d}^c) \tag{11}$$

where $\boldsymbol{n}_c$ and $\boldsymbol{t}_c$ are unit normal and tangential directions of contact c, respectively.

2.3 FEM-DEM coupling method

The algorithmic procedures of the coupled FEM-DEM solver is given in Algorithm 1. The multi-scale coupling method can reproduce the micro-scale mechanical property of granular materials at the macro scale, as macro strain and micro-scale material response obtained from the RVE simulation are fully exchanged at the Gauss points. However, a large number of DEM simulations are required which makes this simulation considerably time-consuming [53, 58]. Most of the computational time is spent on evaluating the micro DEM model for each Gauss point of the macro model. In addition, there is a large amount of statistical data generated in this simulation, including sequences of stress, strain, pore ratio, fabric information, and even the details of the particle packings at every iteration step. It seems that these datasets are underutilised, and most of them are discarded after the concurrent simulation.

The machine learning method can be employed to take full advantage of the intermediate data generated in the concurrent computations and substantially accelerate the multi-scale computation. The rest of the paper is dedicated to developing a neural network-based constitutive model that serves as an efficient surrogate for the concurrent DEM simulation.

Algorithm 1 The FEM-DEM solver

Given: Discretised FEM model, and initialised RVE packings $\mathcal{PK}^{(0)}$

$\left(\boldsymbol{\sigma}^{(0)}, \boldsymbol{D}^{(0)}\right) = \mathcal{PK}\left(\epsilon^{(0)}, \boldsymbol{\varphi}^{(0)}\right)$ ▷ Initialise the stress and matrix $\boldsymbol{D}$

For $n = 1, 2, \ldots, N$ **do** ▷ Loading step

Apply the boundary condition at step n to the FEM model

Set $m = 0$, $e_u = 1$

Get $\Delta \boldsymbol{u}^{(n,m)}$ by solving Eq. (7)

While $e_u > e_{tol}$ **do** ▷ Nonlinear iteration

Compute the strain increment $\Delta \boldsymbol{\epsilon} = \boldsymbol{B} \Delta \boldsymbol{u}^{(n,m)}$

Apply $\Delta \boldsymbol{\epsilon}$ to the RVE packing $\mathcal{PK}^{(n,m)}$

Evaluate $\left(\boldsymbol{\sigma}^{(n,m)}, \boldsymbol{D}^{(n,m)}\right) = \mathcal{PK}^{(n,m)}\left(\boldsymbol{\varepsilon}^{(n,m)}, \boldsymbol{\varphi}^{(n,m)}\right)$

Update $\boldsymbol{R}^{(n,m)}$ and $\boldsymbol{K}^{(n,m)}$ based on $\left(\hat{\boldsymbol{\sigma}}^{(n,m)}, \hat{\boldsymbol{D}}^{(n,m)}\right)$ using Eqs. (5) and (6)

Get $\Delta \boldsymbol{u}$ by solving Eq. (7)

Update $\Delta \boldsymbol{u}^{(n,m)} = \Delta \boldsymbol{u}^{(n,m-1)} + \Delta \boldsymbol{u}$

Compute the error $e_u = \| \Delta \boldsymbol{u} \| / \| \Delta \boldsymbol{u}^{(n,m-1)} \|$

$m = m + 1$

end While

Update $\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \Delta \boldsymbol{u}^{(n,m)}, \boldsymbol{\sigma}^{(n)} = \boldsymbol{\sigma}^{(n,m)}, \boldsymbol{D}^{(n)} = \boldsymbol{D}^{(n,m)}$

end For

3 The neural network-based constitutive model

3.1 Surrogate micro model constructed by neural network

Classical constitutive models are mainly developed based on some phenomenological assumptions. It is generally sufficient to use a small set of experimental data to calibrate the parameters in the model. However, due to the improvement of observation and simulation techniques, these constitutive models may become inadequate to incorporate rapidly increasing high-quality data. Unlike conventional constitutive models, machine learning models directly mine constitutive relationships from data, almost without making any assumptions, since neural network models are sufficiently sophisticated to reproduce various mechanical behaviours of the material [29, 59].

In the FEM-DEM simulation, the particle assembly attached to each Gauss point is saved in the

247 computer RAM for the next loading step to reproduce the history-dependent effect of the granular
248 material, which is also the inherent reason for its huge memory cost. One of the central challenges
249 of the machine learning model is to characterise a general loading history with as few variables as
250 possible. As the network-based model will no longer gain access to the particle assembly, proper
251 internal variables should be included to account for the history-dependent (plastic) state of
252 granular materials in the FEM-ML framework. However, it is hard to explicitly calibrate the
253 plastic state, such as plastic work and yield surface, since they cannot be easily derived from the
254 macro DEM simulated results (i.e. strain, stress, void ratio).
255
256 Unlike traditional internal variables, the history variables used for machine learning can have no
257 physical meaning and can be merely used to quantify the historical influence or sequence order.
258 Thus, they only need to possess some very basic mathematical properties, such as monotonicity. In
259 this work, the accumulation of the absolute strain increments:

260 $$\boldsymbol{\varphi}^{(n)} = \sum_{i}^{n} | \Delta\boldsymbol{\epsilon}^{(i)} | \tag{12}$$

261 which is conveniently available and interpretable, is used as the history variable. Subsequently, the
262 constitutive relationship to be obtained can be expressed as

263 $$\hat{\boldsymbol{\sigma}}^{(n)} = \mathcal{NN}(\boldsymbol{\epsilon}^{(n)}, \boldsymbol{\varphi}^{(n)}) \left(n = 1, ..., N\right) \tag{13}$$

264 where $\mathcal{NN}$ denotes a multi-layer fully connected neural network, the superscript n represents
265 the sample number and N is the total number of training samples. Then the prediction error is
266 measured by the MSE (mean square error) as:

267 $$err = \frac{1}{N} \sum_{n}^{N} || \hat{\boldsymbol{\sigma}}^{(n)} - \boldsymbol{\sigma}^{(n)} ||^2 \tag{14}$$

268 Our experiments show that using the accumulation of the absolute strain increments defined in Eq.
269 (12) as the internal variable seems to be able to encode the unique stress-strain historical state of
270 materials and that the trained network can effectively capture the path-dependent behaviour of
271 granular assemblies.
272
273 In the FEM-DEM framework, the approximated secant matrix is used as the stiffness matrix
274 which is derived from the fabric of the particle assembly (i.e. Eq. (11)), parallel with the stress and
275 other state variables. The requirement of some specific micro-scale information in FEM-DEM

calculation, such as the configuration and dynamic properties of the assembly, will demand much greater computational time and memory resources, dramatically slowing down the multi-scale scheme. There are various features (such as stress, strain, void ratio, and fabric information) generated at every time step in the whole FEM-DEM simulation. However, only necessary features should be involved in machine learning to calculate the operator $\boldsymbol{D}$. Simplifying and encapsulating these features into simple pairs as $(\boldsymbol{\epsilon}, \boldsymbol{\varphi}, \boldsymbol{\sigma}, \boldsymbol{D})$ will alleviate difficulties in the network training and accelerate the computation.

There are mainly two ways to represent the operator $\boldsymbol{D}$ for the nonlinear iteration in a machine learning scheme: (1) tangent operator, and (2) secant operator. The tangent operator will be ideal for the Newton-Raphson solution of the nonlinear problem if the stress-strain function is sufficiently smooth and differentiable. In machine learning-based work, some computations for continuous media use the AD (Automatic Differentiation) in Tensorflow [60] to obtain $\boldsymbol{D} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}$ [28, 35–37] or the automatic differentiation tool AceGen [61] to directly update the stiffness matrix via $\boldsymbol{K} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}$ [62].

Unfortunately, the stress-strain curves extracted from granular material simulations fluctuate dramatically, as shown in Figure 2. The network is trained to capture the main trend of these characteristics but is not intelligent enough to distinguish the valuable constitutive relationship from the fluctuations. Thus, the tangent operator calculated via the auto-differentiation may change sharply and largely deviate from the right direction, which will adversely affect the computational stability.

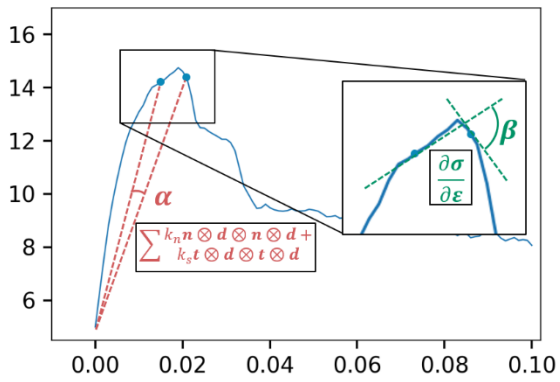


Figure 2: Schematic of the approximate secant matrix and the tangent matrix

The secant matrix can be approximated via Eq. (11) since the contact details in a particle assembly are available in DEM simulation. It is also theoretically available via the machine learning as strain $\boldsymbol{\epsilon}$ and history variable $\boldsymbol{\varphi}$ are both input variables. Therefore, in consideration of the computational stability, the approximate secant operator together with the stress is adopted via the $\mathcal{NN}$ to complete the nonlinear iteration:

$$(\hat{\boldsymbol{\sigma}}^{(n)}, \hat{\boldsymbol{D}}^{(n)}) = \mathcal{NN}(\boldsymbol{\epsilon}^{(n)}, \boldsymbol{\varphi}^{(n)}) \tag{15}$$

3.2 Procedures of the multi-scale coupling method

The procedure of the FEM-ML framework is outlined in Algorithm 2. There are no particle assemblies that need to be initialised and updated since the neural network will directly calculate the stress and the secant operator after the strain and history variables are fed into the trained network. Note that the only physical parameter involved in our neural network-based constitutive model is the particle-scale Hertz contact law for discrete element modelling of granular materials. There are no other phenomenological constitutive models involved in the FEM-ML modelling.

Algorithm 2 The FEM-ML solver

Given: Discretised FEM model, and well-trained neural network $\mathcal{NN}$

$\left(\hat{\boldsymbol{\sigma}}^{(0)}, \hat{\boldsymbol{D}}^{(0)}\right) = \mathcal{NN}\left(\epsilon^{(0)}, \boldsymbol{\varphi}^{(0)}\right)$ ▷ Initialise the stress and matrix $\boldsymbol{D}$

For $n = 1, 2, \ldots, N$ **do** ▷ Loading step

Apply the boundary condition at step n to the FEM model

Set $m = 0$, $e_u = 1$

Get $\Delta \boldsymbol{u}^{(n,m)}$ by solving Eq. (7) with $\hat{\boldsymbol{\sigma}}^{(n-1)}, \hat{\boldsymbol{D}}^{(n-1)}$

While $e_u > e_{tol}$ **do** ▷ Nonlinear iteration

Compute $\boldsymbol{\epsilon}^{(n,m)} = \boldsymbol{\epsilon}^{(n-1)} + \boldsymbol{B}\Delta \boldsymbol{u}^{(n,m)}$

Evaluate $\left(\hat{\boldsymbol{\sigma}}^{(n,m)}, \hat{\boldsymbol{D}}^{(n,m)}\right) = \mathcal{NN}\left(\boldsymbol{\epsilon}^{(n,m)}, \boldsymbol{\varphi}^{(n-1)}\right)$

Obtain $R^{(n,m)}$ and $K^{(n,m)}$ from $\left(\hat{\sigma}^{(n,m)}, \hat{D}^{(n,m)}\right)$ via Eqs. (5) and (6)

Get $\Delta \boldsymbol{u}^{(n,m)}$ by solving Eq. (7)

Calculate $e_u = \| \Delta \boldsymbol{u}^{(n,m)} - \Delta \boldsymbol{u}^{(n,m-1)} \| / \| \Delta \boldsymbol{u}^{(n,m-1)} \|$

$m = m + 1$

end While

Update $\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \Delta \boldsymbol{u}^{(n,m)}$, $\boldsymbol{\sigma}^{(n)} = \boldsymbol{\sigma}^{(n,m)}$, $\boldsymbol{D}^{(n)} = \boldsymbol{D}^{(n,m)}$

end for

3.3 Network architecture

Most of the work in machine learning, especially deep learning, uses fairly complex network structures [37, 38, 49, 63], while traditional phenomenological constitutive models of granular materials generally use around ten parameters to approximately describe the stress-strain relationships. As shown in [64], PINNs (Physics-constrained neural networks) constructed by parsimonious networks can exhibit fairly good prediction accuracy. Using simpler networks can also largely reduce the number of parameters involved. Additionally, we found that simple neural networks are more conducive to computational stability, which may be related to the fact that the predicted values given by a simple network tend to be smoother. By considering the simplicity of deployment, computational stability and efficiency, a simple multilayer fully connected network is chosen in our work to be integrated with FEM calculation, with the expense of slightly lower training accuracy.

Figure 3 shows the neural network used, which is composed of two fully connected hidden layers, each having 40 neural nodes (or neurons). The ReLU (Rectified Linear Unit) activation function is adopted except for the output layer (without activation function). The input layer includes the current strain and history variables, and the output layer produces the corresponding stress and $\boldsymbol{D}$ matrix. The Adam algorithm (with *lr* =1e-3, *betas* =(0.9, 0.999), *eps* =1e-8, *weight_decay* =0) is employed to optimise the weights and biases in the training process. PyTorch is used to implement the neural network. Note that no attempt has been made to optimise the network in terms of the number of hidden layers and neurons in each layer.

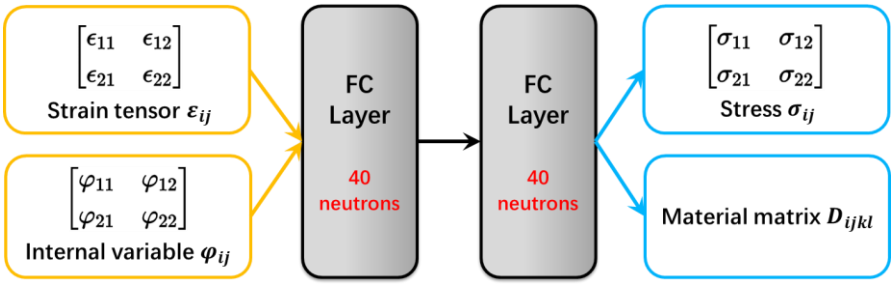


Figure 3: Architecture, input and output features of the neural network

3.4 Sampling

In general, machine learning models are good interpolators but often do not perform well for extrapolation. To develop a machine-learning-based constitutive model suitable for a wide range of strain paths, one solution is to accommodate as many strain-stress pairs as possible, so that all the "extrapolation" become "interpolation". This, however, imposes two challenges. Firstly, it will be extremely challenging to generate sufficient training datasets to fully cover all the possible strain-stress paths for a certain type of granular material. The task becomes practically impossible when further considering all possible loading-unloading combinations. This is a sharp contrast to the conventional constitutive model where the model parameters can be calibrated by a relatively small number of conventional triaxial tests in geomechanics. Secondly, the presence of a large number of datasets is also a challenge for the training of neural networks—a large number of datasets demand expensive computational costs. Thus, it is important to utilise any useful condition to increase the effectiveness of the training dataset, leading to the reduction of sampling and training costs.

For homogeneous materials, some symmetric properties can be exploited for reducing material sampling requirements in the data-driven paradigm [39, 44]. For instance, both strain and stress tensors at any data point pair can be mapped to their principal components via a proper rotation. With this principal space mapping, the dimensions of the strain/stress sampling space are reduced from six to three [39, 63]. In Tang's work [65–67] the 3D sampling space is further reduced to a 1D space. Note, however, that for granular materials, the strain and stress matrix may not be simply mapped to the principal space because the principal directions of the strain and stress start to deviate since plasticity (or non-affine deformation) emerges [8, 68]. This non-coaxial issue is considered in our work by casting all components of the strain and stress tensor into the network model.

3.4.1 Sample generation

Gaussian process is introduced to generate smooth random loading paths in order to cover a large sampling space of $(\boldsymbol{\epsilon}, \boldsymbol{\sigma})$ in our work. In some machine learning frameworks [38, 49], Gaussian process [69] is used in the micro-scale random loading path preparation. However, the DEM simulations may return unreasonable results when the granular materials are over-compressed or pulled up. So, this work applies the randomly generated loading paths as macro boundary conditions instead of the micro-scale RVE. The computed strain at each Gauss point of the macro model is applied as the boundary condition to the corresponding micro-scale RVE. In this way, a large number of samples closely related to the solution space can be directly generated. Applying the boundary condition directly to the macro model is also much simpler, as there is no need to check the compatibility of the generated results.

The macro model used to generate training data is a simple biaxial test discretised by three different finite element meshes. The detail will be described in Section 4.1 and shown in Figure 6. A series of FEM-DEM biaxial simulations subject to randomly constructed boundary loading paths is carried out to generate training data. To avoid over-compression or stretching, the random path is only applied to control the transverse confining pressures for the biaxial compression test. All the REV particle models are initially compressed by an isotropic confinement pressure of 100kPa. Further details will be given in Section 4.

The Gaussian process defines a distribution function $f(\boldsymbol{x})$ that is completely specified by its mean function $m(\boldsymbol{x})$ and covariance kernel function $\boldsymbol{\kappa}(\boldsymbol{x},\boldsymbol{x}')$, i.e., a Gaussian process can be represented as:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}),\ \boldsymbol{\kappa}(\boldsymbol{x},\boldsymbol{x}')) \tag{16}$$

where the mean value $m(\boldsymbol{x})$ is kept to be the initial confinement pressure of 100 kPa. The smoothness of the generated sequence can be controlled by specifying the kernel function. The squared distance function is used as the covariance kernel function and described by:

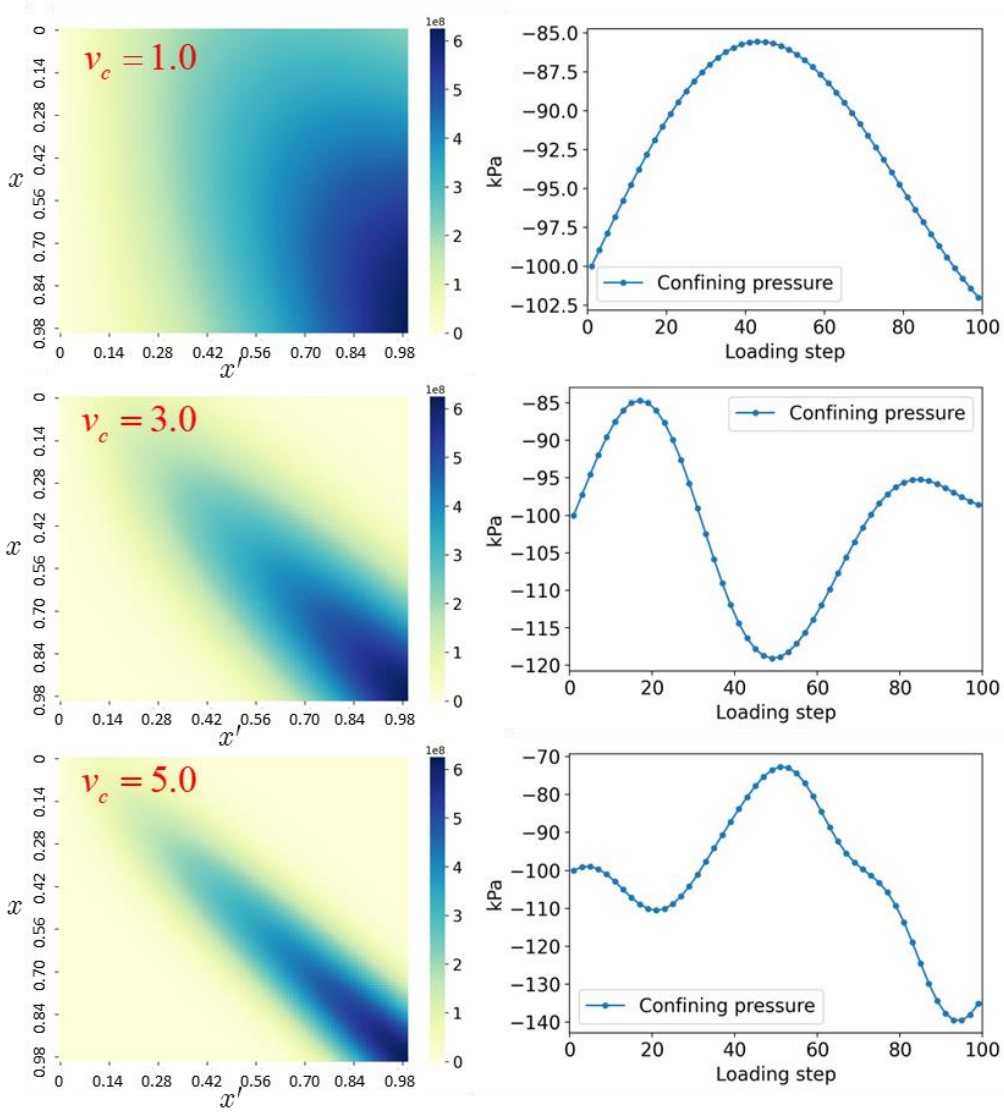
$$\boldsymbol{\kappa}(\boldsymbol{x},\boldsymbol{x}') = \exp(-v_c\theta_k(\boldsymbol{x})\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|^2) \tag{17}$$

where variable $x \in [0,1]$ can be regarded as the pseudo-time to control the loading step, and its discretised values $\boldsymbol{x} = [0.01, 0.02, ..., 1]$, corresponding to 100 loading steps, are used as the vector for kernel $\boldsymbol{\kappa}$ generation; $v_c \in [1,5]$ is a coefficient used to control the curvature of the random path (a larger v_c narrows the band of the kernel, but increases the curvature of the stress curve, as illustrated in Figure 4); and $\theta_k(\boldsymbol{x})$ is a linear function of $\boldsymbol{x}$ starting from 0 to ensure that the random confining pressure starts from the initial consolidation pressure of 100 kPa.

The Gaussian process outlined above is used to generate ten random loading paths for the confinement pressure for each finite element model (Figure 6). The total number of samples is summarised in Table 1, with an overall sum of nearly 17 million data points generated. Due to the nonlinear macroscopic mechanical properties of granular materials, each loading step needs nonlinear iteration. On average, a loading step takes about 20 iterations. Therefore, more than 2000 calculation results are generated in 100 loading steps.

Table 1 Summary of the training samples generated at three mesh levels

Loading path number	Coarse	Medium	Fine
0	76064	308608	1295360
1	75456	316544	1243136
2	75264	308736	1314304
3	81696	309248	1303040
4	77984	304768	1312768
5	77472	317440	1304576
6	76128	303744	1288704
7	75296	310016	1285632
8	75040	313472	1296896
9	75456	331904	1330176
Total training samples		16864928	



411

412 Figure 4: Random loading path generation via Gaussian process: the kernel functions

413 (left) and random paths (right) for different v_c values

414

415 *3.4.2 Active learning-based resampling.*

416 It may not be a good idea to use all the available data points, particularly in huge datasets, for

417 network training. Using all data points may prevent the optimizer from finding the most

418 informative datasets for the targeted training as there may be many data points that are repetitive

419 or very similar. In this work, we adopt an active learning strategy to effectively select samples

420 from our prepared large datasets.

421

The active learning method [70] is introduced for effective resampling from a dataset with massive data points. The key idea behind is that a machine learning algorithm can achieve greater accuracy with fewer training samples if the algorithm is facilitated to choose essential data points according to what it has learned. In addition to resampling, active learning can also guide sample generation, especially when generating a sample is expensive. The active learning method is able to evaluate the network performance at a data point (i.e. strain and internal variable) without knowing the output/response (i.e. stress) and thus can determine whether the data point is valuable to be generated for network training.

The active learning procedure used is as follows. Firstly, a selected number of networks with the same architecture and hyperparameters but different randomly generated initial weights and biases are trained on partial datasets or smaller datasets generated based on a subset (i.e. in the coarse FE mesh here). Subsequently, these trained networks are used to make predictions at different data points (i.e. in the fine FEM mesh), and then evaluate their uncertainty levels ξ, defined to be the standard deviation of all the predictions at each datapoint. The process can be summarised as:

$$\begin{cases} \hat{\boldsymbol{y}}^{(n)} = \dfrac{1}{m}\sum_{i=1}^{m} \mathcal{NN}_i(\boldsymbol{\sigma}^{(n)}, \boldsymbol{\varphi}^{(n)}) \\ \xi^{(n)} = \sqrt{\dfrac{\sum_{i=1}^{m} \| \mathcal{NN}_i(\boldsymbol{\sigma}^{(n)}, \boldsymbol{\varphi}^{(n)}) - \hat{\boldsymbol{y}}^{(n)} \|^2}{m-1}} \end{cases} \tag{18}$$

where m is the number of the randomly trained networks $\mathcal{NN}_i$ ($i = 1, 2, ..., m$), $\hat{\boldsymbol{y}}^{(n)}$ is the averaged output of all the networks at a selected Gauss point n in the fine-meshed model. After evaluation, points with an uncertainty value higher than a certain level will be added to the training samples to retrain the models until a satisfactory level of uncertainty is achieved for all the Gauss points concerned. Then one of the trained models is chosen to be used for prediction in the FEM-ML solver. Thus the active learning in the present work acts as a detector to seek the locations where the randomly trained model performs poorly, i.e. the data points are underrepresented yet. This active learning resampling is referred to as the uncertainty-level based active learning scheme.

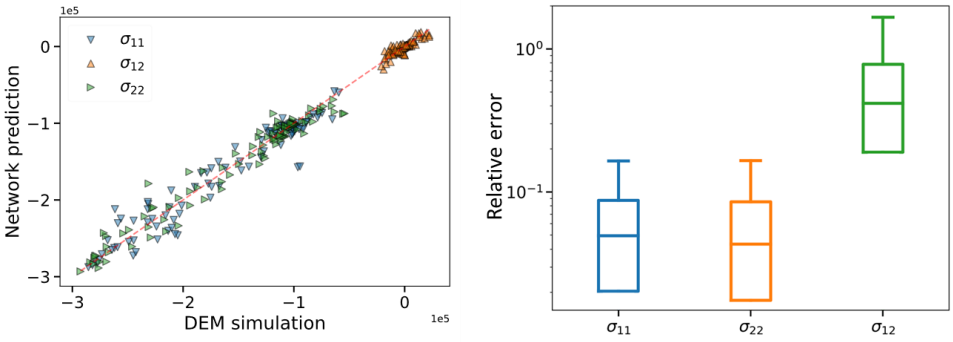
In addition to the above uncertainty-level base active learning scheme, some other indicators can also be utilised to guide the resampling process. An alternative, equivalent shear strain-based

resampling scheme is another possible option for the problems concerned in Section 4. The details of using this option and comparison with the uncertainty-level base scheme are given in Section 4.1.2.

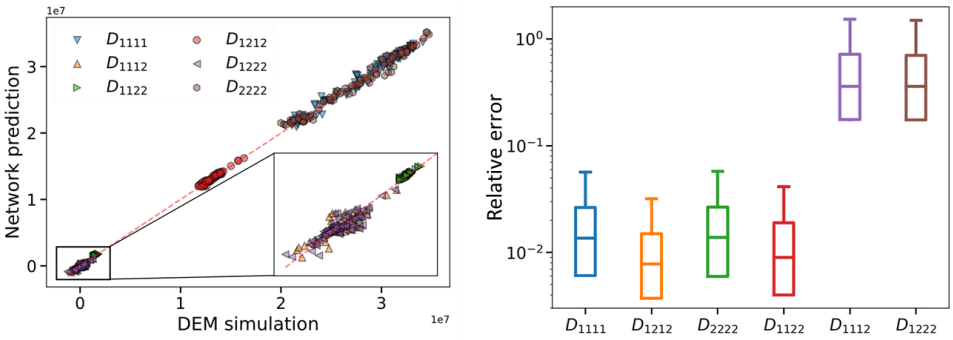
3.5 Training

The neural network is trained on the dataset collected from a random Gaussian process controlled biaxial loading. The datasets are split into training, validating, and testing sets weighing 70%, 15%, and 15%, respectively. The early stopping technique [71] is utilised in the training process to avoid overfitting and thus maximise the generalisation ability of the ML model. The MSE (mean square error) of both training and validation datasets is evaluated every ten epochs. The training process is ceased once the MSE of the validation dataset has no improvement after 1,000 training epochs.

After training, the prediction ability of the model and the prediction error are checked, as shown in Figure 5. The trained $\mathcal{NN}$ model, in most cases, is competent to predict with satisfactory accuracy, especially in the prediction of the two stress components (σ_{11} and σ_{22}) and four components ($D_{1111}, D_{1122}, D_{1212}, D_{2222}$) of the material matrix. However, the $\mathcal{NN}$ model seems not to perform well in predictions of σ_{12} , D_{1112} and D_{1222} , as shown in the right column of Figure 5 where the relative errors reach around more than 100 per cent. Nevertheless, this is because, in an isotropic hyper-elastic model or quasi-elastic stage of the granular material, the components of D_{1112} and D_{1222} should be zero [72] and σ_{12} is also close to zero. Thus, it is more appropriate to check the absolute error for these components. The left column of Figure 5 shows that their predicted values are indeed close to zero, indicating their prediction accuracy is also satisfactory.



(a) Prediction of the stress vector



(b) Prediction of the material matrix

Figure 5: ML prediction versus the results of the DEM simulation – left column: comparison; right column: relative error.

4 Computational examples

4.1 Baseline problem - *the biaxial compression test*

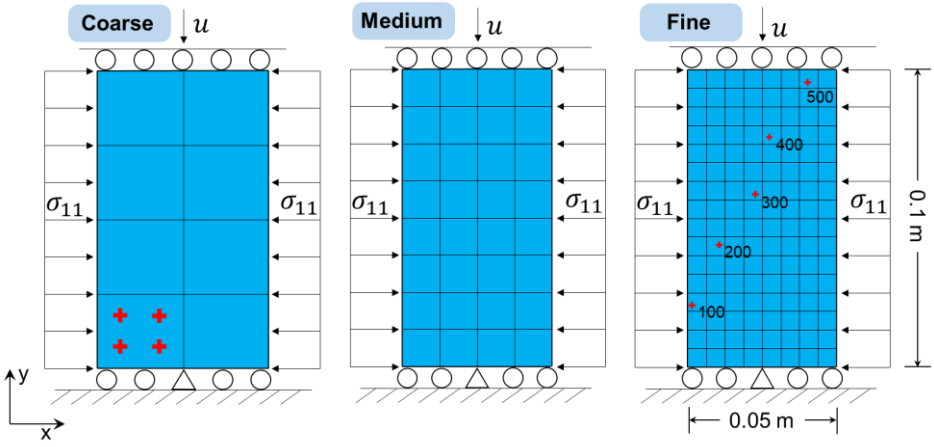


Figure 6: Configuration of the biaxial compression test and three meshes

The biaxial compression test, shown in Figure 6, is simulated as the baseline to validate the proposed FEM-ML framework. Three different meshes are used to discretise the problem domain: coarse (2×4 elements), medium (4×8 elements) and fine (8×16 elements) meshes, with each

element having 4 Gauss points.

Table 2 Parameters of the granular materials

Density (kg/m^3)	Young's modulus (MPa)	Poisson's ratio	Friction angle (rad)	Damping ratio
2650	600	0.8	0.5	0.1

Initially, the vertical displacement-controlled loading is linearly applied to the top boundary of the biaxial model until the macro vertical strain reaches 0.1. The Gaussian process outlined in Section 3 is used to generate ten random loading paths to control the confinement pressure for each meshed model. Parameters used in the lower-scale DEM simulation are shown in Table 2. Both the FEM-DEM and the FEM-ML simulations are carried out for the coarse and fine meshes with an identical loading/boundary condition to assess whether our proposed FEM-ML framework is capable of reproducing the multiscale mechanical response and effectively accelerating the computation. The medium mesh is used later in Section 4.1.4.

4.1.1: Comparison of the prediction results with different meshes.

To comprehensively compare the prediction capability of the proposed FEM-ML framework, six cases that combine the two meshes (coarse and fine) for both the training dataset generations and the FEM-ML simulations are considered and listed in Table 3. The simulated results are displayed in Figure 7 (top force) and Figure 8 (displacement field).

Table 3 The cases using different meshes for training and FEM-ML simulations

Case	Mesh used for obtaining training samples	Mesh used for FEM-ML simulations
A	Coarse	Coarse
B	Fine	Coarse
C	Hybrid	Coarse
D	Fine	Fine
E	Coarse	Fine
F	Hybrid	Fine

These two figures illustrate that all results agree well with the DEM simulations except for Case E. The shear band simulated with the fine mesh is narrower than that with the coarse mesh in the FEM-DEM simulations. The final deformed configuration of Case E is more

similar to Case A instead of the fine-meshed FEM-DEM simulation. In Cases A and E, only datasets collected from the coarse mesh simulations are employed to train the neural network. The Gauss points are clearly sparser in the coarse mesh, thus there are not sufficient points to represent the strain-stress relationship in the fine mesh.

On the other hand, due to the neural network's good interpolation ability, all of the stresses and tangent matrices in the shear bands in the fine mesh can be well approximated by the network trained on the coarse datasets. Although the fine mesh is used for the macro-level simulation in Case E, the strain-stress relationship is still derived from the coarse mesh datasets. Therefore, the shear band obtained in Case E still looks like a duplicate of Case A.

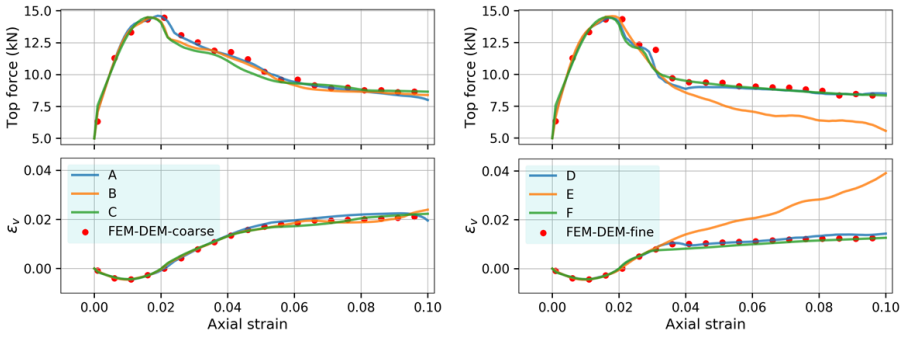


Figure 7: The comparison of the top force between different cases

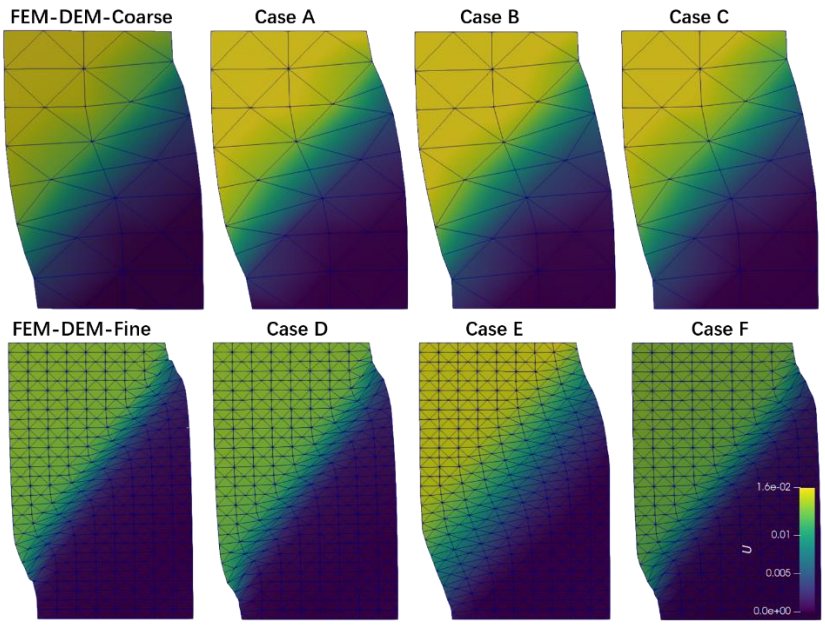


Figure 8: The final deformed configurations of the different cases

Due to the fact that the strain distributions in the upper and lower regions of the domain are very similar for both coarse and fine meshes, attentions are focused on the shear band obtained from different cases. Cases C and F are subsequently carried out to investigate this issue, where the network models are trained on datasets gathered through a hybrid sampling method as shown in Figure 9. In this hybrid sampling method, the data points in the shear band are obtained from the fine mesh simulations, while the data points in the upper and lower triangular regions are collected from the coarse-meshed simulations.

Figure 8 shows that the final deformed configurations of Cases C and F agree well with the FEM-DEM simulations. This suggests that the training datasets in the shearing band significantly influence the performance of the purely data-based model. Only with the proper training datasets, can the network model work well and attain the expected results.

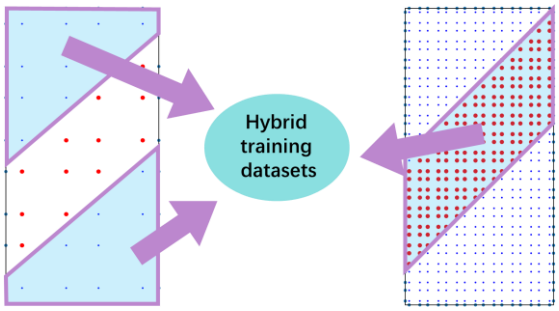


Figure 9: Collecting the training datasets in a hybrid way

4.1.2 Automatic resampling via active learning.

The two hybrid cases (Cases C and F) indicate that the significance of different data points varies greatly. The points in the shear band play a significant role in the current problem. In contrast, constitutive relationships contained by points in the upper and lower triangles of the fine-meshed model can be easily represented by the data points from the coarse-meshed model. To fully utilise these features, the uncertainty-level-based active learning scheme introduced in Section 3.4.2 is used for automatic resampling. The detailed procedure is outlined in Figure 10 and explained

below:

1) Choose five separate networks with the same architecture and hyperparameters as described in Section 3.3 but with different randomly initialised weights and biases which are pre-trained based on the datasets obtained from the coarse mesh.
2) Use the five pre-trained models to predict the stresses and $\boldsymbol{D}$ matrices for all strains of the data points collected from the fine mesh FEM-DEM simulations.
3) Evaluate the prediction uncertainty level at each data point based on the five predicted stresses by Eq. (18).
4) Add 30% of the data points that have the highest uncertainty levels to the training samples used in the pre-training phase.
5) Re-train one pre-trained model on the enriched dataset and use it as the network model for the final FEM-ML simulation.

Note that the number of NNs used in the above active learning resampling cannot be determined analytically but is a result of trial and error in practice. The main principle is that we wish to use as few NN models as possible to find a reliable variance ranking which can effectively recognise the most informative points to improve the current NN predictions. To strike a balance between accuracy and computational costs, 5 is found to be a satisfactory number of the required networks in the current work.

Figure 10 clearly illustrates that the prediction uncertainty is prominent in the shear band, indicating that the data points in the shear band have a stronger influence on the network prediction. Locations of the points with high uncertainty levels agree well with the red data points extracted on the right side of Figure 9, which explains why the result of Case F is improved. Thus, active learning can be used as a filter to identify the locations where the trained network model performs poorly and samples at these locations should be generated or added to enrich the training dataset.

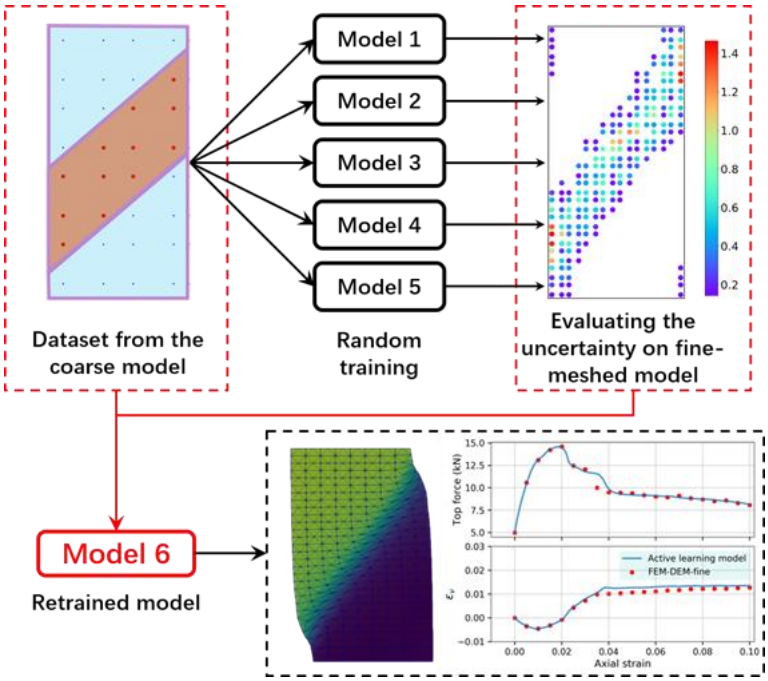


Figure 10: The flowchart of the model training process based on the active learning resampling

As the newly added data points in the above active learning resampling process are almost all located in the shear bands of the deformed configuration, it indicates that the magnitude of shear strain may be utilised as an alternative indicator for the resampling purpose. To investigate the possibility of using the equivalent shear strain as the indicator for active learning resampling, we compute the equivalent shear strains at all Gauss points at each load step. Then 30% of the data points having the highest uncertainty levels at several load steps are depicted on the top row in Figure 11 (where the average uncertainty values are also given), while the bottom row shows the corresponding contour plots of the equivalent shear strain.

The figure shows that the shear band (or strain localisation) starts to emerge at around load step 29 and fully develops from step 59 onwards. The corresponding distributions of the Gauss points with 30% of the highest shear strain for these load steps are indeed around the shear band and these coincide with the prediction by the uncertainty-level based active learning scheme shown in Figure 10. Thus it seems reasonable to conclude that the (equivalent) shear strain can be used as an alternative indicator for the active learning resampling. However, at the beginning of the loading and well before the shear band is formed, the strain field is nearly uniform across the

whole domain. Consequently, the shear strain indicator is unable to distinguish the difference between data points over the whole domain, and the selected data points may not be ideally located. It is clearly displayed for the first two load steps where the selected points are not uniformly distributed across the domain.

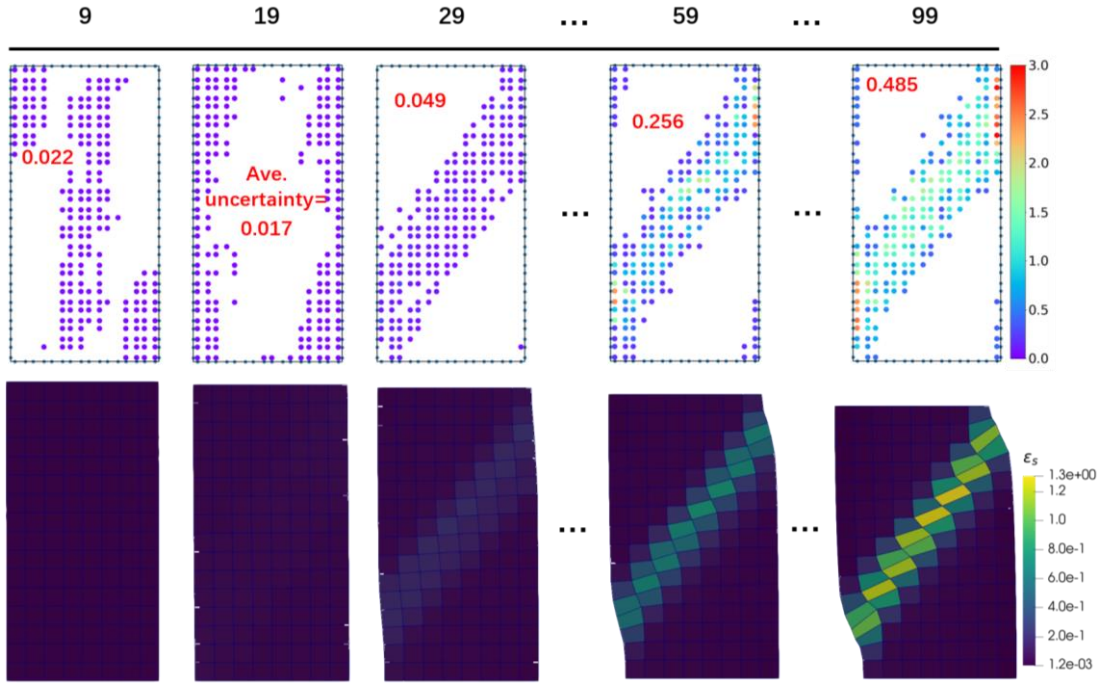
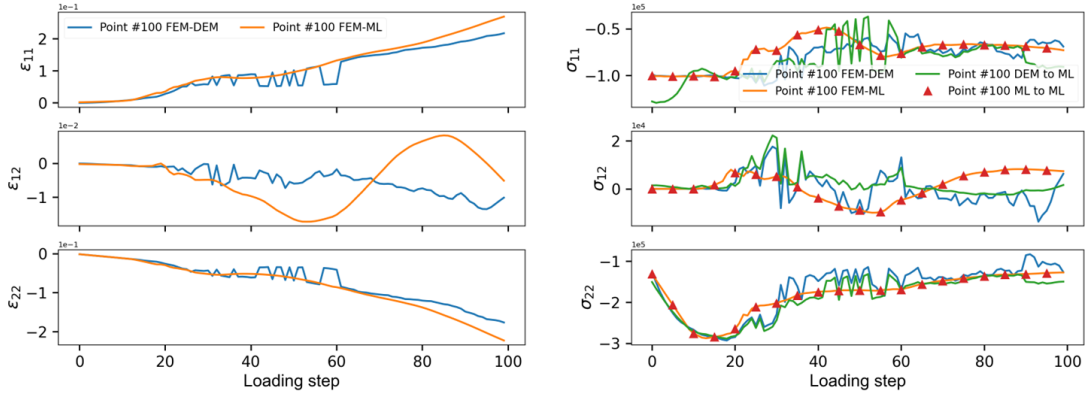


Figure 11 The shear strain based active learning resampling at seven load steps: Top row – 30% of the data points with highest uncertainty levels (the number in red is their average value); Bottom row – equivalent shear strain distribution

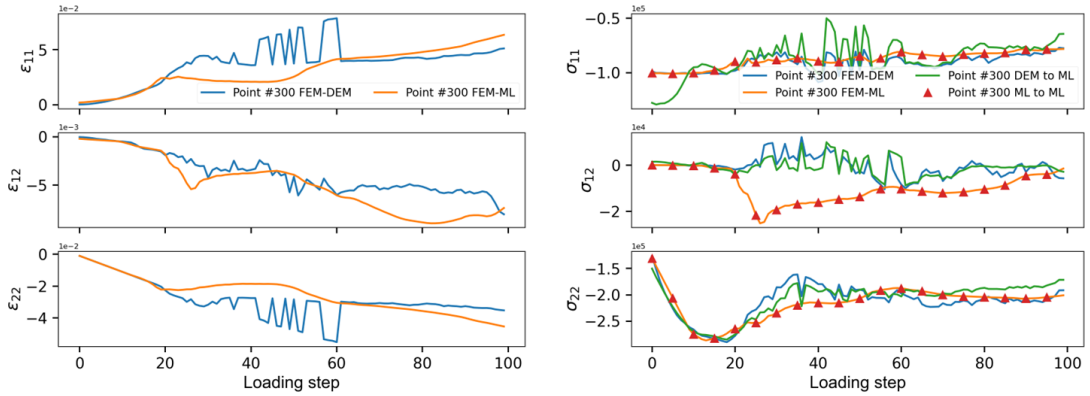
In conclusion, the shear strain can serve as a simple alternative indicator for resampling where large shear strain or shear localisation may be a dominant feature, but it may not be an effective indicator when the shear strain difference is small over the problem domain. More importantly, the shear strain indicator is problem-specific and hence its applicability is rather limited. For different problems, we may need to seek different problem-specific indicators for resampling in an indicator-based sampling scheme. On the contrary, the uncertainty-level based active learning may be slightly more complex in terms of usage, but it is generic and completely based on the evaluation of the uncertainty level of prediction results without any prior knowledge of the datasets concerned.

4.1.3 Comparison of strain-stress predictions at Gauss points

To further assess the validity of the new proposed framework, the strain-stress responses obtained by various models at Gauss point level are checked. The two Gauss points, #100 and #300, whose locations are shown in Figure 6, are chosen for inspection. In the right column of Figure 12, the blue line is the stress evolution history from the FEM-DEM simulation. The green line is the predicted stress from the trained network model fed with the strain obtained from the FEM-DEM simulation. The orange line is the stress from the FEM-ML simulation. The triangles denote the stress predicted by the network model using the strain obtained from the FEM-ML simulation.



(a) Gaussian point #100



(b) Gaussian point #300

Figure 12: Comparison of the local strain and stress responses from different solution schemes

The FEM-ML framework works well in predicting the main trends of the macroscopic stress response of granular materials. However, the trained network model inherently produces a smooth output but cannot reproduce a more oscillating output induced by the transient nature of DEM simulations, as notably visible in the FEM-DEM results. In FEM-DEM simulations, the strain updated via $\Delta\epsilon = \boldsymbol{B}\Delta\boldsymbol{u}$ will also oscillate because the DEM-induced stress fluctuation is

passed on to the FEM solver, as shown in the left column of Figure 12.

It is worth noting that the fluctuating or noisy output is the main obstacle in network model training, especially in high-dimensional problems, since it will aggravate the difficulty to distinguish useful information from the noise. Large and sharp fluctuations make the predicted curve nondifferentiable, which is also the reason that the automatic differentiation method available in most machine learning libraries cannot be directly used to obtain tangent operators in granular material simulation.

The network prediction directly using the strain obtained from the FEM-DEM simulation (green line) seems to have fully captured the fluctuation of the RVE results, except for the initial stress predictions of σ_x. After further inspection, we find that the initial strain returned by the FEM-ML solver is slightly different from the FEM-DEM solver due to the error in the predicted stress at the initial state. As the stress component σ_x is supposed to be near 100 kPa over the whole course of loading, its error in the figure is relatively more visually obvious than the other two.

4.1.4 Performance comparison

It is important to evaluate the efficiency of the FEM-ML framework. The performance of both FEM-DEM and FEM-ML simulations for Case D is compared without including the active learning resampling on a laptop computer (i5-8500 6 Cores@3.00GHz). All six cores are used in the FEM-DEM simulation, while only a single core is used in the FEM-ML simulation.

Table 4 shows that per iteration in the FEM-ML framework is nearly 82 times faster than the FEM-DEM. In addition to the substantial improvement in efficiency, huge computer memory is no longer needed, thus significantly alleviating the memory requirement of the FEM-DEM multiscale computation and also saving the communicating cost for distributed memory parallel platforms.

The number of iterations required at each load step is also recorded in Figure 13 for further comparison. Note that the medium mesh with 4×8 elements, as shown in Figure 6, is also simulated. As shown in Figure 13, both FEM-DEM and FEM-ML methods can converge at every

loading step, but more quickly during the initial (near) elastic stage. Then the number of iterations per loading step significantly increases until the peak is reached at around step 20. Afterwards, the iterations per step decrease steadily towards around 20 iterations.

Table 4: Summary of the number of iterations and clock time consumed in the simulation with the fine-meshed model

	FEM-DEM (multi-cores)	FEM-ML (single)	Speed up
Time (h)	8.02	0.11	72.9
Total iterations	2510	2820	0.89
Time per iteration (s)	11.50	0.14	**82.1**

Compared to the FEM-DEM simulation for the medium mesh, a slightly larger number of iterations are required by the FEM-ML simulation with all three meshes. This results from the error between the ML model prediction and the lower-scale RVE simulation. Because of this error, the FEM-ML framework will in general not converge as fast as the FEM-DEM, but still can obtain the final equilibrium state through iterations.

In summary, the results demonstrate that no notable difference is found in the number of iterations between FEM-ML and FEM-DEM simulations, which further supports the claim that ML models can be a satisfactory surrogate model for lower-scale RVE simulations.

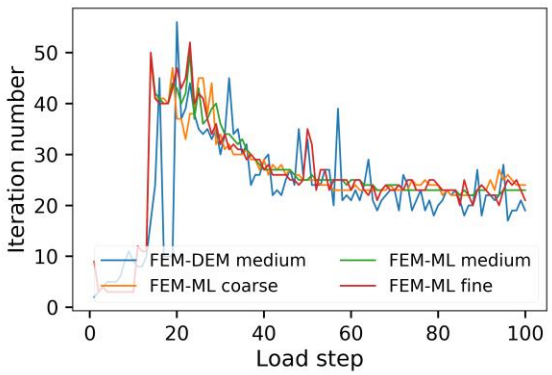


Figure 13: The number of iterations required for the FEM-DEM and FEM-ML simulations with the three meshes

4.2 Retaining wall example

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 problem. The details of the problem are shown in Figure 14, 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.

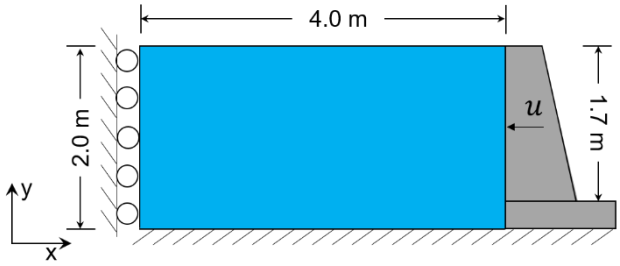


Figure 14: Schematic of the retaining wall problem

Figure 15a shows a prominent cambered shearing band emerging from the FEM-DEM simulation. The total force applied by the retaining wall versus the transverse strain is depicted in Figure 16.

Two FEM-ML approaches have been considered. Note that the first FEM-ML approach, labelled *FEM-ML 1*, is trained only on the datasets collected from the biaxial simulations described in Section 3, while the second approach, labelled FEM-ML 2, is trained based on the enhanced datasets, as will be explained below.

In the *FEM-ML 1* simulation, the solution process breaks down at about the 80th load step. The problem arises due to the accumulated error of the strain and the internal variable when their values are far beyond the network training ranges. The shearing band of the *FEM-ML 1* simulation is approximately lying on a straight line, which largely results from the training dataset gathered from the multiscale biaxial simulations, whose shearing band is a straight line. The failure of the *FEM-ML 1* simulation indicates that the neural network model trained based on the data from the biaxial compression test cannot fully reproduce the micromechanical response in the retaining wall simulation due to the limitation of the loading paths in the training samples.

The *FEM-ML 2* approach uses an enhanced network, where the network used in *FEM-ML 1* is retrained after the datasets of the retaining wall simulations with FEM-DEM are added to the original training samples. Figure 15c and Figure 16 show that the performance of the enhanced network is significantly improved in both displacement and force calculations. Therefore, the proposed methodology is adaptable to upgrade the network model once new datasets are available.

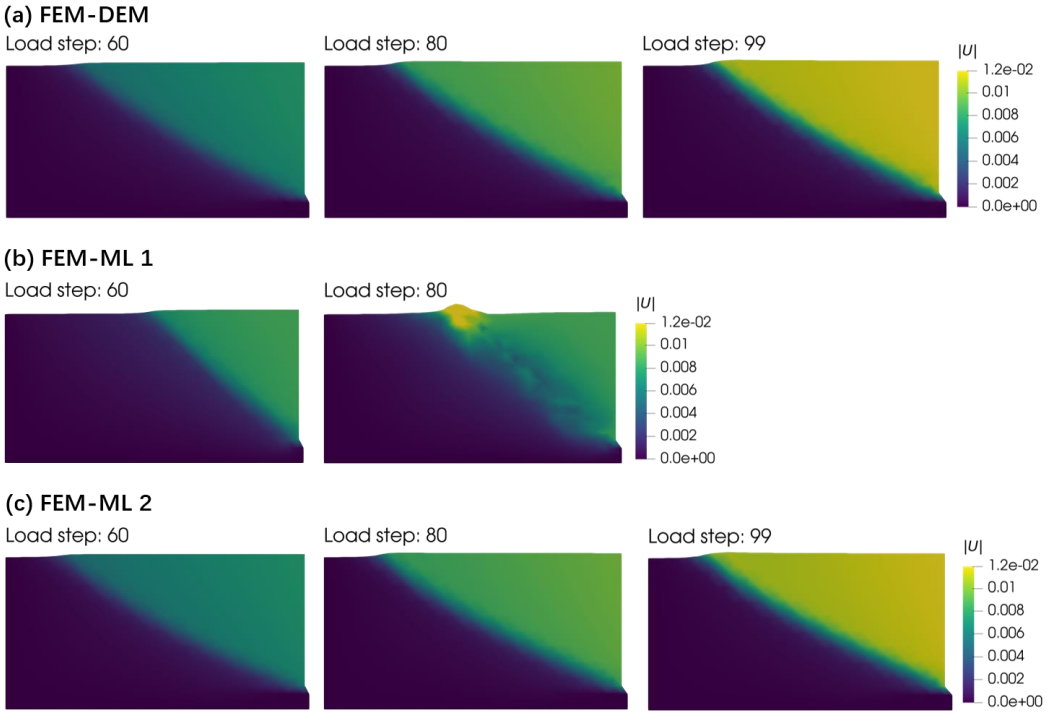


Figure 15: Displacement distributions of the soil at some load steps when compressed by the retaining wall and obtained by (a) FEM-DEM, (b) FEM-ML 1, and the (c) FEM-ML 2

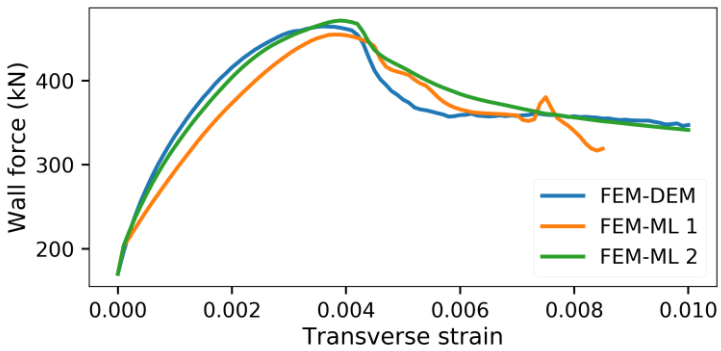
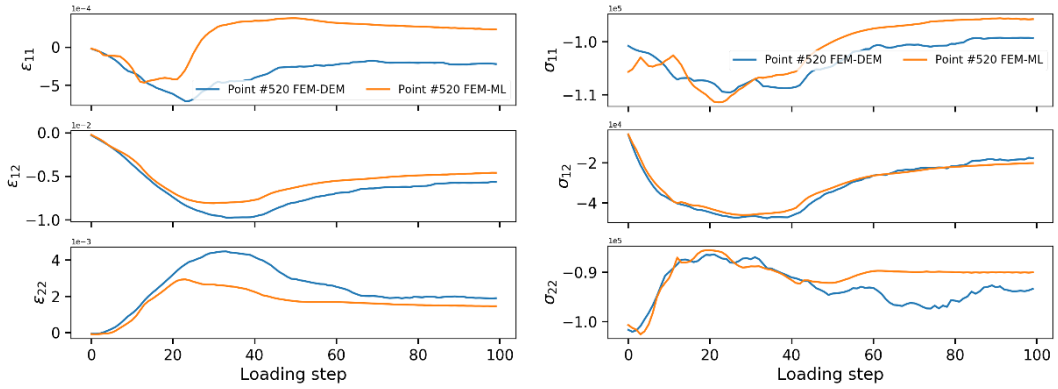


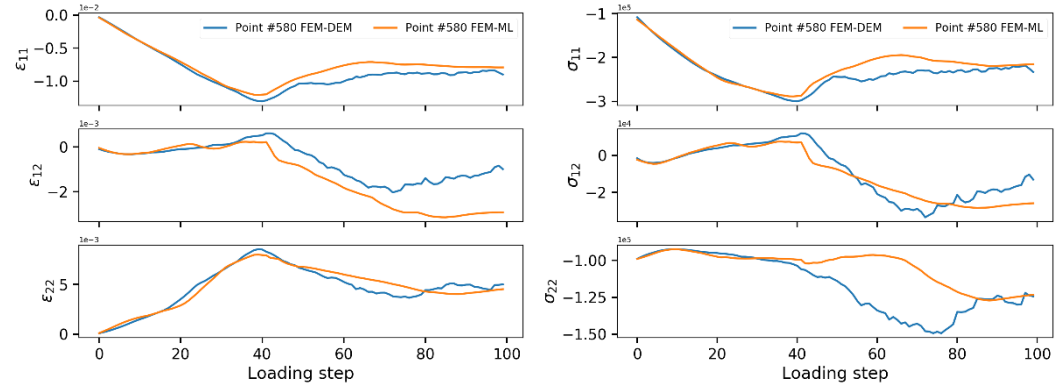
Figure 16: The integrated force on the retaining wall

We have extracted the strain at the Gaussian point of the retaining wall simulation to illustrate that,

as is shown in following figure, unloading has occurred in our example, or that the direction of strain increment has reversed rather than being a single loading. Without the use of internal variables to calibrate the granular material state, the network would not be able to reproduce this path-dependent property.



（a）Points #520 in retaining wall simulation



（b）Points #580 in retaining wall simulation

Figure 17 Curves of stress and strain of Gauss points

5 Conclusion

Our work is primarily devoted to developing a FEM-ML framework and training a network-based constitutive model to replace the micro RVE model via DEM, thus accelerating the classical multiscale FEM-DEM simulation. A multi-layer fully connected neural network, together with the use of the accumulated absolute values of the strain increments as an explicit parametrisation of the strain-stress relationship, is chosen as the surrogate model. This simple network appears to be reasonably able to reproduce the history-dependent mechanical response of granular materials. A Gaussian process-controlled random loading is applied to a biaxial compression problem to

generate training samples. In particular, an uncertainty-level-based active learning scheme is utilised to evaluate the informativeness of data points and select the points with high uncertainty levels to enrich the training dataset. This resampling strategy is generic and proved to be highly effective at least for the current problem concerned. A simpler but problem-specific shear strain-based resampling scheme is also discussed.

The drained biaxial compression tests conducted demonstrate that the FEM-ML framework can genuinely reproduce the micro-scale response of the granular material at a considerably lower CPU cost than the FEM-DEM approach. The generality or extrapolation capability of the proposed framework is also examined in the retaining wall example. The numerical result illustrates that this framework is sufficiently flexible to improve its performance as long as the training datasets are abundant or can be enriched.

The numerical examples provided highlight that a considerable improvement in the computational efficiency can be made based on the trained surrogate network models which may help to extend the multiscale computational framework to practical engineering problems. With the development of advanced numerical simulations and physical experiments, more high-fidelity datasets will be available. This tendency will contribute to developing a more accurate and general network-based constitutive mode and therefore further promote the application of machine learning-based constitutive models.

reference

1. Richard P, Nicodemi M, Delannay R, et al (2005) Slow relaxation and compaction of granular systems. Nat. Mater. 4:121–128
2. Ma G, Regueiro RA, Zhou W, Liu J (2019) Spatiotemporal analysis of strain localization in dense granular materials. Acta Geotech 14:973–990. https://doi.org/10.1007/s11440-018-0685-y
3. Ma G, Zhou W, Chang XL, et al (2016) Formation of shear bands in crushable and irregularly shaped granular materials and the associated microstructural evolution. Powder Technol 301:118–130. https://doi.org/10.1016/j.powtec.2016.05.068
4. Ma G, Zhou W, Zhang Y, et al (2018) Fractal behavior and shape characteristics of fragments produced by the impact of quasi-brittle spheres. Powder Technol 325:498–509. https://doi.org/10.1016/j.powtec.2017.11.030
5. Ma G, Zou Y, Gao K, et al (2020) Size Polydispersity Tunes Slip Avalanches of Granular Gouge. Geophys Res Lett 47:1–9. https://doi.org/10.1029/2020GL090458
6. Li XS, Dafalias YF, Wang ZL (1999) State-dependent dilatancy in critical-state constitutive modelling of sand. Can Geotech J 36:599–611. https://doi.org/10.1139/t99-029
7. Ma G, Zou Y, Chen Y, et al (2021) Spatial correlation and temporal evolution of plastic heterogeneity in sheared granular materials. Powder Technol 378:263–273. https://doi.org/10.1016/j.powtec.2020.09.053
8. Liu J, Zhou W, Ma G, et al (2020) Strong contacts, connectivity and fabric anisotropy in granular materials: A 3D perspective. Powder Technol 366:747–760. https://doi.org/10.1016/j.powtec.2020.03.018
9. Ma G, Chen Y, Yao F, et al (2019) Evolution of particle size and shape towards a steady state: Insights from FDEM simulations of crushable granular materials. Comput Geotech 112:147–158. https://doi.org/10.1016/j.compgeo.2019.04.022
10. Qu T, Feng Y, Wang M (2021) An adaptive granular representative volume element model with an evolutionary periodic boundary for hierarchical multiscale analysis. Int J Numer Methods Eng 122:2239–2253. https://doi.org/10.1002/nme.6620
11. Qu T, Feng YT, Wang Y, Wang M (2019) Discrete element modelling of flexible membrane

boundaries for triaxial tests. Comput Geotech 115:. https://doi.org/10.1016/j.compgeo.2019.103154

12. Azéma E, Linero S, Estrada N, Lizcano A (2017) Shear strength and microstructure of polydisperse packings: The effect of size span and shape of particle size distribution. Phys Rev E 96:1–10. https://doi.org/10.1103/PhysRevE.96.022902

13. Ng TT, Zhou W, Ma G, Chang XL (2018) Macroscopic and microscopic behaviors of binary mixtures of different particle shapes and particle sizes. Int J Solids Struct 135:74–84. https://doi.org/10.1016/j.ijsolstr.2017.11.011

14. Iwashita K, Oda M (2000) Micro-deformation mechanism of shear banding process based on modified distinct element method. Powder Technol 109:192–205. https://doi.org/10.1016/S0032-5910(99)00236-3

15. Oda M (1982) Fabric Tensor for Discontinuous Geological Materials. Soils Found 22:96–108. https://doi.org/10.3208/sandf1972.22.4_96

16. Ken-Ichi K (1984) Distribution of directional data and fabric tensors. Int J Eng Sci 22:149–164. https://doi.org/10.1016/0020-7225(84)90090-9

17. Yang ZX, Wu Y (2017) Critical State for Anisotropic Granular Materials: A Discrete Element Perspective. Int J Geomech 17:04016054. https://doi.org/10.1061/(asce)gm.1943-5622.0000720

18. Li XS, Dafalias YF (2012) Anisotropic Critical State Theory: Role of Fabric. J Eng Mech 138:263–275. https://doi.org/10.1061/(asce)em.1943-7889.0000324

19. Andrade JE, Tu X (2009) Multiscale framework for behavior prediction in granular media. Mech Mater 41:652–669. https://doi.org/10.1016/j.mechmat.2008.12.005

20. Andrade JE, Avila CF, Hall SA, et al (2011) Multiscale modeling and characterization of granular matter: From grain kinematics to continuum mechanics. J Mech Phys Solids 59:237–250. https://doi.org/10.1016/j.jmps.2010.10.009

21. Nitka M, Combe G, Dascalu C, Desrues J (2011) Two-scale modeling of granular materials: A DEM-FEM approach. Granul Matter 13:277–281. https://doi.org/10.1007/s10035-011-0255-6

22. Guo N, Zhao J (2014) A coupled FEM/DEM approach for hierarchical multiscale modelling of granular media. Int J Numer Methods Eng 789–818. https://doi.org/10.1002/nme

23. Guo N, Zhao J (2016) 3D multiscale modeling of strain localization in granular media. Comput

Geotech 80:360–372. https://doi.org/10.1016/j.compgeo.2016.01.020

24. Zhao JD, Guo N (2015) Bridging the micro and macro for granular media: A computational multi-scale paradigm. Geomech from Micro to Macro - Proc TC105 ISSMGE Int Symp Geomech from Micro to Macro, IS-Cambridge 2014 2:747–752. https://doi.org/10.1201/b17395-134

25. Ghaboussi J, Garrett JH, Wu X (1991) Knowledge-Based Modeling of Material Behavior with Neural Networks. J Eng Mech 117:132–153. https://doi.org/10.1061/(asce)0733-9399(1991)117:1(132)

26. Wu X, Ghaboussi J (2002) Neural Network-Based material modelling

27. Ghaboussi J, Pecknold DA, Zhang M, Haj-Ali RM (1998) Autoprogressive training of neural network constitutive models. Int J Numer Methods Eng 42:105–126. https://doi.org/10.1002/(SICI)1097-0207(19980515)42:1<105::AID-NME356>3.0.CO;2-V

28. Hashash YMA, Jung S, Ghaboussi J (2004) Numerical implementation of a neural network based material model in finite element analysis. Int J Numer Methods Eng 59:989–1005. https://doi.org/10.1002/nme.905

29. Sidarta DE, Ghaboussi J (1998) Constitutive Modeling of Geomaterials from Non-uniform Material Tests. Comput Geotech 22:53–71. https://doi.org/10.1016/S0266-352X(97)00035-9

30. Liu WK, Karniadakis G, Tang S, Yvonnet J (2019) A computational mechanics special issue on: data-driven modeling and simulation—theory, methods, and applications. Comput Mech 64:275–277. https://doi.org/10.1007/s00466-019-01741-z

31. Gao W (2018) A comprehensive review on identification of the geomaterial constitutive model using the computational intelligence method. Adv Eng Informatics 38:420–440. https://doi.org/10.1016/j.aei.2018.08.021

32. Zhang P, Yin ZY, Jin YF (2021) State-of-the-Art Review of Machine Learning Applications in Constitutive Modeling of Soils. Arch Comput Methods Eng 28:3661–3686. https://doi.org/10.1007/s11831-020-09524-z

33. Zhang W, Li H, Li Y, et al (2021) Application of deep learning algorithms in geotechnical engineering: a short critical review. Springer Netherlands

34. Zhang W, Phoon KK (2022) Editorial for Advances and applications of deep learning and soft computing in geotechnical underground engineering. J Rock Mech Geotech Eng 4–6.

https://doi.org/10.1016/j.jrmge.2022.01.001

35. Jung S, Ghaboussi J (2006) Neural network constitutive model for rate-dependent materials. Comput Struct 84:955–963. https://doi.org/10.1016/j.compstruc.2006.02.015

36. Ghaboussi J, Sidarta DE (1998) New Nested Adaptive Neural Networks (NANN) for Constitutive Modeling. Comput Geotech 22:29–52. https://doi.org/10.1016/S0266-352X(97)00034-7

37. Ghavamian F, Simone A (2019) Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network. Comput Methods Appl Mech Eng 357:112594. https://doi.org/10.1016/j.cma.2019.112594

38. Mozaffar M, Bostanabad R, Chen W, et al (2019) Deep learning predicts path-dependent plasticity. Proc Natl Acad Sci U S A 116:26414–26420. https://doi.org/10.1073/pnas.1911815116

39. Huang D, Fuhg JN, Weißenfels C, Wriggers P (2020) A machine learning based plasticity model using proper orthogonal decomposition. Comput Methods Appl Mech Eng 365:113008. https://doi.org/10.1016/j.cma.2020.113008

40. Xu K, Huang DZ, Darve E (2020) Learning Constitutive Relations using Symmetric Positive Definite Neural Networks. arXiv 1–31

41. Wang K, Sun WC (2018) A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. Comput Methods Appl Mech Eng 334:337–380. https://doi.org/10.1016/j.cma.2018.01.036

42. Tompson J, Schlachter K, Sprechmann P, Perlin K (2019) Accelerating eulerian fluid simulation with convolutional networks. 5th Int Conf Learn Represent ICLR 2017 - Work Track Proc

43. Giselle Fernández-Godino M, Panda N, O'Malley D, et al (2020) Accelerating continuum-scale brittle fracture simulations with machine learning. arXiv 1–22

44. Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. Comput Methods Appl Mech Eng 304:81–101. https://doi.org/10.1016/j.cma.2016.02.001

45. Conti S, Müller S, Ortiz M (2018) Data-Driven Problems in Elasticity. Arch Ration Mech Anal 229:79–123. https://doi.org/10.1007/s00205-017-1214-0

46. Eggersmann R, Kirchdoerfer T, Reese S, et al (2019) Model-Free Data-Driven inelasticity.

Comput Methods Appl Mech Eng 350:81–99. https://doi.org/10.1016/j.cma.2019.02.016

47. Karapiperis K, Stainier L, Ortiz M, Andrade JE (2021) Data-Driven multiscale modeling in mechanics. J Mech Phys Solids 147:104239. https://doi.org/10.1016/j.jmps.2020.104239

48. Kou B, Cao Y, Li J, et al (2017) Granular materials flow like complex fluids. Nature 551:360–363. https://doi.org/10.1038/nature24062

49. Logarzo HJ, Capuano G, Rimoli JJ (2021) Smart constitutive laws: Inelastic homogenization through machine learning. Comput Methods Appl Mech Eng 373:113482. https://doi.org/10.1016/j.cma.2020.113482

50. Zhang P, Yang Y, Yin Z-Y (2021) BiLSTM-Based Soil–Structure Interface Modeling. Int J Geomech 21:. https://doi.org/10.1061/(asce)gm.1943-5622.0002058

51. Qu T, Di S, Feng YT, et al (2021) Towards data-driven constitutive modelling for granular materials via micromechanics-informed deep learning. Int J Plast 144:. https://doi.org/10.1016/j.ijplas.2021.103046

52. Ma G, Guan S, Wang Q, et al (2022) A predictive deep learning framework for path-dependent mechanical behavior of granular materials. Acta Geotech 0123456789: https://doi.org/10.1007/s11440-021-01419-y

53. Guo N, Zhao J (2014) A coupled FEM/DEM approach for hierarchical multiscale modelling of granular media. Int J Numer Methods Eng 99:789–818. https://doi.org/10.1002/nme.4702

54. Christoffersen J, Mehrabadi MM, Nemat-Nasser S (1981) A micromechanical description of granular material behavior. J Appl Mech Trans ASME 48:339–344. https://doi.org/10.1115/1.3157619

55. Borja RI, Wren JR (1995) Micromechanics of granular media Part I: Generation of overall constitutive equation for assemblies of circular disks. Comput Methods Appl Mech Eng 127:13–36. https://doi.org/10.1016/0045-7825(95)00846-2

56. Wren JR, Borja RI (1997) Micromechanics of granular media Part II: Overall tangential moduli and localization model for periodic assemblies of circular disks. Comput Methods Appl Mech Eng 141:221–246. https://doi.org/10.1016/S0045-7825(96)01110-3

57. Kruyt NP, Rothenburg L (1998) Statistical theories for the elastic moduli of two-dimensional assemblies of granular materials. Int J Eng Sci 36:1127–1142. https://doi.org/10.1016/S0020-7225(98)00003-2

58. Guo M, Hesthaven JS (2019) Data-driven reduced order modeling for time-dependent problems. Comput Methods Appl Mech Eng 345:75–99. https://doi.org/10.1016/j.cma.2018.10.029
59. Hoerig C, Ghaboussi J, Insana MF (2018) Cartesian Neural Network Constitutive Models for Data-driven Elasticity Imaging. arXiv 1–22
60. Abadi M, Agarwal A, Paul Barham EB, et al (1983) TensorFlow: Large-scale machine learning on heterogeneous systems. Methods Enzymol 101:582–598
61. Korelc J, Wriggers P (2016) Automation of finite element methods. Autom Finite Elem Methods 1–346. https://doi.org/10.1007/978-3-319-39005-5
62. Huang DZ, Xu K, Farhat C, Darve E (2020) Learning constitutive relations from indirect observations using deep neural networks. J Comput Phys 416:109491. https://doi.org/10.1016/j.jcp.2020.109491
63. Qu T, Di S, Min YTF, et al Deep learning predicts stress-strain relations of granular materials based on triaxial testing data
64. Desai S, Strachan A (2021) Parsimonious neural networks learn interpretable physical laws. Sci Rep 11:1–9. https://doi.org/10.1038/s41598-021-92278-w
65. Tang S, Zhang G, Yang H, et al (2019) MAP123: A data-driven approach to use 1D data for 3D nonlinear elastic materials modeling. Comput Methods Appl Mech Eng 357:112587. https://doi.org/10.1016/j.cma.2019.112587
66. Tang S, Li Y, Qiu H, et al (2020) MAP123-EP: A mechanistic-based data-driven approach for numerical elastoplastic analysis. Comput Methods Appl Mech Eng 364:112955. https://doi.org/10.1016/j.cma.2020.112955
67. Tang S, Yang H, Qiu H, et al (2021) MAP123-EPF: A mechanistic-based data-driven approach for numerical elastoplastic modeling at finite strain. Comput Methods Appl Mech Eng 373:113484. https://doi.org/10.1016/j.cma.2020.113484
68. Zhou W, Liu J, Ma G, Chang X (2017) Three-dimensional DEM investigation of critical state and dilatancy behaviors of granular materials. Acta Geotech 12:527–540. https://doi.org/10.1007/s11440-017-0530-8
69. Choi S, Lee K, Oh S (2016) Gaussian random paths for real-time motion planning
70. Settles B (2011) Active Learning Literature Survey. Mater Lett 65:854–856.

https://doi.org/10.1016/j.matlet.2010.11.072

71. Raskutti G, Wainwright MJ, Yu B (2014) Early stopping and non-parametric regression: An optimal data-dependent stopping rule. J Mach Learn Res 15:335–366

72. Qu T, Feng YT, Zhao T, Wang M (2019) Calibration of linear contact stiffnesses in discrete element models using a hybrid analytical-computational framework. Powder Technol 356:795–807. https://doi.org/10.1016/j.powtec.2019.09.016

curves of strain and stress of Gauss points in retaining wall simulation

Response to the reviewers:

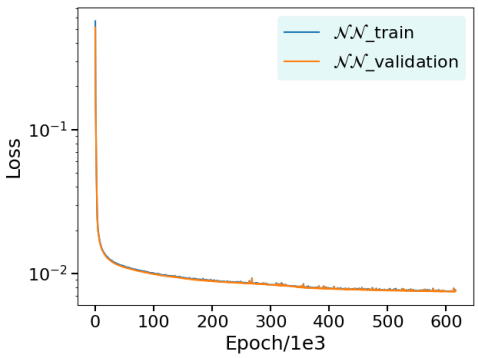
Reviewer #1: The following reviewer comments which were previously mentioned have not been resolved yet, and the authors are encouraged to address them:

We are sorry that we did not reply and express the last revision clearly. Thank you for your interest in our work and for your valuable comments both last time and this time. One important point is the loading/unloading process. The unloading points surely exist in the retaining wall simulation and even in the biaxial compression. We have taken all your suggestions into consideration and thank you for your advice.

1) The number of epochs used and training and test MSE or evolution of the loss function with epochs for test and training datasets need to be provided to demonstrate proper training of the network.

Here is the evaluation of the training loss and the validation loss with the epochs of the network used to predict the datapoints in figure 5. Properly training the network is a very crucial condition for us to complete the ML-based FEM calculation framework. Actually, Figure 5 is intended to show the validity of the network training.

By the way, 'validation' in following figure means the datapoints are split from the training datasets, and never be cast into the training process, only used to make sure the network is not overfitted as is mentioned in Line 450-455. While, points in Figure 5 in the manuscript are totally collected from the test set.



Review2 figure 1 Evolution of the loss with training epochs of the network used in Figure 5

2) The manuscript needs further proofreading. For example, "neutron" has been used instead of "neuron".

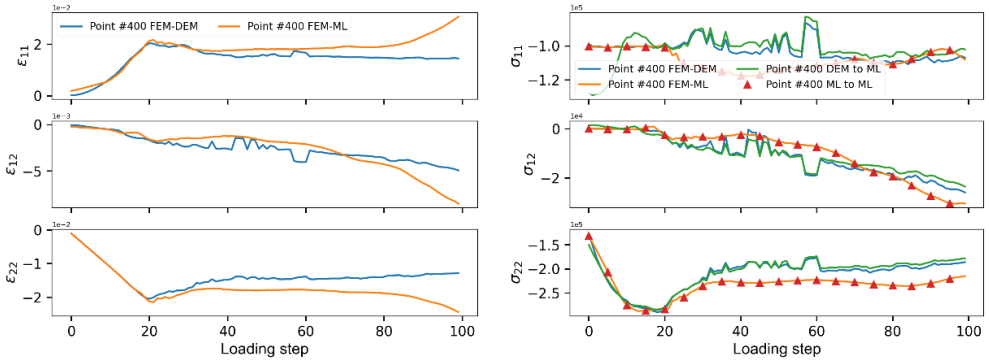
We further proofread the sentences and the logical order of this text.

3) Performance of the model for the case of loading and unloading (e.g. cyclic load) has not been studied in the paper. Therefore, the authors are encouraged to be more precise in their discussion in the paper and explain the extent to which the capability of the model in capturing path-dependent behavior has been studied and potential limitations in capturing path-dependent behavior which are left for future work.

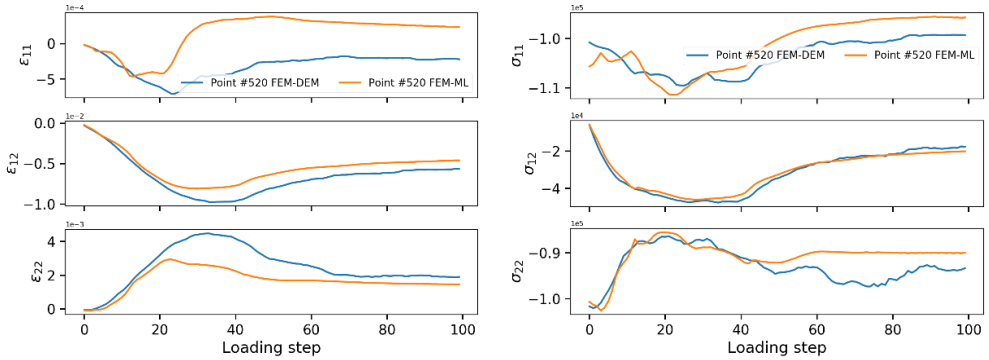
Thanks for your interest in this work. We have introduced the accumulation of the absolute value of strain increment as internal variables to calibrate the state of the granular material. This approach, although not perfect, enables the neural network to describe the path dependent nature of the granular material.

Your concern may be that the biaxial compression example presented in our paper without the add/drop cycle does not reflect the path dependence we mentioned. We have therefore extracted the strain at the Gaussian point of the both simulations to illustrate that, as is shown in following figure, unloading has occurred in our example, or that the direction of strain increment has reversed rather than being a single loading. Without the use of internal variables to calibrate the granular material state, the network would not be able to reproduce this path-dependent property.

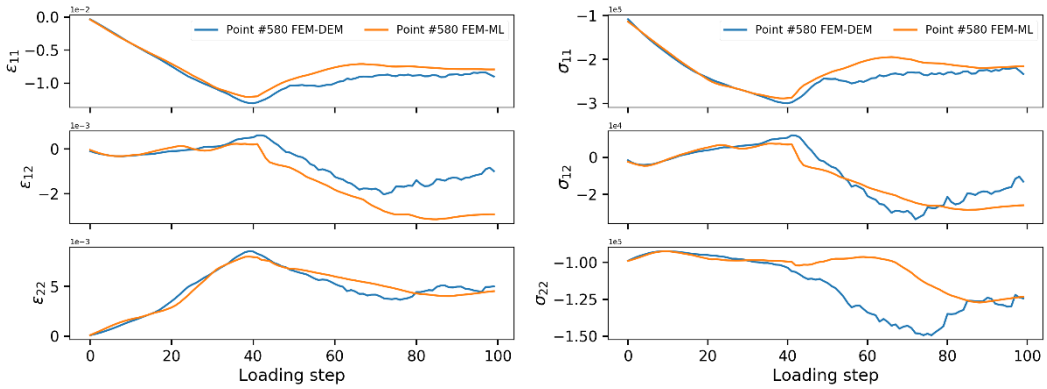
As you can see, we are up to the 100th load step in our loading. It may be necessary to add a few dozen more load steps if we are to finish unloading. A considerable problem with neural networks embedded in BVP calculations is error accumulation because, like recurrent neural networks, the predictions from the previous step are used to calculate the input values for the next prediction. Since our training range is not wide enough, the accuracy of the network drops dramatically when the input values are deviated from the training range. So if longer loading is to be accomplished, there needs to be a corresponding method to reduce the prediction error of the network or slow down the rate of error accumulating. We are still experimenting with this aspect of the work, which is a real challenge for us.



(a) Points #400 in biaxial compression



(a) Points #520 in retaining wall simulation



(b) Points #580 in retaining wall simulation

Review2 figure 2 Curves of stress and strain of Gauss points

4) The authors state in response to Comment 14 that "the network used in FEM-ML 1 is re-trained after the datasets of the retaining wall simulations with FEM-DEM are added to the original training samples". How many samples were added to the training set and what is the size of the final training set?

Also, are the simulation results of the retaining wall BVP in Section 4.2 used for both training (i.e. FEM ML 2 model) and validating the model (i.e. Figure 15(c) and Figure 16)? If so, the outcomes are not valid because a trained network typically performs well on the dataset on which it was trained, thus it cannot be validated using the same dataset used in training.

In the FEM-DEM retaining wall calculations in Section 4.2, we generated close to 2 million additional pairs of data. These data is added to whole the data described in Table 1, from which 0.1 was selected completely at random for training, due to the limited memory of our graphics card and the large number of duplicate samples in the sample. In the end, a total of 1,879,612 pairs of data were used in the training of the FEM-ML 2 computational model.

In fact, we do use all the data from the FEM-DEM retaining wall simulation. Initially we hope that the randomly generated training samples using the Gaussian process described in Section 3.4 would be sufficient to cover the prediction space required for the retaining wall calculations. However, due to the characteristic that granular materials are not able to withstand tensile stresses, it was not possible to design a completely random loading path. FEM-ML 1 did not use the FEM-DEM retaining wall simulation data at all, and thus is not able fully complete the retaining wall calculation, indicating that the sample space is not sufficient. The results of FEM-ML 2 can only be obtained after inputting the FEM-DEM retaining walls datasets.

In response to the question "If so, the outcomes are not valid because a trained network typically performs well on the dataset on which it was trained, thus it cannot be validated using the same dataset used in training.". We have attached plots of the stress-strain curves at Gaussian points calculated in the FEM-ML 2 retaining wall simulation. It can be seen that even when all the data points are put into the network training, the network cannot fully reproduce the results in the FEM-DEM dataset when used again for BVP calculations, and there are even many points where the error at the granular ensemble level is still significant. So it should be a relatively difficult job to reproduce the multi-scale calculation via network calculation even if we have the noisy datasets, especially to calculate at all of the Gauss points and achieve global balance.

Actually, the network's ability to generalise is limited and is only reflected in the prediction of the interpolation range. Once the input exceeds the trained range, the results obtained will be meaningless which is suggested by the retaining wall calculation.

Therefore, this work is used to demonstrate the usefulness of neural networks for improving the

efficiency of multi-scale computation and the challenges of generalisation that a purely data-driven approach will face in BVP computation.

The reviewer has a few additional comments as listed below:

1) Figure 8 shows the simulation results with different coarse and fine meshes from Figure 6. What are the meshes used in the simulations and the presented results?

Yes, the model in Figure 8 is shown in Figure 6. All of the computational details for Figure 8 are taken from Table 3, where we detail the level of grid division for the computational model and what level of grid the data used for network training came from.

2) How are the number of training samples summarized in Table 1 obtained? It seems that 10 loading paths were generated, each with 100 time steps, and were applied as confining pressure of the biaxial BVP in Figure 6. For example, for the coarse mesh, it seems to the reviewer that there are 32 Gauss points, each undergoing 10 loading cases with 100 steps each, leading to 32000 samples.

Sorry we didn't explain the data generation process clearly.

Due to the nonlinear macroscopic mechanical properties of granular materials, each loading step needs nonlinear iteration. On average, a loading step takes about 20 iterations. Therefore, more than 2000 calculation results are generated in 100 loading steps, as is shown in Figure 13. For a coarse mesh, with 32 Gaussian integration points, a single simulation can produce more than 32*100*20=64000 pairs of data.

Reviewer #2: This study is innovative in machine learning use for multiscale computation framework for granular materials. It is interesting and practical. The revision is of considerable quality for publication in AG. It can be accepted in current version.

Thanks for your kind suggestion.